SVU-International Journal of Engineering Sciences and Applications (2023) 4(1): 41-47
Print ISSN 2785-9967 | Online ISSN 2735-4571
DOI 10.21608/svusrc.2022.161642.1073

Robust deep convolutional neural network-Based Classifiers

Y.M Assem [1], Adel B. Abdel-Rahman [2], Mohamed Hassan Essai Ali [3], Samar Hashim Ali [4,] □

Abstract A deep learning convolutional deep neural network (CNN) is employed to build robust classifiers. These classifiers can analyse large amounts of data, find statistical dependencies, learn correlations between features, and generalise their findings. This paper proposes the use of Mean Absolute Error (MAE), Sum of Square of Errors (SSE) and cross entropy loss functions to create two new classification layers, which are employed as the last layer in the proposed convolutional deep neural network -based classifier. A comparative study was conducted to assess the performance of the presented convolutional deep neural network -based classifiers using three different loss functions (cross entropy "conventional", MAE, and SSE) -based classification layers and Adam (adaptive moment estimation), and SGdm (stochastic gradient descent with momentum) optimizers. In addition, the accuracy and loss curves that resulted from the training process are provided for comparison purposes. Handwritten digits dataset was used as a classification case study using the proposed classifiers.

Keywords:. Deep Learning , CNN ,cross entropy ,MAE , SSE, ADAM , SGDM..

1 Introduction[1]

In computer vision, image classification is crucial, It has a huge impact on our work, study and life. Image preprocessing, image segmentation, key feature extraction, and matching identification are all steps in the process of classifying images

The most recent image classification methods enable us to not only obtain image data more quickly than before but also to use it in a variety of fields, including face recognition, medical equipment, traffic identification, and security [1, 2].

A wide range of crucial applications, including data mining, natural language processing, image identification, and expert systems, heavily rely on Machine Learning. Machine learning is regarded as a cornerstone of future civilization since it offers a potential answer in each of those domains. A variety of constructing algorithms used in Machine Learning allow computers to learn from data and make predictions and judgments based on that data. Machine learning's explosive growth over the last few decades has had a significant impact on how we live our daily lives with Examples of machine learning applications include predicting bankruptcy, predicting rainfall, forecasting the weather, self-driving systems, optical character recognition, and more[3] [4]. Even though machine

2. Related work

Convolutional Neural Networks (CNN or ConvNet) are a particular type of multi-layer neural network that were motivated by the operation of visual systems in living creatures. Animal visual cortex cells can perceive light in the tiny receptive field, according to Hubel and Wiesel[6]. This research served as the inspiration for Kunihiko Fukushima's invention of the multi-layered neural network known as the Neocognitron[7] , which can recognize visual patterns hierarchically through learning Theoretically, CNN was theoretically inspired by this network. LeCun and colleagues. established the CNN practical model[8] and created LeNet-5 [9].

Received: 08 September 2022/ Accepted: 09 October 2022
□Corresponding Authors: Yasser Mohamed Asem, yaser.asem@gmail.com
Adel Bedair Abdel-Rahman, Adel.bedair@ejust.edu.eg
Mohamed Hassan Essai Ali, mhessai@azhar.edu.eg
Samar Hashim Ali, engsam016@gmail.com
1. Department of Educational and Information technology, Faculty of Specific Education, South Valley University, Qena, Egypt.
2. Faculty of Engineering, South Valley University, Qena, Egypt and E-JUST, Alexandria, Egypt.
3.Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Qena, Qena, Egypt.
4. Department of Communication and Electronic, The High institute of Engineering and Technology, El-Tod, Luxor, Egypt.

LeNet-5 was able to recognize visual patterns from raw pixel data without the need for a separate feature engineering method thanks to training using backpropagation[10] Additionally, CNN had fewer connections and parameters than traditional feedforward neural networks with a comparable network size, which facilitated model training However, despite having a number of advantages at the time, CNN's performance in challenging tasks like classifying high-resolution images was constrained by a lack of sufficient training data, a lack of an improved regularization technique, and a lack of suitable computational power Modern datasets include millions of highly-resolution labelled data in hundreds of categories, such as ImageNet[11] , LabelMe [12] . With the development of robust GPU machines and improved regularization techniques, CNN now performs brilliantly on picture classification tasks. AlexNet [13], AlexNet [13], a sizable deep convolution neural network created by Krizhevsky et al. in 2012. on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), demonstrating exceptional performance [14]. The success of AlexNet has influenced numerous CNN models over the years, including ZFNet [15], VGGNet [16], GoogleNet [17], ResNet [18], DenseNet [19], CapsNet [20], SENet [21] etc. in the following years.

The Digits handwritten database will be used in research to accurately classify the hand-written numbers. Various learning algorithms, including CNN [22], Support Vector Machine (SVM) [23], Restricted Boltzmann Machine (RBM) [24], and LImited Receptive Area (LIRA) [25], have been used in prior studies to classify handwriting. By altering the database, the MNIST data is also used to categories objects other than numbers. By incorporating more characters into the Digits handwritten database and organizing the alphabets, Cohen et al. [26] develop EMNIST. Xiao and co.

3. Basic CNN architecture

3.1 Basic architecture of CNN

A convolutional neural network(CNN) consists of input, output, and hidden layers [2].

Convolution layers, a pooling layer, and one or more completely connected layers make up the building blocks of hidden layers.[27].

A typical CNN is made up of one or more output layers, one or more fully connected layers, and one or more blocks of convolution and subsampling layers. [28] ,as shown in Fig.1

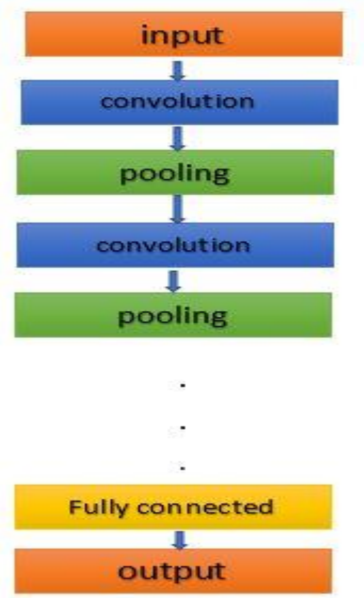

Fig . 1 Block of CNN

3.1.1 Convolutional Layer

The primary component of a convolutional neural network is its convolutional layer, which has regional connections and weights for shared traits. Convolutional layer's objective is to teach input feature representations. Several feature maps make up the convolutional layer. The same feature map's neuron is utilized to extract local attributes from various points in the previous layer, but for individual neurons, its extraction is based on the local features of identical sites in a previous feature map. The input feature maps are convolved with a learnt kernel to produce a new feature, which is then obtained by passing the output through a nonlinear activation function. Applying various kernels will result in distinct feature maps. The sigmoid, tanh, and Relu activation functions are typical.[29].

3.1.2 Pooling Layer

Sample selection is the same as fuzzy filtering. The secondary feature extraction is affected by the pooling layer, It can shrink the size of the feature maps and make feature extraction more robust. Typically, it is positioned between two Convolutional layers. The movement step of the kernels determines the size of the feature maps in the pooling layer The two common pooling techniques are average pooling[30] and max pooling [31]. By stacking many Convolutional layers and pooling layers, we may extract the high-level properties of inputs.

3.1.3 Fully-connected Layer

Typically, a Convolutional neural network's classifier consists of one or more fully connected layers. In fully connected layers, spatial information is not maintained. In fully connected layers, spatial information is not maintained. An output layer follows the final completely connected layer. SoftMax regression is frequently used for classification applications because it produces outputs with a good probability distribution. SVM is another often employed technique that can be used in conjunction with CNNs to address various classification problems.[32].

3.2 How CNN works

The training program repeatedly makes forward and reverse passes through the network. Each layer inputs the outputs of the preceding layers, applies a function, and outputs (forward propagates) the results to the next levels while performing a forward pass through the network

There may be several inputs or outputs for layers. As an illustration, a layer can take X1,..., Xn from several preceding layers and transmit the outputs Z1,..., Zm to the next layers. The output layer determines the loss L between the predictions Y and the actual targets T at the conclusion of a forward pass of the network..

Each layer in a network computes the derivatives of the loss L with respect to the inputs, takes the derivatives of the loss L with respect to the outputs of the layer, and propagates the findings backward. The layer computes the derivatives of the layer weights if the layer has learnable parameters (learnable parameters). The learnable parameters are updated by the layer using the weights' derivatives. The data flow through a layer with a single input X, a single output Z, and a learnable parameter W is highlighted in the accompanying Fig.2, which shows the data flow through a deep neural network.

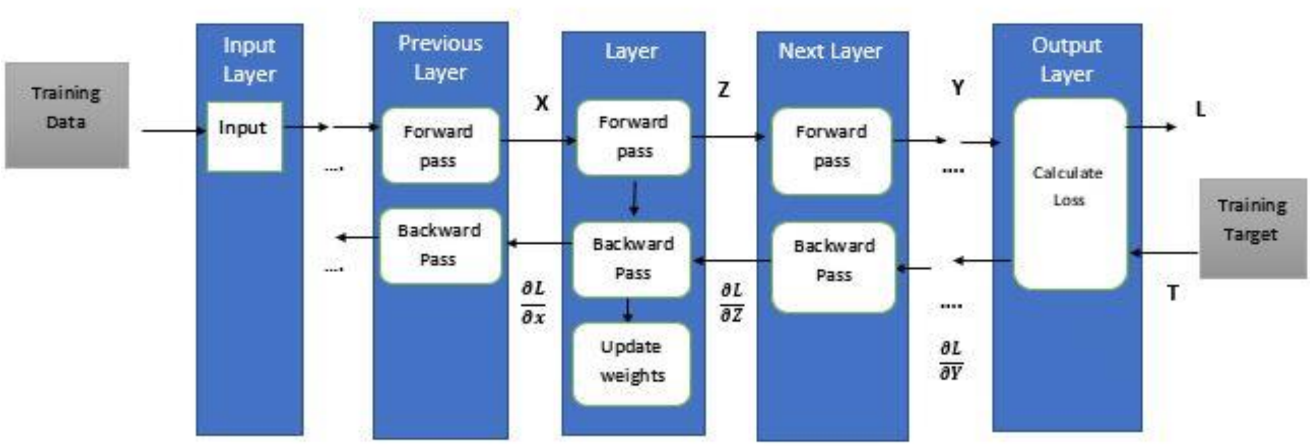

Fig . 2 Work of CNN network

4 proposed convolutional neural network

In this section we discuss the used dataset, and adopted layers, and (MAE, SSE) lose functions-based classification layers.

4.1 DataSet

The 10,000 handwritten digits in the Digits data collection are rendered in synthetic grey. Each image measures 28 by 28 pixels and has a label identifying the digit it represents (0–9). A certain angle has been applied to each image. You can load the image's rotation angle while loading the images as arrays. as shown in Fig.3 samples of Digit dataset.

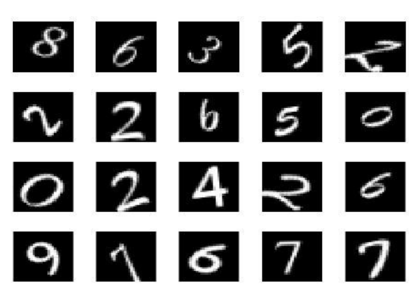

Fig . 3 Sample of digit dataset

4.2 Layers

- Input Layer: Using an image input layer called imageInputLayer, 2-D images are entered into a network and data normalization with a size of (28x28x1) is applied..

- Convolution Layer: applying sliding convolutional filters to the input using the convoltion2dLayer, a 2-D convolutional layer. With a filter size of 5, and 20 total filters, the layer convolves the input by moving the filters along it both vertically and horizontally, computing the dot product of the weights and the input, and adding a bias term.

- Activation Layer: Rectified Linear Unit (ReLU) layer is used, and it conducts a threshold operation on each input element, setting any value less than zero to zero.

- Normalization Layer: utilizing batchNormalizationLayer, which by default has a mini-batch size of 128 and normalizes each input channel throughout the mini-batch. Use batch normalization layers between convolutional layers and nonlinearities, such as ReLU layers, to accelerate the training of convolutional neural networks and lessen the sensitivity to network initialization.

- Fully Connected Layer : A completely connected layer adds a bias vector after multiplying the input by a weight matrix using fullyConnectedLayer.

- Output Layer : A softmax function is applied to the input using the softmaxLayer and classificationLayer.

- classificationLayer : is a classification layer that calculates the cross entropy loss for multi-class classification problems with mutually exclusive classes.

4.3 Loss Functions

In the proposed network use three types of Loss function and work compared to them so as to get the best performance and review how each function works

Crossentropy Function calculates a network performance given targets and outputs, with optional performance weights and other parameters

SSE Function: It measures performance according to the sum of squared errors.

MAE Function: It measures network performance as the mean of absolute errors.

Loss functions can be expressed as

$$\text{crossentropyex}= -\sum_{i=1}^{N}\sum_{i=1}^{c} X_{ii}(k)\log(\hat{X}_{ii}(k)) \qquad (1)$$

$$\text{MAE}= \frac{\sum_{i=1}^{N}\sum_{j=1}^{c}|x_{ij}(k)-\hat{x}_{ij}(k)|}{N} \qquad (2)$$

$$\text{SSE} = \sum_{i=1}^{N}\sum_{j=1}^{c}(X_{ij}(k) - \hat{X}_{ij}(k))^2 \qquad (3)$$

Where N : sample number , c : class number , X_ij is the ith transmitted data sample for jth class and X_ij is the CNN-based classifier response for sample i for class j [33].

4.4 Implementation

The proposed algorithm was implemented on Digit dataset using the MATLAB program , all images are classified into 10 classes (0 to 9). At first, we rescaled all of our 10000 images into constant size of 28x28x1 to be suitable for training of our model and then normalize them. Next, each class or category contain on 1000 , we split each category into to set train set which contain 750 image and validation set which contain 250 images , and test them with different batch-size and different number of epochs to improve the performance of the proposed algorithm .In addition, our optimization algorithm for computing loss function using two : Stochastic Gradient Descent with Momentum (SGDM) ,and Adaptive Moment Estimation (ADAM) with a learning rate of 0.01 and our activation function in each layer was ReLU. Under these settings, we should keep weights and parameters of previous layers fixed. We measure the performance of the classification model using three loss functions: the Cross-Entropy, MAE and SSE to adjust weights and calculate the difference between the output and the label of our image dataset for different epochs and batch size to get the best accuracy.

5 Experimental results

The best accuracy of our model during training and testing was evaluated for optimizer ADAM, epochs 20 and batch size 43.

Table 1 presents the best accuracies that achieved using the proposed classifiers using different optimizers (SGDM, and ADAM) and different loss functions (cross entropy – MAE – SSE)-based classification layers.

Table 1 parameters and results of our proposed method for dataset with different loss functions

optimizer	Epochs	Patch Size	accuracy
Loss function: crossentropy			
SGDM	25	64	99.65%
ADAM	25	64	99.52%
Loss function: MAE			
SGDM	60	43	99.44%
ADAM	40	100	99.60%
Loss function: SSE			
SGDM	25	128	99.68%
ADAM	20	43	99.72%

Fig.4, and Fig.5 depict accuracy, and loss curves of the proposed CNN-based classifiers, that trained using ADAM optimizer and loss functions (cross entropy – MAE – SSE)-based classification layers, respectively.

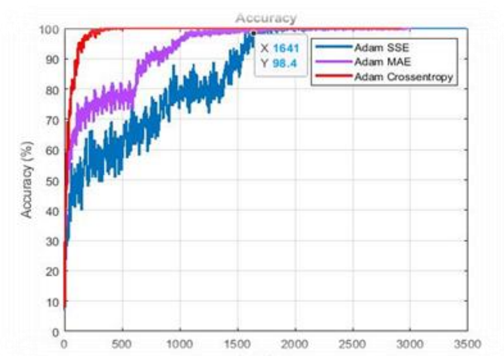

Fig . 4 Accuracy curves of CNN-based classifiers using Adam optimizer, and MAE-, SSE-, crossentropy- based classification layers

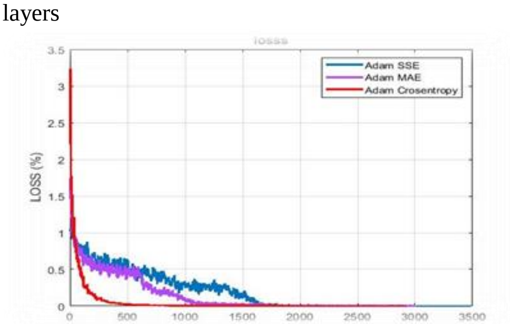

Fig . 5 Loss curves of CNN-based classifiers using Adam optimizer, and MAE-, SSE-, crossentropy- based classification layers

Fig.6 and Fig.7 depict accuracy, and loss curves of the proposed CNN-based classifiers, that trained using SGDM optimizer and loss functions (cross entropy –MAE–SSE)-based classification layers, respectively.

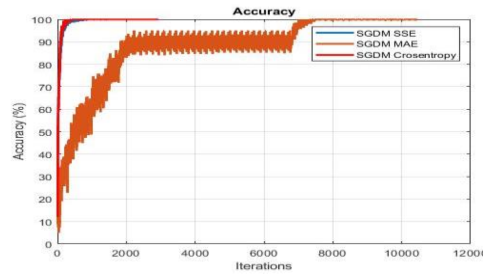

Fig . 6 Accuracy curves of CNN-based classifiers using SGDM optimizer, and MAE-, SSE-, crossentropy- based classification layers**.**

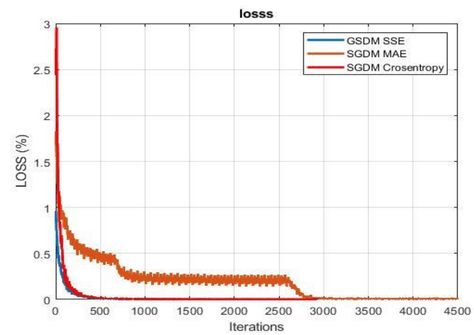

Fig . 7 Loss curves of CNN-based classifiers using SGMD optimizer, and MAE-, SSE-, crossentropy- based classification layers

Fig.8 depict curves of the lowest loss achieved using the proposed classifier for ADAM optimizer with SSE loss function at epochs 20 and batch size 43, and for SGDM optimizer with SSE loss function at epochs 25 and batch size 128.

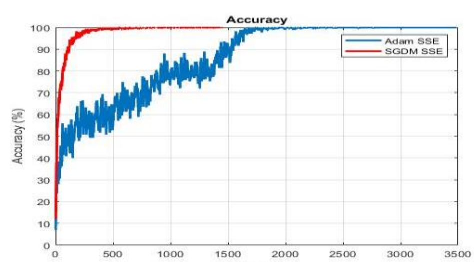

Fig . 8 Loss curves of CNN-based classifiers using Adam, and SGDM optimizers, SSE-based classification layers**.**

And Finally, Fig.9 depict the highest accuracies using the proposed classifier by using chart.

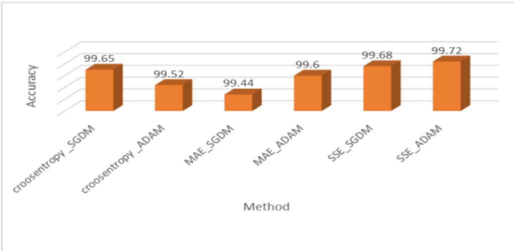

Fig . 9 All the results were represented in the chart

All examined classifiers provide competitive performance. Classifiers using SSE-based classification layer outperform their peers slightly, with true classification accuracies of 99.72% for CNN (Adam, SSE) and 99.68% for CNN (SGDM, SSE).

7 Conclusion

In this paper, the authors have proposed CNN-based classifiers. Two of the presented classifiers make use of MAE- and SSE-based classification layers, which were proposed as new classification layers. Also, the current paper presents a performance study for the proposed classifiers using Adam and SGdm optimizers.
The gotten results showed that the proposed classifiers that use the non-conventional SSE-classification layer and are trained using Adam and SGdm optimizers have achieved accuracy of 99.72% and 99.68%, respectively. The presented results emphasise the importance of developing more loss function-based classification layers and trying different optimizers.

For Future work:

Analyzing the efficiency and precision of the suggested classifiers using various optimizers like Adadelta, Nadam, and Adagrad. Creating more reliable classification layers by using robust statistical cost functions like Huber, Tukey, Welch, and Cauchy. utilising recurrent deep learning neural networks to create more DNN-based classifiers.

References

1. Guo, T., et al. *Simple convolutional neural network on image classification*. in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*. 2017. IEEE.
2. Sornam, M., K. Muthusubash, and V. Vanitha. *A survey on image classification and activity recognition using deep convolutional neural network architecture*. in *2017 ninth international conference on advanced computing (ICoAC)*. 2017. IEEE.
3. Cramer, S., et al., *An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives.* Expert Systems with Applications, 2017. **85**: p. 169-181.
4. Barboza, F., H. Kimura, and E. Altman, *Machine learning models and bankruptcy prediction.* Expert Systems with Applications, 2017. **83**: p. 405-417.
5. Saiharsha, B., et al. *Evaluating performance of deep learning architectures for image classification*. in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*. 2020. IEEE.
6. Hubel, D.H. and T.N. Wiesel, *Receptive fields and functional architecture of monkey striate cortex.* The Journal of physiology, 1968. **195**(1): p. 215-243.
7. Fukushima, K., *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.* Biological Cybernetics, 1980. **36**(4): p. 193-202.
8. LeCun, Y., et al., *Handwritten digit recognition with a back-propagation network.* Advances in neural information processing systems, 1989. **2**.
9. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
10. LeCun, Y., et al. *A theoretical framework for back-propagation*. in *Proceedings of the 1988 connectionist models summer school*. 1988.
11. Deng, J., et al. *Imagenet: A large-scale hierarchical image database*. in *2009 IEEE conference on computer vision and pattern recognition*. 2009. Ieee.
12. Russell, B.C., et al., *LabelMe: A Database and Web-Based Tool for Image Annotation.* International Journal of Computer Vision, 2008. **77**(1): p. 157-173.
13. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**.
14. Russakovsky, O., et al., *ImageNet Large Scale Visual Recognition Challenge.* International Journal of Computer Vision, 2015. **115**(3): p. 211-252.
15. Zeiler, M.D. and R. Fergus. *Visualizing and understanding convolutional networks*. in *European conference on computer vision*. 2014. Springer.
16. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556, 2014.
17. Szegedy, C., et al. *Going deeper with convolutions*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
18. He, K., et al. *Deep residual learning for image recognition*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
19. Huang, G., et al. *Densely connected convolutional networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
20. Sabour, S., N. Frosst, and G.E. Hinton, *Dynamic routing between capsules.* Advances in neural information processing systems, 2017. **30**.
21. Hu, J., L. Shen, and G. Sun, *Squeeze-and-excitation networks, CoRR, vol. abs/1709.01507 (2017).* arXiv preprint arxiv:1709.01507, 2017.
22. Tabik, S., et al., *A snapshot of image pre-processing for convolutional neural networks: case study of MNIST.* 2017.
23. Oliveira, L.S. and R. Sabourin. *Support vector machines for handwritten numerical string recognition*. in *Ninth international workshop on frontiers in handwriting recognition*. 2004. IEEE.
24. Savich, A.W. and M. Moussa. Resource efficient

arithmetic effects on rbm neural network solution quality using mnist. in 2011 International Conference *on Reconfigurable Computing and FPGAs*. 2011. IEEE.

25. Kussul, E. and T. Baidyk, *Improved method of handwritten digit recognition tested on MNIST database.* Image and Vision Computing, 2004. **22**(12): p. 971-981.
26. Cohen, G., et al. *EMNIST: Extending MNIST to handwritten letters*. in *2017 international joint conference on neural networks (IJCNN)*. 2017. IEEE.
27. Yamashita, R., et al., *Convolutional neural networks: an overview and application in radiology.* Insights into imaging, 2018. **9**(4): p. 611-629.
28. Sultana, F., A. Sufian, and P. Dutta. *Advancements in image classification using convolutional neural network*. in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. 2018. IEEE.
29. Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *Icml*. 2010.
30. Wang, T., et al. *End-to-end text recognition with convolutional neural networks*. in *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*. 2012. IEEE.
31. Boureau, Y.-L., J. Ponce, and Y. LeCun. *A theoretical analysis of feature pooling in visual recognition*. in *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
32. Tang, Y., *Deep learning using linear support vector machines.* arXiv preprint arXiv:1306.0239, 2013.
33. Ali, M.H.E. and I.B. Taha, *Channel state information estimation for 5G wireless communication systems: recurrent neural networks approach.* PeerJ Computer Science, 2021. **7**: p. e682.