Journal of Theoretical and Applied Information Technology
15th November 2020. Vol.98. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN OPTIMIZED CONVOLUTIONAL NEURAL NETWORK FOR HANDWRITTEN DIGITAL RECOGNITION CLASSIFICATION

WALID SALAMEH[1], OLA SURAKHI[2]

[1]Princess Sumaya University for Technology, King Hussein School of Computing Sciences, Computer Science Department, Amman-Jordan

[2]University of Jordan, King Abdullah II School for Information Technology, Computer Science Department, Amman -Jordan

E-mail: [1]walid@psut.edu.jo, [2]ola.surakhi@gmail.com

ABSTRACT

A Convolutional Neural Network (CNN) is a deep learning algorithm that designed to demonstrate a high capability of object recognition in image data. This paper has developed a prediction model for handwritten classification problem by using CNN. A better performance has proved noticed where this performance has demonstrated over three main metrics: learning rate, number of iterations and number of hidden layers (model path). The paper suggests a best configuration based on the best reached near accuracy generated from different values for these metrics. The accuracy achieved by the suggested model shows the average of accuracy generated by each fold of 5-fold cross validation where configurations are selected randomly. The experimental results achieved a high score of accuracy with one hidden layer and ten number of epochs. A near state of the performance on MINST handwritten digit recognition task in Python using Keras deep learning library was developed and discovered.

Keywords: *Classification Problem, Convolutional Neural Networks, Deep Learning, Optimization*

1. INTRODUCTION

Computer vision is a huge topic that include different problems such as image classification, object detection and localization. The process of classification the image includes the classification of each pixel into a specific group based on a certain set of rules. In general, the classification can be supervised or non-supervised. Depending on whether there is a prior knowledge about the classes. The supervised training includes teaching the classifier to define the object by training it with dataset. Many techniques in the machine learning field are developed in the literature to achieve that and learn from examples such as Support Vector Machine [1] and K-Nearest Neighbor [2].

For larger dataset for a bigger domain, the features become more complicated. The deep learning is the latest method that finds a complex formation in a large dataset to extract information and assigns objects into different labelled classes from the problem domain [3]. Among deep learning methods and algorithms, convolutional neural network showed an excellent achievement in the classification problem of computer vision domain [4].

Convolutional neural network (CNN) is a deep learning algorithm that is designed to operate a two-dimensional image data. CNN has different architectures. The main two elements of it are convolutional and pooling layers. Although it appears to be simple, but there could be an infinite way to arrange these elements. In order to obtain the best performance for CNN model, (including its architecture and parameters), the hyperparameters tuning is the main process that influence on the prediction performance [5]. The tradition way to do that is manually by testing the performance of the network when changing the value of one parameter while keeping the other fixed. This way is computationally expensive especially when the dataset is large and the available resources are limited.

The success of using machine learning methods for any prediction task depends on finding the best architecture and tuning its hyperparameters in a way that fit the given problem and produce an accurate result.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

How to design the optimal model architecture and tune its parameters to best make the prediction is the challenge part of using any deep learning model [6].

The main contribution of this paper is to propose a deep learning prediction model for handwritten classification problem and by using the convolutional neural networks. The model performance is enhanced by testing its performance over three main metrics: learning rate, number of epochs and number of hidden layers (model depth). The best configuration was selected based on the best accuracy generated from the different values of these metrics.

The rest of this paper is organized as follows: Section 2 describes the methodology followed to build the prediction model. Section 3 gives the experimental results and section 4 presents the conclusions.

1.1 Dataset

The MNIST (Modified National Institute of Standards and Technology) dataset is a standard dataset which was developed for computer vision and deep learning optimization problems [7, 8]. It can be used for as the basis for learning and developing a prediction model using convolutional deep neural network for the image classification problem in order to estimate the model and enhance its performance [9, 10]. The dataset consists of 70,000 black and white small square 28x28 pixel images of handwritten single digit between 0 and 9. The dataset contains 60,000 images are used for training and 10,000 images are used for testing. The aim of using this dataset is to classify images into 10-classes that represent the integer value. An example of four digits from the dataset is shown in Figure 1.

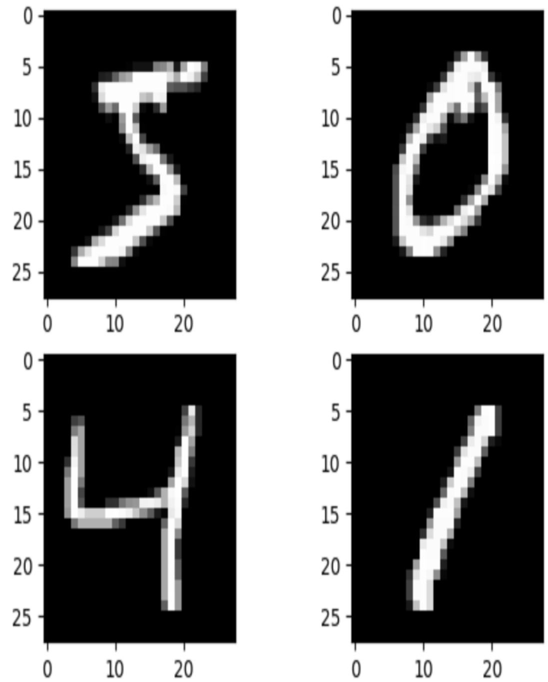

Figure 1: Four digits image from MNIST dataset

Many machine learning methods starting from simple linear classifier to complex deep learning neural network models have been used in the literature to classify images of MNIST dataset. The performance is evaluated by generating a less error rate and increasing the recognition accuracy [11-14].

In this paper, MNIST dataset is used to develop and evaluate a methodology using convolutional neural network in order to solve image classification problem.

1.2 Methods

Convolutional Neural Network

Convolutional neural network (CNN) is a type of deep neural network that can be used in many applications to analysis visual images [15]. Some of these applications include image classifications [16], voice recognition [17], natural language processing [18], time-series forecasting [19] and more.

The architecture of CNNs consist of input and output layers as long as multiple hidden layers that convolve with a multiplication to results as a series of convolutional layers. The architecture can be viewed as a combination of three types of layers: convolution, max-pooling and classification, as shown in Figure 2.

Journal of Theoretical and Applied Information Technology
15th November 2020. Vol.98. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

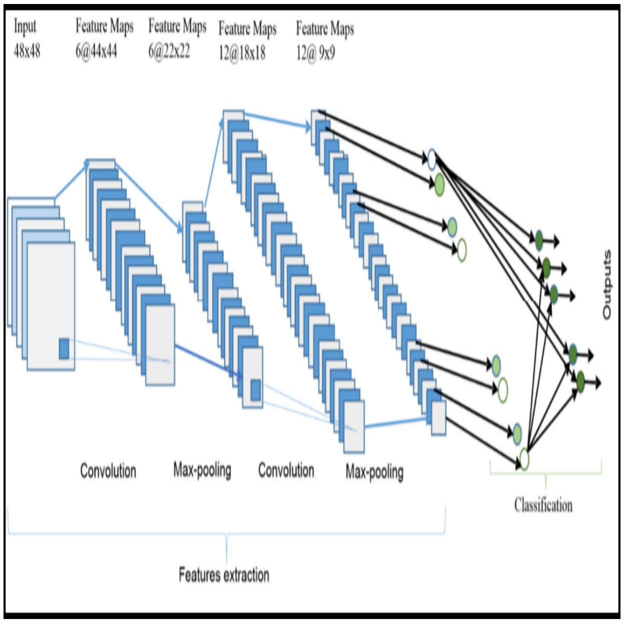

Figure 2: CNN architecture [20]

The convolution and max-pooling layers are feature extractors layers that passes its output as an input to the next layer. The features in the high-level are derived from features propagated from low-level layers, where the dimensions of the features reduced as the features propagated to the highest-level layers. To increase classification accuracy, the number of feature maps may be increased for a better representation of input image features. A fully connected layer in the output layer receives output from the last layer in the feature extractors layers as an input. The feed-forward neural network (FFNN) is used for that to improve performance [21, 22].

The activation function of convolutional layer if usually RELU which are then followed by pooling layer that summarize the output.

The input of CNN is a tensor, which is an algebraic object with a shape that represent (number of images) x (image hight) x (image width) x (image depth). The image is passed to the first convolutional layer in order to extract features. The aim is to reduce the image to an easier form without losing its features. The image is abstracted to a feature map with a defined size. The convolutional layer applies RELU function and passes its output to the next layer in the deep architecture [23, 24].

The pooling layer then gets the output of the convolutional layer to summarize it by reducing its dimension by combining the output of a neuron cluster to a single neuron. It can be achieved by computing the average or max value for the neurons of the cluster. The pooling layer reduces computational power needed to process the data by reducing dimensionality.

After this, the image is flattening into a vector. The vector is fed to a feed-forward neural network and the backpropagation algorithm is applied. After a certain number of repeated epochs, the model can classify the image using Softmax classification technique that applies Softmax function.

Figure 3 shows an example of the CNN architecture.

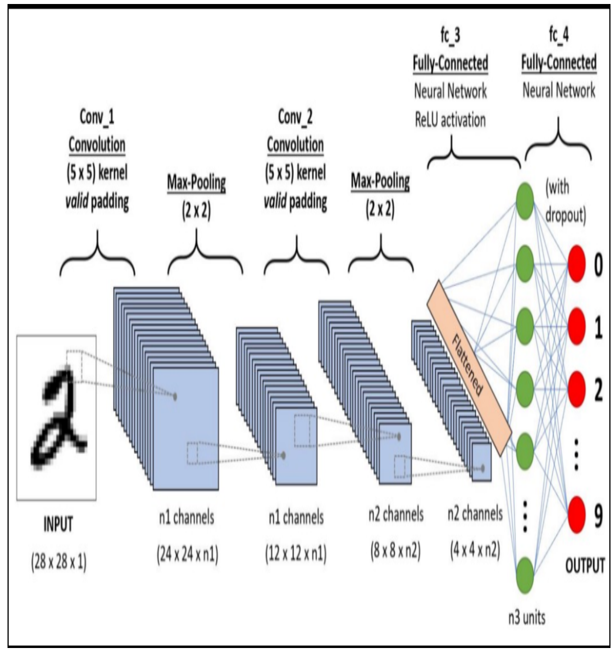

Figure 3: CNN network to classify handwritten images

1.3 Related Works

Different researches have been conducted in the literature in the field of digit recognition and by using different machine learning techniques. In [25], the authors investigated the influence of a number of network parameters such as number of layers, stride size, receptive field and kernel size for CNN handwritten digit recognition. The results show that a fine tuning of hyperparameters can improve CNN model performance. Pham et al. in [26] used Recurrent Neural Network (RNN) with a regularization technique of Dropout method to improve the performance of handwritten recognition model. The results show an improvement in the model performance by decreasing the character error rate and word error rate. Shi et al integrated CNN and RNN for scene text recognition and found that the integration outperforms the tradition methods of recognition

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[27]. A deep convolutional neural network model is proposed in [28] for semantic segmentation. The proposed model architecture consists of decoder-encoder networks.
The CNN is widely used in the literature for recognition tasks, including Arabic language recognition [29], handwritten Tamil character recognition [30], handwritten Urdu text recognition [31, 32], Chinese handwritten text recognition [33, 34], Telugu character recognition [35] and Indic scripts hand written character recognition [36].
The performance of CNN model depends mainly on the choice of network hyperparameters (such as number of hidden layers, activation function, number of epochs, learning rate, etc) [37]. These parameters control the way the algorithm can learn from the data to extract features and make the prediction [38]. A good choice of these parameters guarantees the generation of a satisfiable prediction result. This paper investigates the influence of three CNN hyperparameters which are learning rate, number of epochs and number of hidden layers on the performance of CNN for handwritten digit recognition.

2. METHODOLOGY

A new model is developed from the scratch. The development involves the running of four main steps: 1) Dataset preparation. 2) Model design. 3) Model evaluation. 4) Results presentation. Based on the results generated from the model, the configuration of its architecture is optimized to enhance accuracy and produce better results. The details of each phase are given below.

2.1 The Dataset Preparation

Each image in the dataset is of size 28x28 and each image is of grayscale (the pixel value of each image is between 0 and 255 to represent black and white). The first step of preparing dataset for training is to scale the pixel values for modelling to be within the range [0,1].

The next step is to split the dataset for training and testing. The dataset already divided into two sets: training and testing. as mentioned before, the training set contains 60,000 images, it can be divided into two sets: training and validation in order to estimate the performance of the model using a specific metrics which is k-fold cross validation.

2.2 Model Design

This step involves the creation of a baseline convolutional neural network model able to classify the MNIST dataset. The model can be configured according to the results to enhance its performance.

The model is composed of two main aspects: the first one compromised of convolutional and pooling layers and is responsible of extracting features from each input image, the second classify the output and make the prediction. The first aspect of the model (convolutional and pooling) can be repeated to produce a deep architecture. It can be determined based on the performance and the accepted level of accuracy.

We started with a single convolutional layer with small filter size (3, 3) followed by a max pooling layer. The output layer has 10-neurons to classify image into 10 different classes. The softmax activation function is used for that. A dense layer (fully connected layer) is usually added between convolutional and output layers to interpret the features well.

The neural network has a number of parameters that affect on its performance. The configuration of the network includes the tuning of these parameters to enhance the performance [39]. Some of these parameters include: Learning rate, Momentum, Number of layers, Activation function and more. According to the problem domain, the parameters that are needed to be investigated are chosen. In the problem proposed in this paper, we started with this initial configuration:

Learning rate: 0.01.

Momentum: 0.9.

Number of layers: 1.

Activation function: RELU.

Batch size: 32.

Number of epochs: 10

A five-fold cross validation is used to evaluate the baseline model, where the performance is evaluated by finding the mean of the five folds

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and the standard deviation as discussed in the next section.

2.3 Model Evaluation

The model is evaluated by repeating running five times using five-fold cross validation. Because dataset is of large size, the running will take long time, so the choosing of five-fold is done to reduce running time needed for the evaluation. As mentioned before, the training dataset has 60,000 samples. The training dataset is split into two main groups: Training and Validating with a percentage of 80% to 20% used for them. The validating results are used to evaluate the model each time.

2.4 Results Presentation

After evaluating the model, the results can be represented either by plotting the model performance which will give an idea about model overfitting, underfitting or has a good fit for the dataset. The other way to present results is by evaluating the mean and standard deviation for the classification accuracy collected during each fold of the training.

2.5 Optimization Metrics

In order to improve on the model performance, there are many metrics that can be explored for improvement in the learning algorithm. In this paper, we investigated the effect of the following metrics in the model performance:

1. learning rate: Testing the impact of using large or small value of learning rate.

2. Number of epochs: Evaluating the model performance over different number of epochs.

3. Model Depth: Adding more convolutional and pooling layers.

The evaluation of the model scenario can be continued as long as the time and resources are suitable for that. Many ideas can be applied to improve the performance. At some point, the model should be finalized with the chosen configurations. The model is then can be used to make the prediction on the testing dataset.

3. EXPERIMENTAL RESULTS

The experiments were developed in python using Keras deep learning libraries provided to create a convolutional neural network. The first experiment run over the baseline model where accuracy is evaluated after each fold, then mean and standard deviation are estimated. The results are shown in Table 1.

Table 1: Accuracy achieved by baseline model

Fold	Accuracy
Fold 1	98.467
Fold 2	98.325
Fold 3	98.558
Fold 4	98.825
Fold 5	98.767

The skills are summarized by finding mean and standard deviation as follows: mean = 98.61875, standard deviation = 0.227.

The experiment is repeated to evaluate the model performance over different learning rate, different number of epochs and different depth (more convolutional and pooling layers). The optimization can be determined based on the results. The best configuration will be chosen to represent model architecture. The results of mean and standard deviations are summarized in the tables below and the figures represent performance of each configuration.

Table 2: Accuracy achieved by baseline model over different learning rate

Learning rate	Mean	Standard Deviation
0.01	98.61875	0.227
0.05	98.4532	0.235
0.08	94.6434	2.558
0.1	89.5481	5.341
0.15	86.0762	7.529

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

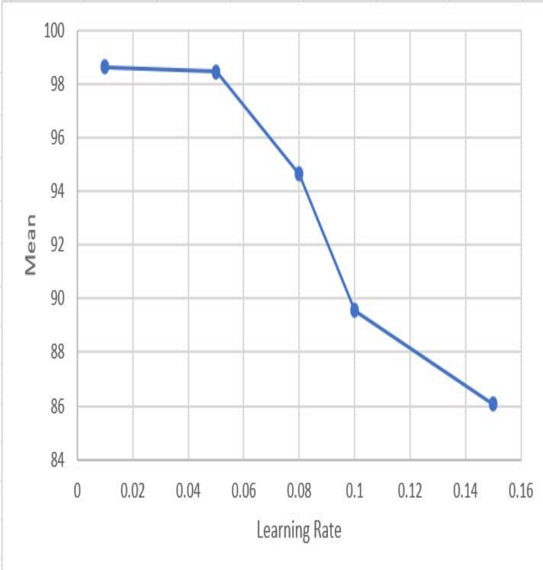

Figure 4: Mean of 5-fold using different learning rate value

Table 2: Accuracy achieved by baseline model over different number of epochs

Number of epochs	Mean	Standard Deviation
10	98.61875	0.227
15	98.785	0.118
20	98.7432	0.145
25	98.695	0.147
30	98.613	0.151

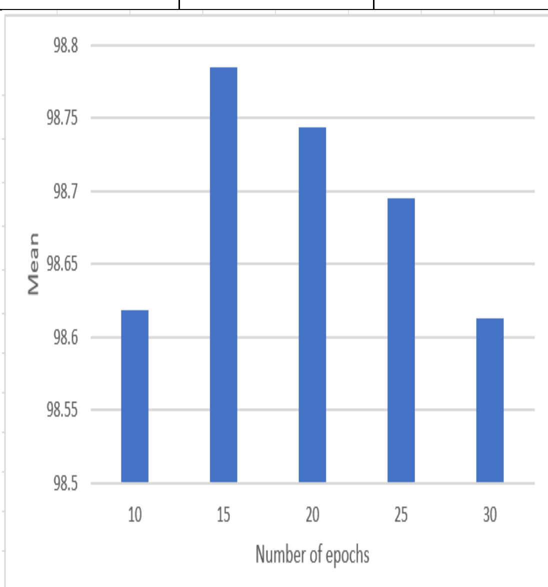

Figure 5: Mean of 5-fold using different number of epochs

Table 2: Accuracy achieved by baseline model over different number of hidden layers

Depth	Mean	Standard Deviation
1-layer	98.61875	0.227
2-layers	98.9052	0.130
3-layers	98.218	0.187

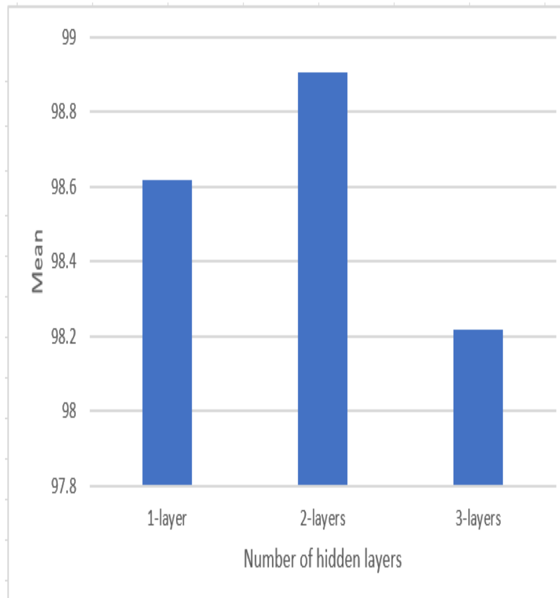

Figure 6: Mean of 5-fold using different number of hidden layers

3.1 Discussion

The model presented in this paper used CNN to classify handwritten images for the MNIST dataset. The accuracy achieved by the baseline model is summarized by finding the average of accuracy generated by each fold of 5-fold cross validation where configurations are selected randomly. In order to evaluate the model and estimate the performance by changing its configurations, three metrics were selected for that: the learning rate value, the number of epoch and the depth of the model architecture.

The best value for the learning rate can be chosen by experiment. The behaviour of model performance depends on the dataset used to build the model, this means that the best value of learning rate that generate the best accuracy can be changed if the problem domain changed and dataset also changed. For the problem which is investigated in this paper, it can be seen from table 2 and Figure 4 that 0.01 value generates the best accuracy. The increasing in the learning rate value reduces classification accuracy of the model. this value can be used as an optimal learning rate value to configure CNN model of handwritten classification problem.

For the number of epochs metric, the results showed that giving the model more time with a greater number of epochs to train does not mean that results will be enhanced. On the other hand, the complexity increased as more time needed to make the prediction. The best accuracy achieved when the number of epochs is 15, this value can be used for the model configuration setting.

The standard deviation can give a view about how much the group values are far from the mean of this values. A high value of it means that data is less reliable and the low value means that data is more reliable and closed to the mean value. It can be seen from tables 2 and 3 that lowest value for standard deviation generated for the best selected learning rate value which is 0.01 and number of epochs which is equal to 15.

The depth of the model architecture can influence on increasing accuracy by adding more hidden layers. Some problems may not need to increase the model depth, it depends on the data used to train the model. for large dataset, increasing the depth of the model helps on capturing data dependency. On the other hand, the complexity increases and the time to train the model will also increase. In this paper, the model performance is investigated over one, two and three hidden layers. The best results achieved for two hidden layers with best value generated for mean and standard deviation as shown in Figure 6.

It can be concluded that the selection of best hyperparameters to be selected for the model configuration depends on the dataset used to build the model, the problem domain and resources available to run the model and make the prediction.

3.2 Advanced Training Methods

For efficient training of CNN, some advanced techniques may be considered for better model learning. This includes: input image pre-processing, weight initialization, batch normalization, activation function and more. A brief discussion for each one of them is given below.

1. Input image pre-processing: Different scaling methods can be applied on the input before feeding it to the network. Some of them includes random cropping, color jittering, sample rescaling and many more.
2. Weight initialization: The weight initialization of the neural network is an important step that has a big effect on the performance of the network [40]. For a high-dimensionality CNN training, the weights should not be symmetrical because of back-propagation process. Therefore, an efficient initialization technique is preferable and important for the training. Some techniques that are proposed through literatures by LeCun and Bengio include scaling the weights by the inverse of the square root of the number of input neurons of the layer [20, 41, 42].
3. Batch normalization: Is a technique used to accelerate network training by automatically standardize the input to a layer in the deep learning to improve the performance of the model. The input is transformed in a way to have a mean of zero and a standard deviation of one [43, 44].
4. Activation function: The traditional activation functions used in the implementation of neural networks are Sigmoid and Tanh. ReLu activation function is recently and widely used to solve vanishing gradient problem by keeping all the values above zero and setting all the negative values to zero [45]. The activation function plays an important role in learning the weights of CNN and other deep neural network architecture. Several improved versions of ReLu have been proposed to enhance learning accuracy such as Leaky ReLu, ELU, MELU and S shape Rectified Linear Activation units [46, 47, 48].

Several methods that can be used for efficient training of CNN and other deep neural networks. A future works for the proposed model for better accuracy may include the using of such methods.

4. CONCLUSIONS

This paper proposed a deep neural network model based on the using of convolutional neural networks to classify the handwritten images from MNIST dataset. The MNIST dataset was used in this experiment for the model training and performance evaluation. The dataset is divided into two groups: training and testing, the training was used for the model training and the testing was used for the performance evaluation where the 5-fold cross validation was used to repeat running of model performance over MNIST dataset. The experimental results achieved a high score of accuracy with one hidden layer and 10 number of epochs. The performance was enhanced by testing the model over different configurations. The configuration metric which are investigated in this paper are: learning rate, number of epochs, model depth. The results show that the choosing of the best learning rate value can be determined by experiment. The increasing in the number of epochs increases the complexity and does not guarantee the enhancement in the model performance. The depth architecture has a great effect on model performance especially for complex and large dataset.

REFRENCES

[1] Le Hoang Thai, Tran Son Hai and Nguyen Thanh Thuy. "Image Classification using Support Vector Machine and Artificial Neural Network" I.J.Information Technology and Computer Science,2012, 5, 32-38

[2] Giuseppe Amato and Fabrizio Falchi. "kNN based image classification relying on local feature similarity". Third International Workshop on Similarity Search and Applications, SISAP 2010,18-19 September2010, Istanbul, Turkey

[3] Lili Zhu and Petros Spachos. "Towards Image Classification with Machine Learning Methodologies for Smartphones". Mach.Learn.Knowl. Extr. 2019, 1, 1039–1057; doi:10.3390/make1040059

[4] Simonyan, K.; Zisserman, A. Very Deed Convolutional Networks for Large-ScaleImage Recognition. arXiv 2014,arXiv:1409.1556

[5] Davy Preuveneers, Ilias Tsingenopoulos and Wouter Joosen. "Resource Usage and Performance Trade-offs for Machine Learning Models in Smart Environments". Sensors 2020, 20, 1176; doi:10.3390/s20041176

[6] Asifullah Khan, Anabia Sohail1, Umme Zahoora, and Aqsa Saeed Qureshi. "A Survey of the Recent Architectures of Deep Convolutional Neural Networks". Published in Artificial Intelligence Review, DOI:https://doi.org/10.1007/s10462-020-09825-6

[7] Support vector machines speed pattern recognition - Vision Systems Design". Vision Systems Design. Retrieved 17 August 2013.

[8] Gangaputra, Sachin. "Handwritten digit database". Retrieved 17 August 2013

[9] Qiao, Yu (2007). "THE MNIST DATABASE of handwritten digits". Retrieved 18 August 2013

[10] Platt, John C. (1999). "Using analytic QP and sparseness to speed training of support vector machines" (PDF). Advances in Neural Information Processing Systems: 557–563. Archived from the original (PDF) on 4 March 2016. Retrieved 18 August 2013

[11] LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-Based Learning Applied to Document Recognition" (PDF). Proceedings of the IEEE. **86** (11): 2278–2324. doi:10.1109/5.726791. Retrieved 18 August2013.

[12] Ciresan, Dan Claudiu; Ueli Meier; Luca Maria Gambardella; Jürgen Schmidhuber (2011). Convolutional neural network committees for handwritten character classification (PDF). 2011 International Conference on Document Analysis and Recognition (ICDAR). pp. 1135–1139.

[13] Wan, Li; Matthew Zeiler; Sixin Zhang; Yann LeCun; Rob Fergus (2013). Regularization of Neural Network using DropConnect. International Conference on Machine Learning(ICML)

[14] Romanuke, Vadim. "The single convolutional neural network best performance in 18 epochs on the expanded training data at Parallel Computing Center, Khmelnitskiy, Ukraine". Retrieved 16 November 2016.

[15] Valueva, M.V.; Nagornov, N.N.; Lyakhov, P.A.; Valuev, G.V.; Chervyakov, N.I. (2020). "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation". *Mathematics and Computers in Simulation*. Elsevier BV. **177**: 232–243. doi:10.1016/j.matcom.2020.04.031. ISSN 0378-4754

[16] Rawat, H. and Wang, Z., "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", Neural Computation, 2017

[17] Mariusz Kubanek , Janusz Bobulski and Joanna Kulawik, "A Method of Speech Coding for Speech Recognition Using a Convolutional Neural Network", Symmetry, 2019

[18] P. Li, J. Li and G. Wang, "Application of Convolutional Neural Network in Natural Language Processing," *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, 2018

[19] Tsantekidis, Avraam; Passalis, Nikolaos; Tefas, Anastasios; Kanniainen, Juho; Gabbouj, Moncef; Iosifidis, Alexandros (July 2017). "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks". *2017 IEEE 19th Conference on Business Informatics (CBI)*. Thessaloniki, Greece: IEEE: 7–12. doi:10.1109/CBI.2017.23

[20] Md Zahangir Alom , Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike,

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen , Abdul A. S. Awwal and Vijayan K. Asari. "A State-of-the-Art Survey on Deep Learning Theory and Architectures". Electronics 2019, 8, 292; doi:10.3390/electronics8030292

[21] Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006, 18, 1527–1554

[22] Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.

[23] Habibi, Aghdam, Hamed (2017-05-30). Guide to convolutional neural networks: a practical application to traffic-sign detection and classification. Heravi, Elnaz Jahani. Cham, Switzerland. ISBN 9783319575490. OCLC 98 7790957

[24] Balas, Valentina E.; Kumar, Raghvendra; Srivastava, Rajshree (2019-11-19). Recent Trends and Advances in Artificial Intelligence and Internet of Things. Springer Nature

[25] Savita Ahlawat, Amit Choudhary, Anand Nayyar, Saurabh Singh and Byungun Yoon. "Improved Handwritten Digit Recognition Using Convolutional Neural Networks (CNN)". Sensors 2020, 20, 3344; doi:10.3390/s20123344

[26] Pham, V.; Bluche, T.; Kermorvant, C.; Louradour, J. Dropout improves recurrent neural networks for handwriting recognition. In Proceedings of the 14th Int. Conf. on Frontiers in Handwriting Recognition, Heraklion, Greece, 1–4 September 2014

[27] Shi, B.; Bai, X.; Yao, C. An End-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 2298–2304

[28] Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 2481–2495

[29] Boufenar, C.; Kerboua, A.; Batouche, M. Investigation on deep learning for off-line handwritten Arabic character recognition. Cogn. Syst. Res. 2018, 50, 180–195

[30] Kavitha, B.; Srimathi, C. Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. J. King Saud Univ. Comput. Inf. Sci. 2019

[31] Ahmed, S.; Naz, S.; Swati, S.; Razzak, M.I. Handwritten Urdu character recognition using one-dimensional BLSTM classifier. Neural Comput. Appl. 2019, 31, 1143–1151

[32] Husnain, M.; Saad Missen, M.; Mumtaz, S.; Jhanidr, M.Z.; Coustaty, M.; Luqman, M.M.; Ogier, J.-M.; Choi, G.S. Recognition of urdu handwritten characters using convolutional neural network. Appl. Sci. 2019, 9, 2758.

[33] Xie, Z.; Sun, Z.; Jin, L.; Feng, Z.; Zhang, S. Fully convolutional recurrent network for handwritten chinese text recognition. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR 2016), Cancun, Mexico, 4–8 December 2016.

[34] Liu, C.; Yin, F.; Wang, D.; Wang, Q.-F. Online and offline handwritten Chinese character recognition: Benchmarking on new databases. Pattern Recognit. 2013, 46, 155–162

[35] Dewan, S.; Chakravarthy, S. A system for offline character recognition using auto-encoder networks. In Proceedings of the International Conference on Neural Information Processing, Doha, Qatar, 12–15 November 2012

[36] Sarkhel, R.; Das, N.; Das, A.; Kundu, M.; Nasipuri, M. A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts. Pattern Recognit. 2017, 71, 78–93

[37] Cui, H.; Bai, J. A new hyperparameters optimization method for convolutional neural networks. Pattern Recognit. Lett. 2019, 125, 828–834

[38] Tso, W.W.; Burnak, B.; Pistikopoulos, E.N. HY-POP: Hyperparameter optimization of machine learning models through parametric programming. Comput. Chem. Eng. 2020, 139, 106902

[39] Surakhi, O., Serhan, S. and Salah, I., "On the Ensemble of Recurrent Neural Network for Air Pollution Forecasting: Issues and Challenges", Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 2, 512-526 (2020)

[40] Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On The Importance Of Initialization And Momentum In Deep Learning. Int. Conf. Mach. Learning. 2013, 28, 1139–1147

[41] Lecun, Y.; Bottou, L.; Orr, G. Efficient Backprop. In Neural Networks: Tricks Of The Trade; Orr, G., Müller, K., Eds.; Lecture Notes

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In Computer Science; Springer: Berlin, Germany, 1524.

[42] Glorot, X.; Bengio, Y. Understanding The Difficulty Of Training Deep Feedforward Neural Networks. In Proceedings Of The Thirteenth International Conference On Artificial Intelligence And Statistics, Sardinia, Italy, 13–15 May 2010; Pp. 249–256

[43] Vedaldi, A.; Lenc, K. Matconvnet: Convolutional Neural Networks For Matlab. In Proceedings Of The 23rd Acm International Conference On Multimedia, Brisbane, Australia, 26 30 October 2015; Pp. 689–692

[44] Laurent, C.; Pereyra, G.; Brakel, P.; Zhang, Y.; Bengio, Y. Batch Normalized Recurrent Neural Networks. In Proceedings Of The 2016 Ieee International Conference On Acoustics, Speech And Signal Processing (Icassp), Shanghai, China, 20– 25 March 2016; Pp. 2657–2661.

[45] Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings Of The 27th International Conference On Machine Learning (Icml-10), Haifa, Israel, 21–24 June 2010; Pp. 807–814

[46] Li, Yang, Chunxiao Fan, Yong Li, Qiong Wu, And Yue Ming. Improving Deep Neural Network With Multiple Parametric Exponential Linear Units. Neurocomputing 2018, 301, 11–24.

[47] Jin, X.; Xu, C.; Feng, J.; Wei, Y.; Xiong, J.; Yan, S. Deep Learning With S-Shaped Rectified Linear Activation Units. Aaai 2016, 3, 2–3.

[48] Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical Evaluation Of Rectified Activations In Convolutional Network. Arxiv 2015, Arxiv:1505.00853