ASPAI
PORTO • 2021

Advances in Signal Processing and Artificial Intelligence

Proceedings of the 3rd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2021)

Edited by Sergey Y. Yurish

IFSA

ASPAI
PORTO • 2021

Advances in Signal Processing and Artificial Intelligence:

Proceedings of the 3rd International Conference on Advances in Signal Processing and Artificial Intelligence

17 - 18 November 2021
Porto, Portugal

Edited by Sergey Y. Yurish

IFSA Publishing

Sergey Y. Yurish, *Editor*
Advances in Signal Processing and Artificial Intelligence
ASPAI’ 2021 Conference Proceedings

E-mail (for orders and customer service enquires): ifsa.books@sensorsportal.com

Visit our Home Page on https://sensorsportal.com/ifsa_publishing.html

Neither the authors nor International Frequency Sensor Association Publishing accept any responsibility or liability for loss or damage occasioned to any person or property through using the material, instructions, methods or ideas contained herein, or acting or refraining from acting as a result of such use.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identifies as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

ASPAI Conference Website: https://aspai-conference.com/

ISSN: 2938-5350
ISBN: 978-84-09-35865-6
BN-20211115-XX
BIC: UYQ

Contents

Foreword

On behalf of the ASPAI’ 2021 Organizing Committee, I introduce with pleasure these proceedings devoted to contributions from the 3rd International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2021).

Advances in artificial intelligence (AI) and signal processing are driving the growth of the artificial intelligence market as improved appropriate technologies is critical to offer enhanced drones, self-driving cars, robotics, etc. Today, more and more sensor manufacturers are using machine learning to sensors and signal data for analyses. The machine learning for sensors and signal data is becoming easier than ever: hardware is becoming smaller and sensors are getting cheaper, making Internet of things devices widely available for a variety of applications ranging from predictive maintenance to user behavior monitoring. Whether we are using sounds, vibrations, images, electrical signals or accelerometer or other kinds of sensor data, we can build now richer analytics by teaching a machine to detect and classify events happening in real-time, at the edge, using an inexpensive microcontroller for processing - even with noisy, high variation data.

According to the Research & Markets recent study, the global artificial intelligence market is expected to grow from US $ 58.3 billion by 2021 to US $ 309.6 billion by 2026 at a compound annual growth rate of 39.6 %. Artificial intelligences currently transforming the manufacturing industry. Virtual reality, automation, Internet of Things (IoT), and robotics are some important features of AI that are benefitting the manufacturing industry. AI has been one of the fastest-growing technologies in recent years. The market growth is mainly driven by factors such as the increasing adoption of cloud-based applications and services, growing big data, and increasing demand for intelligent virtual assistants. The major restraint for such market is the limited number of AI technology experts.

The Series of ASPAI Conferences have been launched to fill-in this gap and to provide a forum for open discussion and development of emerging artificial intelligence and appropriate signal processing technologies focused on real-word implementations by offering Hardware, Software, Services, Technology (Machine Learning, Natural Language Processing, Context-Aware Computing, Computer Vision and Signal Processing). The goal of the conference is to provide an interactive environment for establishing collaboration, exchanging ideas, and facilitating discussion between researchers, manufacturers and users. The first ASPAI conference has taken place in Barcelona, Spain in 2019 and the second – in Berlin, Germany in 2020.

The conference is organized by the International Frequency Sensor Association (IFSA) - one of the major professional, non-profit association serving for sensor industry and academy more than 20 years, in technical cooperation with media partners – IOS Press (journal *‘Integrated Computer-Aided Engineering’*) and World Scientific (*International Journal of Neural Systems*). The conference program provides an opportunity for researchers interested in signal processing and artificial intelligence to discuss their latest results and exchange ideas on the new trends.

I hope that these proceedings will give readers an excellent overview of important and diversity topics discussed at the conference. Selected, extended papers will be submitted to the media partners’ journals and IFSA’s open access ‘*Sensors & Transducers*’ journal based on the proceeding’s contributions.

We thank all authors for submitting their latest work, thus contributing to the excellent technical contents of the Conference. Especially, we would like to thank the individuals and organizations that worked together diligently to make this Conference a success, and to the members of the International Program Committee for the thorough and careful review of the papers. It is important to point out that the great majority of the efforts in organizing the technical program of the Conference came from volunteers.

Prof., Dr. Sergey Y. Yurish
ASPAI’ 2021 Conference Chairman

(002)

A kNN Approach for Melanoma Diagnosis Based on Color Cluster Features

S. Moldovanu [1, 3], F. A. Damian [2, 3] and L. Moraru [2, 3]
[1] University of Dunarea de Jos, Department of Computer Science and Information Technology,
Faculty of Automation, Computers, Electrical Engineering and Electronics, 2 Științei Str.,
800146, Galati, Romania
[2] University of Dunarea de Jos, Department of Chemistry, Physics & Environment, Faculty of Sciences
and Environment, Dunarea de Jos University of Galati, 47 Domneasca Str., 800008 Galati, Romania
[3] The Modelling & Simulation Laboratory, Dunarea de Jos University of Galati, 111 Domneasca Str.,
800102 Galati, Romania
Tel.: +40 336 130 236, fax: + +40 236 470 905
E-mail: moldovanu.simona@ugal.ro

Summary: The study develops an algorithm for skin lesion classification, which is based on twenty-three color clusters. The goal is to differentiate between melanoma and nevus classes. The proposed framework works as follows: firstly, hair removal is carried out using the Dull-razor algorithm and filtering; then, lesion segmentation using the optimal color thresholding on each RGB color channels is executed; finally, statistical features extraction is performed, from the color clusters which were selected using the minimum and maximum RGB intensity and features selection. Three datasets, namely 7-Point (365 images), Med-Node (140 images), and PH2 (120 images) were investigated. The performance measures like accuracy (ACC), Specificity (S) and Precision (P) were valuated using the k-nearest neighbours (k-NN) algorithm. The selected most discriminant color features have archived the following accuracy values: 0.7317 for color cluster 13 and for 7-Point dataset; 0.8196 for color cluster 5 and Med-Node dataset; and 0.8167 for color cluster 15 and PH2 dataset.

Keywords: Melanoma, Nevus, Color cluster features, kNN classifier.

1. Introduction

To diagnose a skin lesion as melanoma - the deadliest type - at the early stages is crucial because the chances to be cured increase by more than 95 %. In the year 2019, reported skin cancer cases, in the USA, were 104,350, from which 62,320 were men and 42,030 women. There were 7320 melanoma death cases [1].

The visual inspection for skin lesion detection involves a screening analysis of the asymmetry, border, color, diameter and how the skin lesion changes over time. Nevi, as benign lesions, are mostly a single shade of brown. Melanoma, as malign lesions, show different shades of brown, tan or black even red, white or blue. The percent of the color in the perceived color space of a lesion can vary substantially. Seidenari *et al.* [2] proposed as relevant a number of twenty-three color clusters that contain almost 88 % of all lesion possible colors. These clusters are used to assess the occurrence of each color either in melanoma or nevi. They were selected based on the minimum and maximum R, G, and B values in each color channel and by considering their contribution to the color space of the skin lesions. An accurate feature extraction is a pre-conditional step for a sound classification process through the artificial intelligence techniques.

Shahi et al. [3] focused on the pre-processing stage (i.e., contrast improvement, histogram adjustment and noise filtering), segmentation, feature extraction and classification using four classifiers (k-NN, SVM, Ensemble and Decision tree) to classify the benignity or malignancy of skin lesions. In this study, research was conducted for the classification of the skin lesions purpose by following the stages: image segmentation, hair removal, color features extraction and prediction with KNN algorithm.

2. Materials and Methods

The proposed methodology is tested and validated on the public 7-Point (68 nevus/297 melanoma dermoscopic images), Med-Node (100 nevus/ 40 melanoma non-dermoscopic images) and PH2 (80 nevus /40 melanoma dermoscopic images) benchmark datasets. The preprocessing and segmentation results are displayed in Fig. 1. The hair removal has been performed using the DullRazor software (Fig. 1(b)) [4] along with a closing operation and a bilinear interpolation. Then, the image is filtered using a median filter. Furthermore, the RGB color space is split into the R, G and B color channels. The image is segmented for each color channel using the color thresholding algorithm provided by the MATLAB 2020 environment. The obtained mask facilitates the segmentation of color images (Fig. 1(c)). According to [2], from the colour histogram, only 23 color clusters are relevant color lesions; these color clusters are distributed between nevi and melanoma (Table 1). This selection was based on the significant difference (>1 %) between the pairs (Rmin, Rmax), (Gmin, Gmax) and (Bmin, Bmax). The weight, i.e. the ratio of pixels numbers belonging to each of each color cluster and total pixels number in the lesion, is computed.

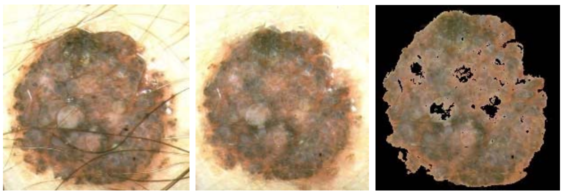

(a) (b) (c)

Fig. 1. Sample segmentation of skin lesions. (a) Nevus image belongs to the 7Point dataset; (b) Hair removal; (c) Result of segmentation based on optimum color thresholding method.

Table 1. The color clusters.

c1	c2	c3	c4	c5	c6
■	■	■	■	■	■
c7	c8	c9	c10	c11	c12
■	■	■	■	■	■
c13	c14	c15	c16	c17	c18
■	■	■	■	■	■
c19	c20	c21	c22	c23	
■	■	■	■	■	

The main purpose of this study is to discern the color cluster features extracted from skin lesions in order to classify the skin lesions into melanoma or benign nevus. Due to its good reputation, we employ a kNN classifier. The images were grouped into two classes (nevi and melanoma) for each color cluster. The kNN classifies each unknown sample. The confusion matrix provides information on the accuracy, $ACC = \frac{\text{TP+TN}}{\text{TP+FN+TN+FP}}$, precision $P = \frac{TP}{TP+FN}$, and Sensitivity $S = \frac{TP}{TP+FP}$, where TP, TN, FN, FP are the true positives, true negatives, false negatives, false pozitives, respectively.

3. Results and Discussions

The pixels' intensity inside of the lesion is summerized based on the relative color histogram bins. The color cluster features selected for each dataset relative to the significant difference (>1 %) between the pairs are presented in Table 2.

The classification perfomance for the selected color cluster is displyed in Table 3.

The accuracy of the classifier is almost the same for Med-Node and PH2 datasets. This suggest that the proposed features provide adequate information for distinguishing between nevi and melanoma in two different image types: non-dermoscopic and dermoscopic. Fig. 2 shows the accuracy values for the weight of all selected color clusters.

Table 2. The relevant color cluster selected for each dataset, minimum and maximum of intensity value for each RGB channel.

Datasets	Cluster No.	Minimum value R/G/B	Maximum value R/G/B
7-Point	c13	159/127/127	128/96/96
Med-Node	c5	96/64/64	127/95/95
PH2	c15	160/96/96	191/127/127

Table 3. Performance measures for each relevant cluster and for all datasets.

Datasets	Cluster No.	ACC	S	P
7-Point	c13	0.7317	0.7172	0.7204
Med-Node	c5	0.8196	0.7934	0.8005
PH2	c15	0.8167	0.7899	0.8083

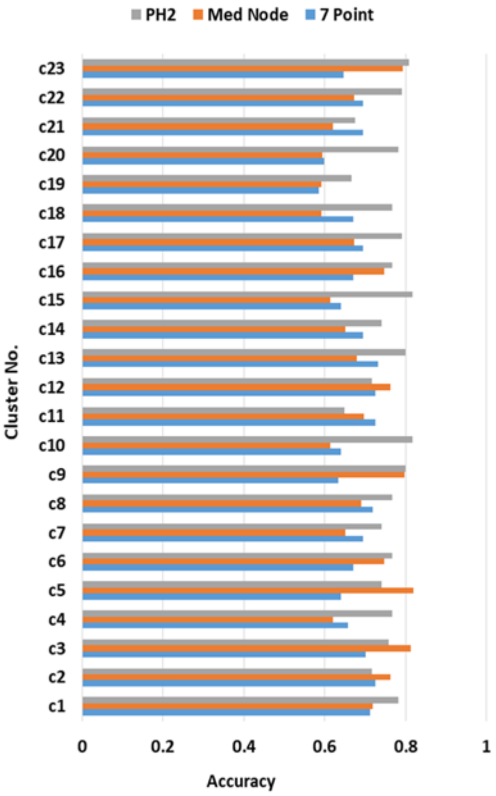


Fig. 2. Plot of accuracy vs. the weight of color clusters.

This approach allows to rank the color dermoscopic features according to their importance in the discrimination between nevi and melanoma. We found that the amount of brown and dark-brown color are relevant for classification problem.

4. Conclusions

We have presented a method based on color clusters features and a kNN classifier which can be used to separate nevus and melanoma from color images. As the accuracy is above 0.800 for Med-Node and PH2 datasets, for c5 and c15 color cluster, respectively, we conclude that the proposed system could be further used as a diagnostic aid for skin lesion images.

Acknowledgement

The project was financed by Dunarea de Jos University of Galati, Grant CNFIS-FDI2021-0443 – Măsuri active de creștere și eficientizare a capacității de cercetare, dezvoltare, inovare și transfer tehnologic în Universitatea Dunărea de Jos din Galați – CEREX-UDJG_2021.

References

[1]. M. E. Celebi, N. Codella, A. Halpern, Dermoscopy image analysis: overview and future directions, *IEEE Journal of Biomedical and Health Informatics*, Vol. 23, Issue 2, 2019, pp. 474-478.

[2]. S. Seidenari, C. Grana, G. Pellacani, Colour clusters for computer diagnosis of melanocytic lesions, *Dermatology*, Vol. 214, 2007, pp. 137-143.

[3]. P. Shahi, S. Yadav, N. Singh, N. P. Singh, Melanoma skin cancer detection using various classifiers, in *Proceedings of 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON'18)*, Gorakhpur, India, 2-4 Nov. 2018, pp. 1-5.

[4]. T. Lee, V. Ng, R. Gallagher, A. Coldman, D. McLean, DullRazor: A software approach to hair removal from images, *Computers in Biology and Medicine,* Vol. 27, 1997, pp. 533-543.

(003)

Revising Pooling in the CNN

J.-Ph. Conge, V. Vigneron and H. Maaref
IBISC EA 4526, Univ. Evry, Universite Paris-Saclay, France
E-mails: {vincent.vigneron,hichem.maaref}@univ-evry.fr, jean-philippe.conge@universite-paris-saclay.fr

Summary: Much of Convolutional neural networks (CNNs) success has been in translation invariance. The other part is that thanks to a judicious choice of architecture, the network is able to take decisions taking into account the whole image. The pooling layer is at the heart of every CNN contributing to invariance to data variation and disturbance. It describes what part of the input image an output layer neuron can see. This approach has rarely been questioned. We propose another way that we named Z-pooling to extend the pooling function, able to extract texture descriptors from images. Z-pooling layers are nonparametric, independent of the geometric arrangement or sizes of image regions, and therefore can better tolerate rotations. Z-pooling functions produce images capable of emphasizing low / high frequencies, outlines, etc. We show in this article that Z-pooling leads to CNN models, which can optimally exploit the information of their receptive field.

Keywords: Deep learning, Pooling operator, Image segmentation, Tumor detection.

1. Introduction

Networks built exclusively from fully connected layers can in theory model any continuous function [4]. This also works for images if they are linearized so that they are represented as a vector x. In practice two problems arise which make the use of fully connected layers undesirable.

The first is that fully connected layers are not *translation invariant*. This is easily understood with an example of image classification. Assume that handwritten numbers have to be detected on an image. Once the network has learned to detect the number ’2’ in the left upper corner of the image it will do just fine. But if the number ’2’ is moved to any other part of the image it will no longer be recognized as the number ’2’. Thus the training set must contain an example of the number ’2’ in any possible position on the image. For handwritten numbers this is doable, but tedious. For a set of natural images it may not be possible to acquire images with all the possible translations of an object and with high resolution images (*i.e.* many possible translations) the size of the training database would become unmanageable.

The second problem is that the number of parameters increases with the input and output size. For image segmentation tasks the input is an image and the output is a (segmented) image as well. An image of resolution 512×512 has 262,144 pixels. With x and y of that size the matrix W of a single perceptron would already have 68,719,476,736 entries. When working with 32 bit numbers this corresponds to 256 GB of memory required. The models are therefore too large for contemporary hardware.

The solution to both problems is the usage of convolutions.

A convolutional layer is written as

$$y = f(W * x + b), \tag{1}$$

where the convolution is defined as the discrete convolution in equation (1). The number of parameters in W now becomes a free choice because the matrix multiplication has disappeared and the convolution is defined for two signals of arbitrary length. This fixes the hardware issues, as the models size can be arbitrarily scaled. Also the input and output no longer needs to be a vector but can be a matrix or any higher order tensor. But using convolutions comes with a cost. In a fully connected layer the output has access to all inputs. In a convolution an output pixel only has access to the part of the input given by the size of the convolution kernel. This part of the input which is available to the final layer of a network is called the *receptive field*. The goal of all CNN architectures is to increase this receptive field until it is large enough for the task at hand. For most image analysis tasks this is the entire input image. The use of large convolution kernels is not possible because the computational complexity and quantity of parameters increase quadratically with the kernel size. Therefore the kernel sizes are set to 3×3 or 5×5 in out architectures (Fig. 1).

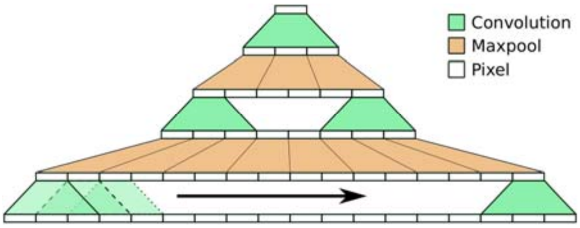


Fig. 1. An input image (bottom) is reduced in resolution with alternating layers of convolution and maximum pooling. Convolution layers have a kernel size of 3 and maximum grouping layers have a window size of 2 and a stride of 2. With 5 operations, the final top pixel has a receptive field of 18. Note that the receptive field grows exponentially with the number of maximum pooling layers. This example also shows the pyramid structure emerging from multi-resolution techniques.

But the receptive field can be increased linearly by stacking convolutional layers. With two 3×3 layers following each other, the last one can ”see” a 4×4 neighbourhood in the input layer. By stacking 3×3 layers, the receptive field increases by one per convolution. Which means a lot of convolutions must be stacked to have a receptive field as large as a reasonable input image. The increase in the convolutional receptive field can be considerably greater when using maximum pooling layers [12] with a stride of 2. The stride induces downsampling of the image to a lower resolution. Fig. 1 illustrates the effect of alternating convolution and pooling layers on the receptive field. With such a pyramidal structure of sufficient depth, the last convolution will have access to a (downsampled) version of the entire image. For image segmentation tasks this downsampled image is once again upsampled to the original size in an inverse pyramid pattern. This concept has been known for a long time [7] and is widely applied in today’s CNNs. Lin et al. [8] have a good overview of CNN architectures which use such pyramid/ multiresolution techniques. Multi-resolution (pyramidal) structures comes from the idea that the network needs to see different levels of detail (resolutions) to produce good results. A CNN stacks four different processing layers: convolution, pooling, ReLU and fully-connected [2, 3] (see Fig. 1). Pooling (i) reduces the number of parameters in the model (subsampling) and calculations in the network while preserving their important characteristics (ii) improves the efficiency of the network (iii) avoids over-learning.

Max-pooling function sub-samples the input representation (image, hidden layer output matrix, etc.), by reducing its dimensionality.

The weaknesses of pooling functions are well identified [15]: *(a)* they do not preserve all the spatial information *(b)* the maximum chosen by the max-pooling in the pixel grid is not the true maximum *(c)* average pooling assumes a single mode with a single centroid. The question is how to take into account optimally the characteristics of the regions (input image) pooled into the pooling operation? Part of the response lies in Lazebnik’s work [6] who demonstrated the importance of the spatial structure of pooling neighborhoods.

This paper proposes a new pooling operator, independent of the geometric arrangement or the size of image regions, and can therefore better tolerate rotations. It is based on Zeckendorf’s integer decomposition theorem and is also simple to implement.

2. Z-pooling Operator

In statistics, ”pooling” means gathering together small sets of data that are assumed to have the same value of a characteristic, *e.g.* a mean.

The goal of pooling is to transform convolution features into a new representation that preserves important information while ignoring irrelevant details.

So, should we pool or not? Or when should we pool and when should we not? The answer depends upon the following considerations, in decreasing order of importance: (i) It would be best to test the *pool ability* before to do so; (ii) Just as ”acceptance” of a null hypothesis does not mean it is necessarily true, ”acceptance” in a pool ability test does not mean that pool ability is necessarily justified; (iii) There is an inadequate number of observations in each of two (or more) subgroups, which would usually necessitate pooling; (iv) Common sense, necessity, etc.

The amount of information extracted from different regions of an image usually depends on the size of the neighborhood, the reading order of the neighbors and the mathematical function that is used to extract the relationship between two neighboring pixels. Most of the descriptors that encode local structures *i.e.* local binary patterns (LBP) [10] and its variants [11], census transform (CT) [16] etc. are dependent on the reading order as they compute the feature value as the weighted sum of mathematical function of neighboring pixels w.r.t their order in the neighborhood. There exists many variants of LBP (see [11, Chap. 2, p. 26] for a summary of the variants) for many types of problems because basic LBP has also some problems that need to be addressed. For example LBP and CT both generates 8-bit string for a 3×3 neighborhood by computing the Heaviside function of the difference of neighboring pixel g_i and the central pixel g_c *i.e.* $(g_i - g_c)$ in case of LBP code and the difference of central pixel and neighboring pixel *i.e.* $(g_c - g_i)$in case of CT code which number is shown in Fig. 2. The only difference between these two descriptors is the reading order of neighboring pixels and the sign of the difference which results in 2 different bit patterns. Given the 8-bit string, the LBP and CT code is calculated as:

$$code_{P,R} = \sum b_{i2}^{i}, \tag{2}$$

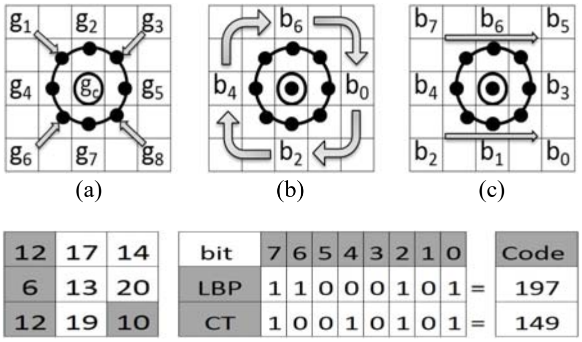
where P is the number of pixels in the neighborhood considering the distance R between central pixel and its neighbors.

The Zeckendorf’s theorem [17] states that every positive integer N can be represented uniquely as the sum of distinct Fibonacci numbers such that the sum *does not include any two non-consecutive Fibonacci numbers* {1,1,2,3,5, 8,...}, *that satisfies the difference equation*

$$x(n) = x(n-1) + x(n-2), \forall n \geq 0 \tag{3}$$

that is $x(n)$ is the sum of the 2 previous values with initial conditions x(0) = x(1) = 1, *e.g.* the number 13 = 8 + 5 = 8 + 3 + 2 or, equivalently, in the Fibonacci base, 010110(fib) or 011000(fib) or 100000(fib). The distinct Fibonacci numbers below 255 are 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233. Based on this

Zeckendorf's additive property of integers, an 8-bit gray scale image has the intensity values in the range of [0, 255]. Each pixel intensity of an image can be represented as a sum of distinct nonconsecutive Fibonacci numbers. For instance the only Zeckendorf representation of pixel value 255 is $(233,21,1)_{\text{Zck}}$. Since there are 12 different possibilities to represent any 8-bit intensity value, therefore each number is represented using 12-bits no consecutive bits are ON *i.e.* 1 because of non-consecutive Fibonacci numbers constraint.



(a) (b) (c)

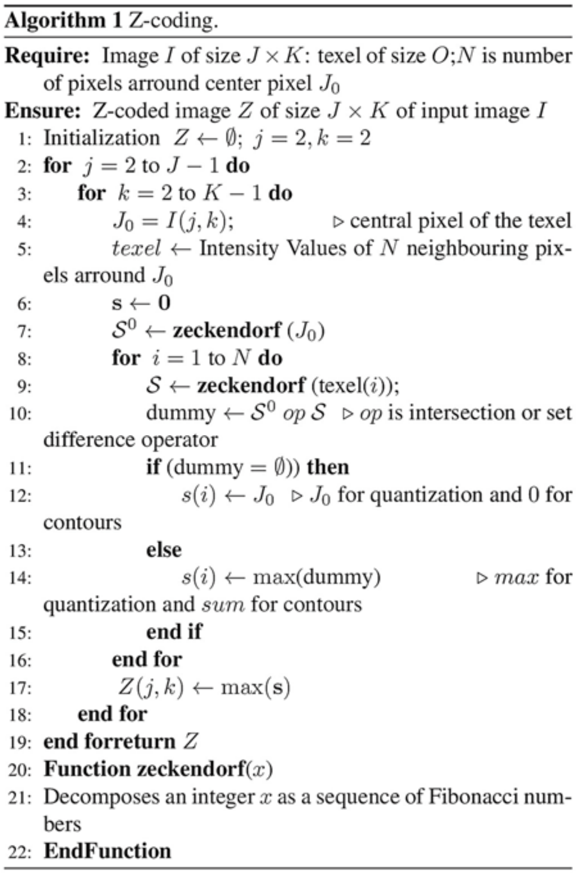
12	17	14
6	13	20
12	19	10

bit	7	6	5	4	3	2	1	0		Code
LBP	1	1	0	0	0	1	0	1	=	197
CT	1	0	0	1	0	1	0	1	=	149

Fig. 2. (a) Neighbors of center pixel g_c participating in code (LBP or CT) generation (b) bit ordering in case of LBP (c) bit ordering in case of CT (d) LBP and CT code generation for a 3×3 neighborhood.

We propose the Algorithm 1 to encode pixel relationship with its local neighborhood we named Z-code. The sequence in which various operators such as supremum (max) or infimum (min) are combined results in images that could be directly used in the computer vision pipeline.

The proposed algorithm results in two different kinds of images based on the initial operator which is applied *i.e.* either set difference or intersection. The *intersection* operator find the similarity among the pixel and its neighborhood Zeckendorf representation and place a value which is common among them thus results in an image that is quantified in terms of their representation as shown in Fig. 3. The set *difference operator* extracts *ultrametric* contours of an image resulting in image segmentation [1] shown in 3rd and 4th of Fig. 3.



Algorithm 1 Z-coding.

```
Require: Image $I$ of size $J \times K$: texel of size $O$;$N$ is number of pixels arround center pixel $J_0$
Ensure: Z-coded image $Z$ of size $J \times K$ of input image $I$
Initialization  $Z \leftarrow \emptyset$;  $j = 2, k = 2$
for  $j = 2$ to $J - 1$ do
    for  $k = 2$ to $K - 1$ do
        $J_0 = I(j, k)$;                ▷ central pixel of the texel
        $texel$ ← Intensity Values of $N$ neighbouring pixels arround $J_0$
        s ← 0
        $\mathcal{S}^0$ ← zeckendorf ($J_0$)
        for  $i = 1$ to $N$ do
            $\mathcal{S}$ ← zeckendorf (texel($i$));
            dummy ← $\mathcal{S}^0$ $op$ $\mathcal{S}$   ▷ $op$ is intersection or set difference operator
            if (dummy = $\emptyset$)) then
                $s(i) \leftarrow J_0$   ▷ $J_0$ for quantization and 0 for contours
            else
                $s(i) \leftarrow \max(\text{dummy})$        ▷ $max$ for quantization and $sum$ for contours
            end if
        end for
        $Z(j, k) \leftarrow \max(\mathbf{s})$
    end for
end forreturn $Z$
Function zeckendorf($x$)
Decomposes an integer $x$ as a sequence of Fibonacci numbers
EndFunction
```

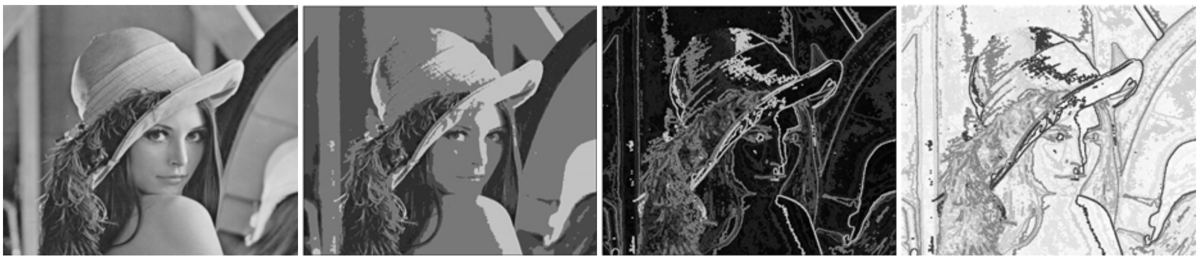


Fig. 3. Z-coded images using Zeckendorff representation – 1st row shows original images, 2nd row shows quantized images obtained by applying intersection operator, 3rd row contains ultrametric contours obtained by applying set-difference operator and last row shows complemented results of 3rd row.

3. Simulations

3.1. Architectures

In these experiments we compare architectures containing the transfer block to a U-Net model [13]. The U-Net was chosen because it is the archetype of modern convolutional networks used for bio-medical image segmentation tasks and achieved good performance in many applications. To prove that the tasks are not too simple and that the transfer layer is responsible for the good results, a simple reference network with two convolutional layers is included. For evaluating the results we not only use the loss but network with two convolutional layers is included. For evaluating the results we not only use the loss but also the Dice coefficient [**?**] which is the standard measure for segmentation quality.

Convolution layers are chosen to keep the output size the same as the input size by padding the input image with zeros. Also all architectures are followed by a SoftMax layer (not explicitly stated below) and cross entropy is used as loss function. The loss is minimised using Adam [5]. The best learning rate for each architecture has been determined experimentally.

U-Net The U-Net [13] (Fig. 4a) consists mainly of a feature extraction pyramid followed by an expanding path which upsamples the features to the space of the original image. A special feature of the U-Net are its skip connections which allow it to preserve fine grained details. Our implementation is almost identical to the original paper, with only two differences. First for the input image more than one channel is allowed. Second the crop operation is not implemented, thus the output image of this U-Net implementation has the same size as the input image.

Z-Net is a CNN which includes a Z-pooling layer instead of max-pooling (Fig. 4b). It has a first 5×5 convolutional layer, then a transfer layer, a second 5×5 convolutional layer and then a 1×1 convolutional layer for the classification, as recommend by [14].

Double Z-Net This sample architecture shows how to chain layers together in order to create deeper architectures (Fig. 3c). It starts with a 5×5 convolutional layer followed by a Z-pooling layer, another 5×5 convolutional layer then another Z-pooling layer and a final 5×5 convolution layer followed by a 1×1 convolution for classification. All other parameters are identical to the Z-Net.

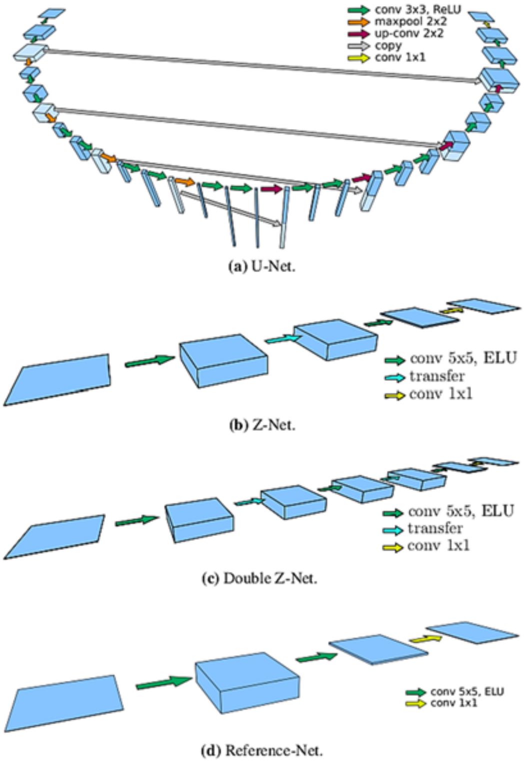
Reference Net (R-Net) is a reference architecture (Fig. 4d) which is used to demonstrate that it is our transfer layer which is responsible for our results and not the fact that two convolutional layers are chained together. It represents the most basic convolution neural network architecture. It suffers greatly from a small receptive field. As expected our first experiment showed that it has trouble with objects that are larger than the convolution kernels.

All architectures are followed by a SoftMax layer (not explicitly stated below) and cross entropy is used as the loss function. For the basic CNN architectures the first convolution increases the number of feature maps to 70, which is kept constant until the last 5×5 convolution which reduces the number of feature maps to 7 and the final 1×1 convolution reduces the number of feature maps to the number of classes. The feature maps all have the same size as the input image, *i.e.* no downsampling occurs.

The following network architectures are implemented in Pytorch on a Tesla VT100 CPU @3.60 GHz with 64 GB of RAM. The best learning rate for each architecture has been determined experimentally.

3.2. MICCAI BraTS Dataset

We chose a real data set which is renowned for its difficulty. The brain tumor segmentation (BraTS) [9] challenge is a recurring challenge attached to the MICCAI Conference. Each year the segmentation results become better, but the problem is an ongoing research. For this experiment we use the high grade glioma part of the BraTS 2017 data-set. It contains multi-modal MRI of 210 patients which were manually segmented my experts, *i.e.* a ground truth is available. On these image three different classes have to be segmented from the background. The enhancing tumor, the necrotic and nonenhancing tumor and as a third class the peritumoral edema.



(a) U-Net.

(b) Z-Net.

(c) Double Z-Net.

(d) Reference-Net.

Fig. 4. Z-net architectures.

This makes it an ideal real data-set for supervised learning of a multi-class segmentation task.

We divided the data-set into 100 patients for learning, 4 for validation and left 106 aside as test set. As we just wanted to demonstrate a concept we never used the test set in the end. Four different MRI modalities (compare Fig. 5a-d) are available, which means that the input image has four channels. The images were normalised to the mean value of healthy tissue, *i.e.* the intensity values were divided by the intensity value of the highest peak in the histogram of brain tissues. As the whole MRI is too large to process in one step it was broken down into patches of size 64×64×10 by the following method: for each patient one patch is taken from the geometrical middle of the tumor (guaranteed tumor patch). Then randomly placed patches with their center in the brain volume are sampled from that patient where the center of each new patch may not be in an already sampled area. Patches are then generated until the whole brain is sampled or a number of 100 patches is reached. This procedure is repeated for each patient. The batches for the training are generated as follows: each odd sample of the batch is the guaranteed tumor patch of a random patient and the following even sample is a random patch of the same patient.

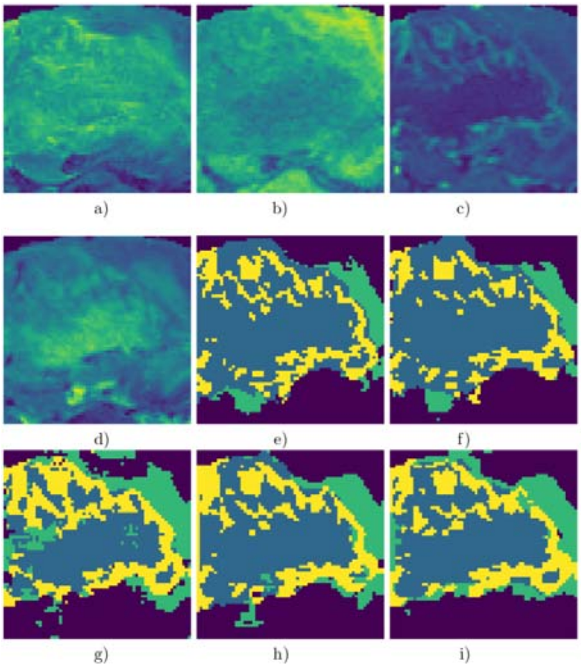


Fig. 5. One patch out of the BraTs validation-set. The input image: a)-d) consists of the MRI modalities Flair a), T1 b), contrast enhanced T1 c) and T2 d). The ground truth is seen in e). The following are the predictions of the five networks: U-Net f), double Z-Net g), Z-Net h), R-Net i).

This sampling scheme guarantees that each batch shows tumor as well as non-tumor regions and effectively combats class imbalance. Purely random sampling would have a chance to generate entire batches which show only background which in turn would slow down the learning process. We use a batch size of four patches per batch, which means that after 50 patches the guaranteed tumor patch of all 100 training patients have been seen. This may be considered an epoch. That all randomly placed patches are seen during training is not guaranteed. For clarification: the input to all tested networks are 2D images, none works on 3D in the current implementation. That means one patch of 64×64×10 is actually 10 samples images of size 64×64. So one batch contains 40 samples taken from two patients.

Table 1. Network configurations for the BraTS experiment and their final dice score on the training and validation set.

Model	U-Net	double Z-Net	Z-Net	R-Net
Kernel Size	3	5	5	5
Parameters	31,032,516	141,929	19,359	19,359
Learning Rate	0.002	0.005	0.002	0.002
Training Speed τ	10,737	13,322	28,761	48,767
Training Dice	**0.97**	0.82	0.96	0.64
Validation Dice	0.81	0.51	**0.89**	0.49

4. Conclusions

The practice of feature extraction has been unchallenged for three decades. Z-pooling can improve the pooling task compared to max-pooling and thus improve any CNN learning task.

References

[1]. P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, New York, NY, USA, 17-22 June, 2006, p. 182.

[2]. I. Arel, D. Rose, and T. Karnowski. Deep machine learning a new frontier in artificial intelligence research, *IEEE Comp. Int. Mag.*, Vol. 5, 2010, pp. 13-18.

[3]. D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUS), in *Proceedings of the 4th International Conference on Learning Representations*, San Juan, Puerto Rico, 2015.

[4]. G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals, and Systems (MCSS)*, Vol. 2, Issue 4, December 1989, pp. 303-314.

[5]. D. P. Kingma, J. Ba, ADAM: A method for stochastic optimization, in *Proceedings of the 3rd International Conference on Learning Representations (ICLR'15)*, San Diego, CA, USA, May 7-9, 2015.

[6]. S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06),* Vol. 2, Feb 2006, pp. 2169-2178.

[7]. Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, W. Hubbard, Handwritten digit recognition: applications of neural network chips and automatic learning, *IEEE Communications Magazine*, Vol. 27, Issue 11, 1989, pp. 41-46.

[8]. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 936-944.

[9]. B. H. Menze, The multimodal brain tumor image segmentation benchmark (BRATS), *IEEE Transactions on Medical Imaging*, Vol. 34, Issue 10, 2014, pp. 1993-2024.

[10]. T. Ojala, M. Pietikainen, D. Harwood. A comparative study of texture measures with classification based on feature distributions, *Pattern Recognition*, Vol. 29, 1996, pp. 51-59.

[11]. M. Pietikainen, A. Hadid, G. Zhao, T. Ahonen, Computer vision using local binary patterns, in Computer Imaging and Vision, Vol. 40, *Springer*, 2011.

[12]. M. Ranzato, F. J. Huang, Y.-L. Boureau, Y. LeCun, Unsupervised learning of invariant feature hierarchies with applications to object recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007, pp. 1-8.

[13]. O. Ronneberger, P. Fischer, T. Brox. U-net: Convolutional networks for biomedical image segmentation, in *Proceedings of the International Conference on Medical Image Computing and*

Computer Assisted Intervention (MICCAI'15), 2015, pp. 234-241.

[14]. J. T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, *arXiv Preprint*, arXiv:1412.6806, 2015.

[15]. D. Yu, H. Wang, P. Chen, Z. Wei. Mixed pooling for convolutional neural networks, in *Proceedings of the International Conference on Rough Sets and Knowledge Technology (RSKT'14)*, 2014, pp. 364-375.

[16]. R. Zabih, J. Woodfill, Non-parametric local transforms for computing visual correspondence, in *Proceedings of the European Conference on Computer Vision (ECCV'94)*, 1994, pp. 151-158.

[17]. E. Zeckendorf, Repre´sentation des nombres naturels par une somme de nombres de fibonacci ou de nombres de lucas, *Bull. Soc. Roy. Sci. Liege*, Vol. 41, 1972, pp. 179-182.

(004)

3D Volume Reconstruction of Brain Images for Common Diseases of Aging

S. Moldovanu [1, 3], L. Pana [2, 3] and L. Moraru [2, 3]
[1] University of Dunarea de Jos, Department of Computer Science and Information Technology, Faculty of Automation, Computers, Electrical Engineering and Electronics, 2 Științei Str., 800146, Galati, Romania
[2] University of Dunarea de Jos, Department of Chemistry, Physics & Environment, Faculty of Sciences and Environment, Dunarea de Jos University of Galati, 47 Domneasca Str., 800008 Galati, Romania
[3] The Modelling & Simulation Laboratory, Dunarea de Jos University of Galati, 111 Domneasca Str., 800102 Galati, Romania
Tel.: +40 336 130 236, fax: + +40 236 470 905
E-mail: luminita.moraru@ugal.ro

Summary: Multiple sclerosis (MS) and Alzheimer's disease (AD) are common diseases of aging. Although there is not much knowledge about the coexistence of MS with AD there are case reports on the co-existence of MS and AD in the same person. Diagnosis of structural brain changes is essential for treatment. 3D volume reconstruction is an approach which can be used to assess the structural brain changes on magnetic resonance imaging (MRI) generated by these neurological conditions. In this study, we integrated the multimodal MRI brain images (T2w and PDw) with filtering operations using a convolutional neural network, median filters and image segmentation for 3D volume reconstruction. The brain tissue segmentation is performed based on a k-means algorithm. Finally, the brain anatomical knowledge on the cerebrospinal fluid (CSF), white matter (WM) and gray matter (GM) will be embedded in a volume similarity metric to assess the filtering efficiency.

Keywords: MRI, Convolutional neural network filter, Median filter, Peak signal-to-noise ratio, 3D volume reconstruction.

1. Introduction

Brain MRI is an important tool used to diagnose many neurological diseases. AD is an irreversible neurodegenerative disease that causes the brain atrophy. The number of patients suffering from it is expected to increase dramatically by 2050 [1]. MS is an inflammatory and demyelinating autoimmune disease which manifests through the brain lesion volumes and induces clinical disability and cognitive impairment [2]. The investigation of both diseases is challenging due to their diffuse appearance and large intensity variations on MRI. Image segmentation leads to an objective measurement of brain tissues volume and aids both the diagnosis and treatment planning. A proper measurement on GM and WM atrophy has an important impact on the patient treatment and is a strong motivation for a correct segmentation of the brain tissues and 3D reconstruction. The artifacts that affect the brain images are usually produced by magnetic field inhomogeneities and subject movement; they lead to a reduced peak signal-to-noise ratios (PSNR). In order to reduce noise while preserving the edges the convolutional neural network (CNNF) and median filters (MF) were used. These operations remove the Rician noise and enhance the quality of the MR images [3, 4]. In order to achieve an optimal brain tissue segmentation, a k-means algorithm is operated. The K-means method is most suitable for noisy images, it is easy to implement, allows a facile interpretation of the clustering results and it is not expensive in terms of computational cost.

2. Materials and Methods

Four patients with clinically definite AD and MS were studied. The MR images belong to the public Harvard's Whole Brain Atlas database (https://www.med.harvard.edu/aanlib/home.html). For each patient, two stacks that contains 20 T2w and 20 PDw images, respectively, were analyzed. We artificially introduced intensity inhomogeneity (linear gradients with various orientations) as data augmentation, so the number of analyzed imaged is doubled. The image stacks are registered to the same coordinate space to help the 3D volume reconstruction. The programming environment is MATLAB R2017a. An image I(x, y) is viewed as a matrix and a convolution product is performed between the matrix I and a kernel matrix of size 3×3. This kernel is sliding across the image. The result of this operation is the 'spike' value from each convolution [5]. The k-means algorithm is used to segment the brain tissue into three classes, namely CF, WM and GM, by using information from the multichannel 2D (T2w and PDw) images [6].

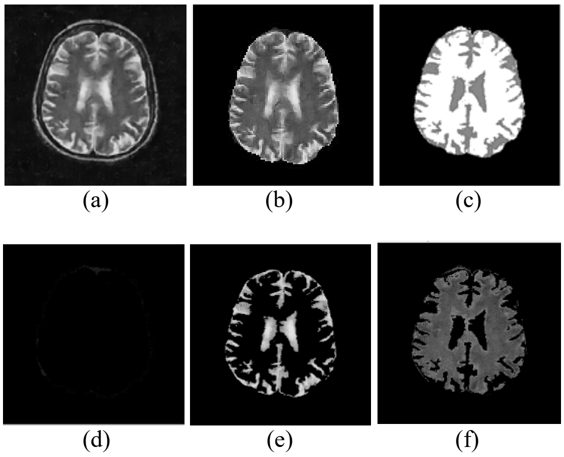
Volume similarity (VS) compares the volumes of two segmented tissues

$$VS = 1 - \frac{|Raw - Fil|}{Raw + Fil} \ (\%),$$

where *Raw* denotes the brain tissues volume in raw slice segmentation and *Fil* is determined for filtered slices.

3. Results and Discussions

We employ the proposed method on 320 MR slices and present the experimental results. Brain tissues alone do not represent a descriptor useful for diagnosis. To generate a correct automatic brain tissues segmentation, we present a comparative analysis between the results of the 3D brain reconstruction using raw and filtered images. Fig. 1 displays an example of a filtered PDw image with CNNF, a skull stripped image and the results of segmentation. K-means identified all three classes in a proper manner. Afterwards, the segmented CF, WM and GM slices are processed with the free software ImageJ (https://imagej.nih.gov/ij/download.html) for 3D reconstruction. Fig. 2 shows a visualization of 3D reconstruction results for an AD (type PDw) image.


(a) (b) (c)

(d) (e) (f)

Fig. 1. Example of segmentation from one subject (AD – PDw image) and brain tissues extraction results. a) CNNF filtered image; b) skull striping; c) clustering results; d) CSF cluster; e) WM cluster; f) GM cluster.

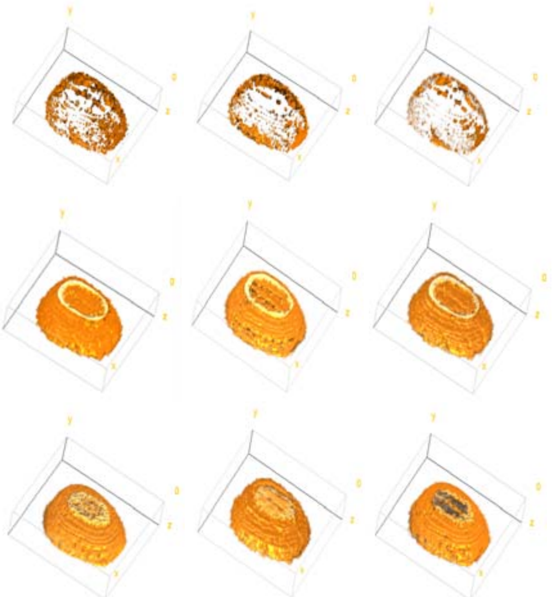


Fig. 2. 3D reconstruction of the brain tissue for image in Fig. 1. First line is for CSF; second line for WM; and third line for GM.

Table 1 shows comparison of PSNR results for filtered images as well as for MR image type and diseases. The quantitative validation is presented in Table 2 in terms of the 3D volume similarity. The differences in the CSF volumes in most of the images are very small and the computed volume does not exceed $0.5 \cdot 10^5$ vx. For WM volumes, the CNNF provides a large deviation for AD and PDw images while, the median filters provides a large deviation for AD disease and both image types. For GM volumes, the results provided by T2w images are almost similar for both filters while PDw images show a higher variability. The CSF was challenging to segment, as the tissue boundaries are missing.

Table 1. PSNR average values for filtered images.

Features	Type of Image	CNNF	MF
PSNR [dB]/ AD	T2w	23.29	33.32
	PDw	23.77	33.32
PSNR [dB]/ MS	T2w	23.48	35.27
	PDw	23.59	35.11

Table 2. Volume similarity VS(%). Data are presented in the sequence of CSF/WM/GM

	AD		MS	
	T2w	PDw	T2w	PDw
CNNF filter	0.9945/ 0.9969/ 0.9949	0.9784/ 0.9102/ 0.8931	0.9836/ 0.9828/ 0.9872	0.9990/ 0.9566/ 0.8665
MF filter	0.9831/ 0.8922/ 0.9194	0.9516/ 0.8855/ 0.8099	0.9427/ 0.9262/ 0.9264	0.9784/ 0.9377/ 0.8774

As MR imaging is an important tool in evaluation of stage or dynamic of neurological diseases, the proposed methodology allowing quantitative analysis using existing MR data.

4. Conclusions

We have presented a method for reliable tissue delineation and 3D volumetric reconstruction of CSF, WM and GM brain tissues which is strongly conditioned by the preprocessing stage. Overall, an accurate 3D reproducibility of brain tissues has been provided by the median filter, for both clinical disabilities. As a future work we plan to used different filters and segmentation techniques to improve the VS for CSF especially.

Acknowledgement

The project was financed by Dunarea de Jos University of Galati, Grant CNFIS-FDI2021-0443 – Măsuri active de creștere și eficientizare a capacității de cercetare, dezvoltare, Innovare și transfer technologic in Universitatea Dunărea de Jos din Galați – CEREX-UDJG_2021.

References

[1]. N. E. A. Khalid, S. Ibrahim, P. N. M. M. Haniff, MRI brain abnormalities segmentation using K-Nearest Neighbors (k-NN), *International Journal on Computer Science and Engineering (IJCSE),* Vol. 3, Issue 2, 2011, pp. 980-990.

[2]. S. Datta, P. A. Narayana, A comprehensive approach to the segmentation of multichannel three-dimensional MR brain images in multiple sclerosis, *NeuroImage: Clinical*, Vol. 2, 2013, pp. 184-196.

[3]. S. Albawi, O. Bayat, S. Al-Azawi, O. N. Ucan, Social touch gesture recognition using convolutional neural network, *Computational Intelligence and Neuroscience*, Vol. 2018, 2018, 6973103.

[4]. J. Yang, J. Fan, D. Ai, S. Zhou, S. Tang, Y. Wang, Brain MR image denoising for Rician noise using pre-smooth non-local means filter, *BioMedical Engineering OnLine*, Vol. 14, 2015, pp. 1-20.

[5]. S. Cohen, Artificial intelligence and deep learning in pathology, Chapter 2, in The Basics of Machine Learning: Strategies and Techniques, *Elsevier*, 2021, pp. 13-40.

[6]. P. Christina, World Alzheimer Report 2018 – The State of the Art of Dementia Research: New Frontiers, https://www.alzint.org/resource/world-alzheimer-report-2018/

(006)

The Matrix-CFAR via Total Bregman Divergence Medians in Signal Detection

Y. Ono, L. Peng and H. Sato
Department of Mechanical Engineering, Keio University, Hiyoshi 3-14-1, Yokohama, 223-8522, Japan
Tel.: +81 (0)45-566-1525
E-mail: yuu555yuu@keio.jp

Summary: In signal detection, the received signal usually contains not only information of the target but also that of nonhomogeneous clutter. For discrimination between the target and clutter, the constant false alarm rate (CFAR) method has been widely used. To overcome its potential shortcoming, the matrix-CFAR was proposed by the combination of the CFAR and matrix information geometry. It assumes each sample data to be modeled as the clutter covariance matrix which is a Hermitian positive-definite matrix. In this study, we define medians associated with the total Bregman divergence (TBD) and apply TBD medians to detection problems. Their performance advantages are shown through numerical simulations.

Keywords: Matrix-CFAR, Total Bregman divergence, Riemannian manifold, Signal detection, Nonhomogeneous clutter.

1. Introduction

The constant false alarm rate (CFAR) is a well-known method conventionally used to remove clutter. A popular CFAR is based on Fourier transform and uses the Doppler spectral density to distinguish targets from clutter. However, this method is rather weak when the Doppler spectra are mixed. One of the popular statistical methods for discriminating between the target signal from clutter is to estimate the covariance matrix and to test the likelihood ratio. There are mainly two methods for estimating the clutter covariance matrix: the sample covariance matrix using the maximum likelihood estimation and the Bayesian method using a prior information. However, it is difficult to obtain sufficient detection performance with these methods because it is difficult to accurately capture the statistical information of clutter and the available homogeneous data is limited because the observation data is usually contaminated.

To compensate for these drawbacks of the conventional methods, the matrix-CFAR was proposed, which needs no a prior information, but makes use of the covariance matrix of the signals [1] [2]. It is based on the fact that the covariance matrix that represents the correlation of the signal is a Hermitian positive-definite (HPD) matrix which is element of HPD manifold. This method was applied to target detection in high frequency X-band radar clutter [3] and drone detection [4]. Recently, the matrix-CFAR with various divergence functions has been proposed, and its detection performance and robustness to outliers were shown better compared with the geodesic distance associated to the affine-invariant Riemannian metric (AIRM) [5, 6]. In this study, the total Bregman divergence (TBD) defined in the HPD manifold is applied to signal detection. In particular, we study the TBD medians associated with the total square loss (TSL), the total von-Neumann (TVN) divergence and the total log-determinant (TLD) divergence. We show their performance advantages and robustness numerically compared with the TBD means and the Riemannian distance (RD) mean.

2. The Matrix-CFAR via TBD Medians

2.1. The matrix-CFAR

The detection problem can be modeled as the following binary hypothesis testing that the null hypothesis H_0 represents that the received data $\boldsymbol{x}$ is only clutter $\boldsymbol{c}$ and the alternative hypothesis H_0 represents that $\boldsymbol{x}$ contains both the known target signal $\boldsymbol{p}$ and clutter $\boldsymbol{c}$, namely

$$\begin{cases} H_0 : \boldsymbol{x} = \boldsymbol{c}, \\ H_1 : \boldsymbol{x} = b\boldsymbol{p} + \boldsymbol{c}, \end{cases} \tag{1}$$

where b denotes the amplitude coefficient which is unknown and complex scalar-valued. The observation data $\boldsymbol{x}$ and the known target signal $\boldsymbol{p}$ denoting the steering vector are defined as follows,

$$\begin{aligned} \boldsymbol{x} &= [x_0, x_1, \dots, x_{N-1}]^{\mathrm{T}} \in \mathbb{C}^{\boldsymbol{N}}, \\ \boldsymbol{p} &= \frac{1}{\sqrt{N}} \begin{bmatrix} 1, \exp(-\mathrm{i}2\pi f_d), \dots, \\ \exp\left(-\mathrm{i}2\pi f_d(N-1)\right) \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{2}$$

where $\mathbb{C}^N$ represents the N dimensional complex space, f_d is the normalized Doppler frequency, i is the imaginary unit and $(\cdot)^{\mathrm{T}}$ denotes the transpose of matrices or vectors.

Conventional sample covariance matrix (SCM) estimator is derived by a set of secondary data $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m\}$ and the maximum likelihood estimation of the circularly symmetric complex Gaussian distribution. It is given by [7]

$$\boldsymbol{R}_{SCM} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{H}}, \boldsymbol{x}_i \in \mathbb{C}^{\boldsymbol{N}}, \tag{3}$$

where $(\cdot)^{\mathrm{H}}$ denotes the conjugate transpose of matrices or vectors. This popular estimator has been utilized in the generalized likelihood ratio test detectors as [8]

$$\frac{\left|\boldsymbol{x}_k^{\mathrm{H}} \boldsymbol{R}_{SCM}^{-1} \boldsymbol{s}\right|^2}{\boldsymbol{s}^{\mathrm{H}} \boldsymbol{R}_{SCM}^{-1} \boldsymbol{s}} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma \tag{4}$$

In the matrix-CFAR, to determine whether $\boldsymbol{x}$ includes $\boldsymbol{p}$ or not, we model the data by the covariance matrix as a Toeplitz HPD matrix. The covariance matrix of the observation data is defined as,

$$\boldsymbol{R} = \begin{pmatrix} r_0 & \bar{r}_1 & \cdots & \bar{r}_{N-1} \\ r_1 & r_0 & \cdots & \bar{r}_{N-2} \\ \vdots & \ddots & \ddots & \vdots \\ r_{N-1} & \cdots & r_1 & r_0 \end{pmatrix}, \tag{5}$$

where the component

$$\begin{aligned} r_l = \mathrm{E}[x_i \bar{x}_{i+l}], 0 \le l \le N-1, \\ 1 \le i \le N-l-1 \end{aligned} \tag{6}$$

is the l-th correlation coefficient of data, $\mathrm{E}[\cdot]$ denotes the statistical expectation and $\bar{r}_l$ is the conjugate of r_l. Because of the ergodicity of stationary Gaussian process, the component, r_l can be approximated by observation data

$$r_l = \frac{1}{N}\sum_{i=0}^{N-1-l} x_i \bar{x}_{i+l}, 0 \le l \le N-1 \tag{7}$$

The covariance matrix of clutter is estimated as $\boldsymbol{R}_g$ by the observations $\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_m\}$, then the problem can be modeled as

$$\begin{cases} H_0: \boldsymbol{R}_{\mathrm{CUT}} = \boldsymbol{R}_g, \\ H_1: \boldsymbol{R}_{\mathrm{CUT}} \neq \boldsymbol{R}_g, \end{cases} \tag{8}$$

where the matrix $\boldsymbol{R}_{\mathrm{CUT}}$ is the covariance matrix of the observation in the cell under test. By the threshold γ, the signal detection is modeled as discriminating $\boldsymbol{R}_{\mathrm{CUT}}$ from $\boldsymbol{R}_g$,

$$d\left(\boldsymbol{R}_g, \boldsymbol{R}_{\mathrm{CUT}}\right) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \tag{9}$$

where $d\left(\boldsymbol{R}_g, \boldsymbol{R}_{\mathrm{CUT}}\right)$ is the difference between $\boldsymbol{R}_{\mathrm{CUT}}$ and $\boldsymbol{R}_g$, such as the Riemannian distance, divergence functions, and so on. The process of the matrix-CFAR is showed in Fig. 1.

2.2. Total Bregman Divergence and TBD Medians

The Bregman divergence for matrices was introduced in [9] and we extended the idea to the total Bregman divergence for matrices in [5].

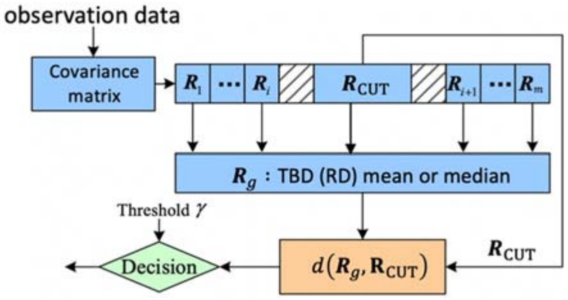


Fig. 1. The process of the matrix-CFAR.

The total Bregman divergence for two matrices $\boldsymbol{X}, \boldsymbol{Y} \in GL(N, \mathbb{C})$, which denotes the general linear group of $N \times N$ invertible matrices, is defined by

$$\delta_F(\boldsymbol{X}, \boldsymbol{Y}) = \frac{F(\boldsymbol{X}) - F(\boldsymbol{Y}) - \langle \nabla F(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y} \rangle}{\sqrt{1 + \|\nabla F(\boldsymbol{Y})\|^2}}, \tag{10}$$

where $\|\cdot\|$ is the Frobenius norm, the Frobenius metric is given by the trace operator $\langle \boldsymbol{A}, \boldsymbol{B} \rangle := \mathrm{tr}(\boldsymbol{A}^H \boldsymbol{B})$, and $F(\boldsymbol{X})$ is a differentiable and strictly convex function in $GL(N, \mathbb{C})$. Next, we are going to introduce some well-known convex functions in HPD manifolds.

Definition 1. Let $F(\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{X}\|^2$, and $\nabla F(\boldsymbol{X}) = \boldsymbol{X}$. The total square loss (TSL) is defined by

$$\delta_F(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2}\frac{\|\boldsymbol{X} - \boldsymbol{Y}\|^2}{\sqrt{1 + \|\boldsymbol{Y}\|^2}} \tag{11}$$

Definition 2. Let $F(\boldsymbol{X}) = -\ln\det\boldsymbol{X}$, and $\nabla F(\boldsymbol{X}) = -\boldsymbol{X}^{-\mathrm{H}}$. The total log-determinant (TLD) divergence is defined by

$$\delta_F(\boldsymbol{X}, \boldsymbol{Y}) = \frac{1}{2}\frac{\ln\det(\boldsymbol{Y}\boldsymbol{X}^{-1}) - \mathrm{tr}(\boldsymbol{Y}\boldsymbol{X}^{-1}) - N}{\sqrt{1 + \|\boldsymbol{Y}^{-\mathrm{H}}\|^2}} \tag{12}$$

Definition 3. Let $\boldsymbol{X}, \boldsymbol{Y}$ be invertible and have no eigenvalues on the negative real line and $F(\boldsymbol{X})$ is

$$(\boldsymbol{X}) = \mathrm{tr}(\boldsymbol{X}\,\mathrm{Log}\,\boldsymbol{X} - \boldsymbol{X}) \tag{13}$$

Its gradient [10] is

$$\nabla F(\boldsymbol{X}) = (\mathrm{Log}\,\boldsymbol{X})^{\mathrm{H}}, \tag{14}$$

where $\mathrm{Log}\,\boldsymbol{X}$ denotes the principal logarithm [11]. Then the total von-Neumann (TVN) divergence is given by

$$\delta_F(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\mathrm{tr}(\boldsymbol{X}(\mathrm{Log}\,\boldsymbol{X} - \mathrm{Log}\,\boldsymbol{Y}) - \boldsymbol{X} + \boldsymbol{Y})}{\sqrt{1 + \|(\mathrm{Log}\,\boldsymbol{Y})^{-\mathrm{H}}\|^2}} \tag{15}$$

In the following, we are going to introduce TBD medians. Let $F(\boldsymbol{R})$ be a differentiable and strictly convex function and δ_F be the corresponding TBD. The TBD median of m HPD matrices $\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_m\}$ is defined by

$$\overline{\boldsymbol{R}} = \underset{\boldsymbol{R}}{\operatorname{argmin}} \frac{1}{m}\sum_{i=1}^{m}\left(\delta_F(\boldsymbol{R},\boldsymbol{R}_i)\right)^{\frac{1}{2}} \quad (16)$$

Proposition 4. If the TBD median (16) exists, then it solves the algebraic equation (17).

$$\nabla F(\boldsymbol{R}) = \sum_{i=1}^{m} \frac{\nabla F(\boldsymbol{R}_i)}{\{\delta_F(\boldsymbol{R},\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{R}_i)\|^2}} \cdot \left(\sum_{j=1}^{m}\frac{1}{\{\delta_F(\boldsymbol{R},\boldsymbol{R}_j)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{R}_j)\|^2}}\right)^{-1} \quad (17)$$

Proof. Denote $L(\boldsymbol{R})$ as the objective function,

$$L(\boldsymbol{R}) = \frac{1}{m}\sum_{i=1}^{m}\left(\delta_F(\boldsymbol{R},\boldsymbol{R}_i)\right)^{\frac{1}{2}} \quad (18)$$

The gradient of function $L(\boldsymbol{R})$ can be immediately calculated and we have

$$\nabla L(\boldsymbol{R}) = \sum_{i=1}^{m}\frac{\nabla F(\boldsymbol{R})-\nabla F(\boldsymbol{R}_i)}{\{\delta_F(\boldsymbol{R},\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{R}_i)\|^2}} \quad (19)$$

Letting $\nabla L(\overline{\boldsymbol{R}}) = 0$, we get the result. ■

However, it is difficult to solve the equation (17) analytically, so we seek for its numerical solutions using the fixed-point algorithm [12]. The fixed-point algorithm is a numerical method for solving fixed-point problems.

Proposition 5. The TSL median $\overline{\boldsymbol{R}}_{TSL}$, the TLD median $\overline{\boldsymbol{R}}_{TLD}$, and the TVN median $\overline{\boldsymbol{R}}_{TVN}$, of m HPD matrices $\{\boldsymbol{R}_1,\boldsymbol{R}_2,\dots,\boldsymbol{R}_m\}$ can be calculated by the fixed-point algorithms (20), (21), (22), respectively.

Proof of Propositions 5 is omitted.

$$\overline{\boldsymbol{R}}_{t+1} = \sum_{i=1}^{m}\frac{\boldsymbol{R}_i}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\boldsymbol{R}_i\|^2}} \cdot \left(\sum_{j=1}^{m}\frac{1}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_j)\}^{\frac{1}{2}}\sqrt{1+\|\boldsymbol{R}_i\|^2}}\right)^{-1}, \quad (20)$$

$$\overline{\boldsymbol{R}}_{t+1} = \left\{\sum_{i=1}^{m}\frac{\boldsymbol{R}_i^{-1}}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_j)\}^{\frac{1}{2}}\sqrt{1+\|\boldsymbol{R}_j^{-1}\|^2}} \cdot \left(\sum_{i=1}^{m}\frac{1}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_j)\}^{\frac{1}{2}}\sqrt{1+\|\boldsymbol{R}_j^{-1}\|^2}}\right)^{-1}\right\}^{-1}, \quad (21)$$

$$\overline{\boldsymbol{R}}_{t+1} = \exp\left\{\sum_{i=1}^{m}\frac{\operatorname{Log}\boldsymbol{R}_i}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\operatorname{Log}\boldsymbol{R}_i\|^2}} \cdot \left(\sum_{i=1}^{m}\frac{1}{\{\delta_F(\overline{\boldsymbol{R}}_t,\boldsymbol{R}_j)\}^{\frac{1}{2}}\sqrt{1+\|\operatorname{Log}\boldsymbol{R}_j\|^2}}\right)^{-1}\right\} \quad (22)$$

3. Main Results

3.1. Numerical Simulations

To confirm the performance of the matrix-CFAR via TBD medians, the following numerical simulations are conducted. For the comparison, we also show the results using the RD mean induced from the AIRM, and the TBD means. In the simulations, the false alarm rate P_{fa} is 10^{-3}, and the dimension of the signal, N, and the number of the observation, m, are both 8. The threshold γ is set to $100/P_{fa}$, and the probability of detection P_d is derived by 2000 independent trails. We consider the model of clutter as the behavior of K-distribution with shape parameter $\alpha = 4$ and scale parameter $\beta = 3$. Fig. 2 shows that the TVN median has the best performance, the TLD median behaves better than the TLD mean, but surprisingly the TSL median is worse than the TSL mean.

3.2. Robust Analysis

In real detections, the received signal may also include outliers, and hence robustness about estimate of the covariance matrix is crucial. To evaluate the robustness, we define the influence function. Assume $\overline{\boldsymbol{R}}$ are the TBD medians (or TBD means, RD mean) of m HPD matrices $\{\boldsymbol{R}_1,\boldsymbol{R}_2,\dots,\boldsymbol{R}_m\}$, and $\widehat{\boldsymbol{R}}$ are the TBD medians (or TBD means, RD mean) of the mixed HPD data including outliers $\{\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_n\}$. By thinking of the outliers as a perturbation $\varepsilon(\varepsilon \ll 1)$ of the mean or median, $\widehat{\boldsymbol{R}}$ can be derived as

$$\widehat{\boldsymbol{R}} = \overline{\boldsymbol{R}} + \varepsilon\boldsymbol{H}(\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_n) + O(\varepsilon^2), \quad (23)$$

and the function

$$h(\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_n) := \frac{\|\boldsymbol{H}(\boldsymbol{P}_1,\boldsymbol{P}_2,\dots,\boldsymbol{P}_n)\|}{\|\overline{\boldsymbol{R}}\|} \quad (24)$$

is defined as the influence function. In order to derive influence functions for the TBD medians, we define $G(\boldsymbol{R})$ as the objective function for mixed data,

$$G(\boldsymbol{R}) = \frac{1-\varepsilon}{m}\sum_{i=1}^{m}\left(\delta_F(\boldsymbol{R},\boldsymbol{R}_i)\right)^{\frac{1}{2}} + \\ +\frac{\varepsilon}{n}\sum_{j=1}^{n}\left(\delta_F(\boldsymbol{R},\boldsymbol{R}_j)\right)^{\frac{1}{2}}, \quad (25)$$

$$\frac{1-\varepsilon}{m}\sum_{i=1}^{m}\frac{\nabla F(\widehat{\boldsymbol{R}})-\nabla F(\boldsymbol{R}_i)}{\{\delta_F(\widehat{\boldsymbol{R}},\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{R}_i)\|^2}}+\frac{\varepsilon}{n}\sum_{j=1}^{n}\frac{\nabla F(\widehat{\boldsymbol{R}})-\nabla F(\boldsymbol{P}_j)}{\{\delta_F(\widehat{\boldsymbol{R}},\boldsymbol{P}_j)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{P}_j)\|^2}} = 0, \tag{26}$$

$$\frac{1}{m}\sum_{i=1}^{m}\frac{1}{\sqrt{1+\|\nabla F(\boldsymbol{R}_i)\|^2}}\frac{d}{d\varepsilon}\bigg|_{\varepsilon=0}\frac{\nabla F(\widehat{\boldsymbol{R}})-\nabla F(\boldsymbol{R}_i)}{\{\delta_F(\widehat{\boldsymbol{R}},\boldsymbol{R}_i)\}^{\frac{1}{2}}}+\frac{1}{n}\sum_{j=1}^{n}\frac{\nabla F(\overline{\boldsymbol{R}})-\nabla F(\boldsymbol{P}_j)}{\{\delta_F(\overline{\boldsymbol{R}},\boldsymbol{P}_j)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{P}_j)\|^2}} = 0 \tag{27}$$

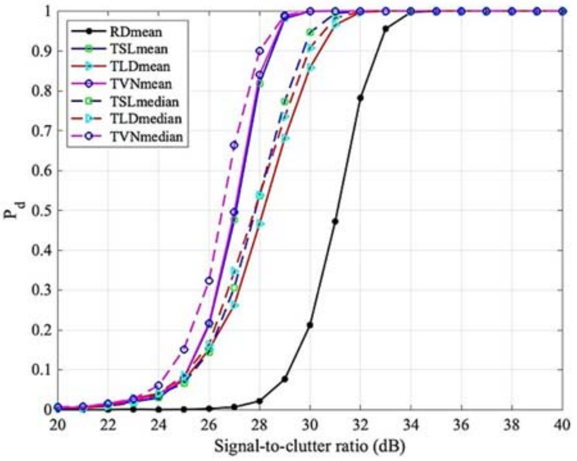


Fig. 2. P_d versus the signal-to-clutter ratio.

Because $\widehat{\boldsymbol{R}}$ are the TBD medians of the contaminated data of HPD matrices $\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_m\}$ and outliers $\{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n\}$, we have $\nabla G(\widehat{\boldsymbol{R}}) = 0$ which is the equation (26). Then we differentiate $\nabla G(\widehat{\boldsymbol{R}}) = 0$ about a perturbation ε at the point of $\varepsilon = 0$ and obtain the equation (27), considering that $\overline{\boldsymbol{R}}$ are the TBD medians of $\{\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_m\}$,

$$\sum_{i=1}^{m}\frac{\nabla F(\overline{\boldsymbol{R}})-\nabla F(\boldsymbol{R}_i)}{\{\delta_F(\overline{\boldsymbol{R}},\boldsymbol{R}_i)\}^{\frac{1}{2}}\sqrt{1+\|\nabla F(\boldsymbol{R}_i)\|^2}} = 0 \tag{28}$$

We compute the influence function of TSL, TLD and TVN medians from the equation individually.

In the simulation, we generate the sample data by the N dimensional complex circular Gaussian distribution with zero-mean and a known covariance matrix $\boldsymbol{\Sigma}$ given by

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_0 + \boldsymbol{I}, \tag{29}$$

where $(i, j = 1, 2, \ldots, N)$.

$$\boldsymbol{\Sigma}_0(i,j) = \sigma_c^2\rho^{|i-j|}\exp\big(i2\pi f_c(i-j)\big)$$

Here, ρ is the one-lag coefficient coefficient, σ_c is the clutter-to-noise ratio and f_c is the clutter normalized Doppler frequency. In the simulation, we set $\rho = 0.9$, $\sigma_c^2 = 20$ dB, $f_c = 0.2$, and $N = 8$.

Firstly, we generate m number of sample data, each of that is an N dimensional complex vector. The TBD medians $\overline{\boldsymbol{R}}$ (or TBD means, RD mean) are calculated from the m HPD covariance matrices of the m number sample data. n outliers are then mixed with the sample data. The outlier is modeled as $b\boldsymbol{p} + \boldsymbol{c}$, where $\boldsymbol{p}$ is the steering vector, and b is the amplitude coefficient derived by the signal-to-clutter ratio which is set to 20 dB. We compute the TBD medians $\widehat{\boldsymbol{R}}$ (or TBD means, RD mean) of the contaminated data and the influence functions. The number of the sample m is 50 and the number of outliers n varies from 1 to 40. In Fig. 3, we show the results by taking the average of 1000 trails except for the TSL mean as its influence function is very large. The influence functions of all medians are smaller than all means, meaning that the TBD medians are more robust compared with all means studied here.

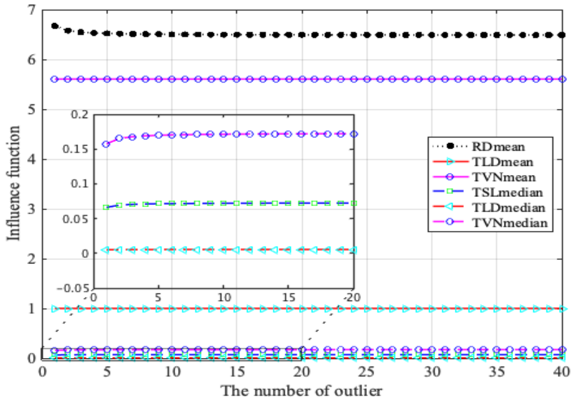


Fig. 3. Robust analysis.

4. Conclusions

We proposed a type of matrix-CFAR detectors via the TBD medians in detection problems, and investigated their detection performance and robustness. It was shown that the TVD median has the best performance from the viewpoint of the probability of detection, and the robustness of the TBD medians is better compared with the TBD means and the RD mean.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number JP20K14365 and JST-CREST Grant Number JPMJCR1914.

References

[1]. F. Barbaresco, Interactions between symmetric cone and information geometries: Bruhat-Tits and Siegel spaces models for high resolution autoregressive Doppler imagery, in *Proceedings of the Conference on*

Emerging Trends In Visual Computing (ETVC'08), Palaiseau, France, 18-20 November 2008, pp. 124-163.

[2]. F. Barbaresco, Innovative tools for radar signal processing based on cartan's geometry of SPD matrices & information geometry, in *Proceedings of the IEEE International Radar Conference (RadarConf'08)*, 2008, pp. 1-6.

[3]. J. Lapuyade-Lahorgue, F. Barbaresco, Radar detection using Siegel distance between autoregressive processes, application to HF and X-band radar, in *Proceedings of the IEEE Radar Conference (RadarConf'08)*, May 2008, pp. 1-6.

[4]. H. Chahrour, R. M. Dansereau, S. Rajan, B. Balaji, Target detection through Riemannian geometric approach with application to drone detection, *IEEE Access*, Vol. 9, 2021, pp. 123950-123963.

[5]. X. Hua, Y. Ono, L. Peng, Y. Cheng, H. Wang, Target detection within nonhomogeneous clutter via total Bregman divergence-based matrix information geometry detectors, *IEEE Transactions on Signal Processing*, Vol. 69, 2021, pp. 4326-4340.

[6]. X. Hua, Y. Cheng, H. Wang, Y. Qin, Y. Li, W. Zhang, Matrix CFAR detectors based on symmetrized Kullback-Leibler and total Kullback-Leibler divergences, *Digital Signal Processing*, Vol. 69, 2017, pp. 106-116.

[7]. N. R. Goodman, Statistical analysis based on a certain multivariate complex Gaussian distribution, *The Annals of Mathematical Statistics*, Vol. 34, Issue 1, March 1963, pp. 152-177.

[8]. E. J. Kelly, Adaptive Detection in Non-Stationary Interference, Part I and Part II, Technical Report 724, Lincoln Laboratory, *MIT*, June 25 1985.

[9]. I. S. Dhillon, J. A. Tropp, Matrix nearness problems with Bregman divergences, *SIAM Journal on Matrix Analysis and Applications*, Vol. 29, Issue 4, 2008, pp. 1120-1146.

[10]. M. Moakher, A differential geometric approach to the geometric mean of symmetric positive-definite matrices, *SIAM Journal on Matrix Analysis and Applications*, Vol. 26, Issue 3, 2005, pp. 735-747.

[11]. N. J. Higham, Functions of Matrices: Theory and Computation, *SIAM*, Philadelphia, 2008.

[12]. M. Moakher, On the averaging of symmetric positive-definite tensors, *Journal of Elasticity*, Vol. 82, Issue 3, 2006, pp. 273-296.

(007)

Binarization for Optical Processing Units via REINFORCE

B. Kozyrskiy [1], **I. Poli** [2], **R. Ohana** [2, 3], **L. Daudet** [2], **I. Carron** [2], **M. Filippone** [1]
[1] Department of Data Science, EURECOM, 450 Route des Chappes, 06410 Biot, France
[2] LightOn, 2 rue de la Bourse, F-75002 Paris, France
[3] Laboratoire de Physique, Ecole Normale Supérieure, 24 rue Lhomond, 75005 Paris, France
E-mail: Bogdan.Kozyrskiy@eurecom.fr

Summary: Optical Processing Units (OPUs) are computing devices which perform random projections of input vectors by exploiting the physical phenomenon of scattering a light source through an opaque medium. OPUs have successfully been proposed to carry out approximate kernel ridge regression at scale and with low power consumption by the means of optical random features. OPUs require input vectors to be binary, and this work proposes a novel way to perform supervised data binarization. The main difficulty to develop a solution is that the OPU projection matrices are unknown which poses a challenge in deriving a binarization approach in an end-to-end fashion. Our approach is based on the REINFORCE gradient estimator, which allows us to estimate the gradient of the loss function with respect to binarization parameters by treating the OPU as a black-box. Through experiments on several UCI classification and regression problems, we show that our method outperforms alternative unsupervised and supervised binarization techniques.

Keywords: Optimization, Random features, Linear regression, Optical processing unit.

1. Motivation

Optical Processing Units (OPUs) are computing devices which perform random projections of input vectors by exploiting the physical phenomenon of scattering a light source through a diffusive medium [1]. The projection operation is particularly useful when approximating kernel functions via random features, a popular technique to implement these models for large-scale problems [2]. OPUs offer the possibility to obtain such approximations with a large number of random features at the speed of light and with low-power consumption, representing a very attractive line of work to further improve scalability of kernel machines. As an example, OPU-based random feature approximations have successfully been proposed to carry out approximate kernel ridge regression in [3].

The main limitations on the generality of this approach are that OPUs require input vectors to be binary and that OPU projection matrices are unknown and can only be retrieved through an expensive calibration procedure. Common approaches for optimization of binarized neural networks, like straight-through estimator or different kinds of a relaxation of the binarization procedure, can be found in the literature on neural networks, where existing methods rely on the possibility to propagate gradient through all operations of the network except binarization [4]. In the literature, there are approaches which address binarization by considering it as a pre-processing step, which happens independently of the regression/classification task [5]. In this case label information is omitted, and this might be suboptimal compared to strategies that take this information into account in the binarization phase.

In this paper, we propose a novel binarization method for OPUs which is learned along with the regression/classification task in an end-to-end manner.

We overcome the main challenge to develop such an end-to-end solution, which is that OPU projection matrices are unknown, by employing the so-called REINFORCE gradient estimator. This allows us to estimate the loss function gradient with respect to binarization parameters by treating the OPU as a black-box. Through experiments on several UCI classification/regression problems, we show that our proposal outperforms alternative unsupervised and supervised binarization techniques.

2. Related Work

In neural networks, binarization is generally targeting intermediate layer activations, and it may also stem from binarization of model parameters; in these cases, binarization is mostly introduced to reduce computational cost and memory consumption [6]. Neural networks with binary hidden layers find applications in binary autoencoders for hashing [7], data compression [5], and hard attention mechanism [8]. The binarization of layer activations is obtained by a suitable choice of activation functions; for instance, the sign or Heaviside functions for the deterministic case, or the sigmoid or tanh functions combined with the Bernoulli distribution for the stochastic case [4, 9]. The most popular technique to propagate gradients through such activation functions is the so called straight-through estimator (STE) [10]. More recently, there have been proposals to replace the STE with another estimator through a relaxation technique, also known as the Gumbel Softmax-trick [11]. Also, different kinds of target propagation are used to learn suitable targets for each binary layer and then train the associated parameters with relaxation techniques or combinatorial optimization [12-14].

In this work, we aim to develop a supervised binarization model which is learned together with the supervised learning task. That is, we aim to provide a training procedure for the heterogeneous model consisting of the kernel ridge regression model approximated with random features and the binarization encoder before the OPU. In this context, a general-purpose framework called Method of Auxiliary Coordinates (MAC) was proposed in [14] with examples of application in [7] and [15]. The authors propose to introduce auxiliary variables into a deep neural network. These auxiliary variables are assigned the role of pre-activations for each layer, and they get replaced during the forward pass. The first step of the optimization targets the auxiliary variables, and, after this step, the parameters of each layer are optimized to regress on these variables, which take the role of layer-specific labels. This is very beneficial when some layers are discrete and vanilla backpropagation is not applicable. In [15], this approach is used to train a fully connected network with binary activation functions, using a STE to propagate a learning signal through the non-differentiable parts. Reference [7] is especially interesting because authors illustrate, how discrete binary layers can be optimized within larger, non-binary model.

While splitting the optimization of the binarization and the model is a viable option, we still need a way to training each part individually. There is a wide variety of ways to obtain a solution for kernel ridge regression with the random feature approximation, so the most difficult point is how to optimize the part consisting of the binary encoder and the OPU, because it combines a non-differentiable function with an implicit random projection. These make the STE from [15] inapplicable. Also, we found that the combinatorial approach used in [7] and [12] is inapplicable for our case for two reasons. First, it is suitable only when the binary dimension is relatively small, which might be a limitation for a general solution. Second, the combinatorial approach combined with MAC converges in one iteration to poor local optima, and this happens because of the model setup which is different from the ones in [7] and [12].

From a different point of view, it is possible to view our problem through the lenses of reinforcement learning, where it is necessary to propagate binary codes through the OPU instead of discrete actions through the black-box environment. Instead of maximizing the reward from the environment, we are trying to minimize the loss function. The classical algorithm to solve this problem is REINFORCE [16]. This allows one to calculate gradients of the reward with respect to parameters of the policy that generates actions. The applicability of this method to other settings with black-box elements was shown in [17]. There are various versions of this algorithm intended to reduce variance of the gradient of the parameters. Very frequently they are based on relaxations of the non-differentiable sampling procedure [18], or approximation of the black-box part of the model [19]. It also worth noting that there exist competitive alternatives to REINFORCE, such as the one in [20], later extended with variance reduction [21] or relaxation [22].

3. Methods

In this paper we consider the kernel ridge regression model. Let $X = x_1, \dots, x_n$ a set of input vectors $x_\text{i} \in \mathbb{R}^\text{d}$ and let $Y = y_1, \dots y_n$ a set of corresponding binary labels. The aim of kernel ridge regression is to establish a mapping between the inputs and the labels by means of functions which belong to the so-called Reproducing Kernel Hilbert Space (RKHS) induced by the choice of a kernel function $k(\cdot,\cdot)$ [23].

Given a choice of kernel function, kernel ridge regression requires evaluating it among all possible pairs of inputs, yielding an n×n matrix K such that $K_{ij} = k(x_i, x_j)$. The solution of kernel ridge regression requires performing algebraic operations with K, and this is problematic when n is large.

A way to avoid these computations and scale kernel ridge regression to large data is to use an approximation based on random features [2]. In this work, we focus in particular on random features produced by OPUs.

OPU performs multiplication of a binary vector $x \in \mathbb{R}^\text{d}$ by a random matrix and applies the activation function $|\cdot|^2$.

$$\phi(x) = \frac{1}{\sqrt{D}} |Rx|^2, \tag{1}$$

where $R \in \mathbb{C}^{\text{D}\times\text{d}}$ is a complex Gaussian matrix with elements $R_{ij} \sim \mathcal{CN}(0,1)$. Performing regression on a linear model using these new random features in (1) gives equivalent results to the original kernel ridge regression problem when $D \to \infty$.

$$\begin{gathered} y^* = W^* \phi(x), \\ W^* = \underset{W}{\operatorname{argmin}} ||\phi(X) W^T - Y||_2^2 + \\ + \frac{\lambda}{2} ||W||_2^2 , \end{gathered} \tag{2}$$

for the training set X, Y. Model (2) is equivalent to the ridge kernel regression with a kernel [3].

$$\begin{gathered} k(\mathbf{x}, \mathbf{y}) \approx \phi(\mathbf{x}) \phi(\mathbf{y}) \overset{D\to\infty}{=} ||\mathbf{x}||^2 ||\mathbf{y}||^2 + \\ + (\mathbf{x}^T \mathbf{y})^2 \end{gathered} \tag{3}$$

We propose to perform the binarization of the input to this model by means of an encoder with parameters W_enc. The output of the encoder parameterizes a multidimensional Bernoulli distribution from which we sample binary vectors and use them as a binary representation of the input data. So, our regression model becomes:

$$\tilde{y} = \mathbb{E}_z[W_{\text{regr}}\phi(z)], \\ \text{where z} \sim \text{Bernoulli}(f(x, W_{\text{enc}})), \tag{4}$$

where W_{regr} are parameters of the linear regression, z is binary representation of the data, $\phi(\text{z})$ are random features. Parameters of the Bernoulli distribution are generated from the input data x by the encoder function f with parameters W_{enc}.

The stochasticity is intentionally introduced to the encoder so that we can employ the so-called REINFORCE gradient estimator. The REINFORCE approach (also called log-derivative trick or score function estimator) aims to estimate the gradient of the expectation of some non-differentiable function f subject to parameters of the distribution of the random variable z:

$$\nabla_\theta \mathbb{E}_{p(z;\theta)} f(z) \approx \tfrac{1}{M}\textstyle\sum_{i=1}^{M} \nabla_\theta log\, p(z;\theta) f(z), \tag{5}$$

where M is number of samples drown from $p(z, \theta)$. For our model, the optimization objective becomes:

$$\min_{W_{\text{regr}}, W_{\text{enc}}} \mathbb{E}_{z\sim\text{Bernoulli}(f(x, W_{\text{enc}}))}[\mathcal{L}(Y, W_{\text{regr}}\phi(z))] + \\ +\lambda_{\text{enc}}||W_{\text{enc}}||^2 + \lambda_{\text{regr}}||W_{\text{regr}}||^2, \tag{6}$$

where $\mathcal{L}\left(Y, \tilde{Y}\right)$ is the quadratic loss for regression problems and the cross-entropy loss for classification problems. The gradient of the first term with respect to W_{enc} becomes:

$$\nabla_{W_{\text{enc}}} \mathbb{E}_{z\sim q(z)}[\mathcal{L}(Y, W_{\text{regr}}\phi(z))] \approx \\ \approx \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}\,(Y, W_{\text{regr}}\phi(z_i))\nabla_{W_{\text{enc}}}\log q(z_i) \tag{7}$$

In order to reduce the variance of this estimator, we can use control variates as proposed in [21]:

$$\nabla_{W_{\text{enc}}} \mathbb{E}_{z\sim q(z)} \left[\mathcal{L}\left(Y, W_{\text{regr}}\phi(z)\right)\right] \approx \\ \approx \frac{1}{M}\sum_{i=1}^{M} \nabla_{W_{\text{enc}}} \log q\,(z_i)\left(\mathcal{L}\left(Y, W_{\text{regr}}\phi(z_i)\right) - v_i\right), \tag{8} \\ \text{where } v_i = \frac{1}{M-1}\sum_{i\neq j} \mathcal{L}\,(Y, W_{\text{regr}}\phi(z_j))$$

Thanks to REINFORCE, we are able to optimize the encoder in an end-to-end fashion. In the remainder of this paper, we refer to this method as End-to-End SE.

In the End-to-End SE in order to estimate the gradient of the loss with respect to W_{enc} it is necessary to pass multiple samples from the encoder through the random projection and the approximate kernel ridge regression model. Depending on the number of random features used for the approximation, this operation can be expensive. To alleviate this computational burden, we propose a variation on the End-to-End SE where we propagate samples only through the random projections layer and then we average them before feeding them to the final linear operation.

$$\tilde{y} = W_{\text{regr}}\mathbb{E}_z[\phi(z)], \\ \text{where } z \sim \text{Bernoulli}(f(x, W_{\text{enc}})) \tag{9}$$

The optimization objective in this case becomes:

$$\min_{W_{\text{regr}}, W_{\text{enc}}} \mathcal{L}(Y, W_{\text{regr}}\mathbb{E}_{z\sim\text{Bernoulli}(f(x, W_{\text{enc}}))}[\phi(z)]) + \\ +\lambda_{\text{enc}}||W_{\text{enc}}||^2 + \lambda_{\text{regr}}||W_{\text{regr}}||^2 \tag{10}$$

So, the gradient of the first term with respect to encoder parameters becomes:

$$\nabla_{W_{\text{enc}}}\mathcal{L} = \frac{d\mathcal{L}}{d(\mathbb{E}\phi(z))}\nabla_{W_{\text{enc}}}\mathbb{E}(\phi(z)), \tag{11}$$

where $\nabla_{W_{\text{enc}}}\mathbb{E}(\phi(z))$ calculated with REINFORCE estimator. We will refer to this method as Isolated Supervised Encoder.

4. Results

We compared the performance of the proposed approaches (End-to-End SE and Isolated SE) against a model based on unsupervised autoencoder proposed in [5], encoder trained with direct feedback alignment (DFA) [23] and Gaussian process (GP) regression based on radial basis function (RBF) kernel over raw and binarized data. Results are reported in Fig. 1 on several UCI regression and classification problems [24]. We want to emphasize that the main competitors of the proposed methods are the ones based on unsupervised autoencoder and encoder trained by DFA, because kernel ridge regression is unable to work with large datasets, and OPU-based regression just approximates this method and is intended to replace it on large datasets.

For Isolated SE and End-to-End SE as an encoding function $f(x, W_{\text{enc}})$ providing parameters for the Bernoulli, distribution we chose a single linear layer with a sigmoid activation:

$$f(x, W_{\text{enc}}) = \sigma(W_{\text{enc}}x) \tag{12}$$

All hyperparameters for the DFA encoder, End-to-End SE and Isolated SE models (size of binary embedding, learning rate, l2 regularization for the encoder and the regression layer, number of training epochs) were chosen with a random search during cross-validation.

In the comparison of binarization strategies we also include Gaussian processes on the original inputs and on the inputs binarized using unsupervised techniques, and we denote these two methods by RBF and binary RBF, respectively. To apply GP-based regression to the two-classes classification problems we represented class labels as -1, 1 and solved a classification problem as a regression one directly. In these cases, GP parameters were tuned by marginal likelihood maximization. This poses computational challenges for large datasets (MiniBoo, MoCap), so we resort to random feature approximations for these cases.

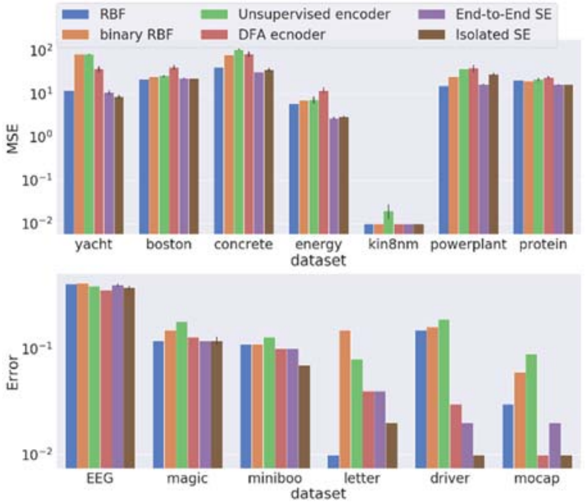


Fig. 1. Mean squared error (MSE) for regression (top) and negative error on classification (bottom) datasets comparison.

For the models involving random features (both Fourier and OPU-generated ones) we have tuned the variance of the distribution that generates these random features. Concretely, assuming that the elements of the R matrix generating the random projections are distributed through the standard Normal distribution, we can obtain a new random matrix R' by multiplying R by any variance, for instance:

$$\phi'(x) = c|R'x|^2 = c|\frac{R}{\sigma}x|^2 = c\frac{1}{\sigma^2}|Rx|^2, \quad (13)$$

with corresponding kernel:

$$k(\mathbf{x},\mathbf{y}) = \frac{1}{\sigma^4}(||\mathbf{x}||^2||\mathbf{y}||^2 + (\mathbf{x}^T\mathbf{y})^2), \quad (14)$$

So, it is enough to multiply the output of the OPU by an additional set of parameters γ, such that $\gamma^2 = \frac{1}{\sigma^2}$, and optimize them with standard gradient descent. The parameter gamma is γ is not equivalent to the lengthscale parameter of the RBF kernel as it has simply a scaling effect on the kernel.

On the regression problems, both proposed methods outperformed their main competitors. On the classification problems, the DFA-based approach was better only on one dataset, and on all other datasets the proposed methods performed better or equally well. Considering the comparison between the proposed methods, we see that End-to-End SE is more stable and requires a significantly fewer number of samples from the encoder, although Isolated SE showed slightly better results on classification problems. We considered including results obtained by running these models on the real OPU (Fig. 2). Unfortunately, the regression problems required such a large number of epochs that we could not perform the experiments in a reasonable amount of time.

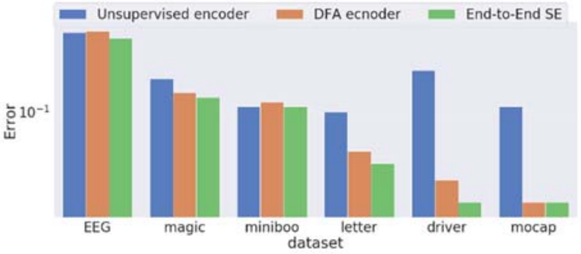


Fig. 2. Error comparison on classification (bottom) datasets for experiments on a real hardware.

We also tested the performance of our approach with respect to the number of samples required to employ REINFORCE. We found that End-to-End SE can achieve good results with a small number of samples from the encoder, and the increase of number of samples does not seem to improve performance. In Fig. 3 we plot the convergence of the loss for one classification and one regression problem. The convergence curves indicate that the convergence speed benefits from the gradient variance reduction.

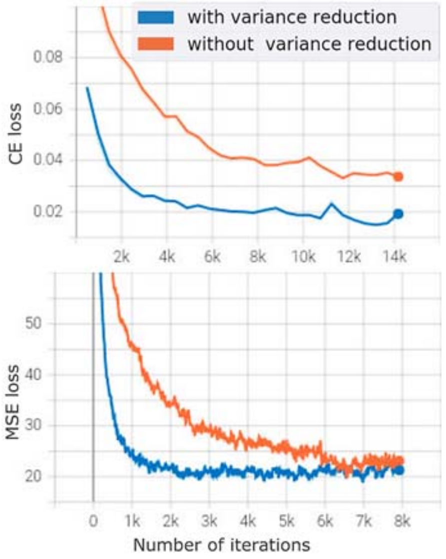


Fig. 3. Convergence of the training procedure on classification problem: mocap dataset (top) and regression problem: Boston dataset (bottom).

5. Conclusion

We proposed a method inspired by reinforcement learning that allows us to use OPUs for approximately solving kernel ridge regression on real-valued data. We have empirically shown that gradient-based supervised optimization of the binarization part is beneficial compared to unsupervised binarization and strategies that do not employ gradient information.

References

[1]. A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Drémeau, S. Gigan, F. Krzakala, Random projections through multiple optical scattering: Approximating kernels at the speed of light, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'16)*, 2016, pp. 6215-6219.

[2]. A. Rahimi, B. Recht, weighted sums of random kitchen sinks: replacing minimization with randomization in learning, in Advances in Neural Information Processing Systems, *The MIT Press*, 2009.

[3]. R. Ohana, J. Wacker, J. Dong, S. Marmin, F. Krzakala, M. Filippone, L. Daudet, Kernel computations from large-scale random features obtained by optical processing units, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*, 2020.

[4]. M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1, arXiv:1602.02830, *arXiv Preprint*, 2016.

[5]. J. Tissier, C. Gravier, A. Habrard, Near-lossless binarization of word embeddings, in *Proceedings of the Conference on Artificial Intelligence (AAAI'19)*, 2019.

[6]. H. Qin, R. Gong, X. Liu, X. Bai, J. Song, N. Sebe, Binary neural networks: A survey, *Pattern Recognition*, Vol. 105, 2020, 107281.

[7]. M. A. Carreira-Perpinán, R. Raziperchikolaei, Hashing with binary autoencoders, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2015.

[8]. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in *Proceedings of the International Conference on Machine Learning (ICML'15)*, 2015.

[9]. J. W. T. Peters, M. Welling, Probabilistic binary neural networks, arXiv:1809.03368, *arXiv Preprint*, 2018.

[10]. Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv:1308.3432, *arXiv Preprint*, 2013.

[11]. E. Jang, S. Gu, B. Poole, Categorical reparameterization with Gumbel SoftMax, arXiv:1611.01144, *arXiv Preprint*, 2016.

[12]. A. L. Friesen, P. Domingos, Deep learning as a mixed convex-combinatorial optimization problem, arXiv:1710.11573, *arXiv Preprint*, 2017.

[13]. D.-H. Lee, S. Zhang, A. Fischer, Y. Bengio, Difference target propagation, in *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD'15)*, 2015.

[14]. M. Carreira-Perpinan, W. Wang, Distributed optimization of deeply nested systems, in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS'14)*, Reykjavik, 2014.

[15]. A. Choromanska, B. Cowen, S. Kumaravel, R. Luss, M. Rigotti, I. Rish, P. Diachille, V. Gurev, B. Kingsbury, R. Tejwani, et al., Beyond backprop: Online alternating minimization with auxiliary variables, in *Proceedings of the International Conference on Machine Learning (ICML'19)*, 2019.

[16]. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, *Machine Learning*, Vol. 8, 1992, pp. 229-256.

[17]. R. Ranganath, S. Gerrish, D. Blei, Black box variational inference, in *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS'14)*, 2014, pp. 814-822.

[18]. G. Tucker, A. Mnih, C. J. Maddison, D. Lawson, J. Sohl-Dickstein, Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models, arXiv:1703.07370, *arXiv Preprint*, 2017.

[19]. W. Grathwohl, D. Choi, Y. Wu, G. Roeder, D. Duvenaud, Backpropagation through the void: Optimizing control variates for black-box gradient estimation, arXiv:1711.00123, *arXiv Preprint*, 2017.

[20]. M. Yin, M. Zhou, ARM: Augment-REINFORCE-merge gradient for stochastic binary networks, arXiv:1807.11143, *arXiv Preprint*, 2018.

[21]. W. Kool, H. van Hoof, M. Welling, Buy 4 REINFORCE samples, Get a baseline for free!, in *Proceedings of the Deep Reinforcement Learning Meets Structured Prediction Workshop at the International Conference on Learning Representations*, 2019.

[22]. Z. Dong, A. Mnih, G. Tucker, DisARM: An antithetic gradient estimator for binary latent variables, arXiv:2006.10680, arXiv Preprint, 2020.

[23]. K. P. Murphy, Machine Learning: a Probabilistic Perspective, *MIT Press*, 2012.

[24]. A. Nøkland, Direct feedback alignment provides learning in deep neural networks, arXiv:1609.01596, arXiv Preprint, 2016.

[25]. D. Dua, C. Graff, UCI Machine Learning Repository, *University of California*, 2017.

(008)

A Novel Python-based Trifold Data Science, Stem, Climate Change Framework

Rafaat Hussein
State University of New York, USA
E-mail: ezpsc@yahoo.com

Summary: Data science and analytics (DSA) present a fresh perspective for tackling the indisputable devastating impacts of climate change (1). The influx of reputable sources has emphasized this reality including the New York Times, the Washington Post, the Guardian, to name a few examples (2). DSA forged with STEM coin a novel diversified framework that directly aim at climate change. The STEM encompasses the vital ingredients for DSA thus was judged to be more fitting to the novel framework than traditional programs. In addition, DSA revolves around data which is drawn from the climate itself. No one can dispute the unimaginable growth nowadays in data. Doesn't this reality imply the massive demands for professionals who can transform the data into useful information rather than just capturing and storing it? How can they be prepared? STEM platforms can be shaped to seamlessly fit DSA and climate change for the coined trifold framework. The primary objective of this paper is to coin a novel trifold perspective that somehow overlooked in the literature. We expect a rapid and widespread adoption of this introduction by the pertinent organizations. It should be noted that climate change and climate change impacts are interchangeably used in this paper.

Keywords: Climate change, Data science, Data analytics, Big data, Python, STEM.

1. Introduction

Climate change and its undeniable devastating impacts on planet earth and its occupants has become the greatest challenge to mankind. Data Science and analytics have also spread to almost all facets of humans' lives. In spite of the abundance of climate data, the data science has little impacts on finding practical solutions to the increasing severity of climate change. This discrepancy stems from the complex nature of climate data as well as the complex questions climate science brings forth.

This paper bridged the gap between data science and analytic, and climate change impacts. It coined a novel thrifold framework. Despite the growing calls to figure out effective strategic frameworks for the future generations of world professionals, the vision(s) for effective pathway(s) is blurred. The paper envisioned the STEM as the incubator for the urgently needed professionals who are prepared to properly stimulate, discover, explore, detect and capture, and facilitate finding sound solutions to climate change impacts.

2. Climate Change

To gain a meaningful framework to the data challenges in climate science, it is helpful to examine elements at the core of the climate analysis [3]: [1] the integration of the data in meaningful presentations; and [3] realistic climate models and simulations supported by real data.

A study by NASA Technical Reports Server [3] provided an in-depth look at how massive amounts of data can be leveraged and analyzed to generate viable solutions to the threat of climate change. In general, the knowledge we gain from the data depends on its generation, dissemination, and analysis. This paper suggests the preparation of future generation of DSA to enhance the understanding of the climate change based on the data records and pertinent models. In other words, better understanding the implications of the vast datasets and the deployment of the right computational resources and useful applications.

3. STEM

The climate challenges have brought the STEM to the fore front in capacity building [5, 6]. With the wide spreading coverage surrounding climate change and its undeniable devastating impacts, the logical question is don't the integration of DSA and STEM hold the promise not only the needed solutions but also the preparation of future professionals. This paper considers such integration is prudent to ensure that the practitioners can convert the overwhelming data to practical advantages. In addition, our novel framework uses STEM as an incubator that to harnesses the boundless data and climate to create positive solutions.

4. Data Science Perspective

Data is any events that can be measured or categorized [7]. Once collected, the data can be studied and analyzed both to understand the nature of the events and very often also to make predictions or at least to make informed decisions.

DAS is a process consisting of several steps in which the raw data are transformed and processed to produce data visualizations and can make predictions using analytic models. So, data analysis is a process chain consisting of the following typical sequence of stages: problem definition, data extraction, data cleaning, data transformation, data exploration, predictive modeling, model validation, visualization and interpretation of results, and deployment. As previously mentioned, the scope of this paper does not cover detailed elaboration on these stages.

One may ask: why Python? Python and Java are sharing the first rank among all commonly used tools worldwide. Python has many advantages including open-source and free, easy to intuitively learn, excellent on line community, integrate well with other packages, and faster than similar tools such as R and MATLAB.

As previously mentioned, this paper is the first in a serious of papers that will provide details on the suggested trifold framework.

5. Conclusions

The inseparable relationship between data and climate sciences connects the alleged disparities between the respective professionals, thus direct them to pursue promising technological discoveries and visions to provide solutions to climate change impacts [8]. DSA has a pivotal role to play within the realm of climate change. Any pattern or trend would be deficient if the used data is inadequately interpreted, analyzed. The logical question is who to take on this task and are they well prepared for it ? This paper answers this question by coining for the first time a novel python-based trifold data science, stem, climate change framework where climate change provides the data (ingredients), DSA provides the recipes (how to), and STEM provides the kitchen for solutions (edible meals).

Acknowledgment

This study received no financial support from any organization. However, the conceptualization of the trifold framework emerged from years of interactions with students and professionals (https://aitechm.wixsite.com/8kbextech04182021esf) to whom the author acknowledges their supports.

References

[1]. T. Donoghue, et al., Teaching creative and practical data science at scale, *Journal of Statistics and Data Science Education*, 2021, pp. S27-S39.

[2]. Climate Change Is the Ultimate Teachable Moment, https://www.edsurge.com/news/2021-08-26-climate-change-is-the-ultimate-teachable-moment

[3]. F. Eggleton, K. Winfield, Open data challenges in climate science, *Data Science Journal*, Vol. 19, Issue 1, 2020, pp. 52-57.

[4]. How Data Science is Driving Innovation in Climate Change Research, https://www.simplilearn.com/how-data-science-is-driving-innovation-in-climate-change-research-article

[5]. Why Colleges Are Offering Data Science Programs Across disciplines, there's a demand for people with an aptitude for crunching numbers, https://www.usnews.com/education/best-colleges/articles/why-more-colleges-are-offering-data-science-programs

[6]. C. Parker, STEM education is the key to raising a generation of climate change leaders, *The Hill Journal*, 2020.

[7]. J. VanderPlas, Python Data Science Handbook, *O'Reilly Publishing*, 2017

[8]. Harnessing Data Science for Climate Change, https://www.simplilearn.com/harnessing-data-science-for-climate-change-article

(009)

Some Like It Tough: Improving Model Generalization via Progressively Increasing the Training Difficulty

Hannes Fassold
JOANNEUM RESEARCH – DIGITAL, Steyrergasse 17, 8010 Graz
E-mail: hannes.fassold@joanneum.at

Summary: In this work, we propose to progressively increase the training difficulty during learning a neural network model via a novel strategy which we call *mini-batch trimming*. This strategy makes sure that the optimizer puts its focus in the later training stages on the more difficult samples, which we identify as the ones with the highest loss in the current mini-batch. The strategy is very easy to integrate into an existing training pipeline and does not necessitate a change of the network model. Experiments on several image classification problems show that mini-batch trimming is able to increase the generalization ability (measured via final test error) of the trained model.

Keywords: Deep learning, Model training, Model generalization, Importance sampling, Curriculum learning, Optimization.

1. Introduction

Training a neural network model which generalizes well (has good performance on unseen data) is a highly desirable property, but is not easy to achieve. Nowadays, most often adaptive gradient methods like the *Adam* optimizer [1] are used for training a model as they are much easier to handle (less sensitive to weight initialization and hyperparameters) compared with mini-batch stochastic gradient descent (SGD). On the other hand, their generalization capability has been observed to be not as good as SGD [2].

In this work, we propose a simple strategy which we call *mini-batch trimming* to increase the generalization capability (measured for image classification problems as the error of the final model on the *test* dataset) of a trained model. The strategy is easy to integrate into an existing training pipeline, does not need a modification of the model structure and is independent of the employed optimizer (so can be used for both SGD and Adam-like methods). Its motivation lies from the fact that humans do learn subjects (e.g. algebra) 'from easy to hard': We first learn the basic concepts of a certain subject and learn the more advanced topics later. In the same way, we want our optimizer to focus in the later training stages on the more difficult samples in the dataset. E.g. for image classification, these are the ones which are harder to classify correctly.

Our strategy has similarities with *curriculum learning* methods and *importance sampling* methods. In curriculum learning (see the survey in [3]), during training the samples are presented in a more meaningful order (e.g. from easy to hard) instead of the default random order. *Importance sampling* methods do not treat all samples in a dataset in the same way, but instead bias the selection of samples via a certain criterion. E.g. in [4], typicality sampling is used to overweight highly representative samples during training. A disadvantage of this approach is that it has a complicated workflow, which involves density clustering (via t-SNE algorithm [5]) in the sample space.

In the following section we will describe our proposed mini-batch trimming strategy, whereas in section 3 experiments will be done on standard image classification problems which demonstrate that the strategy leads to models which generalize better.

2. Mini-batch Trimming

The training of a neural network model is usually done iteratively. In each iteration, a mini-batch consisting of B samples (where B is typically 64 or 128) is drawn randomly from the training set, the mean loss for the mini-batch is calculated in the forward pass and in the backward pass the gradient of the mean loss is utilized to update the model weights.

In order to focus more on the harder samples in the mini-batch, we propose a strategy which we call *mini-batch trimming*. As we cannot quantify the 'hardness' of a sample φ exactly, we take the per-sample loss $L(\varphi)$ as an estimate of its hardness. This makes sense, as the more difficult samples in the training set typically also have a higher loss. We modify the forward pass now in the following way: First the per-sample loss $L(\varphi)$ is calculated for all samples in the mini-batch. Now all samples in the mini-batch are sorted using the per-sample loss as criterion. The mean loss is now calculated *only from a fixed fraction of the samples in the mini-batch with the highest per-sample loss.* So we are calculating sort of a *trimmed mean* instead of the usual mean. For selecting the fraction p of the samples with the highest loss the Pytorch framework provides the *torch.topk* operator, which is also differentiable.

In this way, in each training iteration the update of the model weights is biased towards the more difficult samples. In the fashion of curriculum learning, the fraction p is *linearly decreased* during training. For the first epoch p has the value 1.0 (take all samples in mini-batch into account), whereas in the last epoch p

is set to 0.2 (focus only on the 20 % samples in the mini-batch with the highest loss). Experiments have shown that this is a sensible choice. Note that for neural networks without batch-normalization layers (e.g. transformer architectures for natural language processing), mini-batch trimming *brings also a runtime improvement*, as the backward pass then depends only on a part of the mini-batch[1].

3. Experiments and Evaluation

For the experiments and evaluation, we employ three standard datasets for image classification: SVHN, CIFAR-10 and CIFAR-100. The datasets consist of 32×32 pixel RGB images, which belong to either 10 classes (SVHN and CIFAR-10) or 100 classes (CIFAR-100). We use the Adam optimizer, with learning rate set to 0.001 and weight decay set to 0.0001. The mini-batch size is 128 and training is done for 150 epochs, with the learning rate decayed by a factor of 0.5 at epochs 50 and 100. We perform the experiments with two popular neural network architectures for computer vision, Resnet-34 [6] and Densenet-121 [7].

To measure how well the trained model is able to generalize, we utilize the top-1 classification error of the final model on the test set (which of course has not been seen during training). For each configuration, we do 10 different runs with random seeds and take the average of these 10 runs. We compare the standard training with the variant with mini-batch trimming enabled. Results of the experiments can be seen in Table 1. The evaluation shows that mini-batch trimming is able to improve the generalization capability of the model in nearly all cases, except for one case (Densenet-121 architecture on CIFAR-10 dataset) where there is a slight regression in the model performance.

Table 1. Comparison of training with mini-batch trimming disabled / enabled for various network architectures and datasets. The first value in each cell is the average test error (in percent, averaged over 10 runs) with mini-batch trimming disabled, the second value is with mini-batch trimming enabled. The lower value is marked in bold.

Dataset	**Network architecture**	
	Resnet-34	Densenet-121
SVHN	5.87 / **5.76**	4.62 / **4.42**
CIFAR-10	17.43 / **17.01**	**10.10** / 10.19
CIFAR-100	48.19 / **47.72**	32.95 / **32.18**

3. Conclusion

We presented a novel strategy called mini-batch trimming for improving the generalization capability of a trained network model. It is easy to implement and add to a training pipeline and independent of the employed model and optimizer. Experiments show that the proposed method is able to improve the model performance in nearly all cases. In the future, we plan to investigate and integrate this strategy within a distributed training framework like DeepSpeed.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 951911 – AI4Media.

References

[1]. D. Kingma, J. Ba, ADAM: A method for stochastic optimization, in *Proceedings of the International Conference for Learning Representations (ICLR'15)*, 2015.

[2]. P. Zhou, J. Feng, C. Ma, C. Xiong, S. Hoi, E. Weinan, Towards theoretically understanding why SGD generalizes better than ADAM in deep learning, in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS'20)*, 2020.

[3]. X. Wang, Y. Chen, W. Zhu, A survey on curriculum learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021, preprint.

[4]. X. Peng, L. Li, F. Wang, Accelerating minibatch stochastic gradient descent using typicality sampling, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, Issue 11, 2020. pp. 4649-4659.

[5]. L. van der Maaten, G. Hinton, Visualizing data using t-SNE, *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 2579-2605.

[6]. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 770-778.

[7]. G. Huang, Z. Liu, L. van der Maaten, K. Weinberg, Densely connected convolutional networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 4700-4708.

[1] https://stackoverflow.com/questions/68920059/pytorch-no-speedup-when-doing-backward-pass-only-for-a-part-of-the-samples-in-m

(010)

Data Handling Practices of the Cardiac Echo Core Laboratory at Boston Children's Hospital

Edward Marcus, Meena Nathan, Patrick McGeoghegan, Kevin Friedman
Cardiology Department, Boston Children's Hospital
9 Hope St., Waltham, Massachusetts USA
Tel: (001) 781-216-2532
E-mail: ed.marcus@cardio.chboston.org

Summary: Ultrasound scanner generated 'echo' images have become a prevalent source of data for cardiovascular disease research. The Boston Children's Hospital (BCH, USA) cardiology department has developed echo image handling software for multicenter studies of residual surgical outcomes [1] and cardiovascular Covid 19 effects [2]. To facilitate the enrollment needs of these multi-center efforts, image processing tools have been designed to prepare image transmissions and simplify measurement reporting. In this paper we provide a brief overview of implemented Corelab software tools, and offer approaches to data preparation steps needed by different study pipeline stages.

Keywords: Corelab, HIPAA, Protected health information, Ultrasound images, Structured reporting.

1. Echo Image Data Preparations by Sites

An 'echo Corelab' is organized to analyze ultrasound images collected from participating research sites. Typically, an echo study sent to the Corelab might contain 100 or more image files. When the images are received, they are reviewed by medical experts to measure structural tissue dimensions and assess cardiac disease states.

Before an echo study is generated by the research site, the anticipated images will be assigned a research ID to uniquely identify data before, during, and after transmission. As site images are prepared, this id and subject details such as patient age, height, or weight will also be provided. These additional meta-data elements form the basis of the subject's classification for research statistics.

A challenging aspect of the site's image preparation is to ensure the data released to the Corelab will be purged of information potentially identifying the research subject. A government regulation such as HIPAA [3], might form the context for de-identification steps that will need to be performed at the site before images can be transmitted and released. Shown in Fig. 1, the de-identification processing effectively removes patient ids, names, or other data considered to be protected health information (PHI).

Some challenges are introduced to site de-identification processing that arise from various Digital Imaging and Communication in Medicine (DICOM) image formats implemented by echo scanners at different sites. To have one standard for de-identification, applied software seeks to render every type of scanner image format and provide for intuitive surveying and redaction of PHI. For this process to be seamless, we have introduced tools to quickly define PHI regions from key images of the study, and to expand the key region definitions to redact all study images automatically.

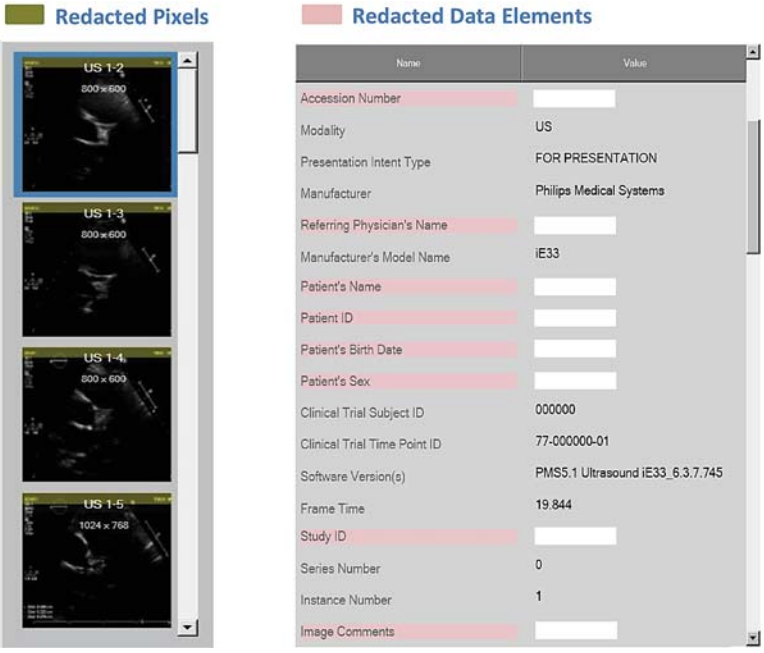
2. Echo Study Review by Corelab

When de-identified files are received at the corelab, the important process of image analysis follows a defined protocol that will be encoded as a structured echo report. The report's items might consist of results from tissue dimensions that are measured. Other items document subjective indications for the 'presence or absence' or functional severity of disease conditions.

The format of each report item begins with a unique concept code to identify the type of measurement the item represents. Each concept code is generated by selecting from entries of a concept dictionary defined by one or more standards [4]. From the concepts selected for the report, the echo report's structure is then formed as a coded data tree.

'Traced' report items are based on reviewer generated outlines drawn as image overlays. These traced measures might include point-to-point distance assessments, heart chamber cross-sectional areas, or the velocity quantifications of blood flow or tissue motion by techniques of Doppler echo processing [5]. Other report items will be additionally generated as derivative results calculated from inputs provided by the tracing measures. These derivative calculations consist of 3D reconstructions of the heart chamber volume [6] and statistical identifications of measurement outliers from normal ranges [7].

Following the completion of traced measures, coded assessments, and derivative calculations, the echo report is finalized with the signature of the corelab reviewer. Signed study findings are then ready for inclusion with results from other completed studies in a statistical pool. All collected data is reviewed by a committee to reach a consensus on achieved study aims and statistics to be published.



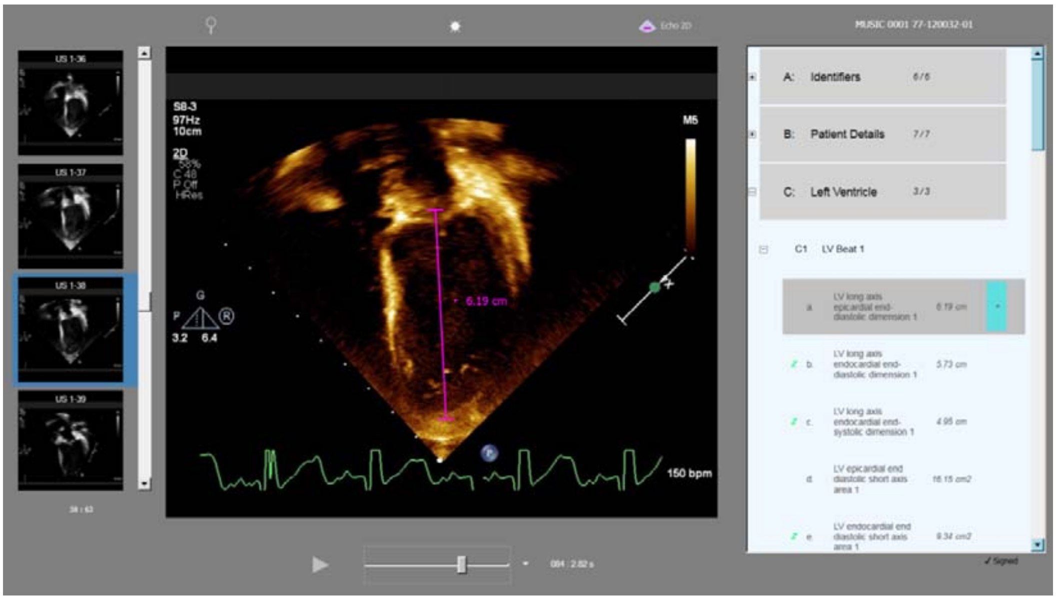
Name	Value
Accession Number	
Modality	US
Presentation Intent Type	FOR PRESENTATION
Manufacturer	Philips Medical Systems
Referring Physician's Name	
Manufacturer's Model Name	iE33
Patient's Name	
Patient ID	
Patient's Birth Date	
Patient's Sex	
Clinical Trial Subject ID	000000
Clinical Trial Time Point ID	77-000000-01
Software Version(s)	PMS5.1 Ultrasound iE33_6.3.7.745
Frame Time	19.844
Study ID	
Series Number	0
Instance Number	1
Image Comments	

Fig. 1. Protected health information (PHI) is redacted by software identifications of PHI sensitive image pixels and data elements.

3. State of the Art Considerations

As the corelab pipeline is comprised of different steps, the processing to be applied at each stage will benefit from a shared data format. In practice, a uniform format is realistically achieved by software components designed to seamlessly exchange data. Simplifications are then achieved, if one installed software package can be applied to handle all pipeline processing requirements.

We suggest that as more research centers become active corelab sites, limitations of personnel or budgets could suggest a need to alter the corelab image flow model. Centralized reviews of received images might be transformed to have images remain with the site, where measurement results would be generated and sent to a central repository. In this *distributed* reviewing model, the requirement for extensive image PHI preparations would be mitigated, along with associated software configuration and training costs.



MUSIC 0001 77-120032-01

A:	Identifiers	6/6
B:	Patient Details	7/7
C:	Left Ventricle	3/3
C1	LV Beat 1	

a.	LV long axis epicardial end-diastolic dimension 1	6.19 cm
✓ b.	LV long axis endocardial end-diastolic dimension 1	5.73 cm
✓ c.	LV long axis endocardial end-systolic dimension 1	4.95 cm
d.	LV epicardial end diastolic short axis area 1	16.15 cm2
✓ e.	LV endocardial end diastolic short axis area 1	8.34 cm2

✓ Signed

Fig. 2. The result of a traced left ventricle long-axis dimension is measured and stored as an echo report item.

In summary, an echo corelab is organized to receive and review ultrasound images generated by research sites. After site images are scanned and generated, the corelab data pipeline implements software to check, prepare, transmit, measure, and report image based findings.

In the future, this centralized data review model might be altered if sites could be provided with reporting capability to export findings instead of images. In this way, site image preparations for transmission might be circumvented - encouraging participation by new sites and expanded levels of patient enrollment.

References

[1]. M. Nathan, *et al.*, Impact of major residual lesions on outcomes after surgery for congenital heart disease, *J. Am. Coll. Cardiol.*, Vol. 77, Issue 19, pp. 2382-2394, May 18 2021.

[2]. D. T. Truong, *et al.*, The NHLBI Study on Long-terM OUtcomes after the Multisystem Inflammatory Syndrome In Children (MUSIC): Design and objectives: Design and rationale of the MUSIC study, *Am. Heart. J.,* Vol. 243, August 18 2021, pp. 43-53.

[3]. W. Moore, S. Frye, Review of HIPAA, Part 1: History, protected health information, and privacy and security rules, *J. Nucl. Med. Technol.,* Vol. 47, Issue 4, December 2019, pp. 269-272.

[4]. National Electrical Manufacturers Association, Digital Imaging and Communications in Medicine (DICOM) Part 16: Content Mapping Resource, http://medical.nema.org/

[5]. W. Armstrong, T. Ryan, Feigenbaum's Echocardiography Eighth Edition, *Wolter's Kluwer Health*, Philadelphia, 2018.

[6]. R. M. Lang, *et al.*, Recommendations for cardiac chamber quantification by echocardiography in adults: An update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging, *J. Am. Soc. Echocardiogr.,* Vol. 28, Issue 1, January 2015, e14.

[7] S. D. Colan, The why and how of Z scores, *J. Am. Soc. Echocardiogr.,* Vol. 26, Issue 1, January 2013, pp. 38-40.

(011)

Classification of SSVEP Signals for Robot Arm Control by Transform Techniques Using LabVIEW

R. S. Sandesh [1] and Nithya Venkatesan [2]
[1] Department of Electronics and Instrumentation Engineering, RVCE, Bengaluru, India
[2] SELECT, VIT University, Chennai, India
Tel.: +919742057790
E-mail: mailsandesh.rs@gmail.com

Summary: This paper proposes the approach of Hilbert Transform (HT) and Time Averaging Techniques (TAT) to classify the Steady State Evoked Potential Signals (SSVEP) for control of robotic arm. Proposed system combines Ag-Agcl electrodes, customized SSVEP acquisition circuit, custom-made simulation panel, NI DAQ card, in LabVIEW environment for real time processing. The experimental results demonstrates the subject's SSVEP controls the robot arm through the Brain Computer Interface (BCI) system with an increased accuracy up to 94.60 % with an increase in Information Transfer Rate (ITR) up to 26.4 bits/min compared to other BCI controlled robot systems.

Keywords: Brain computer interface (BCI), Steady state visual evoked potential signals (SSVEP), Hilbert transform (HT), Time averaging technique (TAT), Robotic arm control.

1. Introduction

Brain Computer Interface (BCI) technology is a foremost communication device connecting the users and systems. BCI is a technique [1] can be used to communicate a user's intentions to external world without involving normal pathway like peripheral nerve system [2]. Typical applications of BCI are in Neuroprosthesis, control of robotic devices for day-to-day applications to regain the degree of independence by the people affected by neurological disorders [3]. The consistent use of BCI has also led to applications in stroke rehabilitation, sleep analysis and detection of other diseases [4]. In BCI, Electro Encephalo Graph (EEG) are often used to extract brain signals due to ease of use, good temporal and spatial resolution. Non-invasive method [5], record the brain waves using electrodes placed on the scalp without surgical incision, often used as compared to invasive method [6]. The sources for EEG signals are event-related synchronization/ desynchronisation (ERS/ERD), visual evoked potential (VEP), slow cortical potentials (SCPs), P300 evoked potentials, μ and β rhythms, etc.

The SSVEP signals are the kind of EEG signals that respond to flickering signals that are greater than 6 Hz [7]. The majority of SSVEP based BCI experimentation is carried out for low and medium flickering frequencies which are less than 30 Hz. The main advantage of such flickering frequencies is that there will be an increase in the amplitude compared to frequencies greater than 30 Hz. The use of high frequencies can reduce the amplitude of the spectrum with a reduction in impulsive activity. Therefore, the presence of medium frequency is suitable for BCI application, since most of the promising results have been obtained for frequency range between 14 Hz and 30 Hz. This research work focuses on SSVEP signals which have 60-70 bits/min, and requires minimum or no training with high Information Transfer Rate (ITR) [8].

SSVEP signals find applications in movement of wheelchairs [9], Neuroprosthesis [10], many more. The developed BCI system can be used to control movement of wheel chair with additional intelligence like interfacing of autonomous navigation system for path referencing [11].To control electrical wheelchair, stimulus used in real time experimentation is LED panel consists of four diodes oscillating at 13, 14, 15, and 16 Hz, associated with left, right, forward and backward directions respectively. Extended work in the control of wheelchair can be found with greater hit rates achieving between 60 and 100 % [12]. SSVEP signals are used to control a robotic car based on ensemble empirical mode decomposition based approach to output three commands like turning left, right and moving forward [13].

Recent studies in SSVEP based BCI enables the subject to serve for themselves, with an increase in accuracy of up to 91.35 % and ITR of 20.69 bits/min. The neural interface system was tested with subject suffering from ALS with questionnaire reply as feedback to improve the performance of the system [14]. Three other functions were integrated like video entertainment, video calling and active interaction, thus making multifunction wireless BCI system providing mean accuracy of 90.91 % and an increase in ITR up to 24.94 bits/min compared to earlier method [15]. Hilbert Transform and Multi Wavelet Transform (MWT) were applied along with neural network and SVM in classifying the SSVEP signals to control robotic arm reaching mean accuracy up to 90 % [16]. In addition to this, many research work were proposed on frequency coding of SSVEP signals, reaching an accuracy up to 97.5 % [17]. The present work focus on phase coding of SSVEP signal for robotic arm control.

With respect to the selection of embedded system, low cost FPGA based system is used for acquiring SSVEP signals to control the multimedia device with an accuracy of 89.2 % [18]. The conventional microcontrollers like MSP 430 [19], DSP processor [20] also finds application in BCI applications. The use of LabVIEW software requires exposure in field of BCI system for neuro rehabilitation application.

The proposed SSVEP based robot arm system recognizes the subject's SSVEP signal by gazing at the stimulus in cyclic order. By identifying the stimulus command, and applying the HT and TAT, the phase value is identified for control of robotic arm, developed using LabVIEW software. The visual stimulus with a flickering frequency of 25 Hz is designed using ATMEGA328P Arduino UNO microcontroller operating with a voltage of 12 volts. The customized SSVEP acquisition system is designed and tested against different forms of EEG signals (delta, theta, alpha, and beta) for operating at voltage of 9 volts battery [21]. The main finding in this paper is to develop and validate the results for attention dependence SSVEP based BCI system with little or no training and to integrate the subject's SSVEP to LabVIEW using DAQ card for a three Degree of Freedom (DoF) robotic arm control. The experimental results obtained proves effectiveness of system by increasing mean accuracy and ITR compared to other works.

2. Material and Methods

2.1. System Configuration

The subject's SSVEP signal is acquired using single channel Ag-Agcl cup shaped electrodes placed at scalp (O_Z, Right mastoid and ground) based on international 10-20 electrode system. The proposed SSVEP based BCI system recognizes Fig. 1 shows the SSVEP signal based simulated robotic arm control. The entire system constitutes of custom made SSVEP simulation panel, customized SSVEP EEG signal acquisition system (battery operated), flickering signal generator using Arduino, simulated robotic hand in a LabVIEW platform, a bio feedback audio buzzer of 5 volts. The customized SSVEP signal acquisition system acquires subject's SSVEP signal, amplifies using a instrumentation amplifier, band pass filtered (0.05 Hz to 30 Hz), notch filter (50 Hz) and amplifies with final stage gain amplifier.

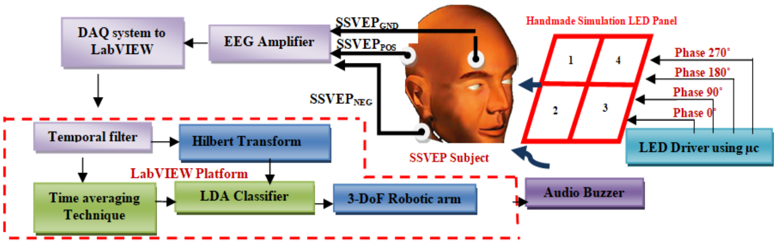


Fig. 1. Application of Hilbert Transform and Time Averaging technique to control simulated robotic arm.

The amplified SSVEP signal is interfaced into LabVIEW environment using data acquisition card. The EEG signal in LabVIEW is then band pass filtered using third order Butterworth IIR filter for the cutoff frequency of 25 Hz. The filtered SSVEP signal is then applied with HT and SAT separately to classify the signal required for robot arm control. Finally, the Linear Discriminant Analysis classifies the SSVEP for control of robotic arm. The entire process is repeated twice and the buzzer will be activated upon the completion of process.

2.2. Arduino Based Simulation Panel

Fig. 2 shows the circuit for handmade SSVEP simulation panel consist of ATMEGA328P Arduino UNO microcontroller board with 4 white LED's (one for each stimuli), transistor (of type SL100), and resistors. The flickering frequency of 25 Hz is decided by subjects participating in the experimentation from a range of frequencies varying from 21- 28 Hz, of which subjects expressed their comfortness to participate in the experimentation.

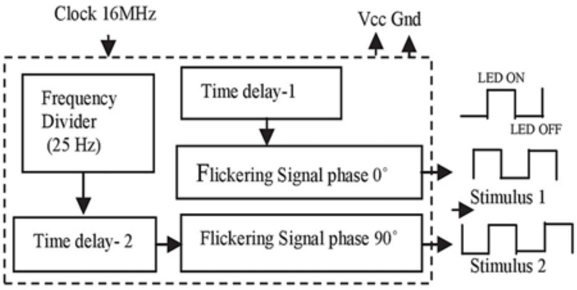

Clock 16MHz
Vcc Gnd
Frequency Divider (25 Hz)
Time delay-1
Flickering Signal phase 0°
Time delay- 2
Flickering Signal phase 90°
LED ON
LED OFF
Stimulus 1
Stimulus 2

Fig. 2. ATMEGA phase encoding circuit.

3. Results and Discussion

3.1. Hilbert Transform and Time Averaging Technique

Once the SSVEP signal is amplified and pre-processed, the next step is to interface the signal with LabVIEW using NI USB DAQ 6009. The interfaced signal is the third order Butterworth band pass filtered.

Let,Wssvep(t) be complex analytical signal and given in equation (1) below,

$$\text{Wssvep(t)} = \text{SSVEP}(t) + jH(\text{SSVEP}(t)), \qquad (1)$$

where $\text{SSVEP}(t)$ is filtered EEG signal and $H(\text{SSVEP}(t))$ is called Hilbert transform of $\text{SSVEP}(t)$ and is defined by equation (2),

$$H(\text{SSVEP}(t)) = \frac{1}{\pi} C.P.V \int_{-\infty}^{\infty} \frac{x(\propto)}{(t-\propto)} \, d\propto \qquad (2)$$

The Instantaneous phase can be calculated from the complex analytic signal Wssvep(t), as shown in below equation (3),

$$\phi_{ssvep}(t) = \begin{cases} \arctan\frac{Hssvep(t)}{SSVEP(t)} \, ssvep(t) \geq 0 \\ \text{Arctan}\frac{Hssvep(t)}{SSVEP(t)} + \pi \\ H(ssvep(t)) \geq 0, \; ssvep(t) < 0 \\ \text{Arctan}\frac{Hssvep(t)}{SSVEP(t)} - \pi \\ H(ssvep(t)) < 0, \; ssvep(t) < 0 \end{cases} \qquad (3)$$

After the phase value is obtained by Hilbert transform as shown in Fig. 3, to increase the accuracy and validate the result, the time averaging technique is also implemented.

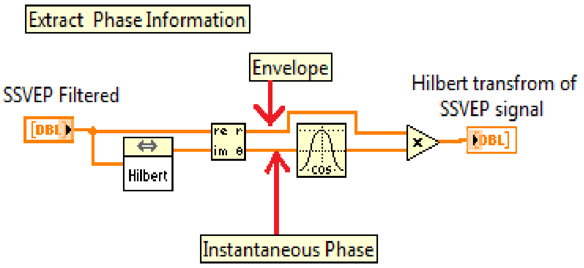


Fig. 3. HT snippet to estimate phase of SSVEP Signal.

The SSVEP filtered signal is averaged to a sampling frequency of 8 KHz to obtain 320 samples per epoch. The subjects were allowed to gaze the stimulus 1 having a phase shift of 0˚ for a time period of ~21 seconds (512 epochs). This signal segment is now averaged for 256 samples to detect the time reference t_{ref} using a waveform peak detector. The predicted practical value is now estimated by using the mathematical equation (4),

$$t^{i}_{practical} = t_{ref} + (i-1) * t_{tinv} \qquad (4)$$

The t_{tinv} between two different phase shift can be estimated using the relation (5),

$$t_{tinv} = [(1/\text{ sampling frequency})*\text{ No of epoch}] / 4 \qquad (5)$$

The estimated time interval t_{tinv} in the above case for sampling rate of 8 KHz is 10 milliseconds. Since this work mainly concentrates on 4 flickering having phase delay of 0˚, 90˚, 180˚, 270˚, the $t^{i}_{practical}$ is estimated in each case and compared with theoretical value. Upon successful comparison of phase value and time averaged values, the classifier classifies the SSVEP signal for translating appropriate commands for robotic hand control.

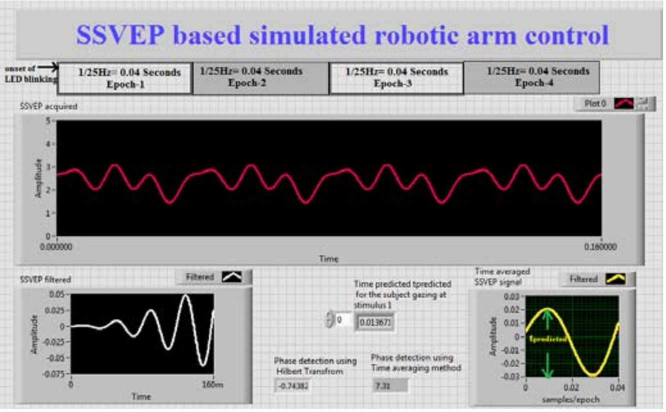
Fig. 4 shows SSVEP signal recorded for time duration of 0.16 seconds (4 epochs), the signal is filtered with a band pass filtered to obtain filtered SSVEP signal.
The both phase and time averaged falls close to that of theoretical value, then translational commands are communicated with the help of producer and consumer design pattern to control the robotic arm. The obtained phase value when Hilbert transform is applied is ~ -0.74˚ and with time averaging technique is ~7.31˚. These results are very close to the value obtained for the subject focusing at stimulus 1. The practical time value for stimulus is ~13 milliseconds is also very close to theoretical value as that ~10 m seconds. This phase information is compared with standard phase values in this work (0˚, 90˚, 180˚, 270˚).

3.2. Producer Consumer Design Pattern for Data Transfer

The classification of SSVEP signal and simulated robotic arm execute between two loops at different time interval. In order to ensure effective communication between the method, first loop, translational commands are obtained – acting as a producer, whereas in second loop, acquires this command to control robotic arm, thus works like the principle of master and slave configuration.

3.3. Simulated Robotic Arm

Fig. 5 shows simulated custom design robotic arm allows the only 3 DoF control by passing θ_1 – base rotation, θ_2 – segment 1 connecting base and one end of segment 2, θ_3 – one end of segment 2 with end effectors. The robotic arm can be operated in one of the three modes- forward kinematics, inverse kinematics and trajectory planning, of which this work concentrate only on first two modes only. Upon the identification of phase value and time predicted value for different stimulus, the 3 DoF of robotic arm is controlled by passing three different arbitrary values of θ with the help of producer/ consumer pattern.



SSVEP based simulated robotic arm control

onset of LED blinking

1/25Hz= 0.04 Seconds Epoch-1

1/25Hz= 0.04 Seconds Epoch-2

1/25Hz= 0.04 Seconds Epoch-3

1/25Hz= 0.04 Seconds Epoch-4

Time predicted tpredicted for the subject gazing at stimulus 1: 0 | 0.013671

Phase detection using Hilbert Transfrom: -0.74382

Phase detection using Time averaging method: 7.31

Fig. 4. HT and TAT for robotic arm control, results for subject 1 gazing at 0 ˚ stimulus.

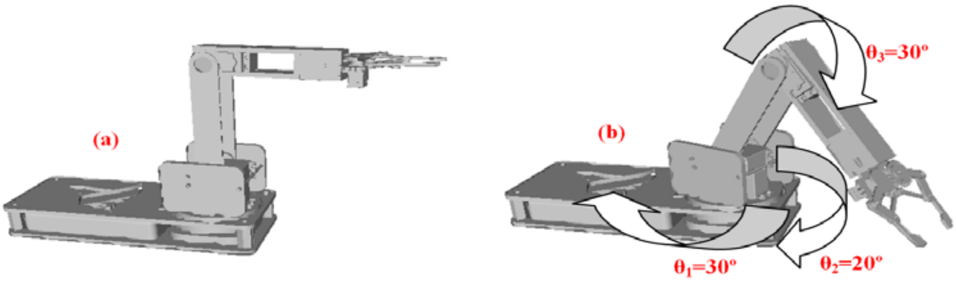


Fig. 5. Simulated robotic arm, (a): rotation of arm at initial position, (b) arm in full action.

The entire 3 DoF robotic arm is controlled as follows, at first, the robotic arm is at or comes to initial position ($\theta_1 = \theta_2 = \theta_3 = 0˚$), when the subject gazes at stimulus 1 If the subject looks at stimulus 4, the three arbitrary angle values $\theta_1 = 30˚$,$\theta_2 = 20˚$and $\theta_3 = 0º$ can be communicated. Intermediate working allows only θ_1 and θ_2 to work for stimulus 2 and θ_2 and θ_3 for stimulus 3. Between each stimulus a time delay of 6 seconds is ensured to record the SSVEP signals accuractely. The entire process is repeated twice for better accuracy and audio feedback is initiated for two seconds and continued to be there for a time period of 6 seconds.

4. Results and Discussion

Seven Subjects of age group starting from 34 to 65 years have participated in the experimentation. The subject and simulation panel were at a distance of 50 cm apart, and the subject is allowed to focus on the stimulus 1 to 4. These stimulus flicker at 25 Hz frequency with predefined phase delays of 0˚, 90˚, 180˚, and 270˚ respectively. The SSVEP acquisition extracts the SSVEP signal for different stimuli, subject gazing at, amplifies and filters the signal. The amplified signal process through the LabVIEW platform, the required result is recognized and simulated robotic arm is controlled.

Table 1 shows the results obtained for the different subjects participating the experimentation. The ITR is a method to evaluate the performance classification of SSVEP based robot control system and is defined by the equation (6) below,

$$ITR_{ssvep} = \begin{bmatrix} (\log_2 N) + A \log_2 A + \\ (1-A)\log_2 \dfrac{(1-A)}{(N-1)} \end{bmatrix} X \frac{60}{B}, \qquad (6)$$

where N defined as the number of stimuli, A is accuracy in percentage and B is Command Transfer Interval (CTIin s/cmd). Out of seven subjects, four subjects showed good accuracy of 100 % and remaining three subjects showed accuracy of 87.5 %. The overall accuracy of the proposed system is with an accuracy of 94.60 %, with an ITR of 26.40 bits/min.

Table 1. Experimental results of SSVEP based robotic arm control.

Sub. (age)	**Error Value**		**t in Sec**	**Acc. % (A)**	**CTI s/cmd (B)**	**ITR bits/min**
	HT	TAT				
1(34)	--	--	27	100	3.37	35.60
2(39)	**180**	--	31	87.5	3.87	19.52
3(44)	--	--	29	100	3.62	33.15
4(49)	--	--	30	100	3.75	33.25
5(58)	--	--	31	100	3.87	34.28
6(64)		**180**	28	87.5	3.5	10.87
7(65)		**270**	32	87.5	4.0	18.88
Average			**29.70**	**94.60**	**3.7**	**26.4**

The average values obtained in this work are compared with results obtained from similar works and are tabulated as shown in Table 2. The identified error values are due to, the subjects are not familiar in the experimentation process or either placement of electrodes resulting in reduction in SNR By using phase coding method, the current work has proven good accuracy and ITR as compared other works.

Table 2. Comparison of experimental results for phase coding with results obtained from similar works.

Ref.	Subjects	Mean accuracy [%]	Mean ITR [bits/ min]	Robot type
[14]	15	92.78	15	Robotic arm
[15]	15	91.35	20.69	NA
[18]	4	89.20	24.67	NA
[22]	86	92.26	17.24	E-puck
[23]	7	90.3	24.7	NAO
[24]	11	91.36	NA	E-puck
[25]	7	88.80	NA	Pioneer 3-DX
[26]	3	73.75	11.36	Lego Mindstorm
[27]	5	84.4	11.40	NAO
[28]	61	93.03	14.07	MRC
[29]	10	80	NA	NA
[30]	15	90.91	24.94	robotic arm
[31]	NA	90	NA	Robotic arm
This work	**7**	**94.60**	**26.40**	**3 DoF robot arm**

5. Conclusion

This work demonstrates to apply SSVEP signals for control of robotic arm in real time, by using handmade visual stimulation panel with Arduino microcontroller to evoke subject's SSVEP signal. Instead of using conventional EEG headsets and acquisition system, this work aims to use the designed, developed and validated customized SSVEP acquisition system. This work also focuses on LabVIEW and USB DAQ to acquire the subject's SSVEP signal and apply signal processing algorithms to identify the phase. Finally the experimental results proves effectiveness of system with obtained results. The current work is an extension of [31] and graphical user interface provides easy to interface between SSEVP signals and robotic arm. Since the results are promising, this work can be extended further to combine different neurological signals to control external devices, providing an end – end solution for patients suffering from different types of neurological disorders.

References

[1]. J. d. R. Millan, et al., Combining brain-computer Interfaces and assistive technologies: State-of-the-art and challenges, *Frontiers in Neuroscience,* Vol. 4, Issue 11, 2010, 161.

[2]. R. Leeb, L. Tonin, M. Rohm, L. Desideri, T. Carlson, J. del R. Millan, Towards independence: A BCI telepresence robot for people with severe disabilities, *Proceedings of IEEE*, Vol. 103, Issue 6, 2015, pp. 969-982.

[3]. P. Ofnerand Gernot, R. Muller-Putz, Using a noninvasive decoding method to classify rhythmic movement imaginations of the arm in two planes, *IEEE Transactions on Biomedical Engineering*, Vol. 62, Issue 3, 2015, pp. 972-981.

[4]. W.-M. Chen, H. Chiueh, et al., A fully integrated 8-channel closed- loop neural-prosthetic CMOS SoC for real-time epileptic seizure control, *IEEE Journal of Solid state Circuits*, Vol. 49, Issue 1, 2014, pp. 232-247.

[5]. M. Mirzaei, M. Tariqus Salam, D. K. Nguyen, M. Sawan, A fully-asynchronous low-power implantable seizure detector for self-triggering treatment, *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 7, Issue 5, 2010, pp. 125-133.

[6]. J. d. R. Millan, F. Renkens, J. Mourino, W. Gerstner, Noninvasive brain-actuated control of a mobile robot by human EEG, *IEEE Transactions on Biomedical Engineering*, Vol. 51, Issue 6, 2004, pp. 1026-1033.

[7]. Y. W. R. Wang, X. Gao, A practical VEP-based brain-computer interface, *IEEE Transaction Neural System Rehabilitation Engineering*, Vol. 14, Issue 2, 2006, pp. 234-240.

[8]. P.-L. Lee, C.-L. Yeh, J. Y.-S. Cheng, C.-Y. Yang, G.-Y. Lan, An SSVEP based BCI using High Duty cycle visual flicker, *IEEE Transaction on Biomedical Engineering*, Vol. 58, Issue 12, 2011, pp. 3350-3359.

[9]. S. M. Torres Müller, T. F. Bastos-Filho, M. Sarcinelli-Filho, Using a SSVEP-BCI to command a robotic wheelchair, in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE'11)*, Jun. 2011, pp. 957-962.

[10]. J. L. Collinger, et al., High-performance neuroprosthetic control by an individual with tetraplegia, *Lancet*, Vol. 381, Issue 9866, 2013, pp. 557-564.

[11]. C. Mandel, T. Luth, T. Laue, T. Rofer, A. Graser, B. Krieg-Bruckner, Navigating a smart wheelchair with a brain-computer interface interpreting steady-state visual evoked potentials, in *Proceedings of the IEEE/RSJ Int. Conference Intell. Robots Syst.*, St. Louis, MO, 2009, pp. 1118-1125.

[12]. S. Dasgupta, M. Fanton, J. Pham, M. Willard, H. Nezamfar, B. Shafai, D. Erdogmus, Brain controlled robotic platform using steady state visual evoked potentials acquired by EEG, in *Proceedings of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ACSSC'10)*, 2010, pp. 1371-1374.

[13]. P.-L. Lee, H.-C. Chang, T.-Y. Hsieh, H.-T. Deng, C.-W. Sun, A brain-wave-actuated small robot car using ensemble empirical mode decomposition-based approach, *IEEE Transactions on Systems Man and Cybernetics A: Systems and Humans*, Vol. 40, Issue 5, 2012, pp. 1053-1064.

[14]. X. Chen, B. Zhao, Y. Wang, S. Xu, X. Gao, Control of 7-DoF robotic arm system using SSVEP based BCI, *International Journal of Neural systems*, Vol. 28, Issue 8, 2018, pp.171-181.

[15]. C.-Y. Chiu, A. K Singh, J.-T. King, Y.-K. Wang, C.-T. Lin, A wireless steady state visually evoked potential-based BCI eating assistive system, in *Proceedings of the International Joint conference on Neural Networks (IJCNN'17)*, 2017.

[16]. O. Friman, T. Luth, I. Volosyak, A. Graser, Spelling with steady state visual evoked potentials, in

Proceedings of the 3rd International IEEE/EMBS Conference on Neural Engineering (CNE'07), May 2007, pp. 354-357.

[17]. R. S. Sandesh, N. Venkatesan, LabVIEW based comparative study on frequency and phase content of SSVEP signal for control of robot hand/ar, in *Proceedings of the 9th IEEE/EMBS, Neural Engineering Conference*, San Francisco, USA, March 20-23, 2019.

[18]. K.-K. Shyu, P. L. Lee, M. H. Lee, M. H. Lin, R.-J. Lai, Y.-J. Chiu, Development of low cost FPGA based SSVEP multimedia control system, *IEEE Transactions on Biomedical Circuits and Systems*, Vol. 4, Issue 2, 2010, pp. 125-133.

[19]. J. A. Mercado, J. Herrera, A. de Jesus Pansza, Josefina, Analysis of medical-sensor signals, *IEEE Journal of Solid-State Circuits,* Vol. 48, Issue 7, 2013, pp. 1625-1637.

[20]. K. H. Lee, N. Verma, A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical sensors, *IEEE Journal of Solid State Circuits*, Vol. 48, Issue 3, 2013, pp. 1625-1637.

[21]. R. S. Sandesh, N. Venkatesan, LabVIEW based control of 5 digit anthropomorphic robotic hand using EEG signals, *International Journal of Biomedical Engineering and Technology-Inderscience,* Vol. 22, Issue 3, 2016, pp. 258-271.

[22]. A. Guneysu, H. Levent Akin, An SSVEP based BCI to control a humanoid robot by using portable EEG device, in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13)*, 2013, Osaka, Japan, pp. 6905-6908.

[23]. J. Zhao, W. Li, M. Li, Comparative study of SSVEP- and P300-based models for the telepresence control of humanoid robots, *PLoS ONE*, Vol. 10, Issue 11, 2015, e0142168.

[24]. C. Kapeller, C. Hintermüller, M. Abu-Alqumsan, R. Prückl, A. Peer, C. Guger, A BCI using VEP for continuous control of a mobile robot, in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'13)*, July 2013, Osaka, Japan, pp. 5254-5257.

[25]. P. F. Diez, V. A. Mut, E. Laciar, E. M. Avila Perona, Mobile robot navigation with a self-paced brain-computer interface based on high-frequency SSVEP, *Robotica*, Vol. 32, Issue 5, 2014, pp. 695-709.

[26]. K. Holewa, A. Nawrocka, Emotive EPOC neuroheadset in brain – Computer interface, in *Proceedings of the 15th International Carpathian Control Conference (ICCC'14),* Velke Karlovice, Czech Republic, 2014, pp. 149-152.

[27]. B. Choi, S. Jo, A low-cost EEG system-based hybrid brain-computer interface for humanoid robot navigation and recognition, *PLoS ONE*, Vol. 8, Issue 9, 2013, e74583.

[28]. P. Stawicki, F. Gembler, I. Volosyak, Driving a semiautonomous mobile robotic car controlled by an SSVEP-based BCI, *Computational Intelligence and Neuroscience*, Vol. 2016, July 26 2016, pp. 1-14.

[29]. Y. Li, G. Bin, X. Gao, B. Hong, S. Gao, Analysis of phase coding SSVEP based on canonical correlation analysis (CCA), in *Proceedings of the 5th International IEEE/EMBS Conference on Neural Engineering (NER'11)*, 27 April-1 – May 2011, pp. 368-371.

[30]. C.-T. Lin, C.-Y. Chiu, A. K Singh, J.-T. King, Y.-K. Wang, A wireless multifunctional SSVEP-based brain computer interface assistive system, *IEEE Transactions on Cognitive and Developmental Systems*, Vol. 11, Issue 3, 2019, pp. 375-383.

[31]. H. Çiğ, D. Hanbay; F. Tüysüz, Robot arm control with for SSVEP-based brain signals in brain computer interface, in *Proceedings of the International Artificial Intelligence and Data Processing Symposium (IDAP'17)*, Malatya, Turkey, 2017, pp. 1-8.

[32]. C.-Y. Chiu, A. K Singh, J.-T. King, Y.-K. Wang, C.-T. Lin, A wireless steady state visually evoked potential-based BCI eating assistive system, *International Joint Conference on Neural Networks (IJCNN'17)*, 2017, pp. 3003-3007.

(013)

Energy-based Optimization for Resource Limited Neural Network Accelerators with Fused-layer Support

Simon Friedrich [1], Robert Wittig [1], Emil Matúš [1] and Gerhard P. Fettweis [1, 2]
[1] Vodafone Chair Mobile Communications Systems
[1] Technical University Dresden, Helmholtzstr. 18, 01069 Dresden, Germany
[2] Barkhausen Institut, Würzburger Str. 46, 01187 Dresden, Germany
E-mail: {simon.friedrich, robert.wittig, emil.matus, gerhard.fettweis}@tu-dresden.de

Summary: Deep Neural Networks (DNNs) show high accuracy for several artificial intelligence tasks. However, resource limitations present major challenges for designers of specific DNN accelerators. On-chip memory needs a high amount of chip area, whereas external memory introduces off-chip transfers which consume significantly more energy. Different memory schemes have been proposed in the literature to shift the trade-off point between off-chip transfers and on-chip memory. Furthermore, the fusion of different layers also leverages this trade-off. However, the impact on the total energy consumption was not considered so far. In this paper, we extend the analysis with an energy model to minimize energy consumption while using the lowest possible amount of on-chip memory. For example, we see a memory reduction of 36 % at an unchanged energy consumption by using 14 fused-layers in *ResNet-50*.

Keywords: Neural network accelerator, Fused-layer CNN, On-chip memory, Memory bandwidth, Energy estimation.

1. Introduction

Deep Neural Networks (DNNs) are widely used within computer vision tasks such as image classification. Dedicated hardware accelerators have been developed to extend the use cases to mobile devices [1]. Since these accelerators do not need to perform DNN training, their design is focused on inference. Therefore, the energy consumption per inference constitutes a key performance indicator. Since the amount of on-chip memory is limited, the data movement mainly contributes to the total energy consumption[2] [2].

In order to parameterize an accelerator efficiently, designers have to trade-off between different memory schemes and the size of the on-chip memory. To locate an optimal design point, we introduce an energy model for on- and off-chip data transfers. This enables us to pinpoint the optimal size of the on-chip memory while reducing the total energy consumption. Furthermore, we are also able to locate the best number of fused-layers for a given architecture.

2. Related Work and Problem Definition

State-of-the-art DNNs mainly consist of convolutions. Their calculations contain several loops which can be executed in different orders. Possible memory schemes exploiting the different computation orders have been analyzed in [3]. Moreover, the effect of on-chip memory sizes and off-chip data transfers was presented as well. An estimation of the energy consumption of a row-stationary dataflow and a given on-chip memory size was done in [4]. The off-chip transfers were further reduced by fusing convolution layers [5]. Consecutive layers are not calculated completely sequentially but the output features of the previous layer are directly reused within the next layer. Several improvements of the fused-layer algorithm have been introduced in the literature [6, 7].

However, the energy consumption of the mentioned memory schemes combined with layer-fusion has not been investigated yet. Therefore, an analysis and comparison of the on-chip memory and energy consumption exploiting different numbers of fused-layers is done in this paper.

The remainder of this paper is organized as follows: In Section 3, the layers of DNNs are briefly described. Section 4 introduces an energy model and the memory schemes that are further investigated. In Section 5, we present the simulation results of the needed on-chip memory and energy consumption. A conclusion is drawn in Section 6.

3. Layers within Deep Neural Networks

Convolution layers are widely used within DNNs. Their working principle is illustrated in Fig. 1. In convolution layers, the feature map of layer l consists of three dimensions $(A_\mathrm{x}, A_\mathrm{y}, C)$ whereas $(A_\mathrm{x}, A_\mathrm{y})$ represent the horizontal and vertical dimensions and C indicates the number of channels. The weights of F filters with a size of (W_x, W_y, C) are applied to the input

[2] The energy required for computation is mainly dependent on the number of operations for a network and neglected in this paper.

feature map of layer l in order to calculate the corresponding output feature map of layer $(l+1)$. Each output feature $a(x,y,c)$ of layer $(l+1)$ is calculated according to equation (1)

$$\begin{gathered} a(x,y,c)^{[l+1]} = \\ = \sum_{j=0}^{W_x^{[l]}-1} \sum_{k=0}^{W_y^{[l]}-1} \sum_{i=0}^{C^{[l]}-1} w\,(j,k,i,c)^{[l]} \cdot \\ \cdot\, a\left(S_x^{[l]} \cdot x + j, S_y^{[l]} \cdot y + k, i\right)^{[l]}, \\ 0 \le x < A_x^{[l+1]}, 0 \le y < A_y^{[l+1]}, 0 \le c < F^{[l]} \end{gathered} \qquad (1)$$

The parameters S_x and S_y indicate the horizontal and vertical stride of the filters of the current layer. The colors in Fig. 1 highlight the calculations of two output neurons as an example. Each output neuron can be calculated independently from the others. Therefore, different orders of the computations can be applied. After calculating an output neuron, either the filter or the input features can be reused to calculate the next output neuron. In the context of this paper, we call the output of a hidden layer intermediate features.

In this paper, we assume that the bias addition and the nonlinear activation function are directly applied after the complete computation of an output neuron. Therefore, these operations do not need additional memory transfers. Moreover, the simulations of the memory schemes also take the other layers of DNNs that require on-chip memory and data transfers, such as pooling, depth wise convolution, and layer addition of shortcut paths into account.

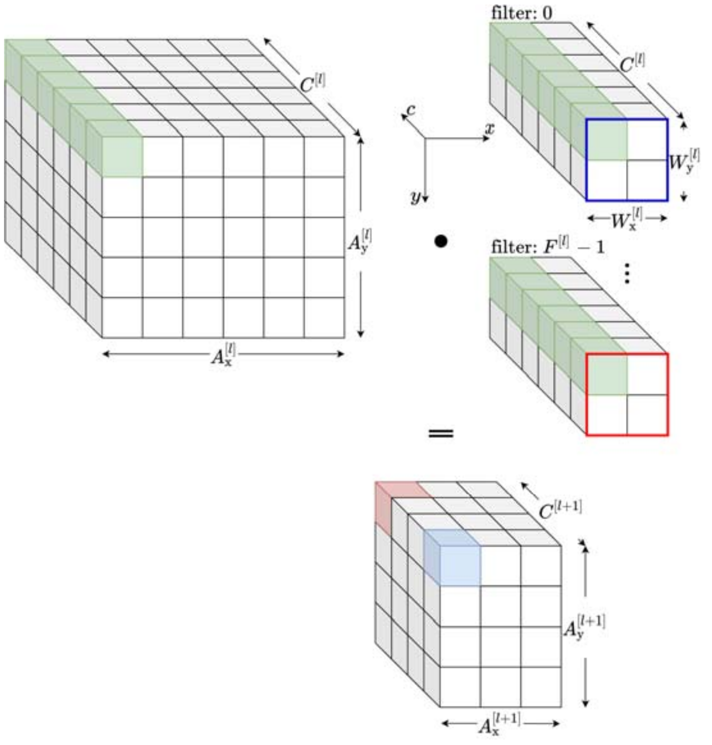


Fig. 1. Working Principle of a Convolution Layer with F Filters. The Colors of the Filters Match with the Resulting Output Neuron.

4. Memory Schemes and Energy Estimation

We assume a DNN accelerator with a hierarchical memory design. The processing elements of the accelerator have access to on-chip memory. As the size of this memory is limited in most designs, the on-chip memory can interact with an additional off-chip memory. In this context, the question of data partitioning and location with respect to cost optimization arises.

In this paper, we divide the inference of a DNN model into two separate parts. The first layers of the DNN are fused [5]. The number of fused-layers starts from the first layer and is variable in our simulation. For these layers, we assume that all filters and a sub-set of intermediate features are stored on-chip resulting in no contribution to the off-chip bandwidth. Otherwise, all filters would have to be reloaded for each computation of a new output line. For the intermediate features, we assume a line buffer approach as presented in [6]. According to Fig. 1, one line represents all horizontal neurons of a specific $y \in [0, A_y)$ of the feature map including all channels. The number of lines is the sum of the filter parameter W_y and the stride S_y.

The non-fused layers are calculated sequentially using different memory schemes. The main distinguishing factor is whether the filters and intermediate features reside on- or off-chip. In case filters are stored off-chip, we still require a small on-chip buffer to store a sub-set of filters for the current calculation. The size of this sub-set of filters depends on how many filters the hardware can compute in parallel. We also assume double buffering to load the next sub-set of filters during computation. In case intermediate features are stored off-chip, a sub-set of the input features of a layer must be stored in the on-chip memory as well. As described for the fused-layers, the additional memory size depends on W_y and S_y. With these assumptions in place, we can distinguish the following memory schemes for the non-fused layers. Table 1 summarizes the different memory schemes.

Full Stationary: Every filter of all layers and the maximum size of the input and output features fit in the on-chip memory. No access to off-chip memory during computation is needed.

Feature Stationary: Only a sub-set of filters and all input and all output features of one layer are stored on-chip. The maximum memory size of all layers is needed. The sub-set of filters is applied to all input features before the accelerator processes the subsequent sub-set.

Table 1. Storage Location of Different Memory Schemes of Non-Fused Layers

Stationarity	**Filters**	**Intermediate Features**
Full	on-chip	on-chip
Feature	off-chip	on-chip
Filter	on-chip	off-chip
Dynamic	on/off-chip	on/off-chip
None	off-chip	off-chip

Filter Stationary: The filters of the next layer must be loaded from off-chip memory during execution. Therefore, the maximum of all filters of a single layer and its subsequent layer must fit in the on-chip memory. Only a sub-set of input features is stored on-chip. All filters are applied to the sub-set of features before the next sub-set is used.

Dynamic Stationary: The storage location of each layer is chosen between the schemes *Feature Stationary* and *Filter Stationary* to minimize the required on-chip memory. Once *Feature Stationary* is chosen, the memory scheme does not change as the size of the filters increases with an increasing number of layers for most DNNs.

None Stationary: Only a sub-set of filters and input features is stored on-chip. All sub-sets of filters are applied to the same feature sub-set before the next features are processed. So, each sub-set of filters has to be loaded multiple times during computation of a layer.

The energy estimation is based on a model of a 45 nm process node of [8], which is applied in other publications about hierarchical memory accesses for accelerators [2]. The existing energy model describes the energy per data access for on- and off-chip memories. It does not include the static energy consumption of the on-chip memory. However, the relative energy costs are similar to other energy analysis of memory accesses of commercial process nodes [9]. We extra- and interpolated the given energy of the on-chip accesses using a logarithmic dependency on the on-chip memory size. In a synthesized design, the on-chip memory is divided into several banks of smaller standalone memories to contain functionalities such as concurrent read and write. In the used model, the energy consumption of an off-chip transfer is between 38 and 21 times larger than the on-chip energy consumption for an on-chip memory size in the range of 0.1 MB to 1.0 MB.

5. Simulation Results

Fig. 2 shows the simulation results of *ResNet-50* [10]. The size of the input image is 400×400 pixels. All simulations are done for an accelerator based on a MAC array calculating 16 channels of 16 output neurons in parallel. The accelerator uses an output stationary dataflow. However, the simulations could be extended to other dataflows as well. We divide the on-chip memory into 8 different banks and the word length of all operands is 8 bit. The simulations do not contain the memory consumption and data transfers of other metadata, e.g. instructions, because filters and features should represent the majority of data.

Fig. 2 (a) illustrates the on-chip memory accesses $D_{\text{on-chip}}$, which involve loading from and saving to the on-chip memory. For *Full Stationary*, the transfers only include the features and filters processed by the MAC array and its outputs. That represents the minimum of data transfers needed to compute a convolution on the accelerator. For the other memory schemes, features or filters have to be loaded from or stored to the off-chip memory. Off-chip data has to reside on-chip before further processing. Therefore, the off-chip transfers contribute to the on-chip memory accesses as well. As a result, the size of the on-chip transfers is larger than the mentioned minimum. This gap is reduced with an increasing number of fused-layers as fewer off-chip transfers occur. Compared to *Filter Stationary* and *Dynamic Stationary,* the on-chip transfers of *Feature Stationary* decrease slower with a higher number of fused layers as only filter data is stored off-chip.

Since the additional on-chip transfers bear a direct relationship to the occurring off-chip accesses, plotting these would be redundant. *Full Stationary* does not have any off-chip transfers. Moreover, *Feature Stationary* shows high reuse of the data stored in the on-chip memory as no intermediate features have to be stored off-chip. In contrast, the off-chip transfers for *None Stationary* are significantly higher.

In general, the size of the total on-chip memory, Fig. 2 (b), increases with a growing number of fused-layers because more filters need to be stored on-chip. However, *Feature Stationary* shows a decline for some fused-layers as the reserved on-chip memory for the intermediate features has to be adapted to the largest non-fused layer. As the intermediate features are getting smaller during interference this effect contradicts the growing amount of filters. Only taking the on-chip memory into account, *None Stationary* achieves the best results.

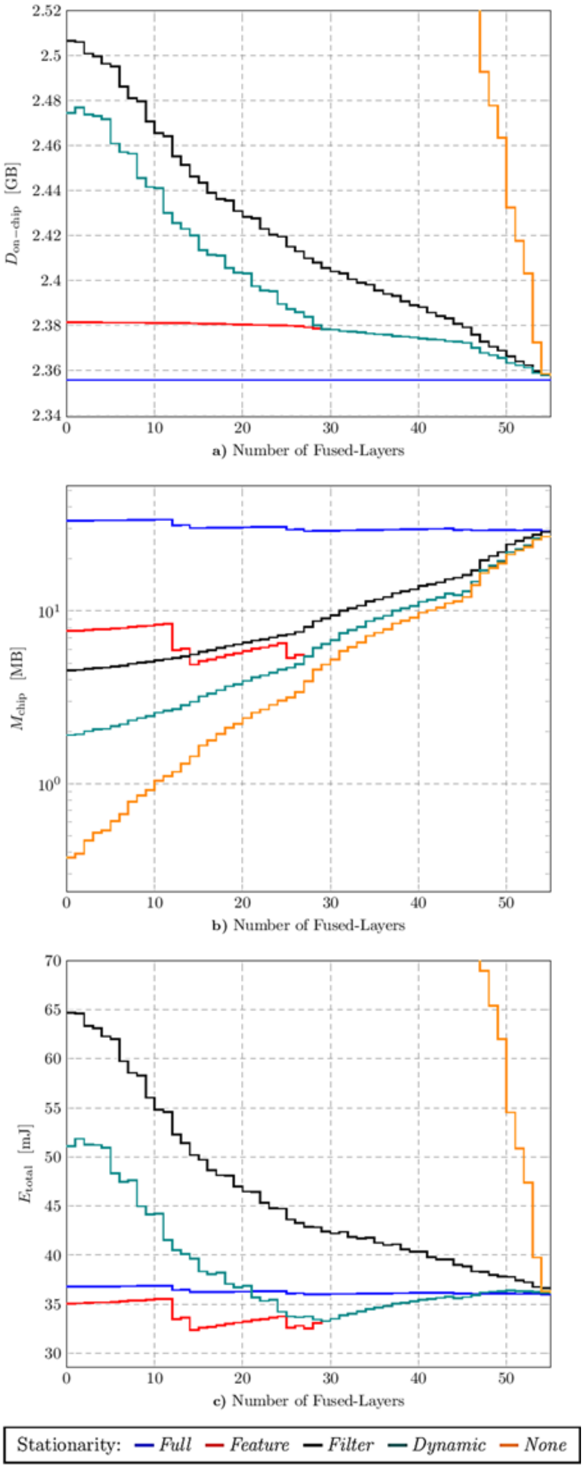


Fig. 2. On-Chip Memory Transfers, Size of On-Chip Memory and Energy Consumption of *ResNet-50* with Input Image Size of 400 × 400 Pixels.

Graph Fig. 2 (c) shows the total data movement energy consumption E_{total} during inference for the different memory schemes. Compared to Fig. 2 (b) *None Stationary* has the highest energy consumption (not shown for few fused-layers as it is around four times larger) due to the high amount of off-chip transfers. Also, *Filter Stationary* performs sub-optimal. Storing filters off-chip and intermediate features on-chip, *Feature Stationary,* results in the best results for a low number of fused-layers. These observations demonstrate the major trade-off between energy consumption and on-chip area. Optimizing for a small area footprint alone can lead to excessive energy consumption. However, M_{chip} can be reduced by 2.78 Mb without an increase of E_{total} by choosing *Feature Stationary* and 14 fused-layers. This corresponds to a memory saving of 36 % compared to using no fused-layers. Nearly the same values of M_{chip} and $\mathrm{E}_{\text{total}}$ can be reached for *Dynamic Stationary* as well. Since more fused-layers are needed in this scenario, *Feature Stationary* can be preferred for designs with limited support of fused-layers. Additionally, the determination of fused-layers should be investigated before implementation as a higher number of fused-layers not always results in a lower $\mathrm{E}_{\text{total}}$ but in a higher M_{chip}. Moreover, the energy consumption of *Dynamic Stationary* can be reduced by 35 % for using 29 fused-layers compared to the non-fused case.

It should be noted that the dimension of the compute array also affects the number of on-chip load/store operations because data can be reused more efficiently. In consequence, the results for the energy consumption can vary for the different cases. For example, it is possible that *Full Stationary* can achieve the highest energy efficiency for big array dimensions.

6. Conclusion

In this paper, we extended the analysis of different memory schemes with an energy model under the assumption of layer-fusion. Optimizing only for a small area footprint was shown to be sub-optimal in terms of energy consumption. Also, there often exists an optimum for the number of fused-layers, which is dependent on the network and the hardware topology. Hence this number should be a flexible design parameter for accelerating various networks and input sizes. As shown by an example, 36 % memory savings for *ResNet-50* can be reached without an increase in energy consumption.

Acknowledgement

This paper contains results of the 'ZuSE KI-mobil' project. It is being financed by the ’Zukunftsfähige Spezialprozessoren und Entwicklungsplattformen (ZuSE)’ initiative of the Federal Ministry for Education and Research of Germany.

References

[1]. E. Talpes, et al., Compute solution for Tesla’s full self-driving computer, *IEEE Micro*, Vol. 40, Issue 2, March 2020, pp. 25-35.

[2]. V. Sze, et al., How to evaluate deep neural network processors: TOPS/W (alone) considered harmful, *IEEE Solid-State Circuits Magazine*, Vol. 12, Issue 3, 2020, pp. 28-41.

[3]. K. Siu, et al., Memory requirements for convolutional neural network hardware accelerators, in *Proceedings of the IEEE International Symposium on Workload Characterization (IISWC'18)*, September 2018, pp. 111-121.

[4]. T.-J. Yang, et al., A method to estimate the energy consumption of deep neural networks, in *Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers*, October 2017, pp. 1916-1920.

[5]. M. Alwani, et al., Fused-layer CNN accelerators, in *Proceedings of the 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'16)*, October 2016, pp. 1-12.

[6]. Q. Xiao, et al., Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on FPGAs, in *Proceedings of the 54th ACM/EDAC/IEEE Design Automation Conference (DAC'17)*, June 2017, pp. 1-6.

[7]. K. Seto, et al., Small memory footprint neural network accelerators, in *Proceedings of the 20th International Symposium on Quality Electronic Design (ISQED'19)*, March 2019, pp. 253-258.

[8]. M. Horowitz, 1.1 Computing's energy problem (and what we can do about it), in *Proceedings of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC'14)*, February 2014, pp. 10-14.

[9]. Y.-H. Chen, et al., EYERISS: A spatial architecture for energy-efficient dataflow for convolutional neural networks, in *Proceedings of the ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA'16)*, June 2016, pp. 367-379.

[10]. K. He, et al., Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, June 2016, pp. 770-778.

(015)

Short-term Residential Load Forecasting with Multi-task Deep Neural Networks Sharing Between Different Users

Yuyao Chen [1, 2], **Christian Obrecht** [1], **David Sierra** [2], **Frédéric Kuznik** [1]
[1] Université de Lyon, CNRS, INSA-Lyon, Université Claude Bernard Lyon1, CETHIL UMR5008, 69621 Villeurbanne, France
[2] ELOW, 10 rue de la Pépinière, 75008 Paris, France
E-mail: yuyao.chen@insa-lyon.fr

Summary: Short-term residential load forecasting is an essential task for electricity load balancing. Due to the volatility of residential load, deep learning has gained success recently because of its ability to capture non-linear relationships within hidden layers. Several techniques such as clustering are added to enhance the performance by sharing the similarities between different users. However, this sharing mechanism mixes all information together inside the neural network without the specification of shared parts. Moreover, it cannot be adjusted by hyperparameters with different levels of similarities. This paper proposes a novel deep learning model more flexible and explainable with multi-task learning to share the similarities between different users provided with hyperparameter. Our deep learning approach combines residual network and long short-term memory recurrent neural network based on prior knowledge of load forecasting. The results show that our multi-task model outperforms single task models with clustering.

Keywords: Short-term load forecasting, Multi-task learning, Deep learning, Clustering, Long short-term memory, Residual connection.

1. Introduction

Short-term residential load forecasting is one of the most difficult task for energy balancing because of its volatility. Clustering based deep neural network has been tested as one of the effective approach to enhance the model performance by sharing the similarities [1]. However, the sharing mechanism in the neural network is ambiguous: similar users are trained together, so that the weights and bias are learned from a mixed dataset which does not distinguish the specific and shared parts between different users. Furthermore, such model cannot be adjusted by hyperparameters with different levels of similarities, for instance, the dynamic time wrapping (DTW) distance, and the ambiguity of the sharing mechanism makes the model less explainable. Therefore we propose a novel deep neural networks with multi-task learning to solve these problems. In this paper, we will compare the common approach of single task deep neural networks with clustering and our residual long short-term memory (LSTM) multi-task learning (MTL) neural network on the CER open data set [2].

2. Residual LSTM

The most common approach of load forecasting with clustering is to separate the clustering and prediction model into two stages: group similar users with clustering algorithm; then train different groups separately [3]. One of the most widely used clustering method for short-term load forecasting is K-means [4]. We adopt K-means as well and the DTW distance to measure the similarity in this paper.

The prediction model may be chosen among a wide variety of possibilities. As a matter of fact, many different architectures have been analyzed for short-term load forecasting: multi-layer perceptron [5], Convolutional Neural Network (CNN) [6], Gated recurrent unit (GRU) [3], LSTM [7], mix of CNN and LSTM [8], residual neural network [9], etc. In this paper, residual connection and LSTM are adopted as the baseline prediction model.

2.1. Residual Connection

Residual connection has been firstly proposed in ResNet [10] for the sake of training deeper neural networks. The reason is that the residual connection smooths the loss landscape of neural networks to make it easier to converge [11]. Because of its effectiveness, many different neural network architectures with residual connection have been designed for load forecasting: Kiprijanovska *et al.* [9] chose fully-connected layer as the inner structure of residual block while Gong *et al.* preferred LSTM as internal layer [12]. Besides, Temporal Convolutional Networks which has gained attention recently for load forecasting [13, 14] also adopts residual connection to ease the training.

Another reason to choose residual connection which has not been shown in the other references is the explanation of prior knowledge for load forecasting. Because the load profile of day *d+1* usually does not change much compared to the day *d*, so that in order to predict the real value of *d+1*, it is convenient to predict the residual of the day *d+1* to the day *d* where the identity shortcut of residual block serves the need.

2.2. LSTM

Recurrent Neural Networks (RNN) is one of the most popular neural network for sequence modeling thanks to the feedback connections to share parameters over time. However due to the vanishing and exploding gradient problem, the variant named gated RNN is more commonly used for long sequences. Short-term load forecasting is not only influenced by the variables at current time but also the past pattern, for instance, the profile of last 24 hours, which explains the popularity of two gated RNN for short-term load forecasting: LSTM and GRU [3, 7, 15, 16]. Therefore, in this paper, we select one of the most popular gated RNN, namely LSTM with residual connection as the baseline model (shown in Fig. 1) for the following comparison of single task learning and MTL. The logic of this residual LSTM block design is to learn the characteristics of residual load with LSTM before addition operation, and then add the identity shortcut to form the real load in order to share with the other load in the next stage. In other words, the sharing stage is manipulated between complete loads rather than residual loads.

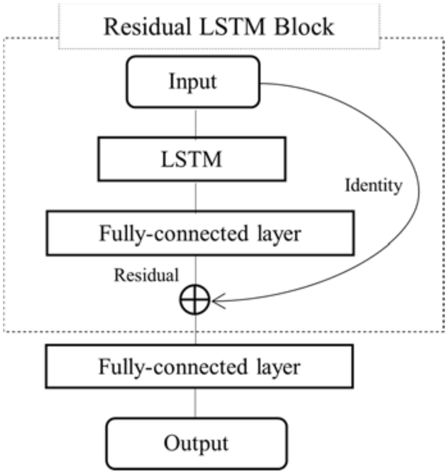


Fig. 1. Baseline model.

3. Multi-task Learning

The common approach with clustering belongs to the single-task learning category which has only one loss function to measure the error. The weights and bias are trained by the mixed data set, so the shared information are represented by these weights and bias but in an ambiguous way. The effectiveness of clustering [1, 3] inspires us to propose MTL as another sharing approach to improve the model performance but in a more clear way.

MTL is a subfield of machine learning that learns multiple tasks jointly to enhance the model performance. It meets success in not only natural language processing [17] but also computer vision [18]. However, the application of MTL in load forecasting is relatively rare: Gilanifar *et al.* [19] proposed regularization based MTL by sharing low-rank structures, which outperformed the clustering-based method and single task learning; Zhang *et al.* [20] investigated the multi-task gaussian process model for short-term load forecasting and showed its superiority over single task learning thanks to the shared hidden variables. MTL has also been used to learn multiple different loads such as electricity, cooling, heat, gas load at same time [21]. However, it is different from our sharing mechanism: they shared different sub loads rather than loads from different users, thus their loads' characteristics of different users are still mixed inside the model. To the best of our knowledge, MTL has never been used to share between different users with neural networks. Our proposed model is presented in Fig. 2 with two branches.

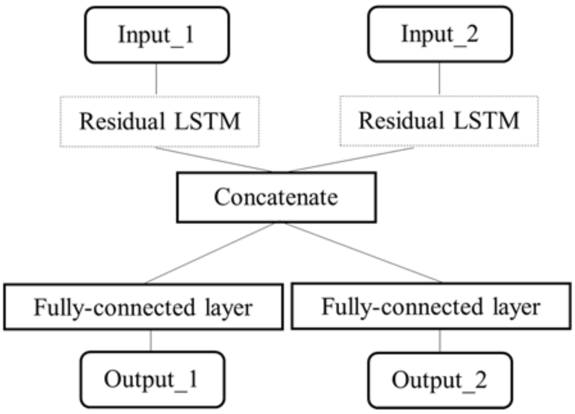


Fig. 2. Residual LSTM Multi-task learning.

These two branches allow each user to keep its specific characteristics before concatenation, and then the weight matrix of the last fully-connected layer keep their sharing information. The hyperparameters to tune the different level of sharing are the weights of the two loss functions, then the MTL loss is calculated by the weighted average of these loss functions. In the following part, we use *ratios* of the weights as the hyperparameter to tune the model.

4. Comparison on the CER Data Set

The CER dataset contains 4232 Irish residential consumers with three variables: meter ID, time and load from 1st July 2009 to 31st December 2010 with time interval of 30 min. In order to enrich the dataset, we preprocess the time variable to 4 variables 'day', 'day of week', 'month', 'hourmin' which encodes the hour and min variables into one integer with interval [1, 48]. The load is standardized during training, but the errors for comparison are recomputed on the original scale. Dataset is split into train set of the first 70 % dataset, validation set of the next 20 % and the last 10 % as test dataset for the prediction model. We aim to predict the next day profile with input sequence length of 48 and prediction horizon of 48.

The number of neurons for LSTM is 48 with *many-to-many* structure followed by a fully-connected layer of 5 neurons. We add the last fully-connected layer of 5 neurons after Residual LSTM block in order to control the variables of comparison with MTL model. These hyperparameters are the same with the MTL model. We set concatenation axis along the variable axis rather than the time axis, thus the output of concatenation layer is (48, 10) to multiply the weight matrix in the next fully-connected layer. Because we assume the residual load is a small value, we thus adopt zero initializer for the fully-connected layer. We use the Adam optimizer with learning rate 0.0001 and stop at the optimal epoch before overfitting. We implement our model with TensorFlow, Tslearn, Numpy and Pandas.

To illustrate the difference between the common approach and our proposed model, we follow firstly the common approach to cluster the dataset into four subsets by K-means and DTW distance (shown in Fig. 3).

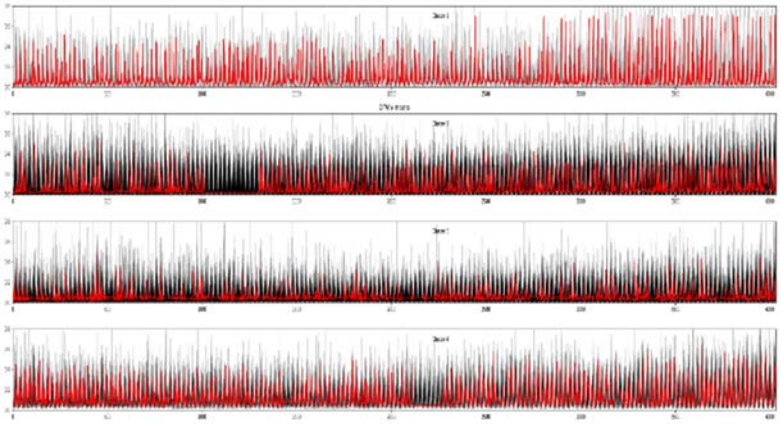


Fig. 3. Four classes clustered by DTW K-means.

For the sake of computation complexity, we choose two users randomly in one of the classes then train them on the baseline Residual LSTM model (shown in Fig. 1). This single task learning result is presented in Table 1 as 'single'.

In order to compare with single task learning, the same two users' loads are trained on the Residual LSTM MTL model (shown in Fig. 2) with different ratios of loss functions. Table 1 lists the results of Mean squared error (MSE), Mean absolute percentage error (MAPE) and Mean absolute error (MAE).

Table 1. Comparison of single task learning and multi-task learning with different ratios.

Ratio A/B	MSE A/B	MAPE A/B	MAE A/B
single	0.370/0.079	0.423/0.340	0.424/0.190
1.0/0	0.232/0.112	0.372/0.541	0.354/0.255
0.8/0.2	0.292/**0.055**	0.374/**0.304**	0.369/**0.166**
0.5/0.5	0.258/0.078	0.340/0.328	0.352/0.191
0.2/0.8	**0.177**/0.138	**0.322**/0.344	**0.307**/0.213
0/1.0	0.469/0.100	0.608/ 0.361	0.531/ 0.215

Ratio 1.0/0 is an extreme case that the weight of user A equals to 1 and the weight of user B equals to 0 which means that the MTL loss function is the same as the loss function of user A single task learning. And the ratio 0/1.0 means that no loss backpropagates to train load of user A. Figs. 4 & 5 detail 2 days prediction performance of user A (1059) and user B (1015). The ratio 0/1.0 of Fig. 4 and ratio 1.0/0 of Fig. 5 show clearly the untrained result.

Except the untrained neural network, all the other MTL with different ratios outperform single task learning which proves the effectiveness of MTL. By analyzing Figs. 4 & 5, MTL adds more power to imitate the pattern of the real load, it contains less lags than single task learning but it needs more capability to predict the peak value. Thus, our future work concentrates on improving peak prediction accuracy. Another interesting result is that the optimal ratio of user A is 0.2/0.8 with the smallest error and the optimal ratio of user B is 0.8/0.2. We think it may relate to the characteristics of load profiles and their correlation. But it needs more tests on different users to demonstrate our analysis which leads to our future research.

5. Conclusions

This paper proposes a novel deep learning approach with multi-task learning to share the similarities between different users. The comparison on the CER data set shows its superiority over single-task learning with the baseline residual LSTM considering the prior knowledge of load forecasting.

The proposed MTL architecture separates the specific and shared parts in neural network with another new hyperparameter *ratio* to tune the model for an optimal performance. The result shows that MTL can imitate the pattern of the load profile however it needs more capability to predict the peak value which leads us to future research. The choice of the optimal ratio requires further research as well.

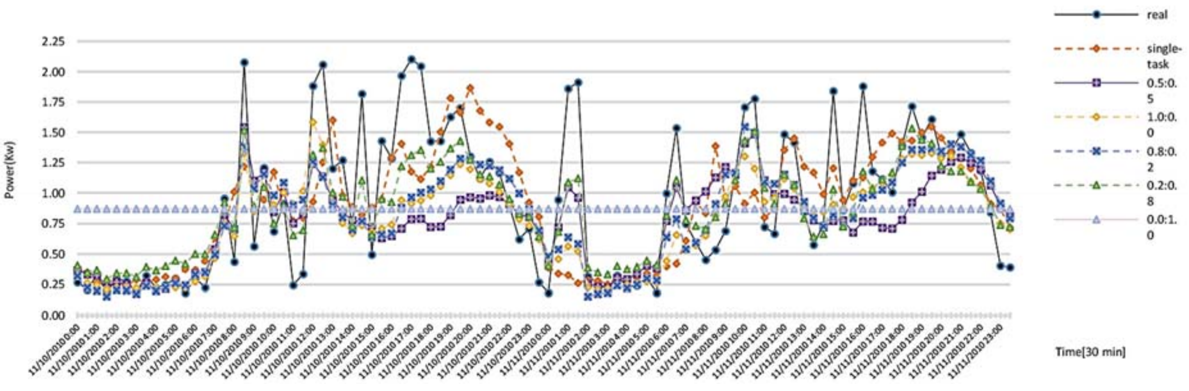


Fig. 4. Load comparison between single task and multi-task with different ratios (User 1059).

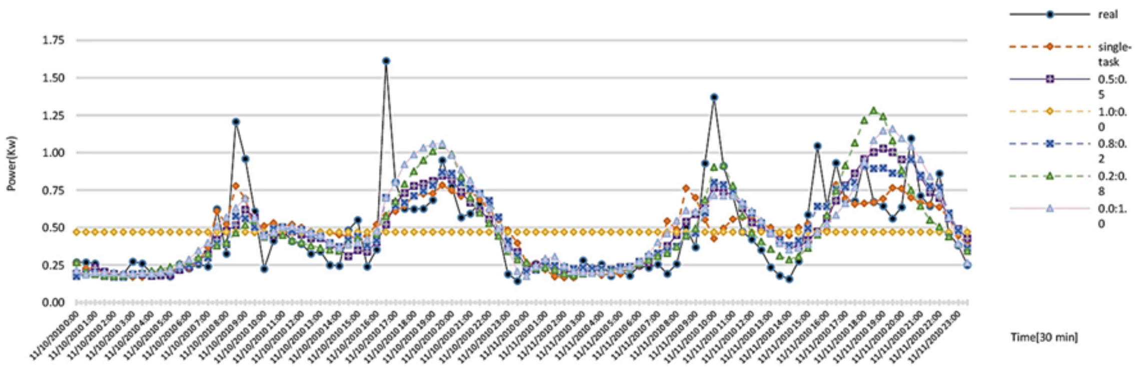


Fig. 5. Load comparison between single task and multi-task with different ratios (User 1015).

References

[1]. T. K. Wijaya, et al., Cluster-based aggregate forecasting for residential electricity demand using smart meter data, in *Proceedings of the IEEE International Conference on Big Data (Big Data'15)*, 2015, pp. 879-887.

[2]. Commission for Energy Regulation (CER). (2012). CER Smart Metering Project – Electricity Customer Behaviour Trial, 2009-2010 [dataset]. 1st Ed. Irish Social Science Data Archive. SN: 0012-00, https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[3]. Y. Wang, et al., Short-term load forecasting with multi-source data using gated recurrent unit neural networks, *Energies*, Vol. 11, Issue 5, 2018, 1138.

[4]. A. Rajabi, et al., A review on clustering of residential electricity customers and its applications, in *Proceedings of the 20th International Conference on Electrical Machines and Systems (ICEMS'17)*, 2017, pp. 1-6.

[5]. D. Kontogiannis, D. Bargiotas, A. Daskalopulu, Minutely active power forecasting models using neural networks, *Sustainability*, Vol. 12, Issue 8, 2020, 3177.

[6]. T. Jasiński, Modelling the disaggregated demand for electricity in residential buildings using artificial neural networks (deep learning approach), *Energies*, Vol. 13, Issue 5, 2020, 1263.

[7]. Y. Cheng, et al., PowerLSTM: power demand forecasting using long short-term memory neural network, in *Proceedings of the International Conference on Advanced Data Mining and Applications (ADMA'17)*, 2017, pp 727-740.

[8]. C. Tian, et al., A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network, *Energies*, Vol. 11, Issue 12, 2018, 3493.

[9]. I. Kiprijanovska, et al., HOUSEEC: Day-ahead household electrical energy consumption forecasting using deep learning, *Energies*, Vol. 13, Issue 10, 2020, 2672.

[10]. K. He, et al., Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016, pp. 770-778.

[11]. H. Li, Z. Xu, G. Taylor, C. Studer, T. Goldstein, Visualizing the Loss Landscape of Neural Nets, *NIPS*, 2018.

[12]. G. Gong, et al., Research on short-term load prediction based on Seq2seq model, Energies, Vol. 12, Issue 16, 2019, 3199.

[13]. F. Dorado Rueda, J. Durán Suárez, A. del Real Torres, Short-term load forecasting using encoder-decoder WaveNet: Application to the French grid, *Energies*, Vol. 14, Issue 9, 2021, 2524.

[14]. Y. Wang, et al., Short-term load forecasting for industrial customers based on TCN-LightGBM, *IEEE Transactions on Power Systems*, Vol. 36, Issue 3, 2020, pp. 1984-1997.

[15]. L. Wen, K. Zhou, S. Yang, Load demand forecasting of residential buildings using a deep learning model, *Electric Power Systems Research*, Vol. 179, 2020, 106073.

[16]. W. Kong, et al., Short-term residential load forecasting based on LSTM recurrent neural network, *IEEE Transactions on Smart Grid*, Vol. 10, Issue 1, 2017, pp. 841-851.

[17]. R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in *Proceedings of the 25th International Conference on Machine Learning (ICML'08)*, 2008, pp. 160-167.

[18]. A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in *Proceedings of the* Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[19]. M. Gilanifar, et al., Multitask Bayesian spatiotemporal Gaussian processes for short-term load forecasting, *IEEE Transactions on Industrial Electronics*, Vol. 67, Issue 6, 2019, pp. 5132-5143.

[20]. Y. Zhang, G. Luo, F. Pu, Power load forecasting based on multi-task Gaussian process, *IFAC Proceedings Volumes*, Vol. 47, Issue 3, 2014, pp. 3651-3656.

[21]. Z. Tan, et al., Combined electricity-heat-cooling-gas load forecasting model for integrated energy system based on multi-task learning and least square support vector machine, *Journal of Cleaner Production*, Vol. 248, 2020, 119252.

(016)

Deep Anomaly Detection Using Self-supervised Learning: Application to Time Series of Cellular Data

Romain Bailly [1, 4], Marielle Malfante [1], Cédric Allier [2], Lamya Ghenim [3], and Jérôme Mars [4]
[1] Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France
[2] Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France
[3] Univ. Grenoble Alpes, CNRS, CEA, INSERM, IRIG, Biomics, 38000 Grenoble, France
[4] Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France
E-mails: [1, 2, 3] firstname.name@cea.fr , [4] fistname.name@gipsa-lab.grenoble-inp.fr

Summary: We present a deep self-supervised method for anomaly detection on time series. We apply this methodology to detect anomalies from cellular time series. In particular, this study focuses on cell dry mass, obtained in the context of lens-free microscopy.

The method we propose is an innovative two-step pipeline using self-supervised learning. As a first step, a representation of the time series is learned thanks to a 1D-convolutional neural network without any labels. Then, the learned representation is used to feed a threshold anomaly detector. This new self-supervised learning method is tested on an unlabeled dataset of 9100 time series of dry mass and succeeded in detecting abnormal time series with a precision of 96.6 %.

Keywords: Self-supervised learning, 1D-CNN, Anomaly detection, Cellular anomaly, Time series, Lens-free microscopy.

1. Introduction

Lens-free microscopy is a recently developed imaging technique [1] overcoming some limitations of classical microscopy. Typically, it allows the rendering of thousands of cells in a single frame with a much less cumbersome device. [2] proposes to analyse sequences of images, from which a dataset of time series of cells' dry mass is built.

The dry mass of a cell, measured in picograms (pg), is related to its metabolic and structural functions. Amongst the thousands of cells in a Petri dish, it may happen that some cells deviate from their typical behaviour, thus influencing their dry mass. It has been shown that cells deviating from healthy trajectories can further drive tissues toward diseases [3]. Detecting abnormal cells automatically is thus crucial.

We propose an innovative method for automatically detecting abnormal cells using their dry mass. Using methods that do not need any manually labelled data is of particular interest especially in the case of time series processing. Indeed, while expert have a good understanding of what a normal cell behaviour is, there is no *a priori* knowledge of what an abnormal cell behaviour is. Working without labels is therefore interesting especially when the datasets are complex or not yet fully understood.

The proposed approach is in two steps: first, a representation of the time series is trained using self-supervised learning. In a second step, an anomaly detection block is used over the learned representation to determine if a cell is abnormal. This self-supervised method benefits from the representation power of deep learning without the usual labelling constraint.

2. Related Works

2.1. Anomaly Detection

Anomaly detection is a broad field of research focusing on the detection of abnormal patterns within a given set of data. We focus on anomaly detection on time series as presented in [4]. In particular, prediction-based anomaly detection techniques, which tries to predict the future of, time series. An outlier score [4] is computed between the prediction and the true value of the time series to determine if it is abnormal.

Multiple predictors can be used such as support vector regression [5], multilayer perceptrons (MLP) [6] or mixture transition distribution [7]. While [8] proposes a vector ARIMA to identify outlier points, other methods focus on discovering multiple outliers such as Gibbs sampling and block interpolation [9] or re-weighted maximum likelihood [10].

2.2. 1D-convolutional Neural Networks

Convolutional neural networks (CNNs) are mainly known for their success in computer vision with AlexNet [11], VGG16 [12] or ResNet [13], since the emergence of huge labelled datasets such as CIFAR100 [14] or ImageNet [15].

Because of the state of the art performances for computer vision achieved by 2D-CNNs, the signal processing community started to renew interest in 1D-CNNs, in the past few years for a wide variety of applications. They range from healthcare with ECG classification [16, 17] to fault detection [18-23] including audio and speech recognition [24] and other fields such as time series forecasting [25], or anomaly detection [26]. 1D-CNNs were introduced in the literature for the first time as Time Delay Neural Network (TDNN) in [27, 28].

2.3. Self-supervised Learning

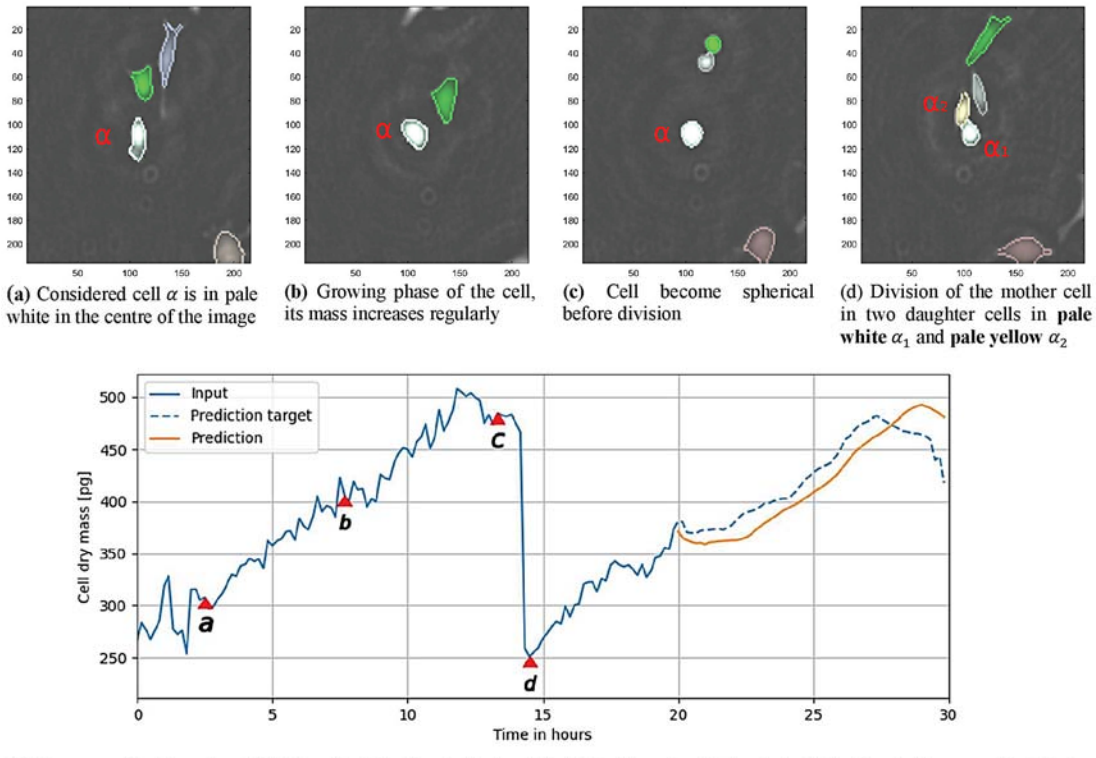
Self-supervised learning [29, 30] is a new training paradigm where supervised methods are used on an unlabeled dataset. The core idea is to automatically obtain a labelled dataset from the initially unlabeled dataset. A pretext task is associated to the self-labelled dataset and allows a supervised training of the neural network. [31] illustrates pretext tasks in computer vision: an input image is rotated $[0, 90, 180, 270]°$ and the neural network has to predict the rotation applied to the image. The network can only succeed if it has learned relevant visual features from within the images.

While a great deal of research exploiting pretext tasks can be found in the field of computer vision [32-36] very little of this work is related to time series processing, with the exception of some papers deeply linked to the temporality of the data. In [37, 38], a set of video images are given in a random order to the network that must order the frames. Finally, another time-related pretext task is presented in [39] where videos with modified playback speeds in range $[-5, +5]$ are given as inputs and the network must predict the playback speed.

3. Dataset

The acquisitions used in this study contain dry mass time series extracted from lens-free images of HeLa cells thanks to an upstream algorithm presented in [40]. A cell dry mass is a measure of how much the cell would weight if it had been deprived of its water. It is directly linked to the proteins content of the cell and is an indicator of its health.

Fig. 1 shows a normal cell behaviour on both the original images (Figs. 1a-1d) and the extracted dry mass time series in blue Fig. 1e. A typical track of dry mass contains a growing phase (Figs. 1a to 1c) where the dry mass increases regularly and a division phase (Figs. 1c to 1d) during which the mother cell is divided in two daughter cells of approximately equal sizes. The division appears as an abrupt decrease on the dry mass time series (between points *c* and *d* Fig. 1e).



(a) Considered cell α is in pale white in the centre of the image

(b) Growing phase of the cell, its mass increases regularly

(c) Cell become spherical before division

(d) Division of the mother cell in two daughter cells in **pale white** α_1 and **pale yellow** α_2

(e) Dry mass of cell number 143. The plain blue line is the input feed into the network, the dashed blue line is the ground truth to be predicted and the plain orange line is the network prediction. Red triangles *a*, *b*, *c* and *d* are respectively the timestamps of figures **1a**, **1b**, **1c** and **1d**

Fig. 1. Tracking of cell number 143 in pale white tagged α which has a normal behaviour. Cell grows (1a-1b) and becomes spherical (**1c**) before division into two daughter cells (1d). One of them is given the same id (143) while the other is given the next available id.

The dataset is split into train, validation and test sets in an 80/10/10 % distribution [41]. Each of those sub-dataset is augmented with window slicing [42]. Every full-length acquisitions is sliced into smaller ones. Every possible smaller time series are extracted from the full-length one *i.e.* there is a one-sample shift between two consecutive time series in the sub datasets.

4. Methods

4.1. Representation Learning Neural Network

The neural network used to learn a representation of the time series is trained in a self-supervised framework. Self-supervision allows the model to learn a deep representation of the signal without any labels. It uses a pretext task, to learn this representation. In our application and in agreement with the experts, we chose the pretext task to be **time series prediction** as presented Fig. 2. In this study, the input vector length is set to 120 time steps and the label vector to 60 time steps.

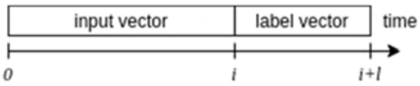


Fig. 2. Time series are split in an input vector of size i and label vector of size l.

A 1D-convolutional neural network architecture is used to capture the representation of the signal. The hyperparameter optimisation for the 1D-CNN representation learning neural network is presented Section 4.2.

The neural network is trained using a Root Mean Squared Error (RMSE) loss eq. (1) between the true future of the time series and the predicted one with y_n the ground truth value at time step n and $\widehat{y_n}$ the prediction value at time step n. Fig. 3 describes the full anomaly detection pipeline, including the representation learning neural network.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \widehat{y_n})^2} \tag{1}$$

Neural networks in this study are trained on a single NVIDIA Titan X with a batch size of 32, a learning rate of 0.001 and with ADAM optimizer.

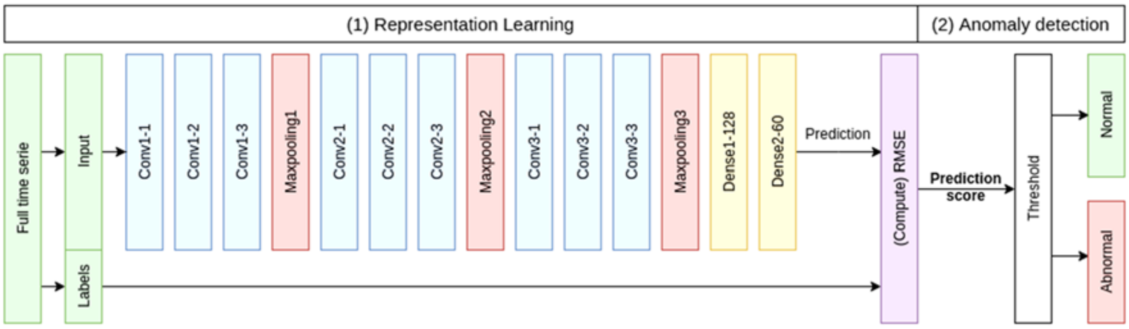


Fig. 3. Full anomaly detection pipeline. A 1D-CNN neural network is trained to predict the future of the time series. The RMSE between ground truth and prediction is compared to a threshold to define is a cell is abnormal.

4.2. Anomaly Detection

The proposed method relies on a second anomaly detection block. Experimental results have shown that the use of a threshold detector over the prediction RMSE allow the model to detect abnormal cells. The threshold τ is computed following eq. (2) such that the metric values outside the 95 % confidence interval of the metrics are flagged abnormal.

$$\tau = \mu_{test} \pm 2 \cdot \sigma_{test}, \tag{2}$$

where μ_{test} and σ_{test} are respectively the mean and standard deviation of RMSEs over the test set. We assume the metric distribution over a dataset to be Gaussian.

4.3. Evaluation

The proposed method is designed to analyse unlabeled datasets. Therefore, it is not possible to fully annotate the dataset nor to compute classical precision/recall curves. We propose an evaluation method based on the annotation on solely positives detections, *i.e.* time series raised as anomalies. The precision is computed following equation (3). While the whole dataset cannot be annotated to compute the recall, we propose an evaluation of the recall $\hat{R}$ equation (4) by labelling a random 5 % sample of the detected-normal cells (Negatives) to estimate the False Negative count.

$$P = \frac{TP}{TP + FP} \text{ (3) } \hat{R} = \frac{TP}{TP + \widehat{FN}} \tag{4}$$

5. Results

5.1. Representation Learning Neural Network

The definition of the best architecture hyperparameter is achieved through an empirical study. Multiples neural networks are trained on the pretext prediction task. All the convolutional layers contain 64 filters and a pooling layer is always added every 3 convolutional layers. The features extracted from the convolutional layers are then fed in a dense layer of 128 neurons. Table 1 shows the validation RMSE obtained for multiple architectures trained for this study.

Table 1. Architecture hyperparameters and their best MSE on the validation set. All the convolutional layers contain 64 filters and a pooling layer is always added every 3 convolutional layers.

# conv layer	kernel size	nb param	RMSE (pg)
3	3	155 708	87.726
3	8	196 988	94.053
3	16	263 036	92.035
3	32	395 132	85.677
3	64	659 324	84.608
3	120	1 121 660	83.089
5	3	82 108	97.824
5	16	295 932	93.178
5	64,32,16,8,4	282 620	93.706
9	3	303 548	80.941
9	**5**	**295 484**	**77.724**
9	8	393 980	85.643
12	3	266 876	81.877
12	5	357 116	86.096
12	6	402 236	88.993

The best 1D-convolutional neural network for the pretext task of prediction in the context of a cellular dry mass dataset is composed of 9 convolutional layers that contains 64 kernels of size 5. The RMSE on the test subset is computed to 76.62 pg.

Fig. 1e shows in blue the input given to the neural network, in dashed blue the ground truth to be predicted and in orange the network prediction. It shows on a specific example that the network is able to predict both a cell growing phase and a cell division.

5.2. Anomaly Detection

The anomaly threshold on RMSE on the test set is computed to $\tau = 230.87$ pg thus raising **208** abnormal tracks. From a fully applicative point of view, the anomalies raised allowed domains experts to identify four possible causes of anomalies:

True positives **TP**:

1. Cellular Anomaly (CA): The cell grows in an unexpected way and should be analysed.

2. Measurement Anomaly (MA): the upstream dataset generation software was not able to track the cell properly.

3. Measurement Anomaly because of a cellular anomaly (CMA): because of a CA, an MA occurred.

False Positives **FP**:

4. Prediction Anomaly (PA): the neural network was not able to predict the cell future correctly whereas the cell is normal.

The category distribution of those abnormal cells is detailed in Table 2. Then, 31 false negatives were counted during the annotation of 447 samples (5 %) of the cells predicted as normal. Anomaly detection has been achieved with a precision $P = 96.6$ % and an estimated recall $\hat{R} = 24.5$ %.

6. Conclusions

We propose an innovative two-step method for automatically detecting abnormal cells using their dry mass time series. This method focuses on unlabeled datasets thanks to the use of self-supervised learning. First, a representation of the time series is learned using a self-supervised 1D-convolutional neural network trained on a pretext prediction task. In a second step, the predicted dry mass value is compared to the ground truth. An anomaly is raised if the RMSE is above a given threshold. A precision of 96.6 % and an estimated recall of 24.4 % are achieved.

Table 2. Expert classification of the anomalies raised.

Anomaly	CA	CMA	MA	PA
Ratio	40 %	31 %	26 %	3 %
	97 %			3 %

References

[1]. T.-W. Su, S. Seo, A. Erlinger, A. Ozcan, High-throughput lens free imaging and characterization of a heterogeneous cell solution on a chip, *Biotechnology and Bioengineering*, Vol. 102, Issue 3, 2009, pp. 856-868.

[2]. C. Allier, *et al.*, Imaging of dense cell cultures by multiwavelength lens-free video microscopy, *Cytometry Part A*, Vol. 91, Issue 5, 2017, pp. 433-442.

[3]. N. Rajewsky, *et al.*, LifeTime and improving European healthcare through cell-based interceptive medicine, *Nature*, Vol. 587, Nov. 2020, pp. 377-386.

[4]. M. Gupta, J. Gao, C. C. Aggarwal, J. Han, Outlier detection for temporal data: A survey, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, Issue 9, Sep. 2014, pp. 2250-2267.

[5]. J. Ma, S. Perkins, Online novelty detection on temporal sequences, in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, Aug. 2003, pp. 613-618.

[6]. D. J. Hill, B. S. Minsker, Anomaly detection in streaming environmental sensor data: A data-driven modeling approach, *Environmental Modelling & Software*, Vol. 25, Issue 9, Sep. 2010, pp. 1014-1022.

[7]. N. D. Le, R. D. Martin, A. E. Raftery, Modeling flat stretches, bursts outliers in time series using mixture transition distribution models, *Journal of the American Statistical Association*, Vol. 91, Issue 436, Dec. 1996, pp. 1504-1515.

[8]. R. S. Tsay, D. Peña, A. E. Pankratz, Outliers in multivariate time series, *Biometrika*, Vol. 87, Issue 4, Dec. 2000, pp. 789-804.

[9]. A. Justel, D. Peña, R. S. Tsay, Detection of outlier patches in autoregressive time series, *Statistica Sinica*, Vol. 11, Issue 3, 2001, pp. 651-673.

[10]. A. Luceño, Detecting possibly non-consecutive outliers in industrial time series, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 60, Issue 2, 1998, pp. 295-310.

[11]. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks,

in Advances in Neural Information Processing Systems 25 (F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds.), *Curran Associates Inc.*, 2012, pp. 1097-1105.

[12]. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv Preprint*, arXiv:1409.1556, 2015.

[13]. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *arXiv Preprint*, arXiv:1512.03385, Jun. 2016.

[14]. A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, *University of Toronto*, 2009, p. 60.

[15]. O. Russakovsky, *et al.*, ImageNet large scale visual recognition challenge, *International Journal of Computer Vision (IJCV)*, Vol. 115, Issue 3, 2015, pp. 211-252.

[16]. S. Kiranyaz, T. Ince, M. Gabbouj, Personalized monitoring and advance warning system for cardiac arrhythmias, *Scientific Reports*, Vol. 7, Issue 1, Aug. 2017, 9270.

[17]. D. Li, J. Zhang, Q. Zhang, X. Wei, Classification of ECG signals based on 1D convolution neural network, in *Proceedings of the IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom'17)*, Oct. 2017, pp. 1-6.

[18]. O. Abdeljaber, O. Avci, M. S. Kiranyaz, B. Boashash, H. Sodano, D. J. Inman, 1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data, *Neurocomputing*, Vol. 275, Jan. 2018, pp. 1308-1317.

[19]. O. Avci, O. Abdeljaber, S. Kiranyaz, D. Inman, Structural damage detection in real time: implementation of 1D convolutional neural networks for SHM applications, *Structural Health Monitoring & Damage Detection,* Vol. 7, 2017, pp. 49-54.

[20]. L. Eren, T. Ince, S. Kiranyaz, A generic intelligent bearing fault diagnosis system using compact adaptive 1D CNN classifier, *Journal of Signal Processing Systems*, Vol. 91, Issue 2, Feb. 2019, pp. 179-189.

[21]. T. Ince, S. Kiranyaz, L. Eren, M. Askar, M. Gabbouj, Real-time motor fault detection by 1-D convolutional neural networks, *IEEE Transactions on Industrial Electronics*, Vol. 63, Issue 11, Nov. 2016, pp. 7067-7075.

[22]. A. Khan, D.-K. Ko, S. C. Lim, H. S. Kim, Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network, *Composites Part B: Engineering*, Vol. 161, Mar. 2019, pp. 586-594.

[23]. W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, *Mechanical Systems and Signal Processing*, Vol. 100, Feb. 2018, pp. 439-453.

[24]. A. van den Oord, *et al.*, WaveNet: A generative model for raw audio, *arXiv Preprint*, arXiv:1609.03499, 2016.

[25]. S. Du, T. Li, Y. Yang, S. Horng, Deep air quality forecasting using hybrid deep learning framework, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 33, Issue 6, 2019, pp. 2412-2424.

[26]. S. Yi, *et al.*, Interference Source identification for IEEE 802.15.4 wireless sensor networks using deep learning, in *Proceedings of the IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'18)*, Sep. 2018, pp. 1-7.

[27]. A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, Phoneme Recognition: Neural Networks vs. Hidden Markov models vs. Hidden Markov Models, https://www.computer.org/csdl/proceedings-article/icassp/1988/00196523/12OmNCdk2By

[28]. A. Waibel, Modular construction of time-delay neural networks for speech recognition, *Neural Computation*, Vol. 1, Issue 1, Mar. 1989, pp. 39-46.

[29]. A. Kolesnikov, X. Zhai, L. Beyer, Revisiting self-supervised visual representation learning, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, Long Beach, CA, USA, Jun. 2019, pp. 1920-1929.

[30]. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*, Vol. 119, Jul. 2020, pp. 1597-1607.

[31]. S. Gidaris, P. Singh, N. Komodakis, Unsupervised Representation Learning by Predicting Image Rotations, https://hal-enpc.archives-ouvertes.fr/hal-01864755

[32]. C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, *arXiv Preprint*, arXiv:1505.05192, Jan. 2016.

[33]. M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, *arXiv Preprint*, arXiv:1603.09246, Aug. 2017.

[34]. G. Larsson, M. Maire, G. Shakhnarovich, Learning representations for automatic colorization, *arXiv Preprint*, arXiv:1603.06668, Aug. 2017.

[35]. S. Jenni, P. Favaro, Self-supervised feature learning by learning to spot artifacts, *arXiv Preprint* , arXiv:1806.05024, Jun. 2018.

[36]. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, Context Encoders: Feature Learning by Inpainting, *arXiv Preprint*, arXiv:1604.07379, Nov. 2016.

[37]. H.-Y. Lee, J.-B. Huang, M. Singh, M.-H. Yang, Unsupervised representation learning by sorting sequences, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*, Vol. 1, Oct. 2017, pp. 667-676.

[38]. I. Misra, C. L. Zitnick, M. Hebert, Shuffle and learn: Unsupervised learning using temporal order verification, in *Proceedings of the European Conference on Computer Vision (ECCV'16)*, 2016, pp. 527-544.

[39]. H. Cho, T. Kim, H. J. Chang, W. Hwang, Self-supervised spatio-temporal representation learning using variable playback speed prediction, *arXiv Preprint*, arXiv:2003.02692, Mar. 2020.

[40]. C. Allier, *et al.*, Quantitative phase imaging of adherent mammalian cells: A comparative study, *Biomed. Opt. Express*, Vol. 10, Issue 6, Jun. 2019, pp. 2768-2783.

[41]. T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, *Springer*, 2009.

[42]. Z. Cui, W. Chen, Y. Chen, Multi-scale convolutional neural networks for time series classification, *arXiv Preprint*, arXiv:1603.06995, May 2016.

(017)

Object Detection Improvement with Morphological Top-hat and LIP (Logarithmic Image Processing) Model Applied to Thermal Images

Maxence Chaverot [1,2], Maxime Carré [2], Michel Jourlin [3], Abdelaziz Bensrhair [1] and Richard Grisel [4]
[1] INSA Rouen - LITIS – 685 Avenue de l'Université 76800 Saint-Étienne-du-Rouvray - France
[2] NT2I – 10 Rue Jean Servanton 42000 Saint-Étienne - France
[3] Hubert Curien Laboratory – 18 Rue Professeur Benoît Lauras, 42000 Saint-Étienne – France
[4] INSA Rouen – 685 Avenue de l'Université 76800 Saint-Étienne-du-Rouvray - France
Tel.: + 0033683753186
E-mail: m.chaverot@nt2i.fr

Summary: Thermal sensors are capable to acquire images in various conditions of weather or daytime, making it a powerful tool for video surveillance or Advanced Driver Assistance Systems, ADAS. The thermal radiation acquired by thermal sensors produces images of lower quality than visible sensors, causing tiny objects in the background to be hardly distinguishable. Enhancement algorithms can be applied to obtain a better image, but it doesn't lead to better object detection performance, particularly with convolutional neural network, CNN, object detectors. In this paper, we propose a method using morphological Top-Hat transform and Logarithmic Image Processing to improve object detection performance on Thermal Images.

Keywords: Thermal images, Top-hat operations, LIP, Object detection, Deep convolutional neural networks.

1. Introduction

Thermal Images low quality is due to their weak resolution and to their lack of texture information and details. Moreover, they commonly appear blurred because of heat radiation, atmosphere and sometimes misfocusing.

In such conditions, tiny objects in the background may be indiscernible, which is a real problem for object detection and dataset building. In the field of ADAS, detecting pedestrians and vehicles further permits to track them and estimate their trajectory or intention more precisely.

To improve the visual aspect of thermal images, spatial domain enhancement, histogram equalization, and frequency domain enhancement are commonly performed. More recently enhancement using CNN has been explored, mainly guided by visible images with the goal of increasing the spatial resolution of thermal images. However, these techniques improve the visual aspects of the thermal image, without showing a significant effect on object detection.

CNN are also widely used for object recognition, with constant renewal of state-of-the-art architecture outperforming the precedent. While such algorithms are generally efficient, their performances are tied to the quality and the quantity of data available. Multiple databases exist for visible images, but a few ones exist for thermal images. CNN are known to perform better with large databases. In our case, available thermal images are in much lesser quantities than visible ones.

When working with small datasets, data augmentation methods can be applied to extend the training data and increase detection performance. Standard techniques such as flipping, rescaling, warping, brightness, and contrast shifting are used to artificially augment the number of objects in the dataset. With the increasing research about CNN, architectures like Generative Adversarial Networks (GAN) have been developed to synthetize artificial images and increase further the quantity of data that can be used for training. Still, more data are needed to train these GAN, adding complexity to the solution.

In such tasks, where data are hard to gather, filter and label, we decided to explore pre-processing methods to enhance performance of already trained networks. Pre-processing methods to enhance CNN performance exist in various domains. Recently, such methods have been applied to enhance the likelihood prediction of COVID-19 in chest X-ray images [1]. In this paper, we propose a multi-scale morphological top-hat technic to improve YoloV4 [2] detection performance of tiny objects in the background of thermal images. This could be applied in various applications such as video surveillance, permitting to detect intrusion in a larger area.

2. Thermal Images Visual Enhancement Methods and Object Detection Performance: An Overview

In the literature, morphological operations to enhance contrast in images have been proposed first by Matheron and Serra [3], then several operations have been declined from this work. Top-hat transform has been used to enhance contrast by detecting bright and dark peaks of an image and adding them to the original image permits to obtain better contrasted images [4]. Morphological operations are highly dependent of the structural element used. Variation of its shape and size gives different outcomes to the operations. Real life scenes could be extremely varied, making the choice of the structural element difficult. Xiangzhi Bai et al. [5] and then Román et al. [6] proposed different multi scale Top-Hat transform algorithms to enhance thermal images, improving contrast and level of details while producing few noised regions. Unfortunately, they did not evaluate any object detection performance improvement.

In recent works, domain adaptation between visible and thermal images has been studied to improve performance detection. Devaguptapu et al. [7] proposed a Generative Adversarial Network to transform thermal images in visible grayscale ones and combining them into a multi-modality neural network. Later, Munir et al. proposed in a same way a Self-Supervised Thermal Network, SSTN [8], learning to maximize the information contained in extracted feature maps between thermal and visible spectra, which are used in CNN object detector. Such approaches permit to improve detection performance on thermal images. These methods require the training of subsidiary networks. To obtain the data needed to train these networks, a multimodality system is used, which is costly and complex, due to the necessity of aligning thermal and visible images. The set up and the processing time of such solutions could be considered as very expensive.

3. Methodology

3.1. FLIR Dataset

The FLIR ADAS Dataset [9] consists of thermal images collected in various day time and weather conditions, around the Santa Barbara region of California, USA. This dataset is composed of around 10.000 images, of which around 9.000 are labelled. Four classes are annotated: Persons, Cars, Bicycles and Dogs. The database constitution is detailed in Table 1. The Dogs class being underrepresented, it hasn't been considered in this paper.

Table 1. FLIR ADAS Classes Distribution on Train and Validation split.

	Person	Bicycle	Cars	Dogs
Train	13725	3297	36642	178
Validation	4955	441	5209	12

The images are provided in two different formats, 14 bits RAW without FLIR post-processing, and 8 bits JPEG after FLIR post-processing. The FLIR post-processing consists of multiple algorithms visually enhancing thermal images. These enhancements increase image noise and pixels values are highly dependent of the scene observed, potentially leading to lower performances of a CNN object detector. We decided to transform the 14 bits RAW images into 8 bits grayscale images with a dynamic expansion centered at a fixed mean of 0.5 and a fixed standard deviation of 0.25. It allows to obtain images close to those of the MS COCO dataset [10] used to pretrain the YoloV4 object detector.

3.2. YoloV4 Fine-tuned on Thermal Images

In a previous work, we have fine-tuned the YoloV4 object detector with thermal images issued of the FLIR ADAS dataset, leading to state-of-the-art results [11]. The YoloV4 object detector is one of the best detectors in term of detection performance and inference time trade off, as shown by Fig. 1.

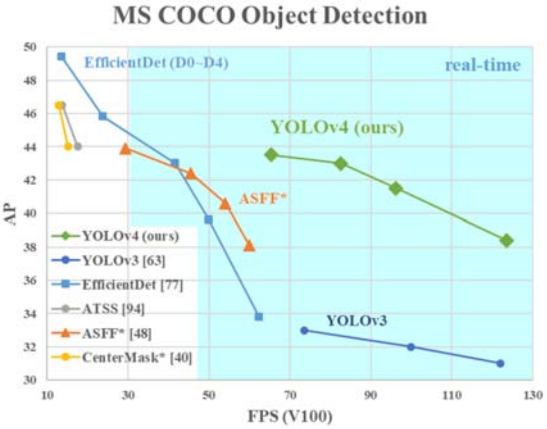


Fig. 1. Comparison of YoloV4 performance against state-of-the-art detectors [2].

The fine-tuning was done with thermal images of size 640*512 pixels. We used a batch size of 64, a stochastic gradient descent, SGD, as optimizer with a learning rate of 0.001. We selected the epoch with the best mean Average Precision, mAP, [12] score on the validation set, and we have kept the train and validation split provided by FLIR. Results obtained outperform the previous ones, see Table 2. This fine-tuned network will be used as detector in the present paper.

Table 2. Average Precision (%) between Faster R-CNN (Resnet-101) [13] (1.), SSD-512 (VGG-16) [14] (2.), and Fine-Tuned YoloV4 (3.) on FLIR ADAS validation set.

	Person	Bicycle	Cars	mAP	w-mAP
1.	54.8	42.76	67.99	55.18	60.77
2.	70.2	53.99	80.55	68.24	74.6
3.	**88.12**	**74.96**	**91.57**	**84.88**	**89.26**

3.3. Multi-scale Morphological Top-hat

One approach to enhance thermal images consists of applying White Top-Hat, WTH, and Black Top-Hat, BTH, transforms, and then adding or subtracting the results to the original image, according to Equation (1). It allows to extend information peaks in the white and in the black, resulting in a better contrasted image (cf. Fig. 2).

$$I^{'} = I + WTH_{k,h}(I) - BTH_{k,h}(I), \qquad (1)$$

where I' is the resulting image, I the original image, $WTH_{k,h}(I)$ the result of a White Top-Hat applied to I with a circular structural element of size k, $BTH_{k,h}(I)$ the result of a Black Top-Hat applied to I with a circular structural element of size k, and h a filter value such as:

$$WTH_{k,h}(I) = \begin{cases} WTH_k(I), WTH_k(I) \geq h \\ \qquad 0, WTH_k(I) < h \end{cases} \qquad (2)$$

This filter permits to ignore information peaks below a chosen value.

We remarked that depending on the value of k, the results of the fine-tuned YoloV4 inference on the image were varying, little objects in the background being detected although undetected by the object detector on the original image.

We propose a method consisting of inferring the object detector on multiple iterations of the Morphological Top-hat method, then applying a non-maximal suppression, NMS, algorithm to select the best bounding box for each object and obtain an overall improvement of detection performance.

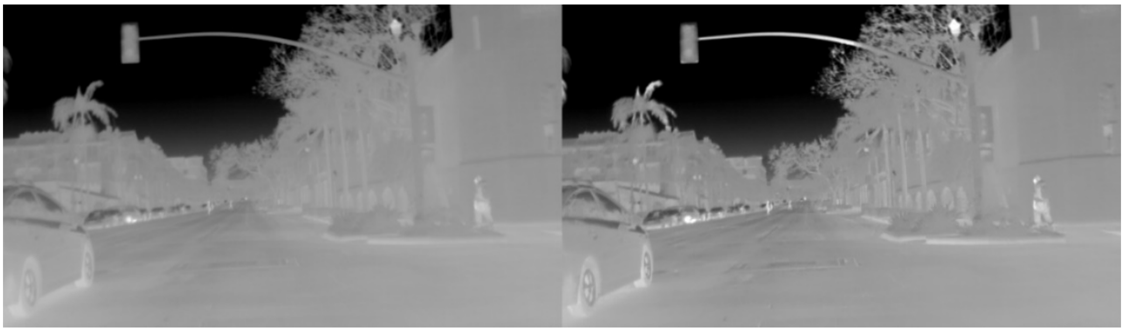


Fig. 2. Result of Top-Hat enhancement, left is original image, right is enhanced image.

3.4. Contribution of the Logarithmic Image Processing Framework

One drawback of these methods is that the addition and subtraction can lead to clipping due to an exceeding of the image range, [0-255] for 8-bits grayscale images, resulting in a loss of information. One solution consists of replacing the standard operations by addition and subtraction defined in the LIP framework. Originally defined by Jourlin and Pinoli [15] to process images in transmitted light, the LIP framework proposes addition and subtraction of two grey level images f and g according to:

$$f \,\triangle\!\!\!\!\!\!+\; g = f + g - \frac{fg}{M}, \qquad (3)$$

$$f \,\triangle\!\!\!\!\!\!-\; g = \frac{f-g}{1-(\frac{g}{M})}, \qquad (4)$$

with M the number of levels in a grayscale image (M = 256 for 8-bits images).

Equation (1) is modified to follow the LIP model requirements. Indeed, in the LIP framework, pixel intensities are inverted, with dark pixels values close to M and white pixels values close to zero. We need to invert the operation for $WTH_k(I)$ and $BTH_k(I)$. Equation (1) becomes:

$$I^{'} = I - WTH_{k,h} + BTH_{k,h}(I) \qquad (5)$$

Morphological top-hat operations applied in the LIP framework guarantee that the pixels values remain in the image range, resulting in more natural images.

4. Experiments and Results

4.1. Image Selection and Novel Annotation

The annotation provided with the FLIR ADAS dataset does not contain little objects in the background which are hardly distinguishable. Computing mAP scores with our novel detection is not accurate if they are not annotated. We decided to select images from the validation dataset, containing interesting scenes with vehicles and bicycles relatively far, and pedestrians crossing the street. See Fig. 3 for some examples. In addition, we have randomly selected other images from the validation set to obtain a test set of one hundred images. Full list of images used and novel annotation files are freely available on GitHub [16].

4.2. Results

We choose to apply the multiscale morphological method with parameter k varying from 1 to 13, by step of 2. We filter the results of the White Top Hat and Black Top Hat operations by a value of h = *5*, to minimize the noise effect of morphological operators which initially also detect the noise peaks. These values are what we found the best suitable for our test set.

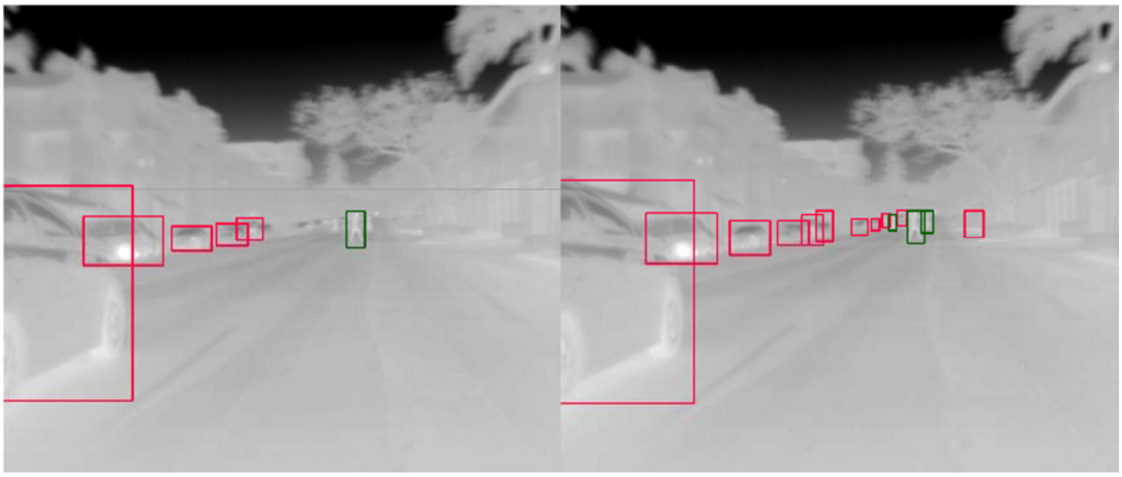


Fig. 3. Comparison of annotation, left is original, right is ours.

We applied the fine-tuned YoloV4 on the test set and we computed the mAP and the weighted-mAP. w-mAP, a mAP weighted by the number of labels by class defined by the following equation:

$$w\text{-}mAP = \sum_{i=0}^{N} AP_i * \frac{l_i}{L}, \qquad (6)$$

where *(i)* is the class index, *(AP_i)* is the Average Precision of the class, *(l_i)* is the number of labeled objects of the class (*i*) in the test set, and *(L)* the total number of labeled objects in the test set.

The w-mAP score is mathematically more precise for computing multi-class mAP. In our case, we use it because of the unbalanced number of labels in the FLIR ADAS validation dataset, see Table 1.

This detection and score calculation have been applied to images enhanced with Top Hat Transform in standard and LIP versions. The following Table 3. presents the average precision obtained from the different experiments.

Table 3. Average Precision (%) between YoloV4 (1.), YoloV4 on Top-Hat enhanced images (2.), and YoloV4 on LIP Top-Hat enhanced images (3.) on selected validation dataset.

	Person	Bicycle	Cars	mAP	w-mAP
1.	55.46	39.91	78	57.79	64.65
2.	57.93	35.15	77.48	56.85	**65.48**
3.	57.47	36.32	78.71	57.5	**65.84**

We obtained an improvement of 0.8 % of the w-mAP score with the standard method and an improvement of 1.2 % with the LIP one. Examples of improved detection can be seen in the Fig. 4.

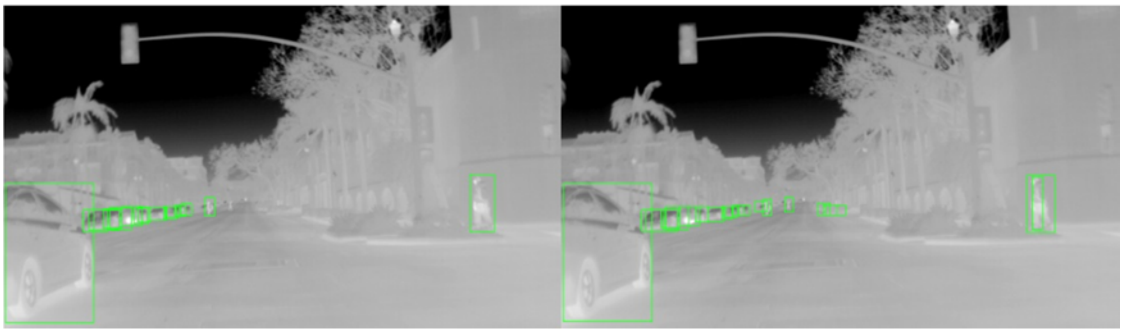


Fig. 4. Detection differences between YoloV4 on left, and our method on right. An additional pedestrian crossing the road and two additional parked cars are detected with our method.

This improvement appears small, but is already significant, being over the typical standard deviation we measure between several training of the YoloV4 network with the same data. One can say that inferring on multiple iterations of one image also increases the false detections, but the score increase shows that it benefits more to newly true positives than to added false positives. Lastly, the main drawback of our method concerns the multiplication of processing time by the number of iterations of the image we are inferencing. In our case, we have done it on seven iterations, meaning we have done seven inferences for only one frame. Such an increase of processing time could be a major drawback in ADAS, where processing power in embedded systems is scarce. But for use cases where we can easily scale up the power and use batch inference, the process could be done in real-time.

5. Conclusions and Perspectives

We have proposed a new method based on Morphological Top-Hat and LIP framework operations, improving detection performance of little objects in the background of thermal images. The contribution of the LIP framework permits to obtain the larger increase in performance score. This could open a research area for LIP application to Thermal images: in fact, the LIP framework proposes plenty of tools linked to contrast, gradients [17], and has been demonstrated consistent with human vision [18]. This method is a pre-processing before neural network execution, meaning it can improve the performance of already trained CNN, and can be useful in use cases where available data for training are difficult to obtain. This method can be improved by finding a solution to combine all the different iterations of the multi-scale morphological top hat enhancement into one image, resulting in computing only one inference per frame and obtaining a large gain in processing time. Finally, it would be interesting to test these methods with visible images and measure the performance gain in this case.

References

[1]. M. Heidari, S. Mirniaharikandehei, A. Z. Khuzani, G. Danala, Y. Qiu, B. Zheng, Improving the performance of CNN to predict the likelihood of COVID-19 using chest X-ray images with preprocessing algorithms, *International Journal of Medical Informatics*, Vol. 144, 2020, 104284.

[2]. A. Bochkovskiy, C. Y. Wang, H.Y. M. Liao, YOLOv4: Optimal speed and accuracy of object detection, *arXiv Preprints*, arXiv:2004.10934, 2020.

[3]. J. Serra, Image Analysis and Mathematical Morphology, Vol. 1, *Academic Press*, 1982.

[4]. P. Soille, Morphological Image Analysis: Principles and Applications, *Springer Science & Business Media*, 2013.

[5]. X. Bai, F. Zhou, B. Xue, Image enhancement using multi scale image features extracted by top-hat transform, *Optics & Laser Technology*, Vol. 44, Issue 2, 2012, pp. 328-336.

[6]. J. C. M. Román, H. Legal-Ayala, J. L. V. Noguera, Top-hat transform for enhancement of aerial thermal images, in *Proceedings of the 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI'17)*, 2017. pp. 277-284.

[7]. C. Devaguptapu, N. Akolekar, M. M Sharma, V. N. Balasubramanian, Borrow from anywhere: Pseudo multi-modal object detection in thermal imagery, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (*CVPR-W'19)*, 2019.

[8]. F. Munir, S. Azam, M. Jeon, SSTN: Self-supervised domain adaptation thermal object detection for autonomous driving, *arXiv Preprint*, arXiv:2103.03150, 2021

[9]. FLIR ADAS, https://www.flir.com/oem/adas/adas-dataset-form

[10]. T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C.L. Zitnick, Microsoft coco: Common objects in context, in *Proceedings of the European Conference on Computer Vision (ECCV'14),* 2014, pp. 740-755.

[11]. M. Chaverot, M. Carre, M. Jourlin, A. Bensrhair, R. Grisel, Object detection on thermal images: Performance of YOLOv4 trained on small datasets, in *Proceedings of the 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN'21)*, 2021.

[12]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, A. C. Berg, Imagenet large scale visual recognition challenge, *International Journal of Computer Vision*, Vol. 115, Issue 3, 2015, pp 211-252.

[13]. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, Issue 6, 2017, pp. 1137-1149.

[14]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, SSD: Single shot multibox detector, in *Proceedings of the European Conference on Computer Vision (ECCV'16)*, 2016, pp. 21-37.

[15]. M. Jourlin, J. C. Pinoli, Logarithmic image processing: The mathematical and physical framework for the representation and processing of transmitted images, *Advances in Imaging and Electron Physics*, Vol. 115, 2001, pp. 129-196

[16]. Selected Validation File List and Annotations, http://github.com/MaxenceChaverot/FLIR_ADAS_Val_Selected_Annotations

[17]. M. Jourlin, Logarithmic image processing: Theory and applications, Advances in Imaging and Electron Physics, Vol. 195, *Academic Press*, 2016.

[18]. J. Brailean, B. Sullivan, C. Chen, M. Giger, Evaluating the EM algorithm using a human visual fidelity criterion, in *Proceedings of Int. Conference on Acoustics Speech and Signal Processing*, 1991, pp. 2957-2958.

(019)

Deep Learning at the Edge: Performance Evaluation of Deep Learning Modelson the Edge Devices

Niloofar Baghdadi and **Jérémie Farret**
Inmind Technologies Inc., Mind in a Box Inc., Montreal, Canada
Tel.: + 1(514) 871-0470
E-mail: nbaghdadi@inmindtechnologies.com, jeremie@mindinabox.ca

Summary: With the consistent growth of the Internet of things (IoT), deep learning applications, and the recent formation of computing paradigms, especially edge computing, performance evaluation of various deep learning tasks on available devices is of great importance. However, there are caveats which concern the deployment of deep learning tasks on edge devices. Due to the low memory, storage, and performance of edge devices, some techniques need to be applied before model deployment. This article will evaluate TensorFlow Lite, TF-TRT, and TensorRT optimization techniques on Nvidia's Jetson TX2, Raspberry Pi 4, and HOMTOM S8 mobile device. The performance of devices is tested with computer vision and natural language processing tasks.

Keywords: Deep learning, Natural language processing, Computer vision, IoT, Edge computing.

1. Introduction

Due to an enormous amount of data generated and streamed by IoT devices, the need for real-time applications to analyze these data is increasing. Hence, exploiting deep learning, which past experiences proved to be successful and accurate, for various applications, including computer vision and natural language processing on the edge devices, is of great significance.

The advent and outbreak of deep learning applications are also impacting businesses in a variety of industrial sectors. Deep learning approaches also outperformed the traditional methods. However, high computational resources for the training and inference of deep learning models are required. Hence, leveraging cloud computing is a common approach to meet the required computational resources for deep learning model training. Nevertheless, moving the data from the edge layer to the cloud layer has some risks and challenges, such as latency, scalability, privacy [1].

The Internet of Things (IoT), fast streaming of data, and the rise in the need for real-time deep learning applications; necessitate data analytics close to the source of data to remove avoidable delays [2].

To address some of the above-mentioned issues, the edge computing paradigm can be helpful. Notwithstanding, training and inference of deep learning models on the edge devices are not simple due to limited memory and storage as well as computational performance. To make use of deep learning models at the edge layer, deployment of some techniques such as model compression and scale reduction, optimal deep learning implementation, and data and model distribution are required [3].

Each edge device has its architecture which needs a specific technique to be deployed on the deep learning models to run the inference task on the device.

This paper deployed two deep learning inference tasks on each edge device with optimization techniques corresponding to that device. The remainder of this paper is organized as follows: Section 2 provides the methodology, Section 3 describes the experimental protocol and architecture, Sections 4 and 5 discuss the experiments and associated results, section 6 shows the application architecture for this study, and Section 7 states the insights from this work.

2. Methodology

Processing the data at the edge enables real-time analysis as well as screening the data before transmission to the Fog or Cloud. In addition, the limited memory and storage of edge devices drive scientists and developers to experiment with various optimization techniques on machine learning models to deploy them on edge devices.

Among the variety of deep learning frameworks, the TensorFlow framework is used in this project which enables the more straightforward usage of TensorFlow Lite.

Applying optimization techniques to models can be helpful for devices with limited memory and computational power in addition to the acceleration of inference. Model size reduction can be beneficial when there is a smaller storage size, a need for smaller download size, and less memory usage. Some quantization can reduce computation to run the inference, resulting in lower latency, which can also impact the power consumption.

Moreover, some of the hardware accelerators can run inference faster if the models are correctly optimized. Each of the hardware accelerators requires a specific way of model quantization. However, optimization techniques come with some expenses, such as accuracy loss. TensorFlow supports optimization through quantization, pruning, and clustering, which pruning, and clustering are good for reducing model download size. Types of quantization when converting a model to TensorFlow Lite format are *post-training float-16*, *dynamic range*, *full integer quantization*, and *quantization aware training*.

In addition to TensorFlow Lite, some other optimization techniques specific to Nvidia's devices, such as TF-TRT and TensorRT, can optimize the TensorFlow Graph hence reducing the inference time. The section below describes the involved process to convert the TensorFlow model to each of the formats mentioned above and the achieved results are described.

3. Experimental Protocol and Architecture

The experimental setup proposes three Edge AI endpoint devices that are part of the ecosystem of a proprietary integrated fog / on-premises solution, Mind in a Box (M/B), supporting coordination, sensing and tuning, and real-time collection of the inference data (Deep Data).

In the course of this project, three devices have been utilized for bench-marking and comparison purposes. Raspberry Pi 4, Jetson TX2, and a HOMTOM S8 mobile phone with an Android operating system. The information about hardware specifications of each of the devices is shown in the Table 1.

4. Experiments and Results

At first, the operating systems for Jetson TX2 and Raspberry Pi 4, and then required environments as well as libraries to execute inference tasks installed. Object detection and natural language processing (NLP) tasks on the devices mentioned above were evaluated during this project.

To execute the tasks on devices, Python and TensorFlow framework is used. Due to limited memory and storage in edge devices, it is highly recommended to use compressed files for execution, like TensorFlow Lite flat buffer (.tflite) file. TensorFlow Lite is an optimization technique for on-device machine learning and addresses some limitations such as latency, privacy, connectivity, size, and power consumption.

TensorFlow Lite also supports multiple platforms, such as Android and iOS devices, embedded Linux, and microcontrollers. Therefore, all of the three devices mentioned above have been evaluated with TensorFlow Lite. Moreover, two other optimization techniques, like TF-TRT and TensorRT applied on Nvidia's Jetson TX2 device.

Below is a brief description of techniques and models used during this project.

4.1. Deep Learning Models

4.1.1. Object Detection

To convert a TensorFlow model to TFLite, TF-TRT, and TensorRT for the object detection task, a pre-trained model trained on COCO dataset was selected from TensorFlow model Zoo. For each object detector, a variety of feature extractor models can be selected. It is important to select a one-stage object detection model and a feature extractor with fewer parameters for real-time applications. Both one-stage object detection and feature extractor with less number of parameters lead to a faster detection rate. Hence among different choices, a Single Shot MultiBox Detector (SSD) [4] model with MobileNet-V2 [5] feature extractor has been selected for this project. SSD and MobileNet-V2 are types of Convolutional Neural Networks (CNN) with a few different functionalities. MobileNet-V2 uses depth-wise separable convolution, which is a lighter version for convolution operation. SSD uses the convolutional filters to predict object classes and runs on input mage only one time. The combination of SSD and MobileNet-V2 is an efficient CNN architecture that performs well on mobile and embedded devices. Fig. 1 depicts the SSD architecture with MobileNet-V2 backbone.

Table 1. Hardware specifications.

	Jetson TX2	Raspberry Pi 4	HOMTOM S8
CPU	Dual-core Nvidia Denver 2.0 and a quad-core ARM Cortex A57	4×Cortex-A72 1.5 GHz	4×1.5 GHz ARM Cortex-A53 and 4×1 GHz ARM Cortex-A53
GPU	256-core Pascal	Broadcom Vide oCore VI@500 MHz	ARM Mali-T860 MP2, 650×2
Memory	8 GB	8 GB	4 GB
OS	Ubuntu	Raspberry Pi OS	Android
Optimization	TF-Lite, TF-TRT, TensorRT	TF-Lite	TF-Lite

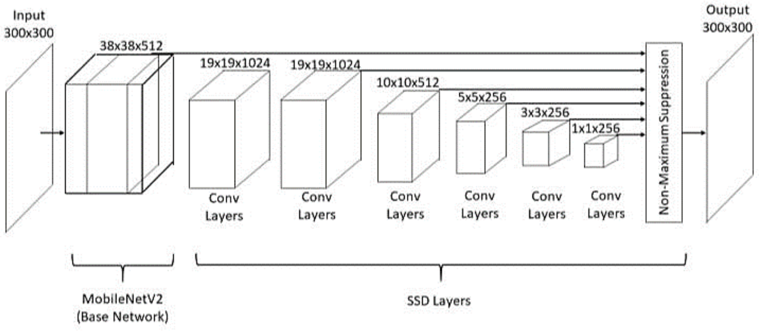


Fig. 1. SSD architecture with the MobileNet-V2 back- bone. Figure from [6].

4.1.2. Natural Language Processing

The inference task for natural language processing was also tested on the aforementioned devices with TensorFlow Lite optimization. For this purpose, Long Short-Term Memory (LSTM), which is a type of Recurrent Neural Network (RNN), is built and trained with the IMDB dataset. LSTM unit (shown in Fig. 2) in an RNN network allows the data to be processed sequentially, and it is suitable for classifying and making predictions on sequential data. The LSTM Recurrent Neural Network used in this project is shown in Fig. 3. The input length is set to 500 words, and 32 LSTM units are used in this network, including other required layers for the network. Afterward, the model was converted to TensorFlow lite format, and the performance of the model has been tested on the edge devices as shown in Fig. 4.

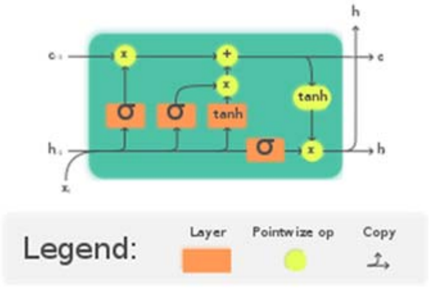
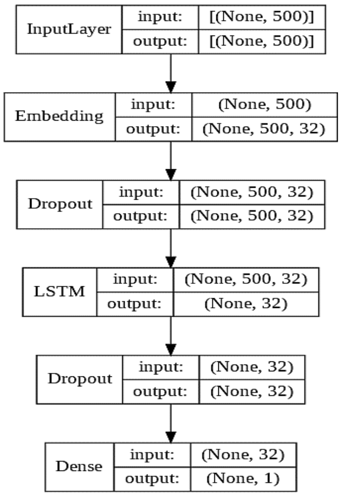


Fig. 2. LSTM Unit. Figure from Wikipedia.



InputLayer — input: [(None, 500)] — output: [(None, 500)]

Embedding — input: (None, 500) — output: (None, 500, 32)

Dropout — input: (None, 500, 32) — output: (None, 500, 32)

LSTM — input: (None, 500, 32) — output: (None, 32)

Dropout — input: (None, 32) — output: (None, 32)

Dense — input: (None, 32) — output: (None, 1)

Fig. 3. LSTM Recurrent Neural Network.

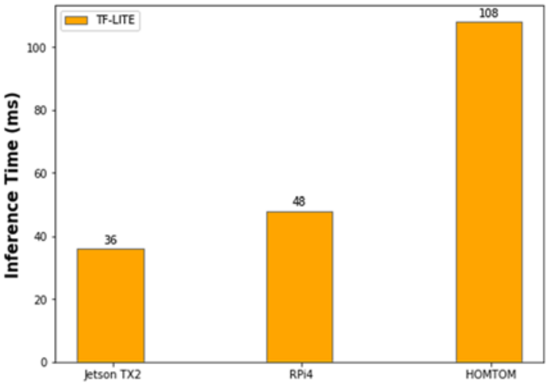


Fig. 4. Result of inference task on different edge devices using Natural Language Processing model.

4.2 Optimization Techniques

4.2.1. TensorFlow Lite (TFLite)

Model quantization is used as an optimization during the conversion process when converting the object detection model to TensorFlow Lite format. So, a single pre-trained object detection model converted to TensorFlow Lite format one time without any quantization and the other times with Dynamic Range, Float-16 and Full integer quantization techniques.

The performance, score, size, CPU, and GPU temperature for each of the converted models are shown in Table 2. The inference task was executed 200 times for each of the optimized models on Raspberry Pi 4 and Jetson TX2. Finally, the model with the lowest latency, CPU, and GPU temperature is kept for comparison with a high detection rate.

4.2.1. TensorFlow-TensorRT (TF-TRT)

TensorFlow™ integration with TensorRT™ (TF-TRT) is another optimization technique. During TensorFlow-TensorRT (TF-TRT) optimization, TensorRT performs and executes several transformations and optimization to the TensorFlow graph wherever possible. The unchanged part of the graph executes by TensorFlow. A saved model or frozen graph of the trained TensorFlow model is required to optimize the TensorFlow graph.

Table 2. Result of TensorFlow Lite on different edge devices using Object Detection model.

RPi4 / Jetson TX2				
Quantization / **Metrics**	**No-Quant**	**Dynamic Range**	**Float-16**	**Full integer**
Inference Time	447 / 474 ms	640 / 556 ms	442 / 470 ms	508 / 240 ms
Model Size	67 MB	17 MB	33 MB	17 MB
Detection Score	96 %	96 %	96 %	96 % / 94 %
TX2 GPU temp increase	2 °C	1.5 °C	1.5 °C	1 °C
CPU temp increase	5 / 3 °C	4 / 2 °C	5 / 2 °C	3 / 0.5 °C

Similar to TensorFlow Lite, TF-TRT can convert tensors and weights to lower precision during optimization. Precision can be set to FP32, FP16, or INT8. Nvidia's Jetson TX2's GPU architecture is not compatible with INT8 precision. Hence, the conversion only took place for FP32 and FP16 precisions. The result of executing the object detection model with FP32 and FP16 precision on Nvidia's Jetson TX2 is shown in Fig. 6.

4.2.3. TensorRT

TensorRT is a software development kit (SDK) for optimizing trained deep learning models, and it consists of an inference optimizer and a runtime engine that improves latency, throughput, and efficiency. Some of the TensorFlow operations are not supported by TensorRT, and they should be replaced with a custom layer plugin node using Onnx, Caffe, or UFF parsers.

In this project, the UFF parser has been used for replacing TensorFlow operations that TensorRT does not support. After that, the TensorRT engine was created, and the inference task was performed. Similar to TensorFlow Lite and TF-TRT, TensorRT can convert the model to lower precision during optimization. The result of running an object detection task on Nvidia's Jetson TX2 is shown in Fig. 6.

5. Result

As it is clear from the hardware architecture and performance of the selected tasks on the devices, Nvidia's Jetson TX2 has more capability than the other two devices. Also, Raspberry Pi 4 performs better than the HOMTOM S8 mobile device, as is shown in Table 3.

Table 3. Result of inference task on different edge devices using Object Detection model.

Device / **Model**	**JetsonTX2**	**RPi4**	**HOMTOMS8**
TensorFlow Lite	240 ms	508 ms	53fi ms
TF-TRT	165 ms	–	–
TensorRT	23 ms	–	–

Nonetheless, knowing the edge devices' performance on a specific task and the metrics associated with it, enables implementing efficient load balancing strategies, thus leveraging the Mind in a Box load balancing capabilities.

6. Applicable Architecture

The setup for this experiment is shown in Fig. 5. As shown in this setup, the real-time data is captured by the edge devices, and the inference tasks are performed in situ by the previously implemented models on edge. The data, result of inference, and information about devices' load, task, temperature, etc. are being sent from the Edge layer to the Mind in a Box in the fog layer. The Mind in a Box gains the capability to retrain the model with the newly captured data, and thus, to update it locally and / or on the edge devices. As for the load distribution itself, having access not only to the computational load to be processed and the queue on each edge devices, but also to multi-sensing information such as temperature, or cooling system activity, allows the Fog level device to implement advanced sensing and tuning strategies (Frequency, cooling system adjustments, etc.), in complement to applying conventional load balancing approaches.

In addition, the Mind in a Box retains the option to run the inference for the queued tasks in the Fog layer if all the devices at the Edge layer are busy or unavailable.

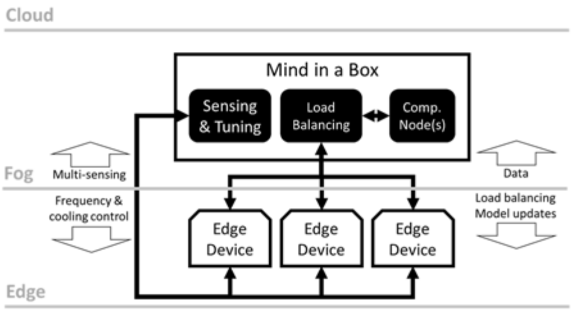


Fig. 5. Experimental setup.

7. Conclusion

In this project, the performance of two deep learning tasks, including object detection and natural language processing on edge devices evaluated. Moreover, Nvidia's Jetson TX2, Raspberry Pi 4, and HOMTOM S8 mobile devices have been assessed with the two deep learning tasks. Based on the evaluation performed in this research, developing advanced load balancing strategies to leverage a Fog level orchestration component, such as the one provided by the Mind in a Box solution, becomes straightforward.

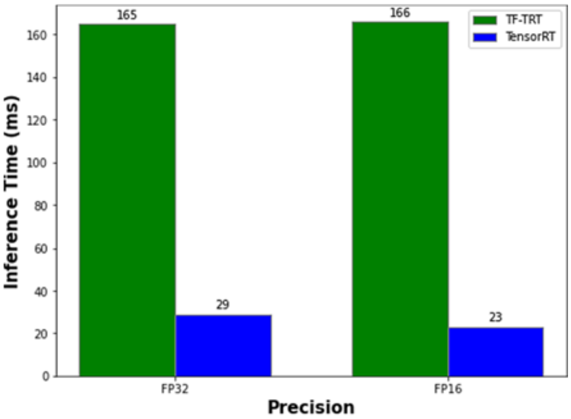


Fig. 6. Result of object detection task on Nvidia's JetsonTX2 using TF-TRT and TensorRT.

References

[1]. J. Chen, X. Ran, Deep learning with edge computing: A review, *Proceedings of IEEE*, Vol. 107, Issue 8, 2019, pp. 1655-1674.

[2]. M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, *IEEE Communications Surveys & Tutorials*, Vol. 20, Issue 4, 2018, pp. 2923-2960.

[3]. S. Voghoei, N. H. Tonekaboni, J. G. Wallace, H. R. Arabnia, Deep learning at the edge, in *Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI'18)*, 2018, pp. 895-901.

[4]. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, A. C. Berg, SSD: single shot multibox detector, http://arxiv.org/abs/1512.02325

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 4510–4520.

[6] T. J. Sheng, M. S. Islam, N. Misran, M. H. Baharuddin, H. Arshad, M. R. Islam, M. E. Chowdhury, H. Rmili, M. T. Islam, An internet of things based smart waste management system using LoRa and TensorFlow deep learning model, *IEEE Access*, Vol. 8, 2020, pp. 793-811.

(020)

Detection of Water Levels in Lake Cerknica Using Sentinel-2 Data and Symmetry

David Jesenko [1], Lukáš Hruda [2], Ivana Kolingerová [2], Borut Žalik [1] and David Podgorelec [1]
[1] University of Maribor, Faculty of Electrical Engineering and Computer Science,
Koroška cesta 46, SI-2000 Maribor, Slovenia
[2] University of West Bohemia, Faculty of Applied Sciences, Department of Computer Science and Engineering,
Technická 8, 301 00, Plzeň, Czech Republic
Tel.: +386-2-220-7476, fax: +386-2-220-7272
E-mail: david.jesenko@um.si

Summary: The Sentinel satellite constellation series, developed and operated by the European Space Agency, represents a dedicated space component of the European Copernicus Programme, committed to long-term operational services in the environment, climate and security. A huge amount of acquired data allow us different surveys. We decided to detect changes in water levels in Lake Cerknica. The multispectral index has been calculated from Sentinel-2 data, and transformed to a 3D point cloud. As shown by the results, symmetry measures of 3D point clouds could be used for the detection of water levels. Linear and quadratic functions have been fitted, and the results $R^2 = 0.9130$ and $R^2 = 0.9135$ have been achieved, accordingly.

Keywords: Sentinel-2, Lake Cerknica, Remote sensing, Water monitoring, Symmetry, Symmetry measure.

1. Introduction

Monitoring open water bodies accurately is an important and one of the basic applications in remote sensing. They are a significant part of the Earth's water cycle, and water bodies such as rivers, lakes and reservoirs are irreplaceable for the global climate system and the ecosystem. Remote sensing has become a conventional approach for monitoring water bodies, as it is real-time, dynamic and cost-effective [1]. The measuring and monitoring of surface water using remote sensing technology is, therefore, an essential topic [2]. In particular, the use of freely available high-spatial resolution optical satellite data is relevant [3]. Such data include the images obtained by the Landsat series [4-6], Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) [7, 8], and Sentinel-2 [1, 9] multispectral imagery. A high extraction accuracy has been achieved in the mapping of surface water bodies, including lakes [10], rivers [11], coastlines [12] and water bodies in rural areas [13.

Among all of the existing water body mapping methods, the calculation of a multispectral index is the most reliable, as it is user friendly, efficient and has low computational cost [14]. The use of the water index is currently accepted to enhance the differences between water and no water bodies, based on combinations of two or more spectral bands using various algebraic operations [15]. The well-known normalised difference water index (NDWI) [13] is sensitive to built-up lands, and frequently results in the overestimation of water bodies in urban areas [16]. The modified NDWI (MNDWI) [17] is used mostly in urban scenes to improve the separability of the built-up areas. The Automated Water Extraction Index (AWEI) [18] highlights the water bodies in urban areas over shadow and dark surfaces. It consists of two separate indices, one for areas where shadow is not important, and a second one, where shadow is significant.

The observed water unit in the presented paper is Lake Cerknica. Lake Cerknica is one of the largest intermittent lakes in Europe. It is located in the southwestern part of Slovenia, caught between the Javorniki hills and the Bloke plateau on one side, and Mount Slivnica on the other. It appears every year on the karst plain, and is present for about eight months of the year. Water usually spreads over a surface of 20 km^2, but, at its fullest, the lake covers a surface of about 26 km^2. The height above sea level is in the range from 546 m to the 551 m, with the maximum depth about 10 m. When full it is the largest lake in Slovenia [19]. The lake is an important wildlife resort, especially as a nesting place for many bird species. During the dry season the lake disappears, which enables hiking and grass mowing, but on the other hand, while present, allows for paddling and fishing. For these reasons it is crucial to detect water levels at different times of the year.

The methodology used in the presented paper is described in Section 2. The results are given in Section 3, while Section 4 concludes the paper.

2. Methodology

This section presents the methodology used in this paper to determine the water levels in Lake Cerknica using Sentinel-2 data and symmetry measure. Firstly, the Sentinel-2 data were acquired from the European Space Agency's (ESA) official website, then the multispectral index was calculated, followed by extraction of the area of Lake Cerknica from the multispectral index and calculation of the symmetry measure of the lake. The water level of Lake Cerknica is predicted finally. Each of these steps is explained additionally in the continuation. A flowchart of proposed method is represented in Fig. 1.

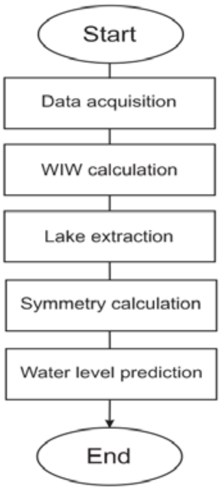


Fig. 1. Flowchart of the proposed approach.

Sentinel-2 data are composed of 13 spectral bands that range from the visible range to the shortwave infrared. Data are freely available on the ESA website, accessible at https://scihub.copernicus.eu. The multispectral index named the Water In Wetlands index (WIW) was calculated from the acquired data [20]. The WIW (Eq. 1) is defined by:

$$WIW = B8A \leq 0.1804 \ \&\& \ B12 \leq 0.1131, \qquad (1)$$

where B8A represents a narrow Near Infra-Red (NIR) band, and B12 is a Short Wave Infra-Red (SWIR) band. In other words, flooded areas could be distinguished from dry areas when pixels satisfy the conditions in Eq. (1) [20].

The area of Lake Cerknica was extracted using a mask, which, in the detail, describes the entire area of the lake. The extracted pixels from the WIW index were converted from 2D to 3D in such a way that a pixel with *x* and *y* coordinates, contained in the area of the flooded lake, stores the intensity (coordinate *z*) of the WIW index.

The symmetry was calculated using the method presented in [21]. The method is based on maximising a specific objective function which we call a symmetry measure (see [21] for details). The function takes a set of points and a plane on the input, and outputs a value that represents the measure of symmetry of the point set with respect to the plane. One of the most important features of the symmetry measure is its differentiability w.r.t. the plane parameters. Candidate symmetry planes are created by pairing points of a simplified version of the input point set, and the symmetry measure is computed for all of them. A small subset of the candidates with the largest symmetry measure is then used to initialise a gradient based optimisation which leads to several local maxima of the measure. Among them, the one with the largest value of the measure is then selected as the resulting symmetry plane.

The linear and quadratic functions are fitted by the obtained symmetry measures and known water levels.

3. Results

Sentinel-2 data processing is a demanding and computer intensive process. An AMD Ryzen 5 4500U with 32 GB of main memory on Windows 10 has been used for these reasons. In Fig. 2 we can see the exact location of Lake Cerknica in Slovenia. Fig. 3 shows the whole area of Lake Cerknica after the calculated WIW index and applied extraction mask. The Sentinel-2 data used for the calculation of WIW presented in Fig. 3 were acquired on 24 February, 2021. At that time, most of the lake was flooded with water (the blue pixels in Fig. 3), and only the east part and a small portion of the centre of the lake were dry. Fig. 4 represents only the flooded parts of the lake. Recently, the flooded parts were converted from 2D to 3D, and in that way prepared for the calculation of the symmetry measure, which was calculated finally.

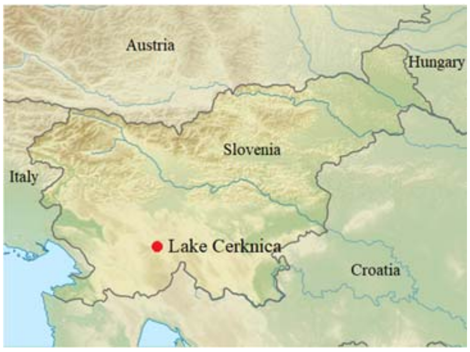


Fig. 2. Location of Lake Cerknica in Slovenia.

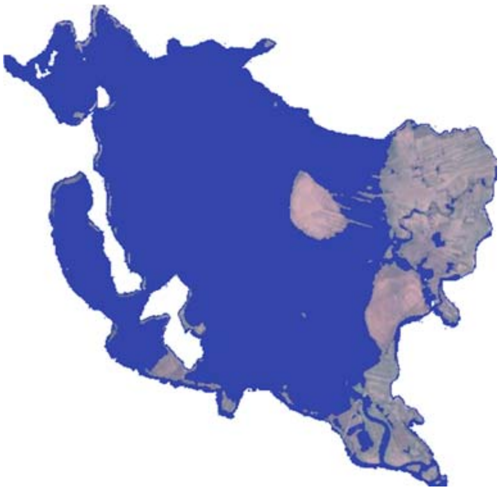


Fig. 3. Area of Lake Cerknica, where blue pixels represent the flooded area of the lake.

The black line in Fig. 4 represents the plane of the 3D object of the flooded Lake Cerknica where the largest symmetry measure value was achieved. The calculated symmetry measure for the flooded parts in Fig. 4 was 1016.89. Then, the several different flooded areas of the lake were tested (see Fig. 5). When the whole area of Lake Cerknica was flooded, the symmetry measure equaled 1,588.41, but, at the same time, when the lake was flooded less than presented in Fig. 4, the symmetry measures were also lower than in Fig. 4. An example of this can be seen in Fig. 5, where the symmetry measure was 983.77. This indicates that the lower the symmetry value is, the lower is the water level in Lake Cerknica.

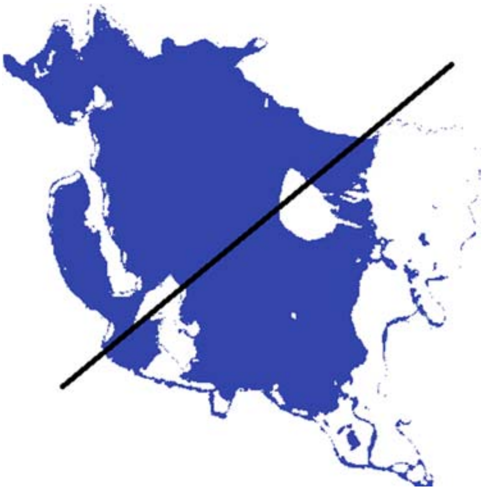


Fig. 4. Only the flooded area of Lake Cerknica, where the black line represents the largest symmetry value.

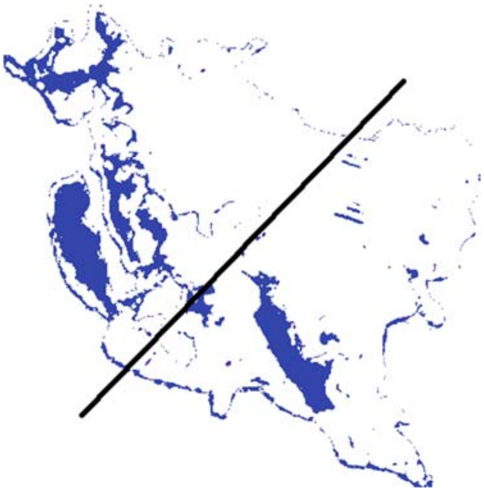


Fig. 5. Only a small part of flooded area of Lake Cerknica, where the symmetry measure was 983.77.

The linear and quadratic functions were fitted based on the calculated symmetry measures and known water levels of Lake Cerknica. For the completely dry lake we assumed that the symmetry measure was equal to 0 and the surface elevation (above the sea level) was 541 m, while, at the surface level of 551 m, the symmetry measure was 1,588.41 (the whole lake was flooded). For the other two measured symmetries (983.77 and 1016.89) we presumed the surface elevations of 547 m and 550 m, accordingly. The linear function (Eq. (2)) is equal to:

$$SE = 139.79 \times SM - 75702.49, \tag{2}$$

while the quadratic function (Eq. (3)) looks like:

$$SE = -1.37 \times SM^2 - 1634.47 \times SM - \\ - 483561.12, \tag{3}$$

where, in both equations (Eq. (2) and Eq. (3)), SE represents the predicted Surface Elevation and SM is the calculated Symmetry Measure. The performance of fitted functions was measured using R^2. R^2 is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model [22]. R^2 for the linear equation was equal to 0.9130, while the quadratic function achieved the result $R^2 = 0.9135$. The learned Eq. (2) and Eq. (3) can now be used for the prediction of surface levels based on the calculated symmetry measures.

4. Conclusion

The methodology for detection of water levels in the intermittent Lake Cerknica using a multispectral index acquired from Sentinel-2 and symmetry measure is presented in this paper. The results presented in the previous Section show that the symmetry measure of the 3D point cloud generated from the WIW index could be used for the prediction of water levels. The learned linear and quadratic functions achieved good results, and can be used for the prediction of surface levels. The use of ground truth data of the measured surface elevations of Lake Cerknica (e.g. sending experts to measure the exact elevation) and a bigger learning set (a combination of measured surface elevations and calculated symmetry measures) will be the main topic of the future work. The influence of other factors, such as shadows and clouds, on the quality of the WIW index will also be part of our future research work.

Acknowledgements

The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0041 and Project No. N2-0181). This work was also supported by the Czech Science Foundation, Project 21-08009K Generalized Symmetries and Equivalences of Geometric Data.

References

[1]. Y. Du, Y. Zhang, F. Ling, Q. Wang, W. Li, X. Li, Water bodies' mapping from Sentinel-2 imagery with modified normalized difference water index at 10-m

spatial resolution produced by sharpening the SWIR band, *Remote Sensing*, Vol. 8, Issue 4, 2016, 354.

[2]. N. Du, H. Ottens, R. Sliuzas, Spatial impact of urban expansion on surface water bodies: A case study of Wuhan, China, *Landscape and Urban Planning*, Vol. 94, Issue 3, 2010, pp. 175-185.

[3]. J. F. Pekel, A. Cottam, N. Gorelick, A. S. Belward, High-resolution mapping of global surface water and its long-term changes, *Nature*, Vol. 540, Issue 7633, 2016, pp. 418-422.

[4]. M. G. Tulbure, M. Broich, Spatiotemporal dynamic of surface water bodies using Landsat time-series data from 1999 to 2011, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 79, Issue 1, 2013, pp. 44-52.

[5]. K. Singh, M. Ghosh, S. R. Sharma, WSB-DA: Water surface boundary detection algorithm using Landsat 8 OLI data, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 9, Issue 1, 2015, pp. 363-368.

[6]. T. D. Acharya, D. H. Lee, I. T. Yang, J. K. Lee, Identification of water bodies in a Landsat 8 OLI image using a J48 decision tree, *Sensors*, Vol. 16, Issue 7, 2016, pp. 1075-1091.

[7]. R. Sivanpillai, S. N. Miller, Improvements in mapping water bodies using ASTER data, *Ecological Informatics*, Vol. 5, Issue 1, 2010, pp. 73-78.

[8]. Y. Zhou, J. Luo, Z. Shen, X. Hu, H. Yang, Multiscale water body extraction in urban environments from satellite images, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 7, Issue 10, 2014, pp. 4301-4312.

[9]. X. Yang, S. Zhao, X. Qin, N. Zhao, L. Liang, Mapping of urban surface water bodies from Sentinel-2 MSI imagery at 10 m resolution via NDWI-based image sharpening*, Remote Sensing*, Vol. 9, Issue 6, 2017, pp. 596-615.

[10]. A. Bhardwaj, M. K. Singh, P. K. Joshi, S. Singh, L. Sam, R. D. Gupta, R. Kumar, A lake detection algorithm (LDA) using Landsat 8 data: A comparative approach in glacial environment, *International Journal of Applied Earth Observation and Geoinformation*, Vol. 28, Issue 1, 2015, pp. 150-163.

[11]. H. Jiang, M. Feng, Y. Zhu, N. Lu, J. Huang, T. Xiao, An automated method for extracting rivers and lakes from Landsat imagery, *Remote Sensing*, Vol. 6, Issue 6, 2014, pp. 5067-5089.

[12]. W. Li, P. Gong, Continuous monitoring of coastline dynamics in western Florida with a 30-year time series of Landsat imagery, *Remote Sensing of Environment*, Vol. 179, Issue 1, 2016, pp. 196-209.

[13]. S. K. McFeeters, The use of the normalized difference water index (NDWI) in the delineation of open water features, *International Journal of Remote Sensing*, Vol. 17, Issue 7, 1996, pp. 1425-1432.

[14]. J.-H. Ryu, J.-S. Won, K. D. Min, Waterline extraction from Landsat TM data in a tidal flat: a case study in Gomso Bay, Korea, *Remote Sensing of Environment*, Vol. 83, Issue 3, 2002, pp. 442-456.

[15]. X. Yang, Q. Qin, P. Grussenmeyer, M. Koehl, Urban surface water body detection with suppressed built-up noise based on water indices from Sentinel-2 MSI imagery, *Remote Sensing of Environment*, Vol. 219, Issue 1, 2018, pp. 259-270.

[16]. X. Huang, C. Xie, X. Fang, L. Zhang, Combining pixel-and object-based machine learning for identification of water-body types from urban high-resolution remote-sensing imagery, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 8, Issue 5, 2015, pp. 2097-2110.

[17]. H. Xu, Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery, *International Journal of Remote Sensing*, Vol. 27, Issue 14, 2006, pp. 3025-3033.

[18]. G. L. Feyisa, H. Meilby, R. Fensholt, S. R. Proud, Automated water extraction index: a new technique for surface water mapping using Landsat imagery, *Remote Sensing of Environment*, Vol. 140, Issue 1, 2014, pp. 23-35.

[19]. H. Šmid, G. Golež, D. Podjed, D. Kladnik, B. Erhartič, P. Pavlin, I. Jerele, Enciklopedija naravne in kulturne dediščine na Slovenskem (Encyclopedia of Natural and Cultural Heritage in Slovenia), *DEDI – Digital Encyclopedia of Natural and Cultural Heritage of Slovenia,* Ljubljana, 2012.

[20]. G. Lefebvre, A. Davranche, L. Willm, J. Campagna, L. Redmond, C. Merle, A. Guelmami, B. Poulin, Introducing WIW for detecting the presence of water in wetlands with Landsat and sentinel satellites, *Remote Sensing*, Vol. 11, Issue 19, 2019, 2210.

[21]. L. Hruda, I. Kolingerova, L. Vaša, Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure, *The Visual Computer*, Vol. 37, Issue 1, 2021, pp. 1-17.

[22]. E. Kasuya, On the use of r and r squared in correlation and regression, *Ecological Research*, Vol. 34, Issue 1, 2019, pp. 235-236.

(021)

DeiT-ADL: Data-efficient Image Transformer for Anomaly Detection and Localization

Hyeonjong Ha and Jongpil Jeong
Department of Smart Factory Convergence, Sungkyunkwan University,
Suwon, Gyeonggi-do 16419, Republic of Korea
Tel.: +82 10-6557-4713
E-mail: wnajrqkq94@naver.com

Summary: We propose an improved transformer-based image anomaly detection and localization network. Our proposed model is a combination of a reconstruction-based approach and patch embedding. The use of transformer networks helps preserving the spatial information of the embedded patches, which is later processed by a Gaussian mixture density network to localize the anomalous areas. Additionally, we introduce a teacher-student strategy specific to transformers. It relies on a distillation token ensuring that the student learns from the teacher through attention. We use this token-based distillation, especially when using a convnet as a teacher. Deit-ADL shows improved results in both accuracy and image processing speed than VT-ADL.

Keywords: Anomaly detection, Anomaly segmentation, Computer vision, Vision transformer, Gaussian density approximation.

1. Introduction

In computer vision, ideal is an image or part of an image that shows a significant difference from predefined normality characteristics. Systems that can intelligently do this task are from video surveillance [6] to defect segmentation [7, 8], inspection [7], quality control [9], medical imaging [10], financial transactions [11], etc. Demand is very high because of the wide range of applications. As the case shows, anomaly detection is particularly important in industries where it can be used to automatically identify defective products. Therefore, anomaly detection is the task of identifying these new samples in a supervised or unsupervised way.

Recently, efforts are being made to improve the anomaly detection task in the field of deep learning. Most tasks try to learn a single class of manifolds [12] representing generic data using an encoding-decoding approach, and the output classifies the input image as normal or abnormal, while less processing the task. Segment the local anomaly in the image [13]. Primarily, this method uses a reconstruction-based approach or is trained on a pre-trained network or end-to-end.

Based on the above facts and industrial needs, Pankaj Mishra [1] developed a Vision-Transformer Network based Image Anomaly Detection and Localization (VT-ADL). Recently, Dosovitskiy et al. The vision transformer network model proposed by [3] is a network designed to operate on image patches during training to preserve positional information. In this work, we use Gaussian Approximation of latent features [14, 15] to show how an adapted transformer network can be used for anomaly localization and how to adjust various configurations to solve some shortcomings of the vision transformer network.

However, these VT-ADLs have disadvantages of low accuracy and slow model processing. In addition, Vision Transformer achieves the performance of existing CNNs only when a huge set of data is applied. Here, an improved VT-ADL model is proposed by referring to a model called Deit [2]. Since the distillization token was used to conduct transfer learning using CNN, a model that increases the computational speed and increases accuracy by reducing the number of parameters of the transformer model is proposed.

The paper is organized as follows. Section 2 describes related work about Image-based anomaly detection and vision transformer. Section 3 details the overall proposed model and knowledge distillation. Section 4 describes datasets and results from the experiment. Finally, Section 5 presents conclusion.

2. Related Work

Image-based anomaly detection has been used in many inspection and quality control systems, but is not a new topic in industrial use cases as it is still being investigated with the latest deep learning technology. Historically, several classical image processing and machine learning methods have been used to perform anomaly detection tasks such as Bayesian networks, rule-based systems, clustering algorithms, etc. However, in recent years, the trend has shifted to the use of deep learning. This is because the convolution layer revolutionized this field. Most of the proposed approaches are based on image reconstruction.

In this case, the network is trained to reconstruct the input image. If the network is trained only on normal data, it is assumed that it will fail to properly reconstruct the anomaly. Network architectures mostly consist of various configurations of autoencoders [16-20] or Generative Adversarial Networks (GANs) [21, 22]. At the image level, the simplest way is to learn using the MSE loss, and as a result expect a higher reconstruction loss for anomalous images. Additional information from the latent space [23] is also used for better classification. However, for anomaly localization, the pixel-reconstructed error is considered an anomaly score. Some methods have also tried to use visual attention maps [24] in latent space. Reconstruction-based methods are very intuitive and explainable, but their performance is limited when it comes to capturing small local anomalies [25].

Recently, Vision transformer (ViT) [3] narrowed the gap with the latest technologies on ImageNet without using convolution. Nevertheless, pre-training steps for a large amount of selected data are required for the learned converter to be effective. Vaswani et al. The Transformer architecture introduced by [26] is the reference model for all current natural language processing (NLP) operations in the case of machine translation. Many of the improvements in convnet for image classification are inspired by transformers. For example, Squeeze and Excitation [27], Selective Kernel [28], and Split-Attention Networks [29] utilize a mechanism similar to the transformers self-attention (SA) mechanism.

Knowledge Distinction (KD) [30] represents a training paradigm in which student models utilize "soft" labels from strong teacher networks. This is not the maximum score, but the output vector of the teacher's SoftMax function and provides a "hard" label. This training improves the performance of the student model. On the other hand, Wei et al. The teacher's supervision in [31] takes into account the effect of data augmentation, which sometimes leads to misalignment between actual labels and images. For example, consider an image with the label "Cat" representing a large landscape and a small cat in a corner. Implicitly change the label of the image when the cat is no longer in the crop of the data augmentation. KD is able to convey inductive biases [32] in a smooth manner in a student model, using a teacher model that integrates in a difficult way.

3. Proposed Model

In Fig. 1, Image is split into patches, which are augmented with positional embedding. The resulting sequence is fed to the Transformer encoder. Then encoded features are summed into a reconstruction vector which is fed to decoder. The transformer encoded features are also fed into a Gaussian approximation network [4], which is later used to localize the anomaly.

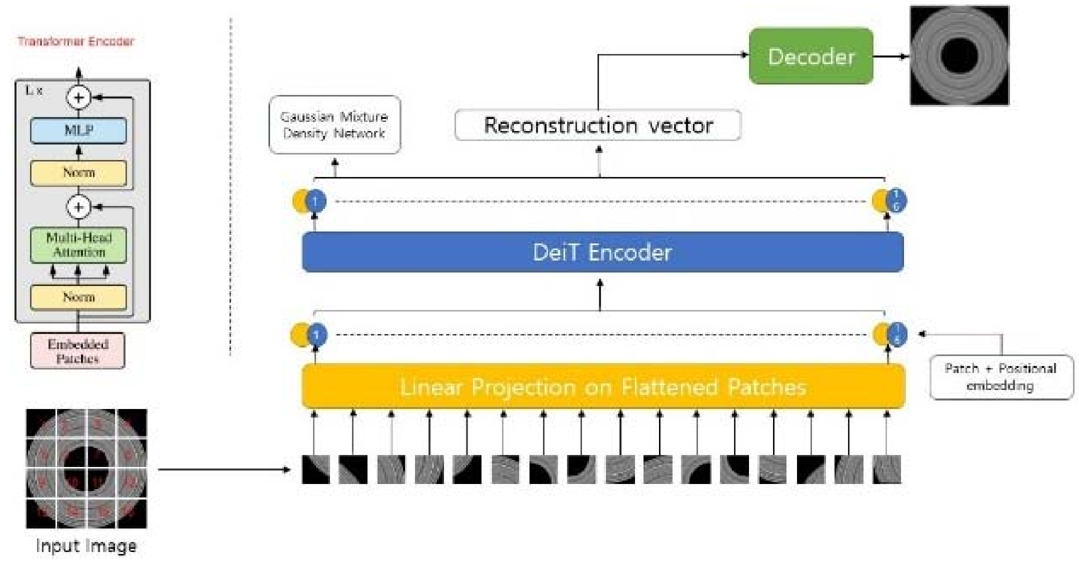


Fig. 1. Model Overview.

The proposed model combines the advantages of traditional reconstruction-based methods and patch-based approaches. Input images are subdivided into patches and encoded using Vision Transformer. The resulting features are then fed to the decoder to reconstruct the original image, allowing the network to learn features representing aspects of the general image (the only data the network has trained on). At the same time, the Gaussian mixed density network models the distribution of transformer-encoded functions to estimate the distribution of normal data in this latent space. Since the function encoded by the converter is connected to the location information, the location can be automatically identified by detecting an abnormality with this model.

The transformer encoder layer is based on the work by Vaswani et al and its application to images by Dosovitskiy et al [3]. The decoder is used to decode the reconstruction vector back to the original image shape. It maps R^{512} → $R^{H\times W\times C.}$ we used 5 transposed convolutional layers, with batch normalization and ReLU in-between, except for the last layer, we use tanh as the final non- linearity.

Gaussian Mixture Density Network estimates the conditional distribution $p(y|x)$ [4] of a mixture density model. In particular, the image embedding (conditional variable x) as the input. The density estimate p^(y|x) follows the weighted sum of K Gaussian functions.

$$\hat{p}(y \mid x) = \\ = \sum_{k=1}^{K} w_k(x;\theta)\mathcal{N}\big(y \mid \mu_k(x;\theta), \sigma_k^2(x;\theta)\big), \quad (1)$$

wherein, $w_k(x;\theta)$ denotes the weight, μk(x; θ) the mean, $\sigma_k^2(x;\theta)$ the variance of the k-th Gaussian. All the GMM parameters are estimated using the neural network with parameters θ and input x. The mixing weights of the Gaussians must satisfy the constraints $\sum_{k=1}^{K} w_k(x;\theta) = 1$ and $w_k(x;\theta) \geq 0$ ∀k. This is achieved using the SoftMax function to the output of weight estimation:

$$w_k(x) = \frac{\exp(a_k^w(x))}{\sum_{k=1}^{K} \exp(a_i^w(x))}, \quad (2)$$

wherein $a_k^w(x) \in \mathbb{R}$ is the logit scores emitted by the neural network. Additionally, standard deviation $\sigma_k\ (x)$must be positive. To satisfy this, a softplus non-linearities applied to the output of the neural network.

$$\sigma_k(x) = \log(1 + \exp(\beta \times x)); \beta = 1 \quad (3)$$

Since, mean μk(x; θ) doesn't have any constraint, we used linear layer without any non-linearity for the respective output neurons.

In Fig. 2, Minimize the KL divergence of the SoftMax distribution of the teacher's model and the SoftMax distribution of the student model. Training student model is done in a way that minimizes the sum of the student loss and the distillation loss. When training the transformer encoder, we use a distillation token to train it to increase the training efficiency.

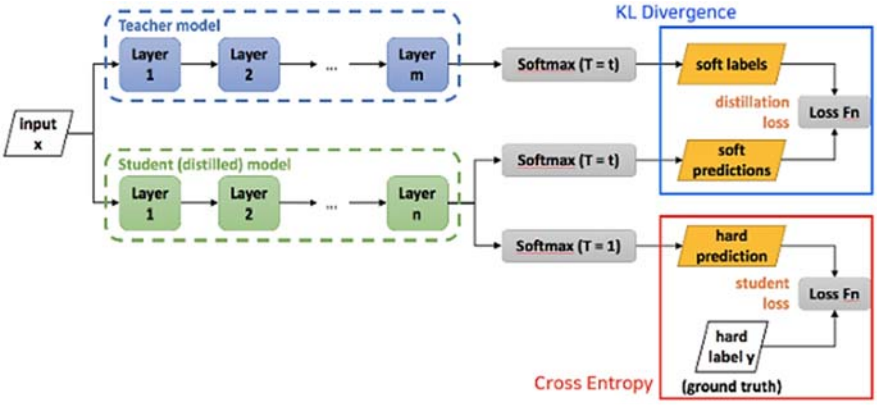


Fig. 2. Knowledge Distillation, taken from [5].

We introduce a variant of distillation where we take the hard decision of the teacher as a true label. Let $y_{\mathrm{t}} = \mathrm{argmax}_c Z_{\mathrm{t}}(c)$ be the hard decision of the teacher, the objective associated with this hard-label distillation is:

$$\mathcal{L}_{\mathrm{global}}^{\mathrm{hardDistill}} = \\ = \frac{1}{2}\mathcal{L}_{\mathrm{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\mathrm{CE}}(\psi(Z_s), y_{\mathrm{t}}) \quad (4)$$

For a given image, the hard label associated with the teacher may change depending on the specific data augmentation. We will see that this choice is better than the traditional one, while being parameter-free and conceptually simpler: The teacher prediction y_t plays the same role as the true label y.

We add a new token, the distillation token, to the initial embeddings (patches and class token). Our distillation token is used similarly as the class token: it interacts with other embeddings through self-attention, and is output by the network after the last layer. Its target objective is given by the distillation component of the loss. The distillation embedding allows our model to learn from the output of the teacher, as in a regular distillation, while remaining complementary to the class embedding.

4. Experiment and Result Analysis

We used MVTec AD Dataset. It's a real-world anomaly detection dataset. It contains 5,354 high-resolution color images of different textures and objects categories. It has normal and anomalous images which showcase 70 different types of anomalies of different real-world products.

In Fig. 3, First row shows the actual anomalous image of bottle, cable, capsule, metal nut and brush. Second row shows the ground truth and third row shows the generated anomaly score and anomaly localization by our method.

Table 1 shows the results for MVTec dataset. The value shows the PRO curve up to an average false positive rate per-pixel of 30 % is reported. It measures the average overlap of each ground truth region with the predicted anomaly region for multiple thresholds.

Our proposed methods performed at par with the most recent state of the art algorithms and even outperformed them in 8 product categories.

There was also an improvement in image throughput (image/s). The existing VT-ADL was 85.9, but our proposed model measured 290.9. These values were calculated as average values after 50 runs. The throughput is measured as the number of images that we can process per second on one RTX 3080 GPU.

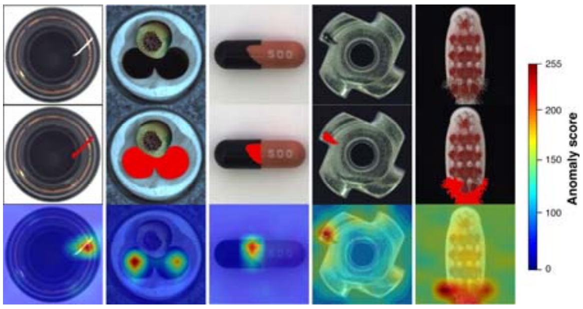


Fig. 3. Anomaly detection on MVTec dataset.

5. Conclusion

In this paper, we propose an improved model for anomaly detection and localization using a vision transformer. Deit-ADL shows improved results in both accuracy and image processing speed.

In future research, we plan to develop a better anomaly detection model by referring to several studies that have improved the performance of the vision transformer.

Acknowledgements

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2018-0-01417) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Table 1. Comparative AUC results on MVTEC Dataset. Comparative results taken from [1].

Category	**1-NN**	**OC SVM**	**K Means**	**AE MSE**	**VAE**	**AE SSIM**	**Ano GAN**	**CNN Feat. Dic**	**Uni. Stud.**	**VT-ADL**	**DeiT-ADL (Ours)**
Carpet	0.512	0.355	0.253	0.456	0.501	0.647	0.204	0.469	0.695	0.773	**0.818**
Grid	0.228	0.125	0.107	0.582	0.224	0.849	0.226	0.183	0.819	0.871	**0.903**
Leather	0.446	0.306	0.308	0.819	0.635	0.561	0.378	0.641	0.819	0.728	**0.836**
Tile	0.822	0.722	0.779	0.897	0.87	0.175	0.177	0.797	**0.912**	0.796	0.851
Wood	0.502	0.336	0.411	0.727	0.628	0.605	0.386	0.621	0.725	0.781	**0.802**
Bottle	0.898	0.85	0.495	0.91	0.897	0.834	0.62	0.742	0.918	0.949	**0.956**
Cable	0.806	0.431	0.513	0.825	0.654	0.478	0.383	0.558	**0.865**	0.776	0.813
Capsule	0.631	0.554	0.387	0.862	0.526	0.86	0.306	0.306	**0.916**	0.672	0.713
Hazelnut	0.861	0.616	0.698	0.917	0.878	0.916	0.698	0.844	**0.937**	0.897	0.912
Metal Nut	0.705	0.319	0.351	0.83	0.576	0.603	0.32	0.358	**0.895**	0.726	0.765
Pill	0.725	0.544	0.514	0.893	0.769	0.83	0.776	0.46	**0.935**	0.705	0.785
Screw	0.604	0.644	0.55	0.754	0.559	0.887	0.466	0.277	0.928	0.928	**0.934**
Toothbrush	0.675	0.538	0.337	0.822	0.693	0.784	0.749	0.151	0.863	0.901	**0.912**
Transistor	0.68	0.496	0.399	0.728	0.626	0.725	0.549	0.628	0.701	0.796	**0.817**
Zipper	0.512	0.355	0.253	0.839	0.549	0.665	0.467	0.703	**0.933**	0.808	0.806
Means	0.64	0.479	0.423	0.79	0.639	0.694	0.443	0.515	0.857	0.807	0.841

References

[1]. P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, VT-ADL: A vision transformer network for image anomaly detection and localization, in *Proceedings of the IEEE 30th International Symposium on Industrial Electronics (ISIE'21)*, 20 Apr 2021.

[2]. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in *Proceedings of the 38th International Conference on Machine Learning (PMLR'21)*, Vol. 139, 2021, pp. 10347-10357.

[3]. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, in *Proceedings of the International Conference on Learning Representations (ICLR'21)*, 2021.

[4]. C. M. Bishop, Mixture Density Networks, *Aston University*, 1994.

[5]. Image Source, https://intellabs.github.io/distiller/knowledge_distillation.html

[6]. C. Piciarelli, C. Micheloni, G. L. Foresti, Trajectory-based anomalous event detection, *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 18, Issue 11, 2008, pp. 1544-1554.

[7]. C. Piciarelli, D. Avola, D. Pannone, G. L. Foresti, A vision-based system for internal pipeline inspection, *IEEE Transactions on Industrial Informatics*, Vol. 15, Issue 6, 2019, pp. 3289-3299.

[8]. P. Chen, S. Yang, J. A. McCann, Distributed real-time anomaly detection in networked industrial sensing

systems, *IEEE Transactions on Industrial Electronics*, Vol. 62, Issue 6, 2015, pp. 3832-3842.

[9]. P. Napoletano, F. Piccoli, R. Schettini, Anomaly detection in nanofibrous materials by CNN-based self-similarity, *Sensors*, Vol. 18, Issue 1, 2018, 209.

[10]. X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, F. Lu, Understanding adversarial attacks on deep learning based medical image analysis systems, *Pattern Recognition,* Vol. 110, 2021, 107332.

[11]. P. Yu, X. Yan, Stock price prediction based on deep neural networks, *Neural Computing and Applications*, Vol. 32, Issue 6, 2020, pp. 1609-1628.

[12]. D. Wulsin, J. Blanco, R. Mani, B. Litt, Semi-supervised anomaly detection for EEG waveforms using deep belief nets, in *Proceedings of the 9th International Conference on Machine Learning and Applications*, 2010, pp. 436-441.

[13]. P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, 2020, pp. 4183-4192.

[14]. C. M. Bishop, Mixture Density Networks, *Aston University*, 1994.

[15]. N. Ueda, R. Nakano, Z. Ghahramani, G. E. Hinton, Split and merge EM algorithm for improving gaussian mixture density estimates, in *Proceedings of the IEEE Signal Processing Society Workshop*, 1998, pp. 274-283.

[16]. P. Mishra, C. Piciarelli, G. L. Foresti, A neural network for image anomaly detection with deep pyramidal representations and dynamic routing, *International Journal of Neural Systems*, Vol. 30, Issue 10, 2020, 2050060.

[17]. W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, O. Camps, Towards visually explaining variational autoencoders, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, 2020, pp. 8642-8651.

[18]. P. Mishra, C. Piciarelli, G. L. Foresti, Image anomaly detection by aggregating deep pyramidal representations, in *Proceedings of the* 25th *International Conference on Pattern Recognition (ICPR'21)*, 2021.

[19]. I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, *MIT Press*, 2016.

[20]. P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in *Proceedings of the ICML Workshop on Unsupervised and Transfer Learning*, 2012, pp. 37-49.

[21]. M. Sabokrou, M. Khalooei, M. Fathy, E. Adeli, Adversarially learned one-class classifier for novelty detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018, pp. 3379-3388.

[22]. S. Pidhorskyi, R. Almohsen, D. A. Adjeroh, G. Doretto, Generative probabilistic novelty detection with adversarial autoencoders, *arXiv Preprint*, arXiv:1807.02588, 2018.

[23]. D. Abati, A. Porrello, S. Calderara, R. Cucchiara, Latent space autoregression for novelty detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 2019, pp. 481-490.

[24]. S. Venkataramanan, K.-C. Peng, R. V. Singh, A. Mahalanobis, Attention guided anomaly localization in images, in *Proceedings of the European Conference on Computer Vision (ECCV'20)*, 2020, pp. 485-503.

[25]. P. Perera, R. Nallapati, B. Xiang, OCGAN: One-class novelty detection using gans with constrained latent representations, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 2019, pp. 2898-2906.

[26]. A. Vaswani, N. Shazeer, et al., Attention is all you need, in Advances in Neural Information Processing Systems, Vol. 30, *Curran Associates Inc.*, 2017.

[27]. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, *arXiv Preprint*, arXiv:1709.01507, 2017.

[28]. X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 2019.

[29]. H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Muller, R. Manmatha, M. Li, A. Smola, RESNEST: Split-attention networks, *arXiv Preprint*, arXiv:2004.08955, 2020.

[30]. G. E. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, *arXiv Preprint*, arXiv:1503.02531, 2015.

[31]. L. Wei, A. Xiao, L. Xie, X. Chen, X. Zhang, Q. Tian, Circumventing outliers of auto augment with knowledge distillation, in *Proceedings of the European Conference on Computer Vision (ECCV'20)*, 2020.

[32]. S. Abnar, M. Dehghani, W. Zuidema, Transferring inductive biases through knowledge distillation, *arXiv Preprint*, arXiv:2006.00555, 2020.

(022)

New Approaches to ECG Reconstruction for Preserving Diagnostic Information

R. Gonzalez Tejeda [1], **C. M. Pais** [2] **and H. L. Rufiner** [2, 3]
[1] Instituto de Bioingeniería y Bioinformática, UNER-CONICET, Oro Verde, Entre Rios, Argentina
[2] Laboratorio de Cibernética, Facultad de Ingeniería, UNER, Argentina
[3] Instituto de Señales, Sistemas e Inteligencia Computacional, SINC(i) UNL-CONICET, Santa Fé, Argentina
Tel.: + 543434617251
E-mail: rgtejeda@ingenieria.uner.edu.ar

Summary: In this work we propose novel algorithms for ECG reconstruction. A subset of 3 of the 12 ECG leads will be selected to reconstruct the remaining ones. Traditional models focus on minimizing the mean square error between the original signal and the reconstructed one. Instead, in this paper we focus on preserving diagnostic information from the ECG. In order to compare these new strategies in relation to traditional reconstruction methods, an error measure called "Weighted Diagnostic Distortion" (WDD) was used. It measures the quality of the reconstructed signal based on the accuracy of the position and amplitude of the main characteristics (such as start point, end and peak) of ECG waves (PQRST). The results show best performance for one of the methods here proposed.

Keywords: Weighted diagnostic distortion, ECG reconstruction, Neural networks, Genetic algorithms.

1. Introduction

Nowadays, the most common method for the diagnosis of heart disease is the analysis of the 12 leads of the ECG (I, II, III, aVR, aVF, aVL, V1, V2, V3, V4, V5, V6). To record 12 lead ECG, 10 electrodes are placed in certain areas of the body. However, some applications such as monitoring (ambulatory and continuous), and remote cardiac care require fewer electrodes. In this context, reconstruction of the missing ECG leads becomes a useful tool.

The most frequently used ECG reconstruction method is to perform a linear transformation on the input signals. Although linear models generally perform well, non-linear models can even improve the quality of the reconstructed signal. This claim is mainly based on the fact that the torso is an inhomogeneous conductor, which was originally demonstrated in the work of Burger and van Milaan [2]. In this context, neural networks are considered one of the most widely used non-linear models. Initially, Atoui [3] proposed a method based on conventional neural networks. A committee of several networks were trained with the backpropagation algorithm, and their outputs were averaged. Recently, in 2020, Lee [4] carried out a more complete work regarding the analysis of results. Neural networks are implemented with slight variations in the structure with respect to Atoui's work, and it demonstrates the robustness of this method in relation to the most relevant methods up to date. The disadvantage of this implementation is the high computational cost for the use of several networks (50 in Atoui's work). To solve this, a convolutional neural network model is proposed [5]. Other machine learning approaches are also proposed with a new training method based on the Monte Carlo algorithm, called General Vector Machine (GVM) [6]. GVM search the global minimum, and therefore doesn't require the use of several networks.

In the state of the art, it can be verified that in most models the aim is to minimize the mean square error (MSE) and maximize the correlation. These measurements are useful in determining the similarity between the reference and the synthesized signal but the most important error criterion is based on similarity in the diagnosis made by expert cardiologists from both signals. In this direction, Zigel [1] introduces the WDD coefficient that compresses the diagnostic information from the parameters of ECG waves. Calculates the coefficient "mean opinion score" (MOS) [9], based on the results of surveys designed for experts to evaluate the quality of the synthesis according to the diagnosis made. Then, Zigel shows that the WDD coefficient is a better quality measure than those traditionally used (MSE and correlation), since it reports a higher correlation with MOS. This coefficient is frequently used [7, 8] to evaluate the proposed methods. Nevertheless, to the best of our knowledge there is no work focused on minimizing the diagnostic error in the training stage. This leads us to propose some new methods using neural networks models, where the loss function is based on the WDD.

2. Materials and Methods

2.1. Pre-processing

Before lead reconstruction, it is usual to do some pre-processing on 12 leads of the ECG. To eliminate low-frequency noise, the baseline of the ECG signal is determined through a low pass filter with a cutoff frequency of 0.7 Hz. A second order Butterworth low-pass filter with a 45 Hz cutoff frequency was applied to 12 leads to remove motion artifacts noise.

2.1. ECG Database

The database used to evaluate the methods is Lobachevsky University Electrocardiography (LUDB https://physionet.org/content/ludb/1.0.1/). This is better in several aspects than other public databases available. Various collections that are currently available in the public domain: MIT-BIH Arrhythmia Database, European ST-T Database, and QT Database, have certain limitations. MIT-BIH Arrhythmia Database, European ST-T Database have a markup only for QRS complexes. In turn, the QT Database contains annotations for P, QRS and T waves, but has only 2-lead recordings. The construction of the new LUDB database aims to eliminate these shortcomings. The database consists of 200 10-second 12-lead ECG signal records representing different morphologies of the ECG signal. The boundaries of P, T waves and QRS complexes were manually annotated by cardiologists for all 200 records. Also, each record is annotated with the corresponding diagnosis.

2.2. ECG Wave Delimitation

The calculation of the WDD coefficient depends on a previous phase to delimit the main waves of the ECG. The algorithm used to the ECG waves delimitation is based on the work of Laguna [10]. Detection of the QRS complex is performed first. T wave is recognized next, and, finally, the P wave. After detecting the ECG waves, it proceeds to search for their limits and peaks. The dyadic discrete wavelet transform (DWT) is calculated at scales 2^k, k = 1,...,5. Then, with the use of thresholds and zero crossings of the DWT values in different scales, the limits and peaks of the ECG waves are found. We use the implementation of the python NeuroKit module(https://pypi.org/project/neurokit/).

2.3. The WDD Measure

The WDD measure is computed from the relevant diagnostic information of the ECG signal mainly distributed in the PQRST waves. The diagnostic features of PQRST waves are location, duration, amplitude, and shape. For each beat detected in the original and reconstructed signal, the features vectors (β for the original signal, $\hat{\beta}$ for the reconstructed signal) are found:

$$\beta = [\beta_1, \beta_2, \ldots, \beta_p], \tag{1}$$

$$\hat{\beta} = [\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p], \tag{2}$$

where p is the number of features in the vector. The diagnostic parameters used are: RR_{int}, QRS_{dur},QT_{int},QT_{Pint}, RR_{int}, P_{dur}, PR_{int}, $QRS_{\text{peak\_no}}$, Q_{int}, QRS_{sign},Δ_{exist}, T_{shape}, ST_{shape}, P_{shape}, RR_{int}, QRS^{+}_{amp}, $ST_{\text{elevation}}$and ST_{slope}. Table 1 shows an overview of all diagnostic parameters. These were chosen with the help of an experienced cardiologist.

The Weighted Diagnostic Distortion between these two vectors is:

$$WDD(\beta, \hat{\beta}) = \Delta\beta \frac{\Lambda}{tr[\Lambda]} \Delta\beta^T, \tag{3}$$

where $\Delta\beta$ is the normalized difference vector:

$$\Delta\beta = \left[\frac{|\beta_1 - \hat{\beta}_1|}{max(|\beta_1|, |\hat{\beta}_1|)}, \ldots, \frac{|\beta_p - \hat{\beta}_p|}{max(|\beta_p|, |\hat{\beta}_p|)}\right], \tag{4}$$

and Λ is a diagonal weighting matrix.

2.4. Genetic Algorithm with WDD Coefficient (GAWDD)

This method consists of a neural network model where the search for optimal weights values is performed with a genetic algorithm. The loss function of the neural network is the value of the WDD coefficient. To calculate the WDD coefficient, it's necessary to estimate the limits of ECG waves previously (see diagnostic features used in Table 1). Since the gradient of the function used to delimit ECG waves cannot be calculated, the use of the backpropagation algorithm to train the network is not possible. This disadvantage justifies the use of the genetic algorithm to find the optimal value as an alternative.

The 2-point crossover method and the tournament selection algorithm are used. The score function used (eq. (5)) includes the value of the WDD coefficient and a diversity measure to avoid rapid convergence. The diversity measure (Eq. (6)) consists of calculating for each individual the average of the distance with the rest of the individuals. The gene values have a 10 bits resolution and the initial population has n = 1400 individuals. A drawback of this method is the high computational cost due to the complexity of the fitness function, which involves the ECG waves delimitation for each beat. To deal with this problem, a variant is used, consisting of randomly selecting 10 % of the beats for each generation.

$$f_i = (1 - w)\left(1 - \frac{WDD_i}{max(WDD_j)}\right) + \\ +w * d_i, \tag{5}$$

where w is weighted parameter and d_i is the diversity measure for individual i:

$$d_i = \frac{\sum_{i! = j} hamming(i,j)}{n-1}, \tag{6}$$

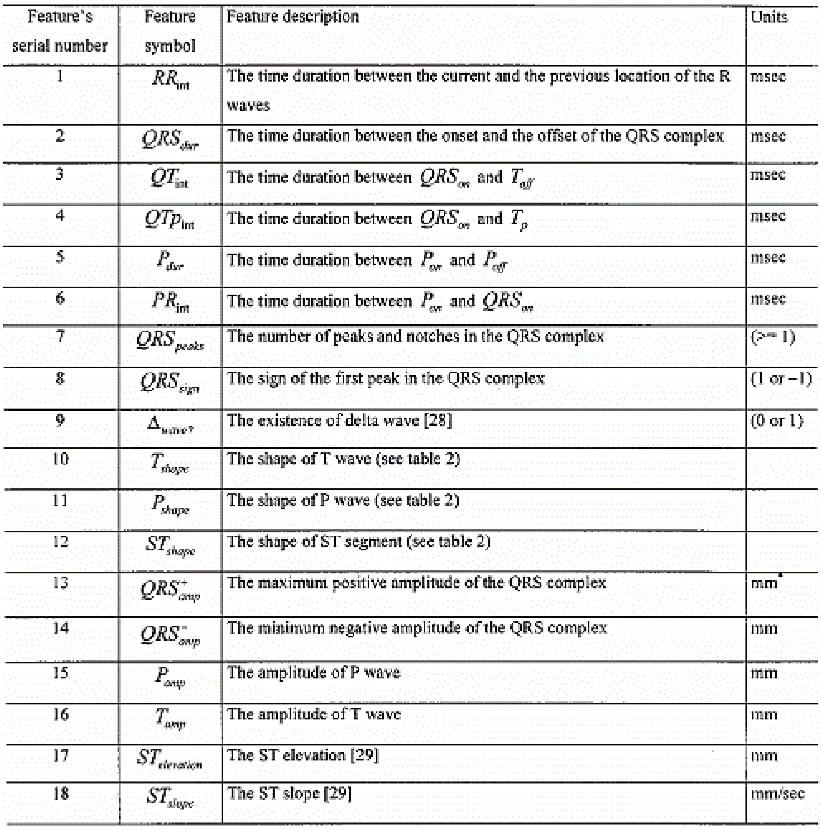
$$hamming(i,j) = \frac{\delta(i,j)}{nbits}, \tag{7}$$

where $\delta(i,j)$ is the number of different bits between individuals i and j, and nbits is the number of bits of the individuals.

2.5 Fine Tuning with Reduced WDD Coefficient (FTWDDred)

This method consists of a neural network model with two training stages. In the first stage, the backpropagation method, with classic error loss function, is used with all points of the ECG signal. In the second stage, the backpropagation algorithm is also used, but only the characteristic points of the ECG waves are used for error computation. This is made possible by the use of the Lobachevsky University Electrocardiography Database with the annotations file of ECG wave marks. So we avoid the use of the wave delimitation algorithm prone to induce artifacts in the neural network learning process due to the difficulty of this task.

Table 1. Description of the diagnostic features (10 mm = 1 mV).



Feature's serial number	Feature symbol	Feature description	Units
1	RR_{int}	The time duration between the current and the previous location of the R waves	msec
2	QRS_{dur}	The time duration between the onset and the offset of the QRS complex	msec
3	QT_{int}	The time duration between QRS_{on} and T_{off}	msec
4	QTp_{int}	The time duration between QRS_{on} and T_{p}	msec
5	P_{dur}	The time duration between P_{on} and P_{off}	msec
6	PR_{int}	The time duration between P_{on} and QRS_{on}	msec
7	QRS_{peaks}	The number of peaks and notches in the QRS complex	(>= 1)
8	QRS_{sign}	The sign of the first peak in the QRS complex	(1 or −1)
9	$\Delta_{wave?}$	The existence of delta wave [28]	(0 or 1)
10	T_{shape}	The shape of T wave (see table 2)	
11	P_{shape}	The shape of P wave (see table 2)	
12	ST_{shape}	The shape of ST segment (see table 2)	
13	QRS^{+}_{amp}	The maximum positive amplitude of the QRS complex	mm*
14	QRS^{-}_{amp}	The minimum negative amplitude of the QRS complex	mm
15	P_{amp}	The amplitude of P wave	mm
16	T_{amp}	The amplitude of T wave	mm
17	$ST_{elevation}$	The ST elevation [29]	mm
18	ST_{slope}	The ST slope [29]	mm/sec

3. Results

For the evaluation of the methods, leads I, II and V2 are used to reconstruct the remaining leads. For the analysis of the results, lead V1 is discarded due to the low performance of the wave delimiter in this lead. For the evaluation of the methods, 25 % of the records from the database are randomly selected. A comparison is made between the two here proposed methods and other two relevant state of the art methods: Atoui's Neural Networks (AtouiNN) and Linear Regression. Tables 2, 3 and 4 show the results of the MSE, Correlation and WDD measures respectively. We observe that the FTWDDred method has the best performance in the WDD and Correlation measures, but not with respect to the MSE measure. The worst performance relative to MSE is due to FTWDDred targeting the diagnostic areas of the ECG in the second phase. Therefore, since the MSE is measured in the entire signal, it is expected that AtouiNN is better with respect to this measure than FTWDDred. We also note that the difference between FTWDDred and AtouiNN is small in the WDD measure. Therefore, the test was repeated 30 times to verify if the differences between the results of these methods are significant with the t-test. The results show that only lead V3 shows significant differences for alpha = 0.05.

Table 2. WDD for each method (lower values are better).

Methods	V3	V4	V5	V6	Mean
Linear Model	**12.57**	12.46	9.26	8.05	10.59
AtouiNN	13.45	9.69	**8.19**	7.22	9.64
FTWDDred	12.95	**8.50**	8.52	**7.17**	**9,28**
GAWDD	12.98	11.57	8.54	7.74	10.21

Table 3. Correlation for each method (higher values are better).

Methods	V3	V4	V5	V6	Mean
Linear Model	0.90	0.86	0.90	0.92	0.89
AtouiNN	0.91	0.88	**0.92**	0.93	0.91
FTWDDred	**0.92**	**0.89**	**0.92**	**0.94**	**0.92**
GAWDD	0.82	0.74	0.72	0.84	0.78

Table 4. MSE for each method (MSE value × 100, lower values are better).

Methods	V3	V4	V5	V6	Mean
Linear Model	0.22	0.29	0.24	0.21	0.24
AtouiNN	0.17	**0.27**	**0.18**	**0.17**	**0.20**
FTWDDred	**0.16**	0.28	0.19	**0.17**	0.20
GAWDD	0.45	0.52	0.46	0.38	0.45

Figs. 1 and 2 show that in some areas of the PQRST waves, the FTWDDred method better approximates the original signal than the AtouiNN. We observe that this improvement occurs mainly in the QRS peaks, also in the T peak, although to a minor extent. This result corresponds to what is expected according to the approach of the FTWDDred method. Since the information on the location, duration and shape of the ECG waves isn't used, only an improvement in the amplitude of the points used to perform the fine adjustment is obtained.

3. Conclusions

A novel approach is proposed and evaluated to improve ECG reconstruction focusing on relevant diagnostic information. The best performance was achieved with the FTWDDred method, but the GAWDD method didn't improve the results of the Atoui neural network. This is caused by limitations of the ECG wave delimitation algorithm. In the analysis of the results, it was shown that the improvements of the FTWDDred are only with respect to the amplitude of the ECG waves, mainly in the QRS. This drawback leads to the study of other alternatives in future works.

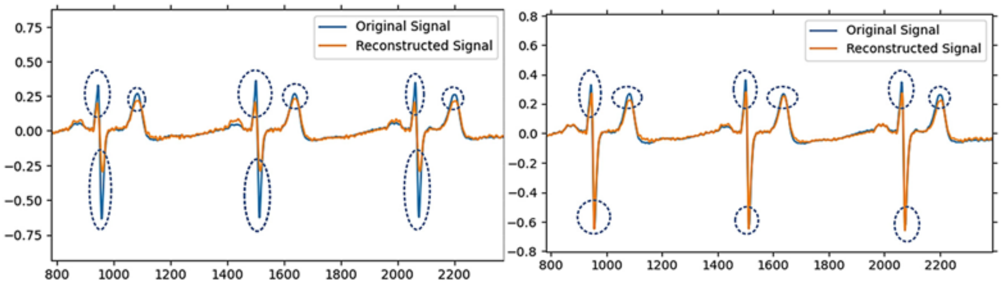


Fig. 1. Record 169 signal reconstructed with AtouiNN (left) and FTWDDred (right) methods.

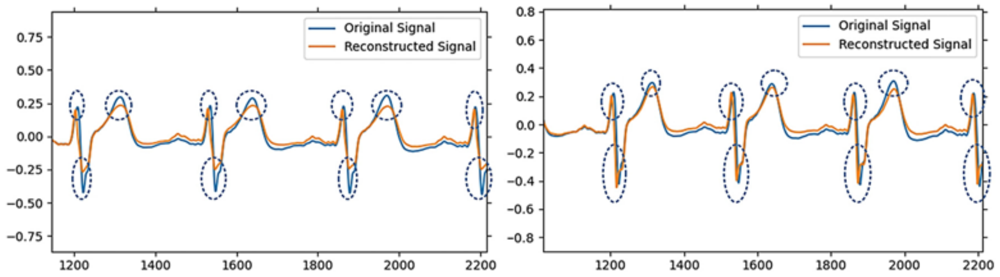


Fig. 2. Record 185 signal reconstructed with AtouiNN (left) and FTWDDred (right) methods.

References

[1]. Y. Zigel, A. Cohen, and A. Katz, A diagnostic meaningful distortion measure for ECG compression, in *Proceedings of the 19th Conference IEEE in Israel*, 1996, pp. 117-120.

[2]. H. C. Burger, J. B. van Milaan, Heart-vector and leads, *Brit. Heart J.*, Vol. 8, Issue 3, 1946, pp. 157-161.

[3]. H. Atoui, J. Fayn, P. Rubel, A novel neural-network model for deriving standard 12-lead ECGs from serial three-lead ECGs: Application to self-care, *IEEE Trans. Inf. Technol. Biomed.*, Vol.14, Issue 3, May 2010, pp. 883-890.

[4]. D. Lee, H. Kwon, H. Lee, C. Seo, K. Park, Optimal lead position in patch-type monitoring sensors for reconstructing 12-Lead ECG signals with universal transformation coefficient, *Sensors (Basel)*, Vol. 20, Issue 4. 963.

[5]. Ld. Wang, W. Zhou, Y. Xing, *et al.*, A novel method based on convolutional neural networks for deriving standard 12-lead ECG from serial 3-lead ECGt, *Frontiers Inf. Technol. Electronic Eng.*, Vol. 20, 2019, pp. 405-413.

[6]. Z. Xu, R. Zhou, Y. Cao, B. Yong, X. Wang, Q. Zhou, Reconstruction of 12-lead electrocardiogram based on GVM, in *Proceedings of the Sixth International Conference on Advanced Cloud and Big Data (CBD'18)*, 2018, pp. 275-280.

[7]. J. J. Nallikuzhy, S. Dandapat, Spatial enhancement of ECG using diagnostic similarity score based lead selective multi-scale linear model, *Comput. Biol. Med*, Vol. 85., June 2017, pp. 53-62.

[8]. J. J. Nallikuzhy, S. Dandapat, Spatial enhancement of ECG using multiple joint dictionary learning, *Biomedical Signal Processing and Control*, Vol. 54, 2019, 101598.

[9]. Y. Zigel, A. Cohen, A. Katz, The weighted diagnostic distortion (WDD) measure for ECG signal compression, *IEEE Trans. Biomed. Eng*., Vol. 47, Issue 11, Nov 2000, pp. 1424-1430.

[10]. J. P. Martinez, R. Almeida, S. Olmos, A. P. Rocha, P. Laguna, A wavelet-based ECG delineator: evaluation on standard databases, *IEEE Transactions on Biomedical Engineering*, Vol. 51, Issue 4, April 2004, pp. 570-581.

(023)

Matrix Beaconing for the Location of Autonomous Industrial Vehicles on a Simulation Platform

A. Ndao, M. Djoko-Kouam and A.-J. Fougères
ECAM Rennes, Louis de Broglie, Campus de Ker Lann, Bruz, Rennes 35091, France
Tel.: 33299058454
E-mail: arame.ndao@ecam-rennes.com

Summary: The use of automated guided vehicles (AGVs) and other autonomous mobile robots is a challenge facing Industry 4.0. While the autonomy of autonomous vehicles has been well characterized in the field of road and road transport, this is not the case for the autonomous vehicles used in industry (autonomous industrial vehicles or AIVs). The establishment and deployment of AIV fleets in industrial companies remain problematic in several respects, including their acceptability by employees, the location of vehicles, the fluidity of traffic, and the perception by vehicles of changing and, therefore, dynamic environments. Thus, simulation serves to account for the constraints and requirements formulated by the manufacturers and future users of autonomous vehicles. In this paper, we present the development of a co-simulation platforms, and a method for estimating the positions of the vehicles simulated in this platform.

Keywords: Autonomous industrial vehicle, Matrix beaconing, Simulation platform, Agent-based simulation, Fuzzy agent.

1. Introduction

Among the challenges facing Industry 4.0 are the development and optimization of the flows of data, products, and materials in production companies. Certain technological bricks have been defined [1, 2], in particular, for the use of automated guided vehicles (AGVs) and other autonomous mobile robots. While the autonomy of autonomous vehicles has been well characterized in the field of road and road transport (6 autonomous driving levels distinguishes by the Society of Automotive Engineers [3]), this is not the case for the autonomous vehicles used in industry (autonomous industrial vehicles or AIVs) [4]. The establishment and deployment of AIV fleets in industrial companies remain problematic in several respects, including their acceptability by employees, the location of vehicles, the fluidity of traffic, and the perception by vehicles of changing and, therefore, dynamic environments. Autonomy has, accordingly, been limited to predetermined trajectories. Thus, the capacity to exchange information among the various AIVs of a fleet should improve this autonomy in terms of:

- adaptation to traffic constraints, especially when the AIV environment changes over time (in the dynamic environments of storage areas, production lines, etc.), with; this adaptive capacity making full use of the development of AI and IoT technologies [5] to perceive the environment;
- Decision-making, even when the information available to an AIV is incomplete, uncertain, or available but fragmented [6];
- Communication with other AIVs in a fleet and with the associated infrastructure or people (commonly referred to as “V2X communications”) [7];
- Reduction (or simply control) of the energy impact, irrespective of any traffic constraints [8].

In this paper, we present our research on these issues and offer a state- of- the- art proposal related to communication among and with autonomous vehicles, the development of simulation platforms, and the location of autonomous vehicles. We then describe the co-simulation platform that we have developed and our method for estimating the positions of the vehicles simulated thereon.

2. State of the Art

2.1. Communication Among Vehicles

The experimental self-driving cars that are already plying roads all over the world to accumulate data and miles do not cooperate with their surroundings. Instead, they rely on on-board sensors [9], such as radar, laser or lidar, cameras, and GPS and information collected internally (through an odometer, assessment of the condition of the wheels, and so on), to acquire raw information with which to construct a representation of their surroundings. A vehicle’s control system then matches its perception with a priori known information, such as a detailed map or a learned representation of the environment in which it is operating [10, 11] to choose a course of action and position itself on the road. The same is true of the AIVs increasingly deployed at industrial production sites, which still have very limited capacity for adaptation.

In recent years, the automotive industry has joined forces with telecommunications players to develop communication standards that facilitate direct cooperation among vehicles through the exchange of structured information [12]. Thus, for instance, a vehicle may start to deaccelerate or brake, not because it observes that it is approaching the vehicle ahead of it, but because the vehicle ahead indicates that it has initiated such an action. This type of coordination saves precious time in reactions to critical events and, therefore, promotes safety in addition to contributing significantly to profitability. Thus, for example, vehicles can be linked for movement on a highway in convoys (platooning) [13] or to optimize passing through intersections [14].

2.2. Simulation Platforms for AIVs

Before full-scale testing of traffic scenarios involving autonomous vehicles in industrial or more complex traffic situations can begin, it is essential to consider the simulation involved. Perhaps the greatest advantage to be gained by running a simulation is that actionable results can be obtained without applying a scale factor [15-19]. As might be expected, there are numerous methods in use for such testing [20].

While progress in the autonomy of automobiles is widely reported [21], including to the general public, studies of AIVs have been relatively few. AGVs and autonomous vehicles more generally have the capacity to adapt to their environs. A combination of computer and physical solutions can facilitate shared communications and, thereby, the autonomy of these vehicles. Agent-based approaches are often presented in this case [22, 23]. We propose here use of the notion of a fuzzy agent to manage the levels of imprecision and uncertainty involved in modeling the behavior of simulated vehicles [24, 25].

2.3. Estimating the Locations and Positions of AIVs

A position estimation provides an approximation of a vehicle's location in relation to its environment. The literature on estimation theory is vast, encompassing a wide variety of techniques and ideas. Naturally, the most common techniques receive frequent attention [26-28]. These general techniques can be applied to a variety of problems, an example being parametric estimation methods such as weighted least squares estimators, maximum-likelihood estimators, minimum mean-square error estimators [26].

Incremental or relative localization [27] makes it possible to determine the position and orientation of a vehicle by taking into account its successive movements from a known starting point. Absolute localization [27], by contrast, involves determining the position of a vehicle or robot in its external or internal environment using exteroceptive sensors. Two strategies are used for localization that rely on either natural or artificial landmarks (e.g., GPS or beacons), respectively. Absolute localization by definition avoids the drift over time that characterizes relative localization; the main disadvantage of this strategy is the loss of visibility of the landmarks in the environment that a vehicle uses to determine its position.

In our study, the measurements necessary for the estimation were susceptible to corruption by noise. The result can be generation of an input that introduces uncertainty into the inference. Uncertainty is, then, at the heart of the estimation problem: in the absence of uncertainty, many problems would have simple algebraic solutions [26].

3. A Co-simulation Platform

3.1. Presentation of the Platform

The simulation platform is composed of (1) a digital simulation framework that is agent-oriented, allowing it to simulate the movements of vehicles in a virtual environment, and (2) a physical platform that serves to develop scenarios for the circulation of vehicles of reduced size or a set of small vehicles. The objective is to visualize the same movements through the virtual and physical simulations. Fig. 1 shows the platform architecture. The obstacles that a vehicle encounters on the physical platform must appear on the software platform. The platform also offers the ability to conduct augmented simulations (for example, adding a new vehicle, a person, or even direct communications between AIVs virtually). We developed two interfaces to follow the evolution of the simulations, one on the server side, for viewing the simulation and managing the components of the two systems, and the physical system, including communication with the vehicles moving on the platform and the virtual system including communication with simulator agents (Fig. 2). This latest HMI allows users to increase the simulation by introducing a new virtual vehicle to the set or making a human operator appear on the vehicle's traffic map.

The AIVs of the physical platform are small and capable of following the road (line tracking), stopping in front of an obstacle, geolocating on the circuit, communicating by radio, and transmitting information (position, speed, etc.) or receiving it from roadside equipment. They can also decide on an action to be taken based on all of the information received from the environment.

The simulator's AIVs are fuzzy autonomous software agents. Thus, they manage their movements while responding to the directives of the server (or of the simulator through the server). To do so, the fuzzy agents communicate with the server or each other.

3.2. Presentation of the Physical Platform

When designing the scenarios involving autonomous vehicles, we were particularly interested in those that favor traffic congestion situations. Accounting for this characteristic led us naturally to diagram the circulation in four loops, as Figs. 2 (the HMI) and 3 (the physical platform) show.

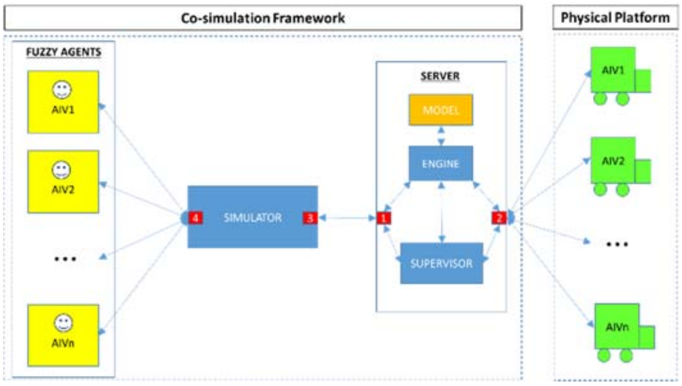
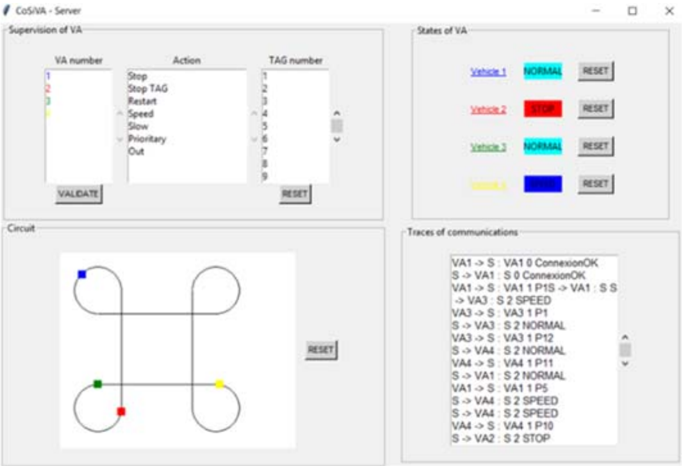


Fig. 1. Architecture of the co-simulation platform.



```
VA1 -> S : VA1 0 ConnexionOK
S -> VA1 : S 0 ConnexionOK
VA1 -> S : VA1 1 P1S -> VA1 : S S
 -> VA3 : S 2 SPEED
VA3 -> S : VA3 1 P1
S -> VA3 : S 2 NORMAL
VA3 -> S : VA3 1 P12
S -> VA4 : S 2 NORMAL
VA4 -> S : VA4 1 P11
S -> VA1 : S 2 NORMAL
VA1 -> S : VA1 1 P5
S -> VA4 : S 2 SPEED
S -> VA4 : S 2 SPEED
VA4 -> S : VA4 1 P10
S -> VA2 : S 2 STOP
```

Fig. 2. Server HMI.

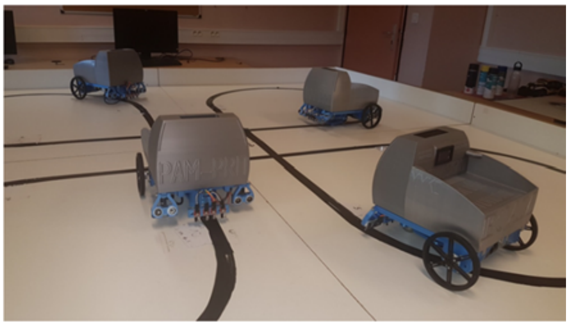


Fig. 3. Autonomous vehicles on the physical platform.

We thus defined a traffic platform as an entity consisting, on the one hand, of a circuit made up of four curving sections and four straight sections placed end- to-end to form what we designated "CircuitLacet4", and, on the other hand, of a set of 12 RFID tags distributed along the circuit. Each quadrant had three markers, with each marker being represented by one RFID tag, so that any nearby vehicle could position itself on the circuit.

Beyond basic functionality, such as line tracking and the detection of nearby obstacles, we incorporated three speed-increasing kinematics into the AIVs. We also developed communication functions (using Wi-Fi) between the vehicles and the server so that each vehicle could send its position back to the server relative to the RFID tags on the road circuit.

3.3. Internal Architecture of an AIV

We equipped each AIV with modules that allowed us to test all of our scenarios of interest. These included controlling modules and modules for detecting RFID tags, indicating the vehicle's status on an LCD display, detecting obstacles, regulating the power supply, transmitting data to the server, receiving instructions from the server regarding movements, operating the motors, and line monitoring. It is important that the electronic module for each of these functions be easily identifiable within the system, just as each function must be easily monitored, when conducting a dynamic search for sources of dysfunction. Thus, for each autonomous vehicle to fulfill the tasks assigned to it, we had to define the organization in as structured a manner as possible from the perspective of the software, the hardware, and the electronics. The result of this work was a complete internal map of the autonomous vehicles. Fig. 4 highlights the various modules as well as the connections among them.

The PRAV module, which controls the detection of RFID tags and the display, is connected physically to both the RFID module and the AFFI module. The ground markings detection (DEMS) and wheel motor control (PMOT) modules are connected to the module dedicated to the line tracking control (PILS) through the BRAC module. Doubly connected in this way, the BRAC module performs the intermediary role of mixing and optimizing the multiple paths of the electrical links, which are already in a certain amount disorder, with the result that it is almost impossible to locate the sources of dysfunction. We designed the BRAC module, being thus located in the center of a star based on DEMS, PMOT, and PILS, to connect with the PILS module through simple superposition according to the form factor of an Arduino board in a connection identified as Bus3 in Fig. 4. The TSVA module, which also controls an obstacle detection module (DOBS) through a direct four-wire connection, provides telecommunications with the remote server. Notably, the TSVA is implemented by means of a raspberry card, which requires a supply of current that is both sufficient and stable. This situation justifies the presence of the power supply regulator module (RALI) that is connected through Bus6 to its input (USB-C). In a multi-connection crossroads, the TVSA is the most- surrounded module in the system, its neighbors being the server by the radio link, the DOBS module by the direct link, RALI by Bus6, the PRAV module by Bus5, and, lastly, PILS by Bus3. Fig. 4 shows clearly the central position of the TSVA module from the perspectives of both its functions and its systemic connections.

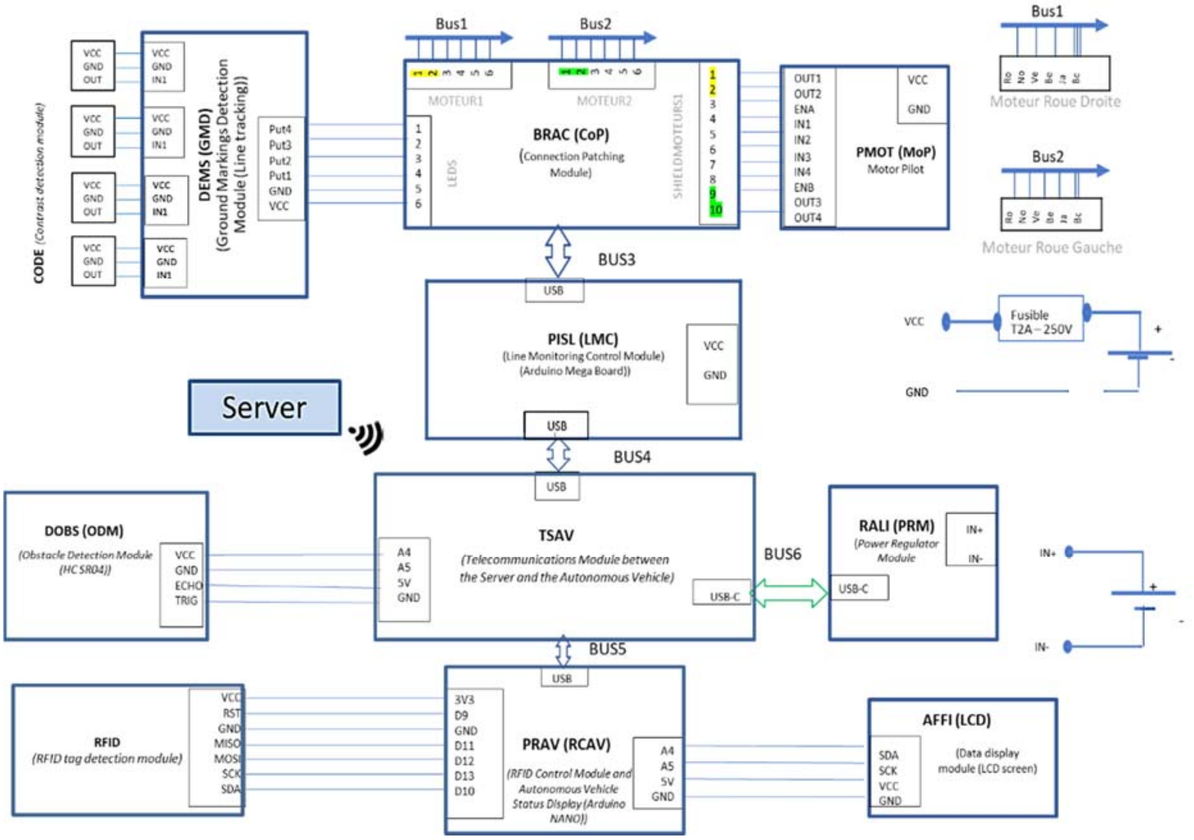


Fig. 4. Internal architecture of the autonomous vehicles.

4. The AIV Position Estimation Method

4.1. The Conceptual Model of the Platform

The co-simulation system includes a simulator capable of reproducing virtually and the AIV evolving in its traffic environment, also called the traffic zone. The conceptual vision of such a device naturally leads to the highlighting of a set of entities, each representing a real object deemed sufficiently relevant for inclusion in the simulation model. The main entities that constitute the static model of the simulation device as a whole are the traffic area, the beacons, the components of the AIVs involved in the estimation of their positions, the circuit that we termed “CircuitLacet4” shown in Fig. 5, and the section. Fig. 6 shows the class diagram of the static model of the entire simulation device.

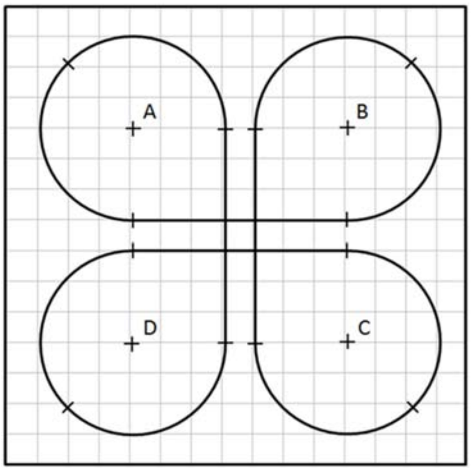


Fig. 5. “CircuitLacet4” circuit profile.

4.2. The Mathematical Position Estimation Model

Our computational approach also makes it possible to obtain the next position of the AIV on the circuit given its current position. Schematically, we defined the current position of the AIV as Pn and the next position as Pn + 1. Our approach was to define an abstract chunk model in which the updating of the current position would only be stated in principle, with no details provided about the concrete implementation. We then defined a concrete traffic section model by building on the aforementioned abstract model. The concrete section model served for the actual calculation of Pn + 1. Thus, we formally defined three types of concrete sections with their associated calculation intelligences: a circular arc section, a horizontal section, and a vertical section.

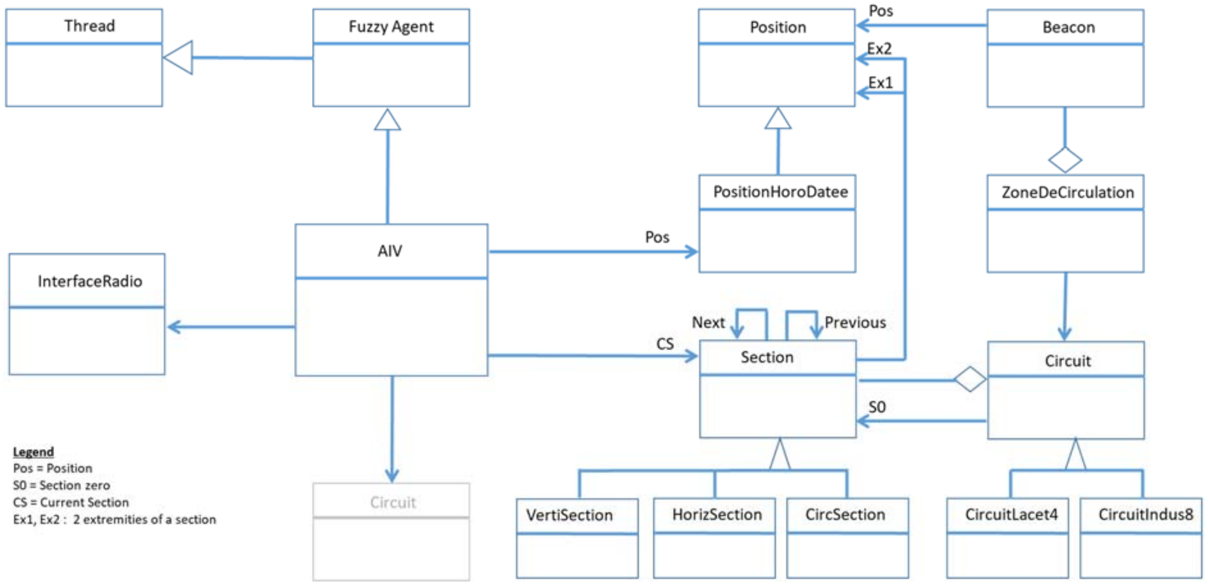


Fig. 6. Static model of the global simulation system.

For a circular arc section with center C and radius R, and for a time step Δt, the update of the position of the AIV is given by the expression (1). For horizontal and vertical sections respectively, expressions (2) and (3) give the required update.

$$\begin{cases} x_{n+1} = x_C + R\cos\left(arctg\left(\dfrac{y_n - y_C}{x_n - x_C}\right) + \dfrac{v}{R}\Delta t\right) \\ y_{n+1} = y_C + R\sin\left(arctg\left(\dfrac{y_n - y_C}{x_n - x_C}\right) + \dfrac{v}{R}\Delta t\right) \end{cases}, \qquad (1)$$

$$\begin{cases} x_{n+1} = & x_n + v\Delta t \\ y_{n+1} = & y_n \end{cases}, \qquad (2)$$

$$\begin{cases} x_{n+1} = & x_n \\ y_{n+1} = & y_n + v\Delta t \end{cases}, \qquad (3)$$

4.3. The AIV Position Update Algorithm

Thus integrated into the concrete section model, the intelligence for calculating the next position of the VA also allows for management of the “handover” of the autonomous vehicle from one section to the next. The flowchart in Fig. 7 shows the algorithm for this processing.

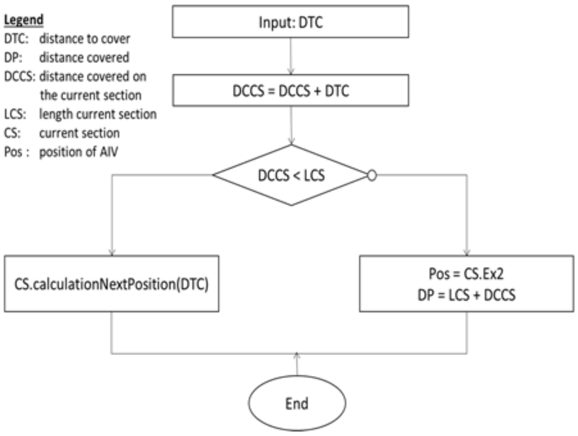


Fig. 7. Algorithm for updating the position of an AIV.

5. Conclusions

There has been a great deal of research on autonomous vehicles, but relatively little of it has concerned industrial vehicles (or mobile robots), with the focus instead remaining on road vehicles. Obvious similarities exist between these two uses of autonomous vehicles, starting with the need to simulate the vehicles and their traffic contexts before developing and deploying them in real environments. In the industrial field, simulation serves to account for the constraints and requirements formulated by the manufacturers and future users of autonomous vehicles.

The development of simulation platforms is, therefore, an important step in improving the autonomy of AIVs. The platforms identified in the literature have been either virtual or physical, but our approach relies on co-simulation platform experiments combining the physical and virtual approaches. On both the physical and virtual levels, it is essential to determine the correct location of the vehicles. We accordingly proposed an approach for estimating the position of AIVs according to the principle of matrix beaconing that we then implemented in our simulation framework.

There are many possible ways in which the co-simulation system can evolve. The main goals are to be able to modify the physical platform freely, for the system to adapt to it autonomously, and, ultimately, to deploy the simulation in any industrial environment. A manufacturer of AIVs could thus provide the simulation to customers and work with them on deployment scenarios specific to their job sites.

References

[1]. H. Lasi, P. Fettke, H. G. Kemper, T. Feld, M. Hoffmann, Industry 4.0, *Business & Information Systems Engineering*, Vol. 6, Issue 4, 2014, pp. 239-242.

[2]. A. C. Pereira, F. Romero, (2017). A review of the meanings and the implications of the Industry 4.0 concept, *Procedia Manufacturing*, Vol. 13, 2017, pp. 1206-1214.

[3]. Y. Wiseman, Autonomous vehicles, in Encyclopedia of Information Science and Technology, Fifth Ed., *IGI Global*, 2021, pp. 1-11.

[4]. H. Andreasson, A. Bouguerra, M. Cirillo, D. N. Dimitrov, D. Driankov, L. Karlsson, A. J. Lilienthal, F. Pecora, J. P. Saarinen, A. Sherikov, and T. Stoyanov, Autonomous transport vehicles: Where we are and what is missing, *IEEE Robotics & Automation Magazine*, Vol. 22, Issue 1, 2015, pp. 64-75.

[5]. H. Khayyam, B. Javadi, M. Jalili, R.N. Jazar, Artificial intelligence and internet of things for autonomous vehicles, in Nonlinear Approaches in Engineering Applications, *Springer*, Cham, 2020, pp. 39-68.

[6]. R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, J. Barata, Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook, *IEEE Access*, Vol. 8, 2020, pp. 220121-220139.

[7]. C. Medrano-Berumen, M. I. Akbaş, Validation of decision-making in artificial intelligence-based autonomous vehicles, *Journal of Information and Telecommunication*, Vol. 5, 2020, pp. 83-103.

[8]. R. Bostelman, E. Messina, Towards development of an automated guided vehicle intelligence level performance standard, in Autonomous Industrial Vehicles: From the Laboratory to the Factory Floor (R. Bostelman E. Messina, Eds.), *ASTM International*, West Conshohocken, 2016, pp. 1-22.

[9]. S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, M. H. Ang, Perception, planning, control, and coordination for autonomous vehicles, *Machines*, Vol. 5, Issue 1, 2017, 6.

[10]. H. Zhu, K. V. Yuen, L. Mihaylova, H. Leung, Overview of environment perception for intelligent vehicles, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 18, Issue 10, 2017, pp. 2584-2601.

[11]. F. Rosique, P. J. Navarro, C. Fernández, A. Padilla, A systematic review of perception system and simulators for autonomous vehicles research, *Sensors*, Vol. 19, Issue 3, 2019, 648.

[12]. F. Arena, G. Pau, An overview of vehicular communications, *Future Internet*, Vol. 11, Issue 2, 2019, 27.

[13]. M. Y. Abualhoul, O. Shagdar, F. Nashashibi, Visible light inter-vehicle Communication for platooning of autonomous vehicles, in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV'16)*, 2016, pp. 508-513.

[14]. O. Grembek, A. Kurzhanskiy, A. Medury, P. Varaiya, M. Yu, Making intersections safer with I2V communication, *Transportation Research Part C: Emerging Technologies*, Vol. 102, 2019, pp. 396-410.

[15]. M. Buehler, K. Iagnemma, S. Singh (Eds.), The DARPA Urban Challenge: Autonomous Vehicles in City Traffic, Vol. 56, *Springer*, 2019.

[16]. M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, J. C. Duchi, Scalable end-to-end autonomous vehicle testing via rare-event simulation, in *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'18)*, 2018.

[17]. B. Y. Ekren, S. Heragu, Simulation based performance analysis of an autonomous vehicle storage and retrieval system, *Simulation Modelling Practice and Theory*, Vol. 19, Issue 7, 2011, pp. 1640-1650.

[18]. F. Bounini, D. Gingras, V. Lapointe, D. Gruyer, Real-time simulator of collaborative autonomous vehicles. in *Proceedings of the Int. Conference on Advances in Computing, Communications and Informatics (ICACCI'14)*, 2014, pp. 723-729.

[19]. D. Kade, M. Wallmyr, T. Holstein, R. Lindell, H. Ürey, O. Özcan, Low-cost mixed reality simulator for industrial vehicle environment, in *Proceedings of the International Conference on Virtual, Augmented and Mixed Reality*, 2016, pp. 597-608.

[20]. W. Huang, K. Wang, Y. Lv, F. Zhu, Autonomous vehicles testing methods review, in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC'16)*, 2016, pp. 163-168.

[21]. S. Liu, L. Li, J. Tang, S. Wu, J. L. Gaudiot, Creating autonomous vehicle systems. *Synthesis Lectures on Computer Science*, Vol. 6, Issue 1, 2017, i-186.

[22]. P. Jing, H. Hu, F. Zhan, Y. Chen, Y. Shi, Agent-based simulation of autonomous vehicles: A systematic literature review, *IEEE Access*, Vol. 8, 2020, pp. 79089-79103.

[23]. K. Dresner, P. Stone, A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, Vol. 31, 2008, pp. 591-656.

[24]. A.-J. Fougères, A modelling approach based on fuzzy agent, *International Journal of Computer Science Issues*, Vol. 9, Issue 6, 2013, pp. 19-28.

[25]. A.-J. Fougères, E. Ostrosi, Fuzzy agent-based approach for consensual design synthesis in product configuration, *Integrated Computer-Aided Engineering*, Vol. 20, Issue 3, 2013, pp. 259-274.

[26]. I. J. Cox, Blanche: Position estimation for an autonomous root vehicle, in Autonomous Robot Vehicles, *Springer*, New York, NY, 1990, pp. 221-228.

[27]. C. Aynau, C. Bernay-Angeletti, R. Aufrere, L. Lequievre, C. Debain, R. Chapuis, Real-time multisensor vehicle localization: A geographical information system based approach, *IEEE Robotics & Automation Magazine*, Vol. 24, Issue 3, 2017, pp. 65-74.

[28]. H. S. Hasan, M. Hussein, S. M. Saad, M. A. M. Dzahir, An overview of local positioning system: Technologies, techniques and applications, *International Journal of Engineering & Technology*, Vol. 7, Issue 3, 2018, pp. 1-5.

(024)

On Alternating and Residuality Language Learning Algorithms

Aziz Fellah
School of Computer Science and Information Systems
Northwest Missouri State University, Maryville, MO 64468 USA
E-mail: afellah@nwmissouri.edu

Summary: Alternation is considered as a natural generalization of nondeterminism whereas residuality is considered as a natural distillation of the essence of the automaton's states language recognition. In this paper, we investigate two crossed semantic properties termed alternation and residuality. Both properties become central to the study of regular languages and related problems, and provoked a tremendous amount of research. Alternation becomes an appealing abstraction and a key ingredient in modeling software systems, including model checking, formal methods, and software programs' verification. Residuality is a framework that allows one to eventually perform the semantic checking of each state of the automaton independently. To name a few applications in residuality: programming languages artificial intelligence, and learning algorithms such as L*, NL, and AL*. We present a new paradigm of learning regular languages represented by a special type of alternating finite automata (AFA), namely reversal AFA (*r*-AFA), which provide a succinct representation of regular languages. Using the residuality property, we introduce residual language equations which exactly correspond to the states of the *r*-AFA. Such a model can be described naturally as a set of residual language equations that parallels the solutions of algebraic equations. Moreover, the solution of such systems of residual language equations is the class of regular languages. Furthermore, we exploit the succinctness relationship between *r*-AFA and DFA (deterministic finite automata), and we develop a new active learning algorithm, called *r*-AL*, which is complemented with an extension of L*.

Keywords: Alternating and residual finite automata, Language learning algorithm, L*, NL, AL*, *r*-AL*, Alternation, Residuality, Residual language equations, Derivative languages.

1. Introduction

The notion of alternation is a natural generalization of nondeterminism. It received its historical details and formal treatment in [1]. This seminal paper and most of the subsequent research focused on a variety of alternating automata in terms of their types, sizes, languages, and computational complexities. The alternating property has played an important role in understanding many questions in complexity theory and model checking. For alternating finite automata (AFA), it is proved that they are precisely as powerful as deterministic finite automata (DFA) as far as language recognition. However, beyond this seemingly negative result the presence of alternation can lead to simplified construction in the area of finite automata [2-6]. Unlike finite automata and Turing machines, alternation increases the expressiveness of pushdown automata, precisely synchronized alternating pushdown automata (PDA). That is, PDA add the power of conjunction over context-free grammar which lay the foundation of one of the most widely used class of languages, context-free languages, in computer science [7-9]. In terms of the number of states, a minimal deterministic finite automata (DFA) might be exponentially larger than a nondeterministic finite automaton (NFA) and double exponentially larger than an alternating finite automata (AFA). All these automata have the same expressive power in terms of language recognition − they accept regular languages but differ in efficiency. Furthermore, AFA have particularly emerged as practical tools in several applications. To name a few: model checking, learning algorithms, formal methods and software programs' verification. Furthermore, learning regular languages has gained widespread in a wide variety of applications such as pattern recognition, data mining and learning algorithms. For example, Angluin-style et al. learning [10-12] well-known learning algorithm styles L* for DFA and NL for NFA have provoked a tremendous amount of research in machine learning, artificial intelligence, and other areas. For many practical applications, it is desirable to work with AFA rather than other types of automaton because of their succinctness and membership problem's efficiency. The uniqueness of the minimality of the DFA play an important role in the language learning algorithm L* [10] where it has been proved that the class of regular languages could be learned efficiently, in polynomial time in terms of the number of states of the DFA. A generalization of L*, NL has been extended to NFA for learning residual NFA (RNFA for short).

Residuality is considered as a natural distillation of the essence of the automaton's states language recognition. It adds a natural meaning to the automaton's states. An automaton A accepting a language $\mathcal{L}$ is residual if every state q of A represents a residual language, specifically $\partial_w(\mathcal{L}) = \{v \in \Sigma^* \mid wv \in \mathcal{L}\}$ of $\mathcal{L}$. Residual automata discern significant facts from the semantics of each state of the automaton. Importantly, every DFA have a useful property termed residuality. Such a semantic property plays an important role for learning finite automata, underlines the algorithm L* [10] for learning DFA, and generalizes the NL algorithm for NFA [11]. In addition, residual automata play an important role in the context of language algorithmic learning, and it is considered as a steppingstone towards language learning algorithms. AFA's structure is more compact than that of NFA and DFA, which makes residuality very desirable when learning AFA. Angluin et. al [12] extended the learning algorithm L^* to residual automata. In this paper, we investigate two semantic properties termed alternation and residuality which complement each other, and we encapsulate them in a model that we refer to as alternating residual automata (ARA). In addition, we show a new perspective on the AL^* algorithm for learning alternating residual automata. We study the efficiency of the algorithms using regular languages which are represented by extended regular expressions. Technically, the states of DFA and ARA have the property of residuality which corresponds to residual languages [13-15]. The learning framework for ARA starts with the learner who can formulate membership queries to test if a word is in $\mathcal{L}$, and equivalence queries to ask if a hypothesis is equivalent to the target language (*i.e.*, target automaton), otherwise a counterexample is returned. The remaining of the paper is organized as follows. In Section 2, we introduce some basic preliminary concepts and definitions. Section 3, alternating finite automata and background are described in some details. Section 4 looks at the related work and background of derivatives of regular expressions and languages. Section 5, the property of residuality is described as an important framework in languages algorithmic learning. In Section 6, we present a new paradigm approach, namely *r*-AL^*, a reversal alternating and residual learning algorithm for regular languages. In addition, we introduce residual language equations that parallels the solution of algebraic equations. Finally, in Section 7 we conclude the paper with a summary and discuss the future work.

2. Preliminaries

In this section we briefly recall some relevant definitions. An alphabet is a finite, nonempty set. The elements of an alphabet are called symbols or letters. A word (string) over an alphabet Σ is a finite sequence consisting of zero or more symbols of Σ. The set of all words (respectively all nonempty words) over an alphabet Σ is denoted by Σ^* (respectively Σ^+). A language over Σ is a (possibly infinite) set of finite words $\mathcal{L} \subseteq \Sigma^*$. The word consisting of zero letters is called the empty word, denoted by ϵ. The length of a word w, denoted by $|w|$, is the number of symbols in w. By definition, $|\epsilon| = 0$. Given a language $\mathcal{L}$ over an alphabet Σ, the Kleene star "$*$") of $\mathcal{L}$ is the set $\mathcal{L}^* = \cup_{i=0}^{\infty} \mathcal{L}^i$ and the positive Kleene plus "+") of $\mathcal{L}$ is $\mathcal{L}^+ = \cup_{i=1}^{\infty} \mathcal{L}^i$. The language $\overline{\mathcal{L}} = \Sigma^* \backslash \mathcal{L}$ is the complement of $\mathcal{L}$. The *concatenation* of two words u and v is the word consisting of the symbols of u followed by the symbols of v, denoted $u \cdot v$ (also written as uv). We say that an extended regular expression e is *nullable* if the language it represents contains the empty string, that is if $\lambda \in \mathcal{L}$ (e). The set of extended regular expressions E over Σ is the subset of $(\Sigma \cup \{\epsilon, \varphi, +, \cdot, *, \neg, \cap, (,)\})^*$ that satisfies the following conditions:

(i) $\Sigma \cup \{\lambda, \varphi\} \subseteq \mathsf{E}$;

(ii) If $e_1, e_2 \in \mathsf{E}$ then $(e_1 + e_2)$, $(e_1.e_2)$, $(e_1{}^*)$, $(e_1 \cap e_2)$, and $(\neg e_1) \in \mathsf{E}$.

The language denoted by an extended regular expressions $e \in \mathsf{E}$, denoted by $\mathcal{L}(e)$, is defined inductively as follows:

(*i*) $\mathcal{L}(\varphi) = \varphi$, $\mathcal{L}(\epsilon) = \epsilon$, and $\mathcal{L}(a) = \{a\}$ for $a \in \Sigma$.

(*ii*) if $e_1, e_2 \in \Sigma$. Then $\mathcal{L}((e_1 + e_2)) = \mathcal{L}(e_1) \cup \mathcal{L}(e_2)$, $\mathcal{L}((e_1 \cdot e_2)) = \mathcal{L}(e_1) \cdot \mathcal{L}(e_2)$, $\mathcal{L}((e_1 \cap e_2)) = \mathcal{L}(e_1) \cap \mathcal{L}(e_2)$, $\mathcal{L}((e_1{}^*)) = \mathcal{L}(e_1)^*$, and $\mathcal{L}((\neg e_1)) = \overline{\mathcal{L}(\mathrm{e1})}$.

Thus, extended regular expressions e and their corresponding languages, $\mathcal{L}(e)$, can be used interchangeably. We denote the *reversal* of a word w by w^R, while the reversal of a language $\mathcal{L}$, denoted $\mathcal{L}^R$, is defined as $\mathcal{L}^R = \{w^R \mid w \in \mathcal{L}\}$. We denote by the symbol B the Boolean semiring, $\mathsf{B} = \{0, 1\}$. Let Q be a set. Then B^{Q} is the set of all mappings of Q into B. Note that $\hat{u} \in \mathsf{B}^{\mathsf{Q}}$ can also be considered as a Q-vector over B.

3. Alternating Finite Automata (AFA)

Alternating finite automata (AFA) has the property of alternation in the following sense: If in a given state q the automaton reads an input symbol a, it will activate all states of the automaton to work on the remaining part of the input in parallel. Once the states have completed their tasks, q will evaluate their results using a Boolean function and pass on the resulting value by which it was activated. A word w is accepted if the starting state computes the values of 1. Otherwise, it is rejected. NFA are a generalization of DFA by allowing a state to have multiple outgoing transitions labeled with the same symbol or a ϵ-transition. A string is accepted by an NFA if there exists some path that leads to an accepting state. In a nondeterministic computation all configurations are *existential* in the sense that there exists at least one successful path that leads to acceptance. An alternating finite automaton (AFA) may have also *universal* configurations from which the computation branches into a number of parallel computations that must all lead to acceptance. We represent existential and universal choices by a Boolean formula. Formally, let Q be a set, we use B^{Q} to be the set of all Boolean formulas over Q. That is, B^{Q} is built from the elements $q \in Q$, 1, and 0 using the binary operations and ($\vee$), or ($\wedge$), and not ($-$). We now formalize this idea.

Definition 1: An alternating finite automaton (AFA) is a quintuple $\mathsf{A} = (\Sigma, Q, s, F, g)$ where (a) Σ is an alphabet, the input alphabet; (b) Q is a finite set, the set of states; (c) $s \in Q$ is the starting state; (d) $F \subseteq Q$ is the set of final states; (e) g is a mapping of Q into the set of all mappings of $\Sigma \times \mathsf{B}^{\mathsf{Q}}$ into B.

We turn to defining the sequential behavior of an AFA. For $q \in Q$ and $a \in \Sigma$, let $g_q(a)$ be the Boolean function defined as:

$$g_q(a, \hat{u}): \Sigma \times \mathsf{B}^{\mathsf{Q}} \rightarrow \mathsf{B},$$

where a $\in \Sigma$ and $\hat{u} \in \mathsf{B}^{\mathsf{Q}}$. Also, for a $\in \Sigma$, $q \in$ Q, and $\hat{u} \in \mathsf{B}^{\mathsf{Q}}$, $g_q(a, \hat{u}) = g_q(a)(\hat{u})$, is equal to either 0 or 1. Later, we also need the mappings $g(a)$ of Q into the set of all mappings of B^{Q} into B and the mappings $g_q(a)$ of B^{Q} into B defined by

$$g(a)(q)(\hat{u}) = g_q(a)(\hat{u}) = g_q(a, \hat{u}),$$
$$\text{for } a \in \Sigma,\ q \in Q, \text{ and } \hat{u} \in \mathsf{B}^{\mathsf{Q}}$$

Now define $f \in \mathsf{B}^{\mathsf{Q}}$ by the condition

$$fq = 1 \Longleftrightarrow q \in F,$$

f is called the characteristic vector of F. We extend g to a mapping of Q into the set of all mappings of $\Sigma^{*} \times \mathsf{B}^{\mathsf{Q}}$ into B as follows:

$$g_q(w, \hat{\mathrm{u}}) \ =$$
$$= \begin{cases} \mathrm{u}_q \text{ if } w \ = \ \epsilon \\ g_q(a, g(v, \hat{\mathrm{u}})) \text{ if } w \ = \ av \text{ with } a\,\Sigma, v \ \in \Sigma^{*'} \end{cases}$$

where $w \in \Sigma^{*}$ and $\hat{u} \in \mathsf{B}^{\mathsf{Q}}$.

Definition 2: Let $\mathsf{A} = (Q, \Sigma, s, F, g)$ be an AFA. A word $w \in \Sigma^{*}$ is accepted by A if and only if $g_s(w, f) = 1$. The language accepted by A is the set $\mathcal{L}(\mathsf{A}) = \{w \mid w \in \Sigma^{*} \wedge g_s(w, f) = 1\}$.

We denote the language of A by $\mathcal{L}(\mathsf{A})$ and the language accepted by a state $q \in$ Q by $\mathcal{L}$. Note that in the same spirit as the characteristic vector of F, we extend g to languages. Thus, we define the characteristic output of A, $g_{\mathsf{A}}(w, \hat{u})$, as follows.

Definition 3: Let $\mathsf{A} = (Q, \Sigma, s, F, g)$ be an AFA and $\mathcal{L}(\mathsf{A})$ the language accepted by A. Then, characteristic output of A is defined as

$$g_{\mathsf{A}}(w, \hat{u}) = \begin{cases} 1 \ if \ g_q(w, \hat{\mathrm{u}}) \ = \ 1 \text{ for all } w \ \in \ \mathcal{L}(\mathsf{A}) \\ \qquad\qquad 0 \ otherwise \end{cases}$$

Example 1: Consider the following AFA $\mathsf{A} = (Q, \Sigma, s, F, g)$ where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $s = \{q_0\}$, $F = \{q_2\}$, and g is given by the following table $F = \{q_2\}$. The example shows a run of the AFA on the input *bab*. See Table 1.

Table 1. AFA: State table.

	a	b
q_0	$q_0 \wedge q_1$	$q_1 \vee q_2$
q_1	q_0	$q_0 \wedge q_2$
q_2	$q_0 \vee q_1$	1

In the same example of AFA, we have drawn the existential states as $\vee$ and the universal states as $\wedge$. The AFA can have multiple runs on a given input where both of these choices coexist. Notice that the run branches in parallel to two states, q_0 and q_2 on the second input symbol b from q_1. See Fig. 1.

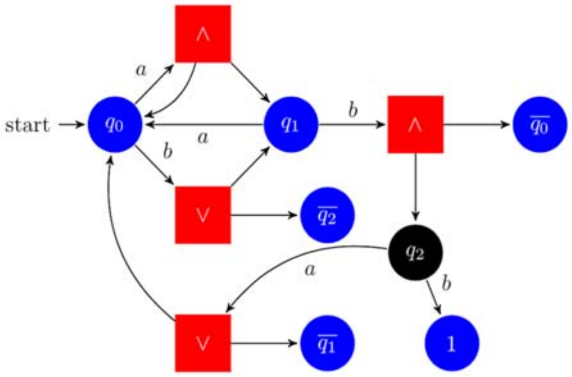


Fig. 1. AFA accepting the string *aba*.

In addition, we can show three separate mappings of B^{Q} into B. That is, $g(q_0)$, $g(q_1)$, and $g(q_2)$. See Table 2.

Table 2. $\mathrm{g}(q_0)$, $g(q_1)$, $g(q_2)$.

	a	b		a	b		a	b
000	0	1	000	0	0	000	1	0
001	0	0	001	0	1	001	1	0
010	0	1	010	0	0	010	0	1
011	0	1	011	0	1	011	0	1
100	0	1	100	1	0	100	1	0
101	0	0	101	1	0	101	1	0
110	1	1	110	1	0	110	1	1
111	1	1	111	1	0	111	1	1

4. Derivatives of Regular Expressions and Languages

The notion of derivative regular expressions has been introduced by Brzozowski [16] in order to find quotient on regular expressions and give corresponding derivatives and their auxiliary functions. Let e be an extended regular expression and u is a word over Σ^{*}. We denote by $\partial_u(e)$ the derivative of e with respect to u, formally defined as

Definition 4: The derivative of an extended regular expression e with respect to a word $u \in \Sigma^{*}$ is defined to as: $\partial_u(e) = \{v \in \Sigma^{*} \mid uv \in e\}$.

Intuitively, ∂_u(e) is the set of all remaining strings obtainable from e by taking off the prefix u, if possible. The derivatives of an extended regular expression e with respect to a symbol $a \in \Sigma$ are defined as follows:

$$\partial_a(\phi) = \phi,$$
$$\partial_a(\epsilon) = \phi,$$
$$\partial_a(a) = \epsilon,$$
$$\partial_\epsilon(a) = a,$$

$$\begin{gathered}\partial_a(b) = \phi \text{ if } b \neq a,\\ \partial_a(e_1 + e_2) = \partial_a(e_1) + \partial_a(e_2),\\ \partial_a e_1^* = \partial_a(e_1) \cdot e_2^*,\\ \partial_a(e_1 \cap e_2) = \partial_a(e_1) \cap \partial_a(e_2),\\ \partial_a(e_1) = \overline{\partial a\ (e_1)}\end{gathered}$$

$$\begin{gathered}\partial_a(\phi) = \phi,\\ \partial_a(\epsilon) = \phi,\\ \partial_a(a) = \epsilon,\\ \partial_\epsilon(a) = a,\\ \partial_a(b) = \phi \text{ if } b \neq a,\\ \partial_a(e_1 + e_2) = \partial_a(e_1) + \partial_a(e_2),\\ \partial_a e_1^* = \partial_a(e_1) \cdot e_2^*,\\ \partial_a(e_1 \cap e_2) = \partial_a(e_1) \cap \partial_a(e_2),\\ \partial_a(e_1) = \overline{\partial a\ (e_1)}\end{gathered}$$

$$\partial \mathrm{a}(e_1 \cdot e_2) = \begin{cases}\partial \mathrm{a}(e_1) \cdot e_2 + \partial \mathrm{a}(e_2) \text{ if } e_1 \text{ is nullable}\\ \partial \mathrm{a}(e_1) \cdot e_2 \text{ otherwise}\end{cases}$$

The last equation takes the symbol a off of the first expression e_1, or from the second regular expression e_2 if $\mathcal{L}\ (e_1)$ is nullable ((*i.e.*, the empty string $\epsilon \in \mathcal{L}\ (e_1)$).

Example 2: Let $\Sigma = \{a, b\}$.

(i) Let $e = a(ab)^*$. Then the derivatives of e with respect to a and b respectively are $\partial_a(e) = (ab)$ and $\partial_b(e) = \phi$.

(ii) Let $e = (ab + a)^*ba$. Then the derivative of e with respect to a using some auxiliary calculations is:

$$\begin{gathered}\partial_a(\mathrm{e}) = \partial_a((ab + a)^*ba) =\\ = \partial_a((ab + a)^*)ba + \partial_a(ba) =\\ = \partial_a((ab + a))(\mathrm{ab} + b)^*ba + \partial_a(b)a =\\ = \partial_a((ab + a))(ab + b)^*ba + \partial_a(b)a =\\ = (\partial_a(ab) + \partial_a(a))(ab + a)^*ba + \partial_a(b)a =\\ = (b + \epsilon)(ab + a)^*ba + \partial_a(b)a =\\ = (b + \epsilon)(ab + a)^*ba + \phi =\\ = (b + \epsilon)(ab + a)^*ba,\end{gathered}$$

$$\begin{gathered}\partial_a(\mathrm{e}) = \partial_a((ab + a)^*ba) =\\ = \partial_a((ab + a)^*)ba + \partial_a(ba) =\\ = \partial_a((ab + a))(\mathrm{ab} + b)^*ba + \partial_a(b)a =\\ = \partial_a((ab + a))(ab + b)^*ba + \partial_a(b)a =\\ = (\partial_a(ab) + \partial_a(a))(ab + a)^*ba + \partial_a(b)a =\\ = (b + \epsilon)(ab + a)^*ba + \partial_a(b)a =\\ = (b + \epsilon)(ab + a)^*ba + \phi =\\ = (b + \epsilon)(ab + a)^*ba\end{gathered}$$

In a similar manner, we can compute the derivative of e with respect to b, that is, $\partial_b(e)$.

In addition, the concept of derivatives apply to languages. For a language $\mathcal{L}$, the derivative of $\mathcal{L}$ with respect to a string $w$is the set of remainder words after having read w from any word in $\mathcal{L}$, as formally defined as follows:

Definition 5: The derivative of a language $\mathcal{L} \subseteq \Sigma^*$ with respect to a word $w \in \Sigma^*$ is defined to as $\partial_w(\mathcal{L}) = \{v \in \Sigma^* \mid wv \in \mathcal{L}\}$.

Proposition 1*:* Let $a \in \Sigma$, $w \in \Sigma^*$, and $e \in \mathsf{E}$. Then $aw \in \mathcal{L}\ (e)$ iff $w \in \mathcal{L}\ (\partial_a(e))$ and $\epsilon \in \mathcal{L}\ (e)$ iff nullable (e).

5. Residuality

The property of residuality introduced by Denis et al. [13] is considered as a natural distillation of the essence of the automaton's states language recognition. Importantly, it adds a foundational meaning for a better understanding of regular language algorithms learning and computational learning theory. A language $\mathcal{L}_r \in \Sigma^*$ is residual of $\mathcal{L}$ if there is $u \in \Sigma^*$ such that $\mathcal{L}_r = \partial_u(\mathcal{L})$. Residual finite state automata (RFA) [13] are a subclass of NFA where each state represents a residual language of the language that is accepted by the automaton. An automaton A accepting a language $\mathcal{L}$ is residual if every state q of A corresponds to a residual language. The class of RFA lies between deterministic (DFA) and nondeterministic automata (NFA). RFA share a number of significant properties with DFA and NFA. For example, they share with DFA a significant property. RFA share with NFA the existence of automata that are exponentially smaller, in the number of states, than the corresponding minimal DFA for the language. These properties make RFA especially appealing in several areas of computer science.

Definition 6: A residual finite automaton (RFA) is a non- deterministic finite automaton (NFA) $\mathsf{A} = (\Sigma, Q, s, F, \partial)$ such that for every state $q \in Q$, $\mathcal{L}q$ is a residual language $\in \mathcal{L}\ (\mathsf{A})$.

The most significant property of residual automata (RFA) is that it performs the semantics of each state independently.

6. r-AL*: A Reversal AL*-like Alternating and Residual Learning Algorithm

With respect to minimization of AFA, we consider a special kind of AFA that we call s-AFA. An s-AFA is an AFA $\mathsf{A} = (Q, \Sigma, s, F, g)$ such as every a $\in \Sigma$ and every $\hat{u} \in \mathsf{B}^{\mathrm{Q}}$, $g_{\mathrm{q}}(a, \hat{u})$ does not depend on $\hat{u}_{\mathrm{s}}$. Intuitively, this means that the starting state s cannot be reached in any computation. Obviously, for every AFA one construct an equivalent s-AFA which has just one more state. On the other hand, if A is a $(k + 1)$-state s-AFA then there exists an equivalent k-state AFA. For example, the language $\{\epsilon, a, a^2\}$ is accepted by a 3-state s-AFA but not any 2-state AFA. s-AFA are particularly useful to simplify certain constructions of regular language learning algorithms.

Theorem 1*:* $\mathcal{L}$ is accepted by an s-AFA with $k + 1$ states if and only if $\mathcal{L}^{\mathrm{R}}$ is accepted by a DFA with 2^{k} states.

Proof: Due to the limited number of pages allowed, we only sketch the proof by construction. The reader may refer to the authors' prior work for details

[2, 6]. Let D be a (k+1)-state s-AFA and $\mathcal{L} = \mathcal{L}(D)$. Let $D = (Q_D, \Sigma, s_D, F_D)$ be a 2^k-state DFA and $\mathcal{L}$. Let $K = \{1, 2,..., k\}$ and $K_0 = K \cup \{0\}$. Without loss of generality, we assume that $Q_D = \mathrm{B}^K$ and $s_D = \{0,...,0\}$. For $\hat{u} \in \mathrm{B}^K{}_0$, let $\hat{u}' \in \mathrm{B}^K$ be defined by $\hat{u}' = \hat{u}$ for all $i \in K$. We know define a (k + 1)-state s-AFA $A = (Q_A, \Sigma, s_A, F_A, g)$ by

$$Q_A = K_0,$$
$$s_A = 0,$$
$$F_A = \begin{cases} \{0\} \text{ if } s_D \in F_A \\ 0 \text{ otherwise} \end{cases},$$
$$g(a, \hat{u})_{\mathrm{i}} = \begin{cases} \partial(\hat{\mathrm{u}}'\, a)_i \; if\; i = 0\; and\; \hat{\mathrm{u}}' \in F_D \\ 1\; if\; i = 0\; and\; \hat{\mathrm{u}}' \notin F_D \end{cases}$$

For $i \in K_0$, $a \in \Sigma$ and $\hat{\mathrm{u}} \in \mathrm{B}^K{}_0$.The function g is well defined since A is deterministic. By induction on the length of $w \in \Sigma^*$, one shows that $\hat{\mathrm{u}} = g(w, f)$ if and only if $\delta(s_D, w^R) = \hat{\mathrm{u}}'$. Since $\hat{\mathrm{u}}_0 = 1$ if and only if $\hat{\mathrm{u}}' \in F_D$, $w \in \mathcal{L}(\mathrm{A})$ if only if $w^R \in \mathcal{L}(\mathrm{D})$. □

Corollary 1: For any AFA, there exists an equivalent s-AFA having at most one additional state.

Corollary 2: Let A be an s-AFA such that $(\mathcal{L}(\mathrm{A}))^R$ is accepted by a minimized DFA with n states. Then A has at least 1 + ⌈(log$_2n$)⌉ states.

Now, we introduce a variation of s-AFA which we call r-AFA (r short for reversal). The new r-AFA is the same as an s-AFA except that the input word is read in reverse. The following theorem is straightforward proved by the above sequence of results.

Theorem 2: For each language $\mathcal{L}$ that is accepted by a DFA with n states, there exists an equivalent r-AFA with at most 1 + ⌈(log$_2n$)⌉ states.

We present a new paradigm of learning regular languages and introduce residual language equations using r-AFA. An r-AFA A = (Q, Σ, s, F, δ) can be described naturally as a set of *residual language equations* that parallels the solutions of algebraic equations. Moreover, the solution of such systems of residual equations is the class of regular languages. We use $\mathrm{X_q}$ to denote a Boolean variable associated with the state q and $\overline{x_q}$ to denote its negation. Let $\mathrm{X_q} = \{\mathrm{x_q} \mid q \in Q\}$. Then the following system $\mathcal{L}(\mathrm{A})$ of residual language equations can be used to describe A:

$$\mathrm{A} = \{\mathrm{X_q} = \textstyle\sum_{a\epsilon\in} a.\, g_q(a, \mathrm{X}) + \epsilon(f_q)\}_{q\,\epsilon\, Q}, \qquad (1)$$

$$\epsilon(f_q) = \begin{cases} \epsilon\; if\; f_q = 1 \\ \phi \text{ otherwise} \end{cases}$$

In the system $\mathcal{L}(\mathrm{A})$ of equations, the Boolean function $g_q(a, \mathrm{X})$ is considered as being given by a Boolean expression in $\mathrm{B}^{\mathrm{X_Q}}$. Any system of residual language equations of the above form has a unique solution for each X_Q, $q \in Q$. Furthermore, the solution for each X_q is regular. The system of equations (1) corresponds to the set of residual language equations of $\mathcal{L}$. That is, each residual language equation exactly corresponds to the states of A. That is, there is a bijection between the residual language equations of $\mathcal{L}$ and the states of the minimal r-AFA.

6.1. Learning from Reversal AFA, Membership, and Equivalence Queries

We present a new active learning algorithm called r-AL* which is complemented with an extension of L* [10-12] and complemented with *membership* and *equivalence queries*. The r-AL* algorithm can be seen as a game between two players − *the learner* and *the teacher* (*oracle*). The learner guesses a set of rules that define the language $\mathcal{L}$. The learner aims to construct the minimal r-AFA and consequently deriving the minimal DFA for an unknown regular target regular language $\mathcal{L}$ over Σ in polynomial time. We introduce an extension of the membership query called *reversal membership queries*. That is, the input is consumed in reverse. In this querying algorithmic framework, a learner wants to learn a regular language from a teacher who knows the regular language and can answer queries from the learner. The teacher is assumed to be fully knowledgeable in answering the queries. The algorithm can guarantee to learn a correct r-AFA which recognizes the target regular language. When the learner asks whether a word in the language, the teacher's answer is very simple, *yes* or *no*. If the teacher replies yes to an equivalence query, then the algorithm terminates, as the hypothesis r-AFA is correct. Otherwise, the teacher must supply a counterexample, that is a word in the symmetric difference of $\mathcal{L}$ with respect to $\mathcal{L}$(r-AFA). The following summarizes the main steps of the r-AL* algorithm with a focus on the r-AFA.

Teacher

(*i*) Convert AFA to r-AFA then to DFA;

(*ii*) Find the residual language equations $\mathcal{L}_q$ for $q \in Q$.

Learner-Teacher

(*i*) r-AL* *Reversal Membership Query:* Is a given reversal word w^R in the target language, *i.e.*, $w \in \mathcal{L}$?

(*ii*) *Reversal Equivalence Query:* Does a given hypothesis automaton provided in the form of r-AFA recognizes the target language, $\mathcal{L} = \mathcal{L}$($r$-AFA)?

7. Conclusion

A new active learning algorithm called r-AL* for learning AFA in the spirit of L* algorithm was introduced in this paper. Its time complexity is almost the same as that of L* [10]. Although this result extends the current results of regular languages algorithmic learning, a further improvement would be needed to conduct additional practical experiments. Using the residuality property, we introduced residual language equations which exactly correspond to the states of the r-AFA. That is, there is a bijection between the residual language equations of $\mathcal{L}$ and the states of the minimal *r*-AFA. Such models can be described naturally as a set of residual language equations that parallels the solutions of algebraic equations. Moreover, the solution of such systems of residual language equations is the class of regular languages. Furthermore, we exploit the succinctness relationship between *r*-AFA and DFA to develop a new active learning algorithm.

References

[1]. A. K. Chandra, D. C. Kozen, L. J. Stockmeyer, *Alternation*, *J. ACM*, Vol. 28, Issue 1, 1981, pp. 114-133.

[2]. A. Fellah, H. Jurgensen, S. Yu, Constructions for alternating finite automata, *Int. J. Comput. Math*., Vol. 35, Issue 1-4, 1990, pp. 117-132.

[3]. A. Fellah, Real-time languages, timed alternating automata, and timed temporal logics: Relationships and specifications, *J. Procedia Computer*, Vol. 62, 2015, pp. 47-54.

[4]. M. L. S. Berndt, M. Liskiewicz R. Reischuk, Learning residual alternating automata, in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 1749-1755.

[5]. L. J. Kavitha, G. Sethuraman, Descriptional complexity of alternating finite automata, in *Proceedings of the Int. Workshop on Descriptional Complexity of Formal Systems*, 2016, pp. 188-198.

[6]. S. Yu, Regular languages, in Handbook of Formal Languages (A. Salaomaa, Ed.), Vol. 1, *Springer-Verlag*, Berlin, 1997

[7]. A. Okhotin, Conjunctive grammars, Journal of Automata, Languages and Combinatorics, Vol. 6, Issue 4, 2001, pp. 519-535.

[8]. T. Aizikowitz, M. Kaminski, Linear conjunctive grammars and one-turn synchronized alternating pushdown automata, *Int. Journal of Foundations of Computer Science*, Vol. 6, Issue 25, 2014, pp. 781-802.

[9]. F. W. Vaandrager, Model learning, *Commun. ACM*, Vol. 60, Issue 2, 2017, pp. 86-95.

[10]. D. Angluin, Learning regular sets from queries and counterexamples, *Information and Computation*, Vol. 75, Issue 2, 1987, pp. 87-106.

[11]. B. Bollig, P. Habermehl, C. Kern, M. Leucker, Angluin-style learning of NFA, in *Proceedings of the Int. Joint Conference on Artificial Intelligence (IJCA'I9)*, 2009, pp. 1004-1009.

[12]. D. Angluin, S. Eisenstat, D. Fisman, Learning regular languages via alternating automata, in *Proceedings of the 24th Int. Joint Conference on Artificial Intelligence (IJCAI'15)*, 2015, pp. 3308-3314.

[13]. F. Denis, A. Lemya, A. Terlutte, Residual finite state automata, *Fundamental. Inform*., Vol. 51, Issue 01, 2002, pp. 339-368.

[14]. F. Denis, A. Lemay, A. Terlutte, Residual finite state automata, in *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'10)*, 2010, pp. 144-157.

[15]. J. Moerman, M. Sammartino, Residual nominal automata, in *Proceedings 31st International Conference on Concurrency Theory (CONCUR'20)*, 2020, pp. 44:144:21.

[16] J. A. Brzozowski, Derivatives of regular expressions, *J. ACM*, Vol. 11, Issue 4, 1964, pp. 481-494.

(025)

Simulation and Analysis of the Relative Electromagnetic Field Strength in a Portable Microwave Breast Cancer Detection Device

Debarati Nath [1] **and Stephen Pistorius** [2]
[1] Biomedical Engineering, University of Manitoba
[2] Physics and Astronomy, University of Manitoba
E-mails: nathd1@myumanitoba.ca, stephen.pistorius@umanitoba.ca

Summary: While early detection of breast cancer can reduce mortality, women in low- and middle-income countries often have limited access to breast cancer screening programs. Microwave-based techniques can provide portable, low-cost, and user-friendly systems that use machine learning to detect the presence of breast lesions. To optimize the design of a portable system, this study analyzed and quantified the effects of a point-like scatterer (PLS) on the electric fields in a microwave breast cancer screening device. Electromagnetic (EM) fields were simulated as a function of frequency and the PLS position inside the sensing chamber. An analytical approach was developed to describe the relative field strength agreed with simulated results to within -1.3 % ± 10 %. The fit returned an $r^2 = 0.92$ ($p < 0.01$) for PLS positions within ±6 cm in both x and y directions and frequencies from 2-8 GHz. Analyzing the EM field strength variations with the position of a PLS facilitates optimal antenna placement and will allow data to be generated for transfer learning of machine learning networks.

Keywords: Breast cancer, Microwave sensing, Electromagnetic (EM) simulation, Point-like scatterer (PLS), Electric field (E-field), Mathematical fitting.

1. Introduction

Breast cancer is the second most frequently diagnosed cancer in Canada [1], representing 25 % of all new cancer cases in Canadian women [2]. In Canada, the 5-year survival rate for breast cancer in females is 88 % [2], aided in part by established screening programs. Women living in remote communities and low- and middle-income countries (LMIC) often lack access to early detection, resulting in increased breast cancer mortality. The age-standardized mortality rate due to breast cancer is estimated to be between 18.0 and 22.3 per 100000 for Middle, Northern, and Western Africa, whereas Australia and New Zealand have a mortality rate of approximately 12.1 per 100000 [3]. In Canada, a 60 % mortality-to-incidence rate is found in rural areas of Manitoba, while urban Manitoba has a 37 % mortality-to-incidence rate [4]. Breast cancer detection systems such as X-ray Mammography, Magnetic Resonance Imaging (MRI), and Ultrasound are costly, time-consuming, lack sensitivity or specificity, and require highly trained personnel [2]. They are therefore not ideal for screening in regions with limited human and capital infrastructure.

Microwave-based breast cancer sensing is an approach that can overcome many of the drawbacks of X-ray Mammography, MRI, and Ultrasound [5]. It employs non-ionizing radiation, is potentially more sensitive and specific, lower in cost and compact, and does not require breast compression [6, 7]. However, most existing microwave breast imaging systems are not suited for rugged, remote locations, and hence, an inexpensive, self-contained and portable microwave breast cancer detection system that will increase accessibility, is needed for the remote areas.

A portable 2D simulated microwave breast cancer detection system that uses machine learning techniques was proposed [8]. The proposed system consisted of twelve solid-state sensors and a transmitting antenna. To optimize the design of the prototype device, a preliminary experimental and simulation study was conducted [9]. The prototype system has a transmitting antenna and thirteen microstrip patch antennas on a semi-circular sensor array. The responses of the receiver arrays to a point-like scatterer (PLS) as a function of microwave frequency and the receiver antenna positions were examined. In that study, the output voltages obtained from the antenna array were well correlated with the E-field magnitudes from the simulated system [9].

The current work investigated the effect of changing the location of a PLS on the electromagnetic (EM) field within the proposed portable microwave breast cancer detection device. Understanding the relationship of a PLS on the EM fields will inform and advance future designs and improve tumour detection using machine learning networks that take these factors into account. To optimize the system, the relative EM fields need to be estimated in a computationally efficient way. A mathematical model was derived to estimate the magnitude of the E-field at the receiver array when a PLS is present in the sensing chamber, relative to that without the PLS (open space).

2. Methods

2.1. Simulated Portable Microwave System

CST Studio Suite® [10] was used to calculate the EM fields as a function of frequency and the positions of the PLS inside the sensing chamber of the simulated system. The simulated portable microwave system was designed with a horn antenna as a transmitter. The simulation complexity was reduced by replacing the patch antennas described in [9] with 37 point-like receivers. The geometry of the simulated system is shown in Fig. 1. The distance between the transmitter antenna and the midpoint of the receiver array was 20 cm. The X-axis (y = 0) was 11 cm from the transmitter antenna, and the hemispherical 37-point receiver array was symmetric to the Y-axis (x = 0) at 9 cm from the central axis (y = 0). An Aluminum (Al) rod was used as the PLS and placed at various x and y positions in the sensing chamber. The intensity of the E-field was measured at the receiver array for every frequency and each Al rod's position.

The ratio of E-field intensity with the rod present to the open-space E-field intensity was evaluated for various rod positions and frequencies. The ratios of E-field intensities as a function of rod position averaged over all frequencies were also calculated.

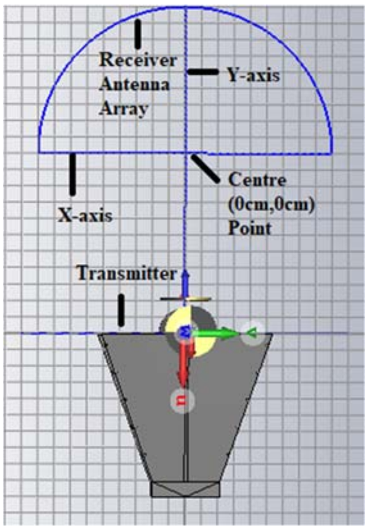


Fig. 1. The geometry of the simulated system illustrated the relative position of the transmitting and receiving antennas.

A mathematical equation was derived and fitted to the E-field intensity ratio as a function of four variables. These were - (i) Frequency from 2 GHz to 8 GHz; (ii) X-axis values for the Al rod position over a range of ±6 cm, (iii) Y-axis values for the Al rod position over ±6 cm, and (iv) Receiver positions over ±90 degrees.

2.2. Mathematical Modeling for a Point-like Scatterer (PLS)

Electromagnetic waves travel at the speed of light (c) in open space with the synchronized propagation of electric and magnetic fields in the $\hat{z}$ direction. The electric field (E) of the propagated electromagnetic wave in Vm^{-1} and the power per unit area (S) in Wm^{-2} can be written as (1) and (2) respectively, where f is the frequency, U is the amplitude of the propagated wave, t is the time, and ∅ is a phase shift.

$$E = U\cos\left(\frac{2\pi}{c}fz - 2\pi ft + \emptyset\right)\ (1)\ S = = \varepsilon_0 cE^2 = \varepsilon_0 c\left(U\cos\left(\frac{2\pi}{c}fz - 2\pi ft + \emptyset\right)\right)^2 \qquad (1)$$

Since the intensity (Wm^2) is the average power per unit area, it can be described by Eq. (3).

$$I = \langle S\rangle = \langle \varepsilon_0 cE^2\rangle = \frac{1}{2}\varepsilon_0 cU^2 \qquad (2)$$

Under a far-field approximation, the electric field (E$_r$) at the Al rod can be approximated by Eq. (3).

$$E_r = \frac{U}{4\pi u^2}\cos\left(\frac{2\pi}{c}fu - 2\pi ft + \emptyset\right), \qquad (3)$$

where the coordinates for the Al rod with respect to the centre axis are given by (x, y), where x and y are the X-axis, and Y-axis values of the Al rod position, l = 0.11 m is the distance from the transmitter to the centre point, and u is the distance from the transmitter antenna to the Al rod given by Eq. (4).

$$|u| = \sqrt{(l+y)^2 + x^2} \qquad (4)$$

The field intensity (I$_r$) at the Al rod is given by Eq. (5),

$$I_r = \frac{1}{32}\varepsilon_0 c\frac{U^2}{\pi^2 u^4} \qquad (5)$$

The electric field at the receiving antenna for the open space condition (E_{OS}), and in the presence of the Al rod (E_{rs}) can be approximated by Eq. (6) and (7), respectively.

$$E_{OS} = \frac{U}{4\pi v^2}\cos\left(\frac{2\pi}{c}fv - 2\pi ft + \emptyset\right), \qquad (6)$$

$$E_{rs} = \frac{\varepsilon_0 cU^2}{128\pi^3 u^4 w^2}\cos\left(\frac{2\pi}{c}fw - 2\pi ft + \varphi\right), \qquad (7)$$

where ∅ and φ are the phase shifts due to the path length directly from the transmitter antenna to the receiver, and from the transmitter to the receiver via the rod position, $|v| = \sqrt{(l + q\cos\theta)^2 + q\sin\theta^2}$ is the distance from the transmitter antenna to the receiver array, $|w| = \sqrt{(q\cos\theta - y)^2 + (q\sin\theta - x)^2}$ is the distance from the Al rod to the receiver array, q = 0.09 m is the distance from the centre (0,0) point to the midpoint of the receiver array, and θ is the sensor angle in radians.

The electric fields received by the receiver array of the simulated device in Eqs. (6) and (7) must be summed. The sum of the electric fields at the receiver array can be written as Eq. (8).

$$E = \frac{U}{4\pi v^2}\left(\begin{array}{c}\cos\left(\frac{2\pi}{c}fv - 2\pi ft + \emptyset\right) + \\ + \frac{\varepsilon_0 cUv^2}{32\pi^2 u^4 w^2}\cos\left(\frac{2\pi}{c}fw - 2\pi ft + \varphi\right)\end{array}\right) \tag{8}$$

The intensities of the electric fields at the receiver array in the presence of the Al rod (*I*) and for the open space condition (*I_{OS}*) are given by Eq. (9) and (10), respectively.

$$I = \frac{\varepsilon_0 cU^2}{16\pi^2 v^4}\left(\begin{array}{c}\frac{1}{2} + \frac{\varepsilon_0 cUv^2}{32\pi^2 u^4 w^2}\left(1 + \cos(\frac{2\pi}{c}fv - \frac{2\pi}{c}fw + \emptyset - \varphi)\right) + \\ + \frac{\varepsilon_0^2 c^2 U^2 u^4 w^2}{2048\pi^2 v^2}\end{array}\right) \tag{9}$$

$$I_{OS} = \frac{1}{32}\varepsilon_0 c\frac{U^2}{\pi^2 v^4} \tag{10}$$

The ratio (R) of electric field intensity (I) with the Al rod present at (x, y) to the intensity for open space conditions (I_{OS}) is given by Eq. (11).

$$\boldsymbol{R} = 1 + \left(\frac{C_1 v^2}{2u^4 w^2}\right)^2 + + \left(\frac{C_1 v^2}{u^4 w^2}\right) cos\left(\frac{2\pi f}{c}(v - w) + C_2\right), \tag{11}$$

where $C_1 = \frac{\varepsilon_0 cU}{16\pi^2}$ (m^4), and $C_2 = \emptyset - \varphi$ (radians).

Eq. (11) was used to extract four parameters, A, B, C, and D which have the following characteristics.

A is a unitless value that scales the Al rod's position to account for the experimental near-field conditions instead of the far-field assumptions used in the derivation of Eq. (11). F$_2$ is equivalent to C$_2$ of Eq. (11) divided by 2π and represents the phase shift (constrained to 0 to 2π) at the sensor due to the rod position. 2πB/c is the phase shift at the reference centre (x = 0, and y = 0) and theoretically should be equal to π/2. F$_1$ is equivalent to C$_1$ of Eq. (11) and is ideally a constant but was allowed to vary as a function of the rod position using an empirical relationship that is controlled by C (m^4) and D (m^2).

In Eq. (12), x' and y' are the X-axis and Y-axis values of the Al rod position in meters scaled by A, f is the frequency in GHz, θ is the sensor angle in radians, and l = 0.11 m and q = 0.09 m, describe the system geometry while c = 3×10^8 m is the speed of light. Here, u', $v', and\ w'$ represent u, v, and w of Eq. (11) scaled by A.

$$\boldsymbol{R} = 1 + \left(\frac{F_1.(v')^2}{2.(u')^4.(w')^2}\right)^2 + + \left(\frac{F_1.(v')^2}{2.(u')^4.(w')^2}\right)\cos\left(\frac{2\pi f}{c}(v' - w') + 2\pi F_2\right), \tag{12}$$

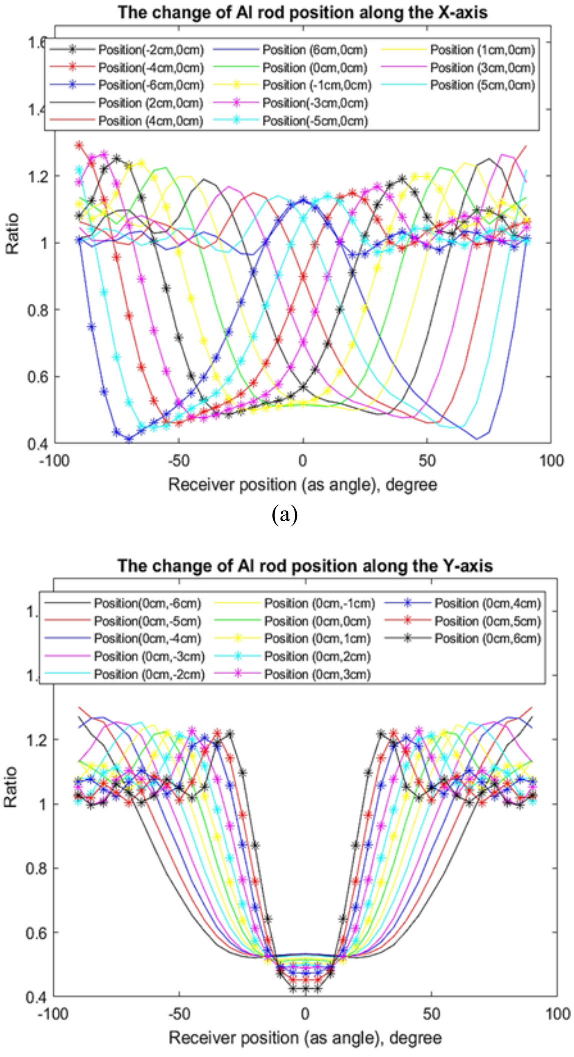
where $F_1 = C - D.((x')^2 + (y')^2)$ (m^4);
$v' = \sqrt{(q\sin\theta)^2 + (q\cos\theta + l)^2}$ (m);
$w' = \sqrt{(q\sin\theta - x')^2 + (q\cos\theta - y')^2}$ (m);
$F_2 = \left(1 - \frac{F_3}{c}\right) + \left|\left\lfloor\frac{F_3}{c}\right\rfloor\right|$ (unitless);
$F_3 = B - f.\left(\sqrt{((x')^2 + (l + y')^2)}\right) = 10^9 \times (B - fu')$ (ms^{-1}).

The fitting constants were obtained using a robust multi-dimensional fitting procedure [11].

3. Results and Discussion

3.1. Simulated Portable Microwave System

The ratios of the simulated E-field intensity averaged over frequency as a function of the Al rod position on the X- and Y-axis and receiver positions are displayed in Fig. 2. Fig. 2(a) illustrates that the peaks of the ratios shift in angle as the Al rod's position shifts along the X-axis, with Fig. 2(b) showing that the peaks move to smaller angles as the Al rod shifts in the positive direction (toward the receivers) on the Y- axis.



(a)

(b)

Fig. 2. The ratio of the E-field intensities (averaged over all frequencies) when the Al rod was present to the open space E-field intensities as a function of rod position on (a) the X-axis (y = 0) and (b) the Y-axis (x = 0).

3.2. Mathematical Modeling

The fitting returned an $r^2 = 0.92$ for Al rod positions situated within ±6 cm along the X-axis (y = 0) and Y-axis (x = 0). Eq. (12) agreed with simulated results to within -0.00343 (-1.3 %) ± 0.092 (± 10 %), the p-values for all parameters were less than 0.01, and F-value was 30183.28. Outliers were removed using Chauvenet's criterion. The residuals following the final fit were normally distributed and fell within ±3 sd.

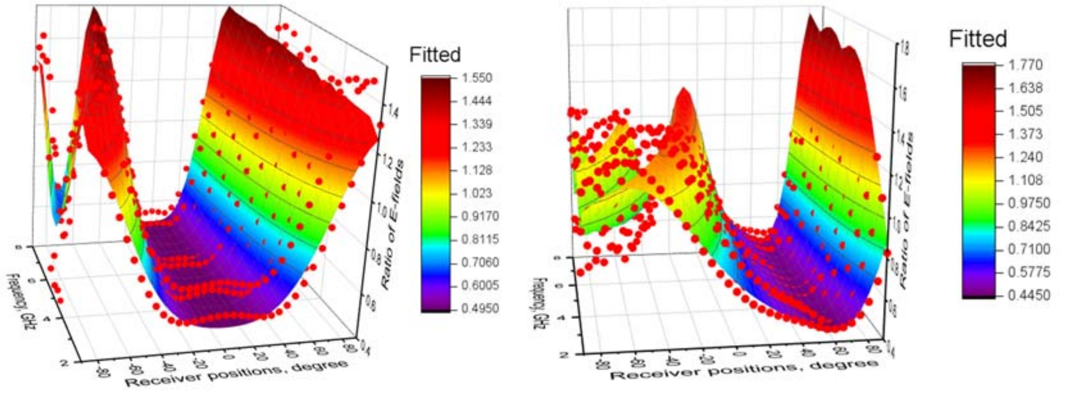
Fitting Eq. (12) to the data returned the following values:

$A = 0.871 \pm 0.002$ (unitless),
$B = 0.748 \pm 0.0002$ (ms^{-1}),
$C = 1.451 \times 10^{-03} \pm 6.99 \times 10^{-06}$ (m^4),
$D = 0.228 \pm 0.004$ (m^2).

The values of A, B, C, and D have small uncertainties, with A being consistent with the ideal value of l and B being close to the theoretical value of 0.75. While F_1 is ideally a constant = C, a term (controlled by D) was necessary to improve the agreement with the simulated data by reducing the ratio (primarily through the third term of Eq. (12)) as the rod moves away from x = 0 and y = 0.

Figs. 3 and 4 illustrate the simulated data obtained from CST Studio Suit® and the results of the analytical function Eq. (12) using the fitting parameters as a function of Al rod positions, frequencies, and receiver positions. In Fig. 3(a), the ratio of the E-field intensity is plotted as a function of frequency and receiver positions for rod positions of (0 cm, 0 cm) and in Fig. 3(b), for (4 cm, 0 cm).

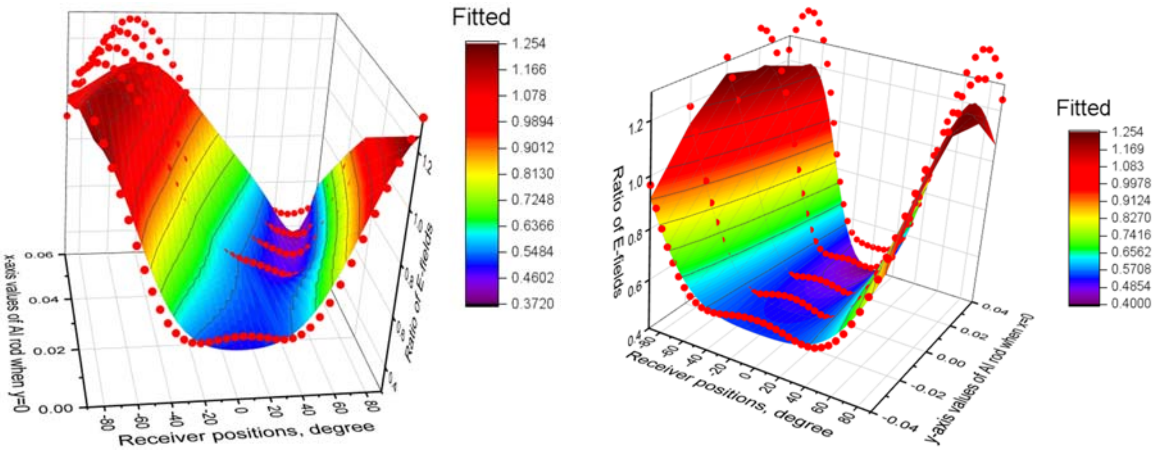
Fig. 4 (a) illustrates the ratio of the E-field intensity for a frequency of 2 GHz as a function of receiver positions and rod positions on the X-axis, while Fig. 4 (b) shows the results for rod positions on the Y-axis.



(a)

(b)

Fig. 3. The ratio of the E-field intensity obtained from the fit of Eq. (12) (surface) and simulated data (points) as a function of frequency and receiver position for an (a) Al rod position (0 cm, 0 cm) and (b) Al rod position (4 cm, 0 cm).



(a) Different positions of Al rod along the X-axis (y = 0) for frequency = 2 GHz.

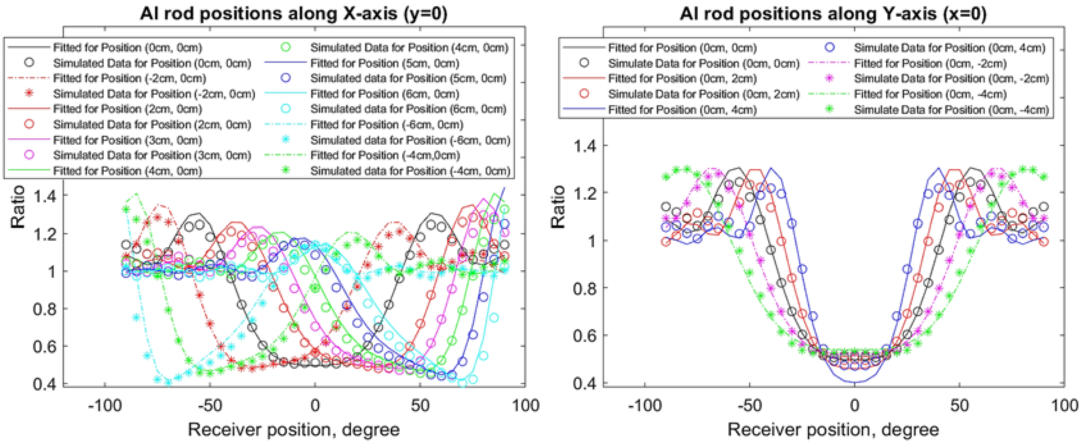
(b) Different positions of Al rod along the Y-axis (x = 0) for frequency = 2 GHz.

Fig. 4. For frequency = 2 GHz, fitted (surface) and simulated data (points)as a function of receiver position for (a) positions of Al rod along the X-axis and (b) positions of Al rod along the Y-axis.

The ratio of E-field intensities from the simulated data and those derived from Eq. (12) were averaged over all frequencies and are shown as a function of receiver position and rod position on the X-axis in Fig. 5(a) and along the Y-axis in Fig. 5(b). It can be observed that the ratio curves obtained from Eq. (12) display a similar pattern to the ratios obtained from the simulation. The angles at which the maxima and minima of the fitted and simulated ratio curves were similar, with the results from Eq. (12) agreeing with the simulated data to within ±5 degrees for all rod positions.

4. Conclusions

A prototype portable microwave-based breast cancer detection system is being developed to improve breast screening cancer access and reduce the mortality rate in remote regions and low-income countries. To optimize the system's design and improve the machine-learning-based detection, the system was simulated to explore the behaviors of the electromagnetic (EM) fields due to the presence of the point-like scatterer inside the sensing region. A mathematical equation was derived to estimate the ratio of the open electromagnetic (EM) field intensity to that with the inclusion of a point-like scatterer (PLS), and it was shown to agree well with the simulated results over a suitable range of PLS positions and frequencies. This equation can be used to optimize the antenna placement within the system and to rapidly generate datasets for future machine learning purposes.



(a) (b)

Fig. 5. The fitted and simulated ratios of the averaged E-field intensity as a function of receiver position for various positions of the Al rod on (a) the X-axis (y = 0) and (b) the Y-axis (x = 0).

References

[1]. Canadian Cancer Statistics 2019, Toronto, Canadian Cancer Society, cancer.ca/Canadian-Cancer-Statistics-2019-EN

[2]. L. Irwig, N. Houssami, C. van Vliet, New technologies in screening for breast cancer: A systematic review of their accuracy, *British Journal of Cancer*, Vol. 90, Issue 11, 2004, pp. 2118-2122.

[3]. Breast Cancer Facts & Figures 2019-2020. American Cancer Society, https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2019-2020

[4]. Department of Epidemiology and CancerCare Manitoba Cancer Registry, Cancer in Manitoba, 2012 Annual Statistical Report, https://www.cancercare.mb.ca/Research/epidemiology-cancer-registry

[5]. E. C. Fear, S. C. Hagness, P. M. Meaney, M. Okoniewski, M. A. Stuchly, Enhancing breast tumor detection with near-field imaging, *IEEE Microw. Mag.*, Vol. 3, Issue 1, Mar. 2002, pp. 48-56.

[6]. J. Bourqui, J. M. Sill, E. C. Fear, A prototype system for measuring microwave frequency reflections from the breast, *Int. J. Biomed. Imaging*, Vol. 2012, 2012, pp. 1-12.

[7]. N. K. Nikolova, Microwave imaging for breast cancer, *IEEE Microwave Magazine*, Vol. 12, Issue 7, 2011, pp. 78-94.

[8]. J. Sacristán, S. Pistorius, A comparison of classifiers for detecting tumours using microwave scattering in numerical breast models, *CMBES Proc.*, Vol. 40, Issue 1, May 2017.

[9]. M. M. Rana, D. Nath, S. Pistorius, Sensitivity analysis of a portable microwave breast cancer detection system, in *Proceedings of the 6th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'20)*, Aug. 2020, ICBES 112.

[10]. CST Studio Suite 3D EM Simulation and Analysis Software, https://www.3ds.com/products-services/simulia/products/cst-studio-suite/

[11]. TableCurve 3D, https://www.environmental-expert.com/software/tablecurve-3d-version-40-automoted-curve-fitting-analysis-software-598

(027)

Automatic Recognition of Epileptic Patterns in EEG via Neural Network

S. Chaibi
Faculty of Science of Monastir, Avenue of the Environment, 5019, Tunisia
Tel.: + 21655385405
E-mail: sahbi.chaibi@yahoo.fr

Summary: EEG signals recorded from human epileptic brains contain t*wo main abnormalities* well known clinically as epileptic spikes and epileptic High Frequency Oscillations (HFOs). The visual examination of these two kinds of epileptic patterns associated with long routine EEG recordings is a tedious task, very time consuming, requires a great deal of mental concentration and *experienced reviewers*. Therefore, to tackle this drawback, the automatic computer-based algorithms for the *detection* of epileptic patterns can be considered as an efficient, fast and reliable tool. In the present study, we have proposed and evaluated the performance of an artificial *neural network based machine learning model* dedicated to detect both spikes and HFOs.

The evaluated *sensitivity*, specificity and FDR related to spikes detection were respectively 89.53 %, 91.79 % and 08.33 %. However, the performance assessments for HFOs detection, were respectively 95.37 %, 90.70 % and 08.83 %. The proposed method may be considered helpful in the *localization* of epileptogenic zone.

Keywords: Epilepsy, EEG, Spike, HFO and neural network.

1. Introduction

Since its inception in the 1950s, Artificial Intelligence (AI) has been emerged as *mathematics and informatics disciplines* were originally intended to reproduce human intelligence. Its applications and services concern all human activities, allow in particular to improve the quality of life for so many unhealthy patients. The concept of Machine Learning in conjunction with the successful advances of signal and image processing theories reflected a significant evolution in biomedical engineering field, in particular in the diagnosis of several neurological diseases such as Alzheimer's, Parkinson's and Epilepsy [1-3]. Indeed, each neurological disease is characterized by its own abnormalities, which are manifested especially in electroencephalographic (EEG) signals. In particular, EEG recordings of epileptic patients contain t*wo main abnormalities* well known clinically as epileptic spikes and epileptic High Frequency Oscillations (HFOs). International Federation of Societies for Electroencephalography and Clinical Neurophysiology (IFSECN) defined a spike as a peak, clearly distinguished from the background activity, in which its duration is comprised between 20 ms and 70 ms [4-6]. However, an HFO is defined as a spontaneous rhythmic wave that persists for at least three or four periods or cycles, with established frequency ranged from 80 up to 500 Hz [7].

The visual marking or the identification of these epileptic bursts is a very difficult task. In addition, their examination require also a lot of time and mental focus. Moreover, the visual marking should be done with good inter-rater reliability. In this study, we have proposed a framework for joint spikes and HFOs detection. We have chosen as a machine learning technique the neural network. The structure of the paper parts is ordered as follows: Section 1 presents Introduction. Section 2 describes the clinical used database and visual marking of target events. In Section 3, the various details of the proposed method are provided. Section 4 presents the performance metrics. Section 5 presents the discussion and the different results. Last section presents conclusion and our outlines future work.

2. Database and Visual Marking

In order to test the functionalities and the execution of the *current proposal*, we used the same clinical database employed in [8]. Briefly, this database was recorded in the Montreal Neurological Institute and Hospital (MNI), Canada. A *Stellate Harmonie-Routine EEG system* with a sampling frequency of 2000 Hz was used for the iEEG data acquisition. In the present study, the variety of implemented techniques, algorithms and used tools are fully programmed with Python environment. Here, expert visual marking is serve as the benchmark or the gold standard. Both spikes and HFOs were visually and separately annotated by a one neurologist. All our results are considered as a ground truth for evaluating the performance measures of the proposed present method.

3. Detection of Epileptic Patterns by Artificial Neural Network (ANN)

Machine Learning (ML) is a popular supervised learning model that associate essentially two successive steps, training phase followed by testing phase. In our context of epileptic bursts detection, the objective of machine learning technique is to distinctly classify data into two binary classes: (HFO, background) as well as (spike, background). The neural network is chosen in our case as a ML technique for the automatic recognition of the events of interests (spikes and HFOs). A neural network is an approach, which its functioning is inspired from biological neurons in the human brain.

The neural network architecture is made up of three layers: an input layer, hidden layers and an output layer. Each layer is made up of several neurons. The optimal setting parameters of ANN structure were obtained based on relevant marked clinical HFOs and spikes, which have been considered as a ground truth. In our case, the main internal architecture of neural network is composed of a one input layer, an output layer and two hidden layers. The used activation function is the relu. In Fig. 1, a detailed flowchart of our proposal is shown. All theoretical aspects and details about the implementation of the different blocs used for automatic HFOs and spikes detection are described in the next sections.

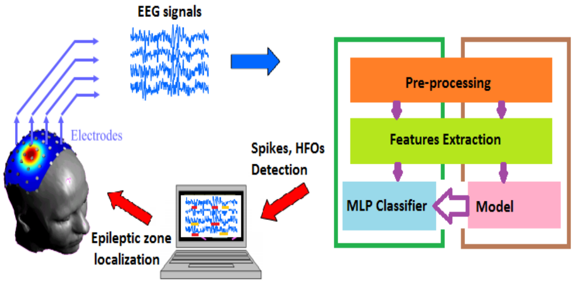


Fig. 1. The proposed methodology for the automatic detection of epileptic bursts based on Neural-Network.

3.1. Pre-processing Step

In the context of machine learning, it is important to do so many pre-processing steps like *denoising, equalization* matching and filtering. In our case, four domains are used for feature extraction that are mainly based on the following measurements: *time series* analysis, *frequency*-traces, time-frequency representation and Hilbert spaces. In our context, the band-pass filter was chosen to be between 80 and 500 Hz for HFOs detection and 30-70 Hz for spikes.

3.2. Features Extraction

A precise method of extracting informative characteristics is very important to extract the attributes that well describe the behavior and the traces of epileptic patterns in EEG signals.

The following measurements are computed for different labeled events (spikes, HFOs and background). We used 14 features that are: the mean of filtered signal, the std of filtered signal, the power of filtered signal, the mean of FFT spectrum, the peaks number in FFT space and their mean value, the Shannon entropy, Zeros-crossing rate, mean of *Hilbert instantaneous a*mplitude, mean of *Hilbert instantaneous frequency*, mean of *Hilbert instantaneous phase, Normalized energy* of time-frequency map, normalized number of pixels in time-frequency map different to zero, normalized number of local peaks or maxima in time frequency map. After that, a Recursive Feature Elimination (RFE) is applied in order to keep only the *most informative features* and *remove irrelevant* features. As a result, only the following features are retained for HFOs detection: the std of filtered signal, the power of filtered and the mean of *Hilbert instantaneous a*mplitude. For spikes detection, the retained features are respectively: the mean of filtered signal and the mean of *Hilbert instantaneous a*mplitude.

3.3. Training-testing Phases with Cross-validation

Cross-validation is one of the most popular techniques used for testing the effectiveness of a machine learning model. This is a procedure used to evaluate a model if the used database is limited. We used the k-fold cross-validation method with k is equal to 10. Our data partitioned into *10* equal sized subsets. Then, we train iteratively our learning model on all the subsets except a one of the 10 subsets. The last one is used for testing the model. So, in each iteration 90 % of the data are used for training and the 10 % remaining data are employed for testing phase.

3.4. Performance Measure

There are several metrics employed to assess the performance of the proposed approach. These metrics are calculated based on the following parameters set: TP, TN, FP and FN. TP (True Positives) counts the cases where the epileptic pattern is detected by the algorithm and by the expert. FP (False Positives) counts the cases where the background activity is marked visually by the expert and misclassified as an epileptic pattern by the algorithm. TN (True Negatives) counts the cases where the background activity is detected by the expert and the algorithm. FN (False Negatives) counts the cases where the epileptic pattern is marked visually by the expert and misclassified as a background activity by the algorithm.
The following metrics [9] are used to evaluate the performance of the proposed method:

$$Sensitivity = \frac{TP}{TP + FN} \tag{1}$$

The sensitivity evaluates how good the algorithm is correctly detecting spikes or HFOs.

$$Specificity = \frac{TN}{TN + FP} \tag{2}$$

The specificity provides information on the ability of the algorithm to detect negative backgrounds events.

$$FDR = \frac{FP}{FP + TP} \qquad (3)$$

It reflects the ability of the proposed approach to detect false epileptic patterns coming essentially from the filtering of *sharp waves and artifacts*.

4. Results and Discussions

A first example of spike detection by the proposed neural network based approach is provided in the Fig. 2.

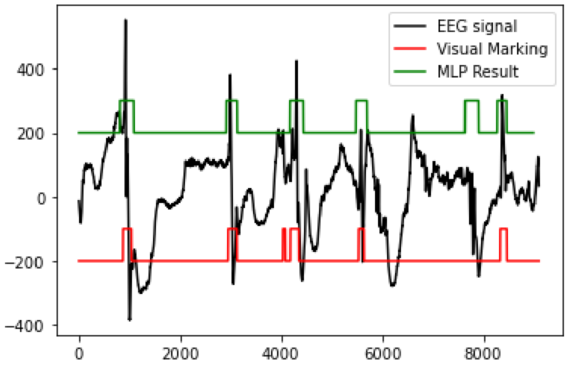


Fig. 2. An example of spikes detection by Neural Network approach.

A second example of HFOs detection by the proposed neural network based approach is provided in the Fig. 3.

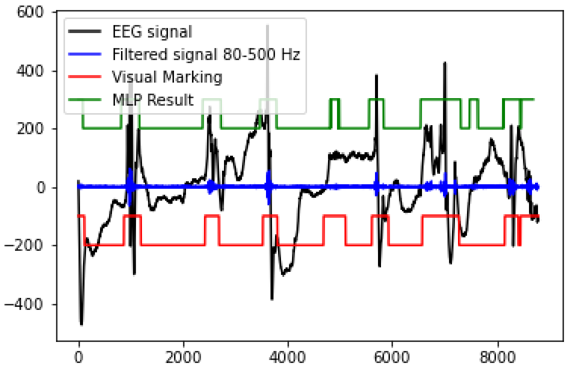


Fig. 3. An example of HFOs detection by Neural Network based technique.

After the execution of *different approaches* of neural network used for both spikes and HFOs detection, the obtained confusion matrix are provided in Table 1.

For the detection of spikes, we obtained the evaluated sensitivity, the specificity and the FDR as follows: 89.53 %, 91.79 % and 08.33 %. However, for HFOs detection, the performances are respectively 95.37 %, 90.70 % and 08.83 %. Overall, the proposed approach may be considered efficient and helpful for detecting both spikes and HFOs in EEG signals.

Table 1. Confusion matrix for both epileptic spikes and HFOs detection.

		Predicted Class			
		Spike	***HFO***	***Background***	
Real Class	***Spike***	TP_{Spike} =231		FN_{Spike} =27	
	HFO		TP_{HFO} =392		FN_{HFO} =19
	Background	FP_{Spike} =21		TN_{Spike} =235	
			FP_{HFO} =38		TN_{HFO} =371

5. Conclusions

The detection of epileptic patterns in EEG records by advanced methods such as machine learning become an efficient, fast and reliable tool in the diagnosis and treatment of epilepsy compared to the visual marking by a neurologist. Indeed, the visual marking of epileptic patterns associated with long EEG recordings is a tedious task and time consuming process. To tackle this drawback, the development of an automatic system based on artificial intelligence is necessary in order to accelerate the diagnosis of epilepsy. Our work is to propose and then evaluate the robustness of our ANN approach dedicated to build an automatic system of detection of epileptic patterns. The obtained results show that the ANN classifier is efficient and accurate to detect spikes and HFOs in EEG signals.

References

[1]. S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, D. Feng, L. Siqi, L. Sidong, C. Weidong, Early diagnosis of Alzheimer's disease with deep learning, in *Proceedings of the 11th International Symposium on Biomedical Imaging (ISBI'14)*, Beijing, China, 29 Apr. 2014 – 2 May 2014, pp. 1015-1018.

[2]. E. W. Abdulhay, N. Arunkumar K. Narasimhan, E. Vellaiappan, V. Venkatraman, Gait and tremor investigation using machine learning techniques for the diagnosis of Parkinson disease, *Future Gener. Comput. Syst.*, Vol. 83, 2018, pp. 366-373

[3]. A. Subasi, J. Kevric, M. A. Canbaz, Epileptic seizure detection using hybrid machine learning methods, *Neural Computing and Applications*, Vol. 31, Issue 1, 2019, pp. 317-325.

[4]. F. E. Abd El-Samie, T. N. Alotaiby, M. I. Khalid, S. A. Alshebeili, S. A. Aldosari, A review of EEG and MEG epileptic spike detection algorithms, *IEEE Access*, Vol. 6, 2018, pp. 60673-60688.

[5]. J W Puspita, G Soemarno, A I Jaya, E Soewono, Interictal Epileptiform Discharges (IEDs) classification in EEG data of epilepsy patients, *IOP Conference. Series: Journal of Physics*, Vol. 943, 2017, 012030.

[6]. S. Chaibi, C. Mahjoub, F. Krikid, A. Karfoul, R. Le Bouquin Jeannès, A. Kachouri, Pitfalls of spikes filtering for detecting high frequency oscillations (HFOs), in *Proceedings of the 18th International Multi-Conference on Systems, Signals & Devices (SSD'18)*, 2018.

[7]. S.-C. Wu, C.-W. Chou, C. Chen, S.-Y. Kwan, Y.-C. Su, Epileptic high-frequency oscillations: Detection and classification, *Multidimensional Systems and Signal Processing*, Vol. 31, 2020, pp. 965-988.

[8]. S. Chaibi, T. Lajnef, Z. Sakka, M. Samet, A. Kachouri, A comparison of methods for detection of high frequency oscillations (HFOs) in human intacerberal EEG recordings, *American Journal of Signal Processing*, Vol. 3, Issue 2, 2013, pp. 25-34.

[9]. R. Chander, Algorithms to detect high frequency oscillations in human intracerebral EEG, PhD Thesis, Department of Biomedical Engineering, *McGill University Montreal*, Canada, 2007.

(030)

Deep Learning Classification of EEG Signals from Alcoholics and Non-alcoholics in a Language Recognition Task

Victor Borghi Gimenez [2], Suelen Lorenzato dos Reis [1] and Fábio Marques Simões de Souza [1, 2]
[1] Federal University of ABC, Neuroscience, Center for Mathematics, Computing and Cognition, Alameda da Universidade s/n - Bairro Anchieta, CEP: 09606-045, São Bernardo do Campo - São Paulo, Brazil
[2] Federal University of ABC, Computer Science, Center for Mathematics, Computing and Cognition, Avenida dos Estados, 5001- Bairro Bangú, CEP: 09210-580, Santo André - São Paulo, Brazil
E-mail: victor.gimenez@ufabc.edu.br, suelen.lorenzato@ufabc.edu.br, fabio.souza@ufabc.edu.br

Summary: This work consists of the development of a MLPNN (Multi-Layer Perceptron Neural Network) for the classification of an original dataset from alcoholics and non-alcoholics during a task of verb recognition. We used either 300 (dataset with transformation) or 3072 (dataset without transformation) neurons in the input layer, two hidden layers with ReLU activation function and an output layer with 1 unit and two output labels: alcoholic (1) and control (0). The dataset of electroencephalogram (EEG) signals was divided in epochs of event-related potentials (ERPs) in response to verb stimulus. We performed Fast Fourier transforms (FFTs) and created two datasets being one with FFT and another one without any transformation. Initially, the MLPNN was able to successfully classify the dataset with 84 % of accuracy.

Keywords: Deep learning, Neural network, Neuroimaging data, Signal processing, Alcoholism automated diagnosis.

1. Introduction

Alcoholism is a psychiatric disorder responsible for a large number of addicted subjects around the globe. Actually, there are several neuroinformatics studies working with alcoholic x non-alcoholic related problems, such as [1] and [2].

The present work is related to a previous study [3] using fNIRS (Functional Near-Infrared Spectroscopy) neuroimaging technique to collect mirror neuron data from volunteers in a language recognition task, and the data was processed and classified through a SLPNN (Single-Layer Perceptron Neural Network) using four output neurons for labeling different verbs.

2. Materials and Methods

We used two laptops for performing the stimuli presentation and EEG recording respectively. We used an Easy-cap (Brain Products) with 16 scalp electrodes connected to the amplifier V-Amp (Brain Products®) for the EEG recordings. One laptop was dedicated to run the software generator of visual and auditory stimuli that sends a synchronizing trigger to another notebook connected to V-AMP and running the EEG acquisition software Open Vibe. Both notebooks run under the battery to reduce AC line sinusoidal oscillations in the EEG signal. For training the artificial neural network, we used a workstation PowerEdge T640 (Dell), and we used a Sony VAIO SVE14A15FBB for the programming of the Brazilian Portuguese stimuli related subroutines to be included in the stimulation software, as well as modifying the MLPNN architecture source code.

2.1. Participants

6 non-alcoholic and 9 alcoholic Brazilian subjects were recruited to participate on this project. Each participant was recorded twice in the same task. Subjects were divided into alcoholic and non-alcoholic groups through their answers from AUDIT [6] and BIG-5 [7] tests.

2.2. Multi-Layer Perceptron Neural Network

We defined four scenarios in our experiments, for the two first scenarios were used a default setup with 150 epochs, batch size of 32, validation split of 30 % (0.3), dropout of 50 % (0.5), learning rate of 1×10^{-3}, momentum of 80 % (0.8), SGD (Stochastic Gradient Descendent), binary cross entropy as cost function, two hidden layers being the first one containing 100 neurons with ReLU activation function and an output layer containing one neuron with sigmoid activation function. From the third scenario the number of hidden layers raised to three, being only changed the numbers of the neurons as 1000 instead of 100 at the first one and was added a hidden layer using 64 units with two regularizers: L1 and L2 to avoid the data overfitting in the validation data. The MLPNN was developed using Keras with TensorFlow frameworks in Python [11]. It was used 300 and 3072 input neurons X respectively for FFT and raw dataset (non-FFT) and two output neurons Y considered as Alcoholic (1) and Control (Non-Alcoholic) (0), our dataset contains 25040 ERP's (samples).

2.3. Software Generator for Visual and Auditory Stimuli

We used a Psychtoolbox framework based innominated software in MATLAB/Octave developed by Researchers from Shahid Beheshti University in Tehran, Iran with the purpose of displaying 33 verbs divided in 3 blocks being 11 verbs per block to the subject and obtaining information about response times in seconds. The software at the front-end induces the subject to react to the spoken or graphical language with the aim to test the language comprehension and the visualization of the appeared stimuli from each verb. In addition, the task helps to activate different brain areas that can be detected by the EEG electrodes to collect the neural signals without the need to execute any movement and guarantee the acquisition of VEP's (Visually Evoked Potentials) and AEP's (Auditory Evoked Potentials).

The back-end part recorded the order of verbs, list of the stimuli that were displayed and display time from each stimulus among other parameters. The retrieved data combined with the subject's answers from AUDIT [6] and BIG-5 [7] tests were used in statistic analysis to discover correlations between their answers x behaviors.

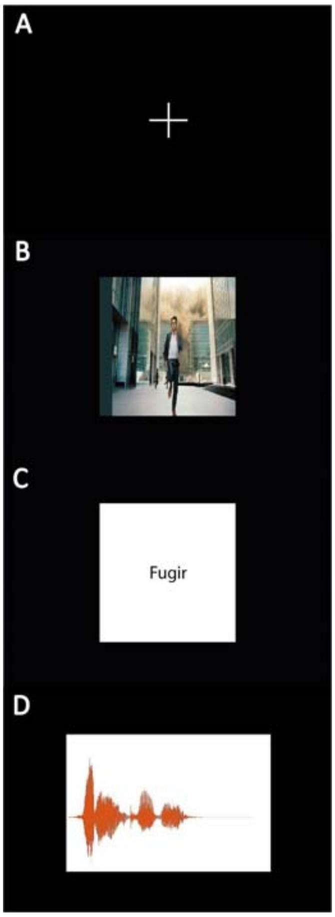


Fig. 1. Scheme from the four kinds of stimuli: Rest image (A) that is displayed before the appearance of each stimuli, including the visual descriptive stimulus (B), the visual written stimulus (C), the auditory stimulus (D) and the classification question.

2.3.1. Brazilian Portuguese Speech and Image Generation

Initially, the original verbs contained in the software loading folder were in Farsi language. Displayed verbs are divided in 3 different groups: Alcoholic, control and emotional of 11 verbs each.

Table 1. Verbs used in the experiment and their correspondent translations.

#	Class	Brazilian Portuguese	English
1	Alcoholic	Entornar	Pour
2	Alcoholic	Cochilar	Nap
3	Alcoholic	Cambalear	Stagger
4	Alcoholic	Emborcar	Quaff
5	Alcoholic	Esvaziar	Empty the wine glass
6	Alcoholic	Encher	Fill the wine glass
7	Control	Engolir	Swallow
8	Control	Brindar	Toast
9	Control	Comprar	Buy
10	Control	Procurar	Seek
11	Emotional	Gritar	Shout
12	Control	Cair	Fall
13	Emotional	Humilhar	Humiliate
14	Emotional	Arrepender	Regret
15	Emotional	Chorar	Cry
16	Emocional	Sorrir	Smile
17	Controle	Relaxar	Relax
18	Controle	Esquecer	Forget
19	Controle	Lembrar	Remember
20	Emocional	Cuspir	Spit
21	Emocional	Machucar	Hurt
22	Emotional	Fugir	Flee
23	Alcoholic	Parar	Stop drinking
24	Control	Correr	Run
25	Emotional	Sofrer	Suffer
26	Alcoholic	Pegar	Pick up
27	Emotional	Quebrar	Break
28	Alcoholic	Festejar	Party with drinking
29	Alcoholic	Beber	Drink
30	Alcoholic	Exceder	Exceed
31	Emotional	Chutar	Shoot
32	Control	Olhar	Look
33	Control	Falhar	Fail/Not succeed

We developed a Brazilian version to meet the needs of the local researches. The auditory verb files were generated with the NaturalReaders website (https://www.naturalreaders.com/online/), and the recording was done through the Ubuntu sound recorder Audiorecorder. The original Farsi audio files have lengths of 47.232, 91.008, one file with 132.784, two channels (stereo) and frequency of 44100 Hz. Initially, it was purposed to set the same amplitude peak normalization equal the division between the highest absolute value of the Portuguese generated signal x'_n, and the highest absolute value of the original Persian audio signal x_n and the result of them is multiplied by the entire Portuguese generated signal, the developed version below is a adapted version from [12] and [13]:

$$(s_n) = \frac{max\{|\{(x_n)\}|\}}{max\{|\{(x'_n)\}|\}}\{(x'_n)\} \qquad (1)$$

The original audio signals were analyzed and it was noticed that the speech can happen at the beginning of the sound (sound wave with only one part: speech + silence), middle of the sound (three parts: silence +

speech + silence) or only at the final of the auditory signal (two parts: silence + speech), it was purposed a subroutine that reads the original Persian audio signal x_n and returns the number of halves available (border between the cell in the two channels equal to 0 and cell in the two channels not equal 0 or vice-versa), i is the position where the last cells are equal 0 before the next cells not equal 0 and j is the position where the last cells are not equal 0 before the next cells that are equal 0. There is a silent period in the way the generated audio was recorded through the computer. Thus, it was necessary to exclude the silent period ranges in the audio signal before and after the period that contains sound without silence in the two channels, the input parameter and output parameter of this subroutine is the s_n signal obtained in equation (1). The last subroutine at the Equation (2) below creates the Brazilian Portuguese audio signals with the speech starting in the same period of time of the original Persian audio signals x_n, same halves and same length between *newArr* and x_n, N is the length of any an audio signal and i was the obtained value previously:

$$\begin{cases} newArr \leftarrow \\ \left[(s_n)_{2\times N_{s_n}} \quad 0_{2\times(N_{x_n}-N_{s_n})} \right] & \text{if halves} = 1 \\ newArr \leftarrow \\ \left[(s_n)_{2\times 2\times N_{s_n}} \right] & \text{else if halves} = 2 \\ newArr \leftarrow \\ \left[0_{2\times(1\leq N\leq i-1)} \quad (s_n)_{2\times N} \quad 0_{2\times(N_{x_n}-(1\leq N\leq i-1)+N_{s_n})} \right] & \text{else} \end{cases} \tag{2}$$

After that, the new audio signal obtained in newArr is written in the computer with the same frequency of the original Farsi audio signal. The image files were created using the GIMP (GNU Image Manipulation Program) software using a white background as can be seen in the Fig. 2 item D, The parameters used were height x width of 720 pixels × 960 pixels, Calibri font with size 16 in black color and each one of the inserted words were centralized in the image document.

2.4. EEG Data Acquisition

All EEG data was registered two times using the 10-20 cap pattern with 16 electrodes distributed through the scalp in each subject. The EEG signals were recorded using the Psychtoolbox framework stimulation software (PFSS) and the data was captured with OpenViBE [8]. The EEG signal was synchronized with the stimuli triggers generated by the PFSS to isolate the ERP's. The data analysis was performed using EEGlab [9] in MATLAB/Octave. The data was pre-processed through a band pass filter with frequencies ranging between 1 to 50 Hz for noise removal. The ERP signals were separated by image, sound and text stimuli. The trials with eye movement artifacts, head movement and chewy noise were excluded. We performed spectral analysis and ERSP (Event-Related Spectral Perturbations) [10], and parametric comparisons were used between alcoholic and
non-alcoholic conditions. P-value < 0.01 was considered significantly different.

2.4.1. Data Analysis and Transformation

We generated two datasets being one using FFT transformation with a total of 25.040 lines which represents the amount of EEG registers from ERP (Event Related Potentials) to different kinds of stimuli by 300 columns which represents the number of component points from the FFT frequency, and a raw dataset with the same number of rows as the previous one by 3072 columns representing the number of component points in the time domain. For the raw dataset all of his values are in a range between [-483.36 347.01].

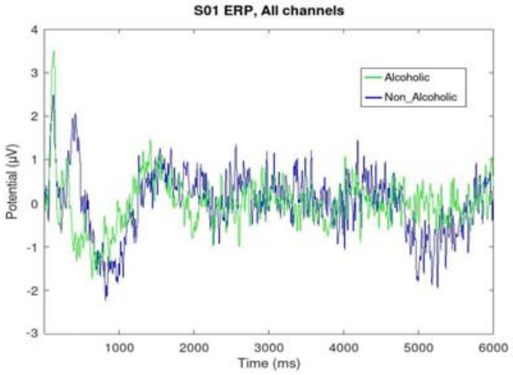


Fig. 2. Averaged ERP of the raw dataset in the time domain (without application of the FFT algorithm) of the alcoholic and non-alcoholic epochs.

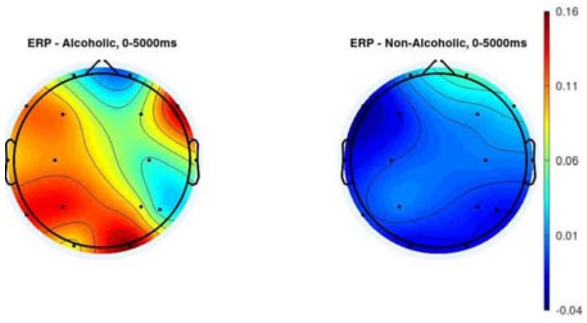


Fig. 3. Topographic averaged ERP signal of the raw dataset in the time domain of the alcoholic (left panel) and non-alcoholic (right panel) epochs.

Using the same concept of the DFT (Discrete Fourier Transform) [14] it was applied in the raw dataset a FFT (Fast Fourier Transform) where his time complexity is equal $O(Nlog_2N)$ which means that FFT makes faster computations than DFT $O(N^2)$ and his purpose consists in removing the negative values, at the FFT dataset the range of the values are between [0 50.586] and negative values were removed from the dataset

$$S(f) = \sum_{t=0}^{N=1} x(t)e^{-2\pi ift/N} \tag{3}$$

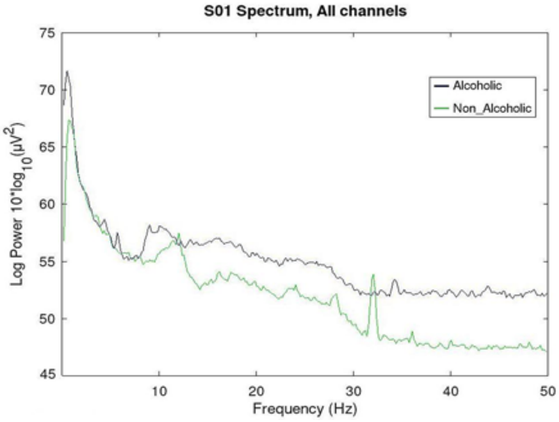


Fig. 4. Averaged frequency domain spectrum of the ERP of the alcoholic and non-alcoholic datasets submitted to the FFT transformation with a frequency between 0 and 50 Hz.

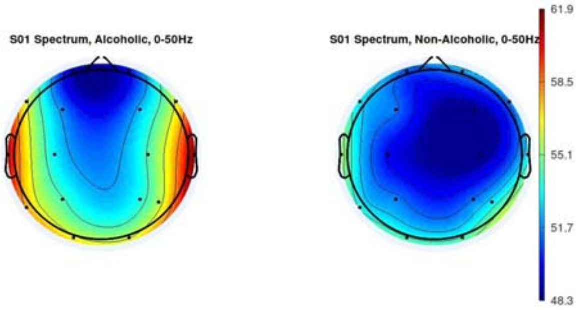

S01 Spectrum, Alcoholic, 0-50Hz
S01 Spectrum, Non-Alcoholic, 0-50Hz
61.9
58.5
55.1
51.7
48.3

Fig. 5. Topographic averaged frequency spectrum of the alcoholic (left panel) and non-alcoholic (right panel) epochs submitted to the FFT transformation.

Two plus datasets were generated to the raw and FFT ones being one with Min-Max normalization algorithm where his purpose consists in normalizing each value, letting all of his values in a range between [0 1]:

$$norm = \frac{X - min(X)}{max(X) - min(X)} \tag{4}$$

And another dataset without normalization was generated for the datasets with FFT and without FFT.

3. Results

3.1. Training and Test Data (67 %-33 %)

For the first scenario we decided to use the default settings already used in the example. The execution times for the algorithm with raw and FFT data were respectively equal 131 seconds and 114 seconds, the overall accuracies and losses for the raw dataset kept stagnant with a fast increase until the epoch 50 while the accuracies for the FFT dataset grew slowly from the epoch 30 and unexpectedly the FFT dataset got a lower accuracy than the raw dataset one.

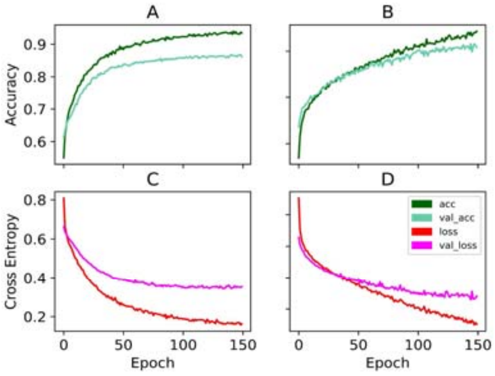


Fig. 6. Test accuracy over the epochs: (A) Dataset without application of any transformation algorithm, (B) Dataset with application of FFT. Test loss over the epochs: (C) Dataset without application of any transformation algorithm, (D) Dataset with application of FFT.

Table 2. Accuracy and Loss Total Averages for Training and Test Data Scale of (67 %-33 %).

	Test Acc Avg	Test Loss Avg
Non-FFT	0.88	0.323
FFT	0.818	0.416

3.2. Min-max Normalized Dataset With Same Previous Training-test Scale Set

With the application of the Min-Max normalization function and using the same parameters that were used at the previous scenario. The execution times in seconds were 138 and 125 seconds to the raw and FFT datasets respectively.

For the raw dataset his accuracies as long as the epochs were passed by began with a low percentage in relation to the first setup and until the last epoch tends to keep his accuracies stagnated with lots of ups and downs.

The dataset created with FFT transformation tends to have an accuracy similar to the previous execution but his training accuracy stays higher than the training one until the 60 epoch, so, his accuracy grew higher than in his previous scenario.

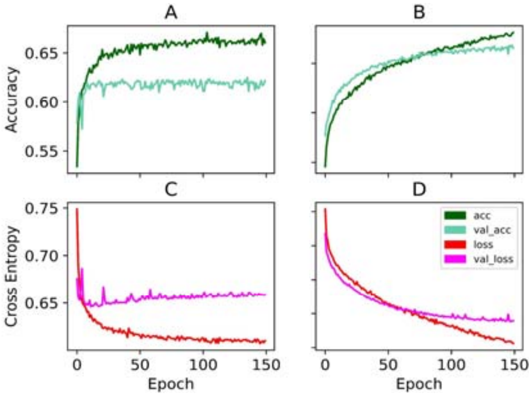


Fig. 7. Test accuracy over the epochs: (A) Dataset without application of any transformation algorithm, (B) Dataset with application of FFT. Test loss over the epochs: (C) Dataset without application of any transformation algorithm, (D) Dataset with application of FFT.

Table 3. Accuracy and Loss Total Average with the Min-Max Function.

	Test Acc Avg	Test Loss Avg
Non-FFT	0.627	0.65
FFT	0.831	0.383

3.3. Training – Validation – Test Data (96 %-2 %-2 %) using L1 and L2 Regularizers

The execution time of the raw dataset took almost 7 minutes being it the time for the training and the whole execution time was equal 2543 seconds where each epoch lasted 14 seconds approximately, by his turn, the FFT dataset execution time took 2 minutes for execute the training of the dataset and the full execution time of 1675 seconds what is a result very meaningful to this scenario, at the same time it was got a accuracy mean close to the 90 % using the same dataset without the need of modify or include more content, here each epoch lasted between 11 and 8 seconds. Different from the previous scenarios, the accuracies and losses obtained in the raw dataset grew and decreased a little, maintaining the validation accuracy and validation loss with stagnated values and eventually ups and downs.

Table 4. Accuracy and Loss Total Averages for Training and Test Data Scale of (67 %-33 %).

	Test Acc Avg	Test Loss Avg
Non-FFT	0.591	0.729
FFT	0.9	0.378

3.4. Applying Dataset Percentage Decrease

We performed a dataset size reduction to estimate the influence of a percentage reduction in the accuracy of the classification. We verified that the average test accuracy has an approximate quadratic increase with the size of the dataset (Figs. 8 and 9).

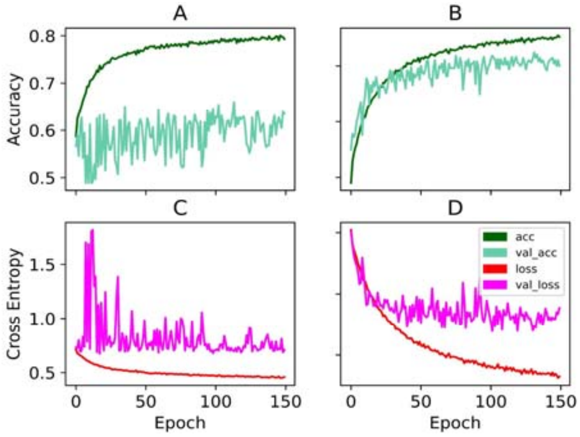


Fig. 8. Test accuracy over the epochs: (A) Dataset without application of any transformation algorithm, (B) Dataset with application of FFT. Test loss over the epochs: (C) Dataset without application of any transformation algorithm, (D) Dataset with application of FFT.

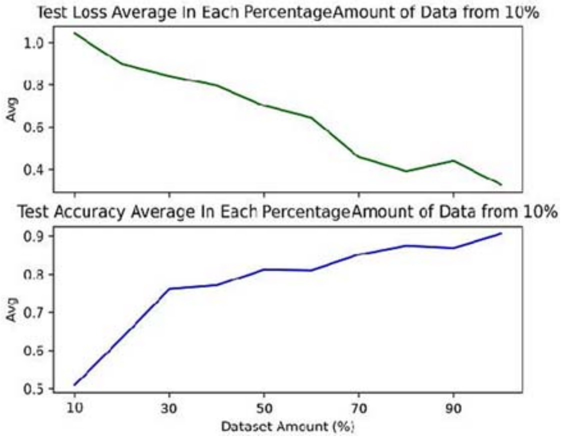


Fig. 9. Average variation obtained when the entire FFT dataset is submitted to a percentage reduction.

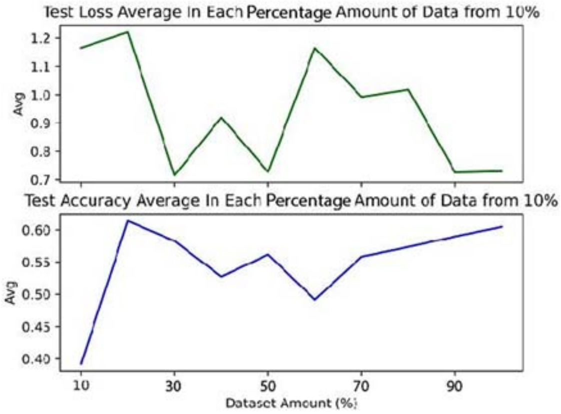


Fig. 10. Average variation obtained when the entire raw dataset (Non-FFT) is submitted to a percentage reduction.

4. Discussion

During the data analysis and artifact removal it was hard to remove some kinds of artifacts like chewing.

It was evidenced that the use of L1 and L2 regularizers associated with the use of more training samples increased the test/validation accuracy of the data to a relevant value next 95 % and drastically reduced the validation/test lost amounts, almost reaching zero. On the other hand, the test data accuracy for a raw dataset (Non-FFT) in the first scenario had an accuracy higher than the FFT dataset test data accuracy without the use of regularizers, normalization or any other improvement.

The use of Min-Max normalization had an imperceptible improvement in the second scenario on the overall accuracy in both datasets and the overall lost was small, also, it was executed a classification in the first scenario without use of normalization and with training-test scale: 98 %-2 % and it was obtained an accuracy of 91 % on the raw dataset (Non-FFT) that was bigger than the FFT dataset.

There was a significant impact at the FFT dataset while the raw dataset kept oscillating without any constant progression in accuracy when we reduced the percentage of the two datasets.

5. Conclusions

The main idea was to establish an initial study in order to verify how accurate a MLPNN architecture using a framework could classify a relatively small dataset with 25040 samples obtained from 30 EEG recordings in a total of 15 subjects being a total of ERP's (samples) equal to 12480 control and 12560 alcoholic, with the results it was not necessary to establish an artificial expansion of the dataset. Larger datasets with more training samples would allow the neural network to achieve higher accuracies as indicated by study varying the size of the dataset. More extensive studies will allow using 2D topographic images acquired from the same EEG signals and using two neural networks: CNN (Convolutional Neural Network) architecture to classify the 2D topographic images and a RNN (Recurrent Neural Network) to analyze deeply the dynamic behavior, as well as developing applications for diagnosis and prognosis for specific alcohol use disorders. We concluded that the FFT dataset had a better efficacy in the training of the MLPNN than the non-FFT dataset when used L1 and L2 regularizers, Min-Max normalization and more training percentual amount than validation + test data, on the other hand, the MLPNN had a better efficacy than non-FFT dataset without the use of additional techniques.

Funding Statement

This work was funded by Fapesp grant #18/06504-4. Victor Gimenez and Suelen Dos Reis were funded respectively by Fapesp fellowships #2019/04104-1 and #2020/02273-8.

Acknowledgements

The authors acknowledge the Science Valley Research Institute (https://svriglobal.com) for providing the infrastructure and technical assistance for this work. Fabio Simoes de Souza acknowledges Cristiane Salum and Rodrigo San Martin for the general support and advices regarding the EEG equipment. Fabio Simoes de Souza acknowledges Dr. Najafi, Dr. Bigdeli, Dr. Khosrowabadi and Dr. Varandi for helping with the study design and for sharing the Psychtoolbox code.

References

[1]. U. R. Acharya, et al., Automated diagnosis of normal and alcoholic EEG signals, *International Journal of Neural Systems*, Vol. 22, Issue 3, 2012, 1250011.

[2]. J. C. Rodrigues, et al., Classification of EEG signals to detect alcoholism using machine learning techniques, *Pattern Recognition Letters*, Vol. 125, 2019, pp. 140-149.

[3]. V. B. Gimenez, DdS. Souza, T. Carthery, F. M. Simões de Souza, Deep learning analysis of hemodynamic mirror system responses during the semantic recognition of verbs and actions, in *Proceedings of the Bernstein Conference*, 2020.

[4]. L. Deng, The MNIST database of handwritten digit images for machine learning research, *IEEE Signal Processing Magazine*, Vol. 29, Issue 6, 2012, pp. 141-142.

[5]. M. A. Nielsen, Neural Networks and Deep Learning, Vol. 25, *Determination Press*, San Francisco, CA, 2015.

[6]. D. F. Reinert, J. P. Allen, The Alcohol Use Disorders Identification Test (AUDIT): A review of recent research, *Alcohol. Clin. Exp. Res.*, Vol. 26, Issue 2, Feb. 2002, pp. 272-279.

[7]. B. Rammstedt, O. P. John, Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German, *Journal of Research in Personality*, Vol. 41, Issue 1, Feb. 2007, pp. 203-212.

[8]. Y. Renard, et al., OpenViBE: An open-source software platform to design, test and use brain-computer interfaces in real and virtual environments, in Presence: Teleoperators and Virtual Environments, *Massachusetts Institute of Technology Press (MIT Press),* Vol. 19, 2010, pp. 35-53.

[9]. A. Delorme, S. Makeig, EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis, *J. Neurosci. Methods*, Vol. 134, Issue 1, Mar. 2004, pp. 9-21.

[10]. S. Makeig, Auditory event-related dynamics of the EEG spectrum and effects of exposure to tones, *Electroencephalography and Clinical Neurophysiology*, Vol. 86, Issue 4, Apr. 1993, pp. 283-293.

[11]. TensorFlow Developers, TensorFlow, *Zenodo*, 2021.

[12]. E. Tarr, Peak Normalization, https://www.hackaudio.com/digital-signal-processing/amplitude/peak-normalization/

[13]. E. Tarr, Hack Audio: An Introduction to Computer Programming and Digital Signal Processing in MATLAB, *Routledge*, 2018.

[14]. A. Delorme, Time-frequency analysis of biophysical time series, in *Proceedings of the EEGLAB Workshop*, Aspet, France, 2017.

9 788409 358656