



# Optimization with ADAM and RMSprop in Convolution neural Network (CNN): A Case study for Telugu Handwritten Characters

B. Soujanya<sup>1</sup>, T. Sitamahalakshmi<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, Institute of Technology, GITAM (Deemed to be University, India, sbuddhar@gitam.edu

<sup>2</sup>Department of Computer Science and Engineering, Institute of Technology, GITAM (Deemed to be University, India, stummala@gitam.edu

## ABSTRACT

Handwritten character recognition is the prime and challenging research area in the area of image processing. This recognition can be defined as the ability of a computer to take the input as handwritten input from various primary sources like paper documents, photographs, touch screens and other devices which are both online and offline. Although a large extent of work has been done in many languages to name a few English and Asian languages such as Japanese, Chinese etc. for handwriting recognition and less amount of work was done on Indian languages like Hindi, Tamil, Telugu, and Kannada etc. This paper explains the methodology for telugu handwritten character recognition algorithm using two optimizers namely Adaptive Moment Estimation (ADAM) and Root Mean Square Propagation (RMSprop) implemented in Convolution neural network (CNN) which has high recognition, accuracy and minimum training and classification time. CNN is dynamic tool to overcome the limitations which had come across in the basic machine learning approaches. To construct the proposed model, we have built our own data set for telugu character (Ka) taking its guninatham as different samples of input. It is observed that RMSprop optimizer outperforms with 89% accuracy compared with ADAM optimizer..

**Key words :** Handwriting recognition, CNN, Telugu, RMSprop, ADAM.

## 1. INTRODUCTION

In the present scenario internet is piled with lots of pictures and video representations which provides ample opportunities for developing various research applications for the analysis [1] of image and videos for showing people about the more advanced content and their procedures. The area of machine learning has taken a substantial step in recent times with the increase of Artificial Neural Network (ANN). These ideas increase the model to go beyond the performance of the

prior forms of artificial intelligence in common machine-learning tasks. One of the inspiring forms of ANN architecture is Convolution Neural Network (CNN). As the ANN performance decreased in the regions of object detection and image classification problems, and with the availability of better CNN, studies using CNN in the areas of image processing has increased substantially [2-4] which gave clarification about many concepts and also provided with the solutions in different ways. CNN's have achieved significant success in the areas of computer vision, natural language processing, speech recognition etc.

CNN is a leading area among multiple machine learning models and its domain is expanded by using some mathematical representations to resolve many visual image applications, object classification and speech recognition. It is a trainable multi-layer network structure which is a composition of many layers [33]. The network can take the input as raw pixel values rather than feature vectors which are generally used in machine learning. There exists an large variants of CNN architecture but however the basic components [5] in the traditional CNN architecture are given by convolutional layer, pooling layer and fully connected layer as shown in figure 1.

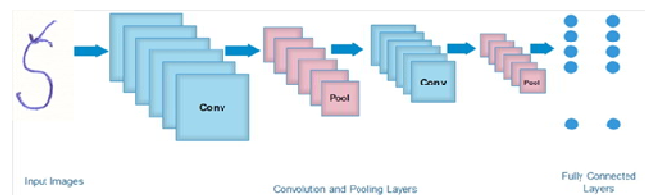


Figure.1: CNN Layers

**1.1 Convolution Layer:** It is the major building block in CNN. It behaves like a filter for images, and detects the characteristics from them, which are used for finding the matched characteristic points of the image during the testing. To traverse over the entire image for a single feature a convolution operation is needed with minimal parameters. In CNN, the convolution is performed on the input data with the use of a filter or kernel then produces a feature map.

**1.2 Pooling Layer:** The features which are extracted are passed to this layer. It takes the larger images and shrinks it while maintaining the most key data in it. It retains the maximum value from each window, by keeping the best fit value obtained within the window. This function gradually reduces the spatial size of the image in minimizing the number of parameters and calculations in the model. In pooling layer, the common approach is Max Pooling. It is an operation which chooses the maximum element from the area of the feature map covered by the filter.

**1.3 Fully Connected Layer:** The last layer is the fully connected layer, in which it inputs the high-level filtered images and renders them into categories with labels. It is also a traditional neural network layer in which every neuron of the previous layer is connected to the present layer. Any typical architecture has recurrences of the stack with many convolutional layers and pooling layers.

**2. Related work**

CNN have a great performance in different applications which inspired the researchers work on it in prime areas like natural language processing (NLP), image classification and Face recognition, predictive analytics etc. P.V. Ramana Murthy *et al.* [6] has developed a model which recognizes online handwritten telugu letters for different domains and organizations by which they developed a model that gives 98.3% accuracy which exceeded their expectations. P. Sujatha *et al.* [7] have identified few deep learning techniques in recognizing telugu and hindi scripts using CNN architecture. They have also proposed a different architecture that identifies low level textual features of the handwritten characters. Buddharaju Revathi *et al.* [8] also made a survey on Optical Character Recognition (OCR) for telugu language which showed the developments of processing the characters stage by stage and performed the operations like segmentation and processing etc. that gave higher accuracy.

B. Hari Kumar *et al.* [9] also performed script identification of telugu language using the data from different sources. Konkimalla Chandra Prakash *et al.* [10] used CNN for the recognition of Telugu script. In their work, a record of Telugu fonts, a client-server solution for the algorithm’s online deployment and deep learning-based OCR algorithms were provided. The segmentation algorithm can be enhanced so that every character is segmented together with its guninatham and vattu. Chirag I Patel *et al.* [11] emphasized a technique to identify the characters in a given digitalized documents and read the changing effects of the Models using ANN by applying the back proliferating neural network to rise the script recognition’s accuracy. A.Ram Bharadwaj *et al.* [12] constructed a model for telugu text extraction and recognition using both CNN and Recurrent neural network (RNN) with their own data set which gave them accuracy of 81% choosing 100 random words from the validation set.

**3. Proposed method**

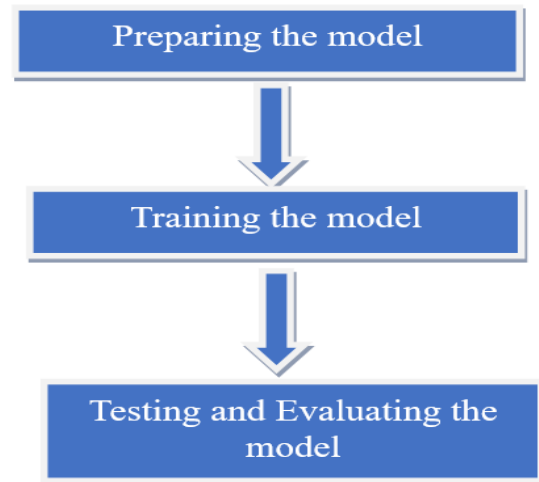
The architecture of CNN is neural network that is extensively used for image-based classification. CNN’s have a number of

filters comprising of trainable parameters which are combined to the image in recognizing the features like edges and shapes. These high-level filters are widely used in capturing spatial attributes of the image which are built on learned weights from the spatial attributes at every ensuing level. This architecture is interpreted by parameters known as Hyper Parameters. These variables affect the network structure and the training of the network, which are regulated before the training process.

Hyper Parameters used for building the network are

- **No. of hidden layers:** Layers in the middle of input and output.
- **Dropout:** It is used to avoid overfitting.
- **Activation Function:** To introduce non linearity.
- **Learning Rate:** It defines how network updates its parameters.
- **Momentum:** It prevents oscillations.
- **Number of Epochs:** One complete cycle of predictions of neural network.
- **Batch Size:** It is the number of sub samples given to the network.

The proposed model is implemented in three different phases as shown in Fig 2.



**Figure 2:** Process Flow

All the images in the build data set are categorized and resized to 70 x 70, converted to grey scale and the data has been saved in .npz format for reading the data at faster rates while training the CNN model. CNN with three hidden layers are chosen to train and test the telugu ka guninatham dataset. The network comprises of 3 convolutional layers along with Maxpooling layers and a fully connected layer. The initial hidden layer is a Convolution2D layer. 256 filters have been used with kernel size of 3×3 each with zero padding and stride equals to 1, followed by the Maxpooling layer with a pool size of 2×2. The architecture employs padding which preserves the original input size.

The next step is the second convolution layer with 256 filters with kernel size 3×3 each followed by Maxpooling layer with a pool size of 2×2. A third convolution layer with 256 filters with kernel size 3×3 each with padding and stride equals to 1 along with a Maxpooling layer of pool size 2×2 has been

implemented. A Flatten layer is used to convert the two-dimensional matrix data into one dimensional vector before building the fully connected layers. Subsequently we have used a fully connected layer with ReLu activation function. Then a regularization layer known as Dropout is configured to arbitrarily eliminate 20% of the neurons in the layer in reducing overfitting. Finally, the output layer having 16 neurons and a sigmoid activation function is implemented. The activation function is an important part of CNN, they define the output of a neuron based on the given set of inputs. The purpose of the activation function is to introduce non-linearity in the model. Choosing a proper activation function improves the performance of a CNN model. The proposed model uses ReLu and Sigmoid activation functions.

**3.1 Rectified Linear Unit (ReLU):** - It is an activation function with strong biological and mathematical support [13]. ReLu returns 0 for any negative input, and the value itself if it is positive (Eq 1). The max operation in ReLu allows it to compute faster than other activation functions. It is generally the default activation function for many types of neural networks.

$$f(x) = \max(0, x) \tag{1}$$

Generally, this is implemented element-wise to the output of some other function, such as a matrix-vector product. This function is required for CNN, since every layer in the network applies a nonlinearity.

**3.2 Sigmoid:** - It is a probabilistic approach towards decision making and ranges between 0 to 1 defined as shown in Eq 2. To predict an output, we have chosen this activation function because the prediction would be more accurate.

$$f(x) = 1/(1+e^{-x}) \tag{2}$$

It is particularly used for the models which predict the probability as an output. Sigmoid function which is differentiable is the correct choice since probability of anything ranges between 0 and 1. So that the use of this function provides the slope of the curve at any two points.

After each epoch, training and validation error have been calculated. Once there is no substantial change in training and validation errors for a pre-set number of epochs, the training stops. Training and validation errors were noted, and the network was estimated on test set. To increase accuracy and to reduce losses the use of optimizers is necessary. Optimizers update the weight parameters to diminish the loss function. The purpose of optimization is to achieve the best design relative to a set of prioritized criteria or constraints. The proposed system used two optimizers, Adaptive Moment Estimation (Adam) and Root Mean Square Propagation (RMSprop).

**3.3 Adaptive Moment Estimation (Adam):** Adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. It stores the exponentially decaying average of the previous squared gradient like momentum, while retaining that of previous historical gradients. It is also a replacement optimization algorithm for stochastic gradient descent systems in

updating the network weights in training data. To estimate the moments, it utilizes exponentially moving averages, computed on the gradients which are evaluated by using the Eq, 3 and 4, Where  $m_t$  and  $v_t$  are moving averages,  $g$  is gradient,  $\beta_1$ ,  $\beta_2$  are gradient forgetting features, second moment of gradients and index  $t$  is the current training iteration.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{4}$$

The recommended Adam configuration parameters[14] of  $\beta_1=0.9$  and  $\beta_2=0.999$ .

**3.4 Root Mean Square Propagation (RMSprop):** This is similar to gradient descent algorithm having momentum, limiting the oscillations to vertical direction. Therefore, by increasing learning rate the algorithm takes substantial steps in the horizontal direction in converging rapidly. It utilizes the magnitude of the recent gradient descents to normalize the gradient. The learning rate gets attuned automatically by choosing different learning rate for individual parameter. The parameters are updated using the Eq. 5, 6, where  $\eta$  is learning rate which holds the value 0.001,  $v_t$  is exponential average of squares of gradients,  $g_t$  is the gradient at time  $t$ .

$$v_t = \rho v_{t-1} + (1 - \rho) * g_t^2 \tag{5}$$

$$\Delta\omega_t = -\frac{\eta}{\sqrt{v_t + \epsilon}} * g_t \tag{6}$$

**4. Experimental results**

**4.1 Data set:**

As there is no publicly available training data for telugu characters to train a classifier, we have generated our own dataset. Initially ka guninatham handwritten is collected from many people in various styles, scanned and classified into different individual symbols. Table 1 shows a sample copy of scanned handwritten ka guninatham. For each letter 128 unique renderings are generated for training and testing purposes. The total images in the built-in data set are 2048 (16 letters x 128 images).

**Table 1:** Notations used in the model


**4.2 Experimental setup:**

The proposed model is executed using Keras library, Tensor flow 2.2 as backend engine. Implementation of CNN was performed on the built-in dataset. The experiments are performed on Intel ® Core (TM) i7-4770 CPU @ 4.00GHz with 16GB RAM 64-bit operating system.

**4.3 Results**

The suggested model is implemented on the built-in dataset. This work focuses on the solution for telugu handwritten character recognition, which shows a powerful implementation of classification. Table 2 shows the layer wise specifications of the suggested model. Total parameters to be handled by the proposed network is 1,986,640.

**Table 2:** Parameters of CNN Model

Layer	Output Shape	Parameters
conv2d_16 (Conv2D)	68, 68, 256	2560
activation_21 (Activation)	68, 68, 256	0
max_pooling2d_	34, 34, 256	0
conv2d_17 (Conv2D)	32, 32, 256	590080
activation_22 (Activation)	32, 32, 256	0
max_pooling2d_	16, 16, 256	0
conv2d_18 (Conv2D)	14, 14, 256	590080
activation_23 (Activation)	14, 14, 256	0
max_pooling2d_	7, 7, 256	0
flatten_6 (Flatten)	12544	0
dense_11 (Dense)	64	802880
dropout_6 (Dropout)	64	0
dense_12 (Dense)	16	1040
activation_24 (Activation)	16	0

The proposed system is evaluated using the metrics like, Recall, Precision and Accuracy [32].

- Precision gives a measure of positive patterns that are suitably predicted from the total predicted patterns in a positive class.

$$Precision = \frac{truepositives}{truepositives + falsepositives}$$

- Recall computes the fraction of positive patterns that are appropriately classified.

$$Recall = \frac{truepositives}{truepositives + falsenegatives}$$

- Classification Accuracy provides the fraction of correct predictions made by the model. This can be calculated by using the given formula.

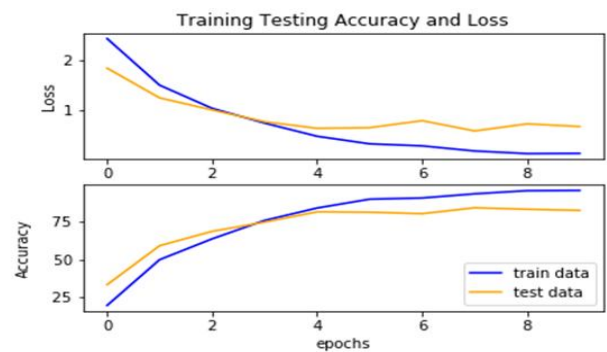
$$Accuracy = \frac{truepositives + truenegatives}{totalexamples}$$

Table 3 shows the values obtained using different optimizers like Adam and RMSprop. RMSprop optimizer produced better result compared to Adam. Using RMSprop, the learning rate is increased as it takes substantial steps in the horizontal direction which converges faster. The loss function quantifies the similarity between predicted output given by the network through forward propagation and specified ground truth labels.

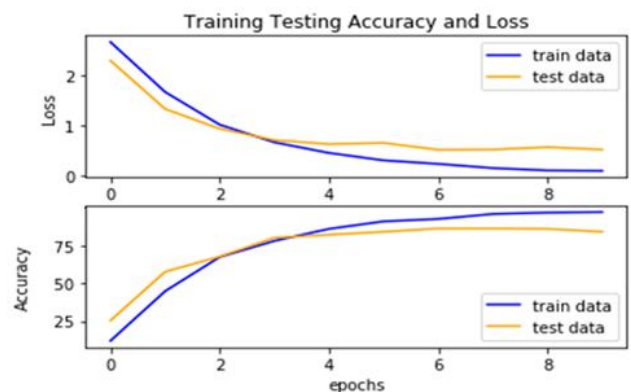
**Table 3:** Different optimizers used in proposed model.

Optimizer	Training Accuracy	Testing Accuracy	Loss	Precision	Recall
Adam	94.5%	84.63%	0.63	0.835	0.812
RMSprop	96.0%	89.0%	0.39	0.895	0.890

Training, testing accuracies and loss versus epochs using Adam optimizer are shown in Fig 3. In the similar manner, training, testing accuracies and loss versus epochs using RMSprop optimizer are shown in Fig 4.



**Figure 3:** Loss and Accuracy using Adam optimizer



**Figure 4:** Loss and Accuracy using RMS prop optimizer



## 5. CONCLUSION

The objective of the paper is to improve the quality of identifying handwritten telugu ka guninatham using CNN. The model gave significant improvement in performance and accuracy with the use of optimizers like Adam and RMSprop. Moreover, it utilizes dropout in avoiding the problem of overfitting along with the activation functions ReLu and Sigmoid. RMSprop optimizer implementation in CNN model gave much better accuracy compared to that of Adam optimizer. This model can be further implemented to enhance the accuracy of identifying handwritten telugu characters. We are also focusing to continue the work in future with other methods to upgrade the accuracy.

## REFERENCES

1. Kou, F., Du, J., He, Y., & Ye, L. “**Social Network Search Based on Semantic Analysis and Learning.**” *CAAI Transactions on Intelligence Technology.* 1 (2016) 293-302
2. Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., & Babu, R. V. (2016) “**A taxonomy of deep convolutional neural nets for computer vision.**” <https://doi.org/10.3389/frobt.2015.00036>
3. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014) “**Object detectors emerge in deep scene cnns.**” <https://arxiv.org/abs/1412.6856>
4. Wang, Y., & Wu, Y. “**Scene Classification with Deep Convolutional Neural Networks.**” (2014)
5. B. R. Kavitha and C. Srimathi, “**Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks**”, *Journal of King Saud University – Computer and Information Sciences*, <https://doi.org/10.1016/j.jksuci.2019.06.004>. (Inpress)
6. P.V. Ramana Murthy, Ch. G.V.N. Prasad “**Recognition of Online Handwritten Telugu Letters For Different Domains And Organizations**” *Journal of Critical Reviews* 6 (2019.) 33-40
7. P. Sujatha, D. Lalitha Bhaskari “**Telugu and Hindi Script Recognition using Deep learning Techniques**” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* 8 (2019) 1758-1764
8. Buddaraju Revathi, G.Naveen Kishore, V Dheera “**A Survey On OCR For Telugu Language**” *International Journal Of Scientific & Technology Research* 8 (2019) 559-562
9. B. Hari Kumar, P. Chitra “**Survey Paper Of Script Identification Of Telugu Language Using OCR**” *International Journal of Electronics and Communication Engineering (IJECE)* 8 (2019) 15-20
10. Prasanta Pratim Bairagi, “**Optical Character Recognition for Hindi**” *International Research Journal of Engineering and Technology (IRJET)* 5 (2018) 3968-3973
11. Chirag I Patel, Ripal Patel, Palak Patel “**Handwritten Character Recognition using Neural Network**” *International Journal of Scientific & Engineering Research* 2 (2011) 1-6
12. A Ram Bharadwaj, A. Venugopal, Ch. Surya Kiran, M. V. Nageswara Rao “**Telugu text extraction and recognition using convolutional and recurrent neural networks**” *International Journal of Engineering and Advanced Technology (IJEAT)* 8 (2019) 1449-1451
13. Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung “**Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit**” *Nature* 405 (2000), 947.
14. Diederik P. Kingma, Jimmy Ba, “**Adam: A Method for Stochastic Optimization**”, arXiv:1412.6980
15. Li Chen, Song Wang, Wei Fan, Jun Sun, Satoshi Naoi, “**Beyond human recognition: A CNN-based framework for handwritten character recognition**” 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (2016), IEEE. 695-699
16. Durjoy Sen, Maitra, Ujjwal Bhattacharya, Swapan K. Parui “**CNN based common approach to handwritten character recognition of multiple scripts**” 2015 13th International Conference on Document Analysis and Recognition (ICDAR) 23 November 2015, IEEE 1021-1025, 10.1109/ICDAR.2015.7333916
17. A Yuan, G. Bai, L. Jiao and Y. Liu, “**Offline handwritten English character recognition based on convolutional neural network,**” 2012 10th IAPR International Workshop on Document Analysis Systems, Gold Coast, QLD, (2012) 125-129, doi: 10.1109/DAS.2012.61.
18. J. Bai, Z. Chen, B. Feng and B. Xu, “**Image character recognition using deep convolutional neural network learned from different languages**” 2014 IEEE International Conference on Image Processing (ICIP), Paris, (2014) 2560-2564, doi: 10.1109/ICIP.2014.7025518.
19. D. C. Ciresan, U. Meier, L. M. Gambardella and J. Schmidhuber, “**Convolutional Neural Network Committees for Handwritten Character Classification,**” 2011 International Conference on Document Analysis and Recognition, Beijing, (2011) 1135-1139, doi: 10.1109/ICDAR.2011.229.
20. T. Matsumoto et al., “**Several image processing examples by CNN**” IEEE International Workshop on Cellular Neural Networks and their Applications, Budapest, Hungary, (1990) 100-111, doi: 10.1109/CNNA.1990.207512.
21. E. Saatici and V. Tavsanoğlu, “**Multiscale handwritten character recognition using CNN image filters,**” *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, Honolulu, HI, USA, (2002) 2044-2048 vol.3, doi: 10.1109/IJCNN.2002.1007454.
22. Meiyin Wu and Li Chen, “**Image recognition based on deep learning,**” 2015 Chinese Automation Congress

- (CAC), Wuhan, (2015) 542-546, doi: 10.1109/CAC.2015.7382560.
23. S. Wang, L. Chen, L. Xu, W. Fan, J. Sun and S. Naoi, "**Deep Knowledge Training and Heterogeneous CNN for Handwritten Chinese Text Recognition**," 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, (2016) 84-89, doi: 10.1109/ICFHR.2016.0028.
  24. Li, Z., Teng, N., Jin, M. et al. "**Building efficient CNN architecture for offline handwritten Chinese character recognition.**" IJDAR 21 (2018) 233–240
  25. K. Chandra Prakash, Y. M. Srikar, G. Trishal, S. Mandal and S. S. Channappayya, "**Optical Character Recognition (OCR) for Telugu: Database, Algorithm and Application**" 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, (2018) 3963-3967, doi: 10.1109/ICIP.2018.8451438.
  26. N. B. Muppalaneni, "**Handwritten Telugu Compound Character Prediction using Convolutional Neural Network**" 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, (2020) 1-4, doi: 10.1109/ic-ETITE47903.2020.349.
  27. Mrs. Safna K M, "**English Handwritten Character Recognition using Convolutional Neural Network (CNN)**", IJSRD - International Journal for Scientific Research & Development 6 (2018) 3391-3398.
  28. F. P. Such, D. Peri, F. Brockler, H. Paul and R. Ptucha, "**Fully Convolutional Networks for Handwriting Recognition**" 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, (2018) 86-91, doi: 10.1109/ICFHR-2018.2018.00024.
  29. K. Dutta, P. Krishnan, M. Mathew and C. V. Jawahar, "**Towards Spotting and Recognition of Handwritten Words in Indic Scripts**" 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, (2018) 32-37, doi: 10.1109/ICFHR-2018.2018.00015.
  30. Zhang, Zijun. "**Improved Adam optimizer for deep neural networks.**" In 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), pp. 1-2. IEEE, 2018.
  31. Neha Sharma, Vibhor Jain, Anju Mishra "**An Analysis Of Convolutional Neural Networks For Image Classification**" Procedia Computer Science 132 (2018) 377–384
  32. Praveen Kumar Kollu, R. Satya Prasad "**Intrusion Detection System Using Recurrent Neural Networks and Attention Mechanism**", International Journal of Emerging Trends in Engineering Research 7(2019) 178-182
  33. Artha Andriyanto, Antoni Wibowo NS Norhaslinda Zainal Abidin "**Sectoral Stock Prediction Using Convolutional Neural Networks with Candlestick Patterns as input Images**" International Journal of Emerging Trends in Engineering Research 8(2020) 2249-2252