Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 4, July 2021: 1493-1503

DEEP LEARNING BASED CONVOLUTIONAL NEURAL NETWORK FOR HANDWRITING RECOGNITION

Keerthi Prasad G[1], Asha K[2], Naveen R Chanukotimath[3], Dr. Manjula G M[4]

[1,2,3]Assistant Professor, Department of ISE, GMIT
[4]Assistant Professor, Department of Mathematics, GMIT
keerthiprasad@gmit.ac.in / ashak@gmit.ac.in / naveenrcm@gmit.ac.in / manjulagm@gmit.ac.in

ABSTRACT
Handwriting is a taught skill that has been used for thousands of years as a way of communication and recording. Natural handwriting is a fairly simple way for computers and humans to exchange information. Because of their complex typing nature, it is difficult to input data for computers using Indian language scripts.This research investigates the use of convolutional neural networks based on deep learning for the recognition of handwritten script. The proposed technique has demonstrated 75 percent recognition accuracy on average for handwritten English numerals and shows promise for Kannada numerals.

Keywords: CNN, Deep learning, handwriting recognition, machine learning, neural network, pattern recognition, Tensor Flow.

I. INTRODUCTION

Computers take raw data and human instructions and turn them into useful information. The raw data and instructions are entered through a variety of input devices, the most common of which is the keyboard. The normal keyboard is primarily built for Latin characters, and we can input a character by pressing a single key. Handwritten languages with a larger number of characters than Latin characters, on the other hand, should require a combination of keys to enter a single character.

Inputting data through handwritten papers and speech are two quick and natural ways for users to communicate with computers. Speech recognition has limitations in noisy environments, particularly when an individual's privacy is desired.As a result, handwriting recognition systems are in demand. While substantial research has been done in this area for foreign languages such as English, little work has been done for Indian languages due to hardware limitations in accepting a huge character set of Indian scripts.

In the field of handwriting identification, machine learning models are gaining traction, and one such model is the deep learning-based neural network. The technical specifics of employing a deep learning methodology for Kannada script are explored in this study.

II. RELATEDWORK

Asha K, Krishnappa H K [1] developed a Kannada handwriting recognition system, and documents from ICDAR-2013 and ICDAR-2015 were used in the experiment, with 100 percent line segmentation accuracy and 96 percent character segmentation accuracy.

Asha K, Kiran Kumar Gowda R, Diksha M M, Soniya B S [2] This paper describes the technical aspects of the design of a Kannada handwriting recognition system. Tensorflow - an open source machine learning framework – is used to demonstrate the stages involved in developing a handwriting recognition engine.

Keerthi Prasad G, Vinay Hegde[3] The performance of Principal Component Analysis (PCA) and Dynamic

Time Wrapping (DTW) techniques for recognizing online handwritten isolated Kannada letters is the subject of this paper. The writer independent model suggested in this paper identifies fundamental 50 Kannada characters, comprising 16 vowels and 34 consonants.

Asha K, Krishnappa H K [4] proposed a system for the recognition of handwritten script, a system based on convolutional neural networks (CNN) and deep learning was presented. The proposed technique has demonstrated 99 percent recognition accuracy for handwritten English numerals and shows promise for Kannada numerals.

Asha K, Krishnappa H.K. [5] the authors provide technical details on the suggested design and implementation. The implementation is based on the Convolutional Neural Network (CNN) model, which was trained on the Chars74K dataset. For a handwritten document with non-overlapping lines of characters, the system achieved 98 percent accuracy.

AnirudhGanesh*etal.*,[6] Deep Learning Approach for Recognition of Handwritten Kannada Numerals has been proposed. The authors used the LeNet model for Convolutional Neural Networks and Deep Belief Network models to investigate the performance of Convolutional Neural Networks and Deep Belief Networks in classifying Handwritten Kannada numerals from the Chars74 k dataset.They found that convolutional neural networks converge faster than deep belief networks, with an accuracy of roughly 97.76 percent using CNNs and a similar accuracy of 98.14 percent using DBNs.

PardeepKumar*etal.*,[7]Handwriting Recognition using Tensor Flow and Convolutional Neural Networks was proposed for handwritten numerical recognition. The suggested approach extracts features using CNN and achieves a 98 percent accuracy rate.

Salma Shofia Rosyda and TitoWaluyoPurboyo[8] "A Review of Various Handwriting Recognition Methods" was presented as a paper. Convolutional Neural Network (CNN), Semi-Incremental Segmentation, Incremental, Lines and Words, Parts, Slope and Correction Slant, Ensemble, and Zoning are the eight approaches discussed in this study. The benefits and drawbacks of each of the eight ways are discussed, and it is determined thatThe time required for extensive training owing to the Convolutional Neural Network (CNN) is included in the deep learning study, yet CNN has good accuracy for handwriting identification because more CNN training will result in more accurate writing recognition.

Darmatasia and Mohamad Ivan Fanany [9] For English, they presented “Handwriting Recognition on Form Document Using Convolutional Neural Networks and Support Vector Machines.” Both the training and testing datasets for this project were acquired from NIST SD 19 2nd edition. Convolutional Neural Network (CNN) as a powerful feature extraction and Support Vector Machines (SVM) as a high-end classifier are used in this learning model.Their proposed technique achieves a recognition rate of 98.85% for numeric characters, 93.05 percent for uppercase characters, 86.21 percent for lowercase characters, and 91.37 percent for numeric and uppercase characters combined. On 10 separate test form documents, the system achieves an accuracy of 83.37 percent.

Batuhan Balci*etal.*,[10]For English, Deep Learning Handwritten Text Recognition was proposed. Convolutional Neural Networks (CNNs) are used for classification, and LSTMs with convolution are utilized to create bounding boxes for character segmentation.The IAM Handwriting Dataset is used for training, and it contains handwritten text from over 1500 forms (a form is a piece of paper containing lines of text) from over 600 authors, totaling 5500 sentences and 11500 words.All associated form label metadata is provided in related XML files, and the words were then segmented and carefully validated. For word-level classification, experiments are undertaken with four different models: VGG-19, RESNET-18, and RESNET-34, and for character classification, experiments are conducted with the Char-Level model. VGG-19 has a 20% accuracy rate, RESNET-18 has a 22% accuracy rate, RESNET-34 has a 25% accuracy rate, and Char-Level has a 31% accuracy rate.

III. DEEPLEARNING

Deep Learning is a new branch of Machine Learning research that aims to bring the field closer to one of its primary goals: Artificial Intelligence. Deep learning has had a comeback in recent study, and it has been proved to give state-of-the-art outcomes in a variety of applications.Deep learning algorithms are built in a hierarchy of increasing complexity and abstraction, unlike typical machine learning algorithms, which are linear.

The learning process in classical machine learning is supervised, and the success rate is entirely dependent on the programmer's ability to appropriately construct a feature set for classification.The benefit of deep learning is that it allows the software to create the feature set on its own, without the need for human intervention. Unsupervised learning is not only faster, but also more accurate in most cases. Figure 1 depicts the generalized stages of the Deep Learning process.

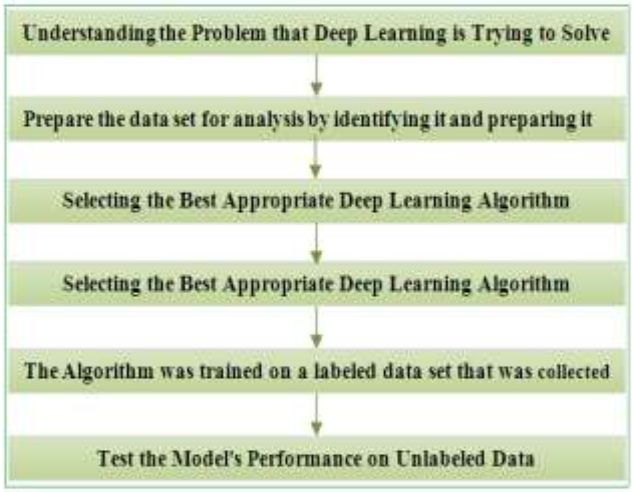

Figure 1.Stages in the Process of Deep Learning

Stage 1: The first stage is to determine whether or not the given problem can be solved using a deep learning methodology. Because deep learning is not used to solve all of the problems.

Stage 2: The next stage is to find and prepare suitable data sets for analysis. The success rate of deep learning is mostly determined by the data set used to train the model.

Stage 3: Choosing the deep learning architecture to be used. Deep neural networks, deep belief networks, and recurrent neural networks are among the deep learning architectures that can be chosen.

Stage4: After you've chosen a deep learning architecture, you'll need to train the model with a huge amount of labeled data.

Stage 5: Test the trained model with unlabeled test input in real time to see how well it performs in terms of proper classification accuracy.

III. CONVOLUTIONAL NEURAL NETWORK

Recurrent neural networks, convolutional neural networks, artificial neural networks, and feed forward neural networks are some of the numerous types of neural networks, each with its own set of benefits for certain use situations.They all work in a similar way, by putting data into the model and allowing it to determine whether it has made the correct interpretation or judgment about a given data element.

For image classification problems, convolutional neural networks (CNNs) are the current state-of-the-art model architecture.CNNs use a series of filters to extract and learn higher-level features from an image's raw pixel data, which the model can then use for classification.Figure 2 shows the three components of a CNN.

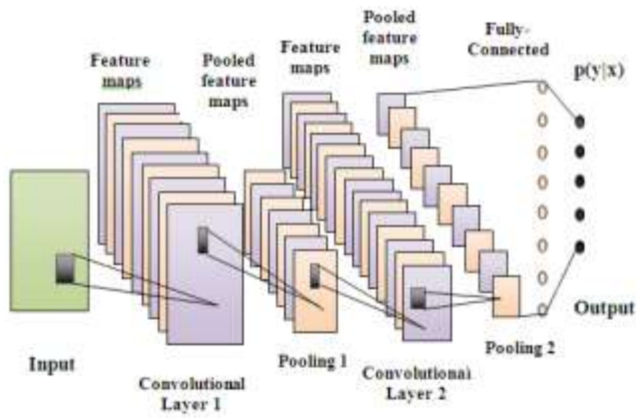

Figure 2. Convolutional Neural Network

Convolutional layers are used to apply a set of convolution filters on an image. The layer executes a sequence of mathematical operations for each subregion to produce a single value in the output feature map. To inject nonlinearities into the model, convolutional layers often apply a ReLU activation function to the output. The activation function in a neural network is responsible for converting the node's summed weighted input into the node's activation or output for that input.

The rectified linear activation function, or ReLU, is a piecewise linear function that, if the input is positive, outputs the input directly; else, it outputs zero. Because a model that utilizes it is quicker to train and generally produces higher performance, it has become the default activation function for many types of neural networks.

It is mathematically defined as y = max (0, x). Visually it appears as in Figure 3.

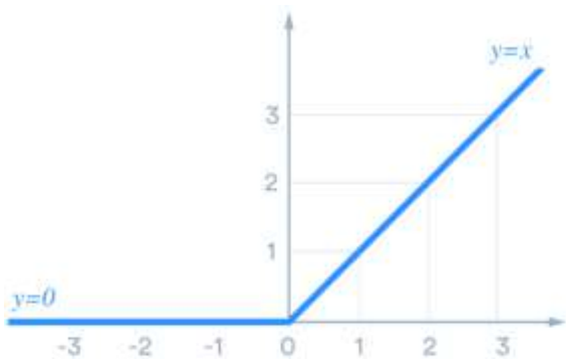

Figure 3. The ReLU activation function is visualized.

Pooling layers, After the convolutional layer, a new layer called a pooling layer is introduced. After a nonlinearity (such as ReLU) has been applied to the feature maps produced by a convolutional layer, this down samples the picture data produced by the convolutional layers in order to lower the dimensionality of the feature map and hence speeds up processing.

The pooling operation entails sliding a two-dimensional filter across each channel of the feature map and aggregating the features that fall inside the filter's coverage zone.

The dimensions of output received after a pooling layer for a feature map with dimensions $\mathbf{n_h \times n_w \times n_c}$ are given in equation (1).

$$(n_h - f + 1) / s \times (n_w - f + 1)/s \times n_c \qquad (1)$$

Where,

n_h - height of feature map

n_w - width of feature map

n_c - number of channels in the feature map

f - Size of filter

s - Stride length

The feature maps' dimensions are reduced by using pooling layers. As a result, the number of parameters to learn and the amount of processing in the network are both reduced.

The features contained in a region of the feature map generated by a convolution layer are summed up by the pooling layer. As a result, rather than precisely positioned features created by the convolution layer, following actions are conducted on summarized features. As a result, the model is more resistant to changes in the position of features in the input image.

Max Pooling, shown in Figure 3. Pooling that selects the maximum element from the region of the feature map covered by the filter is known as max pooling. As a result, the output of the max-pooling layer would be a feature map with the most prominent features from the previous feature map.

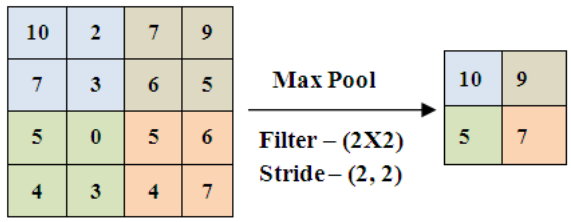

Figure 3. Max Pooling

Average Pooling, shown in Figure 4. The average of the items present in the region of the feature map covered by the filter is computed using average pooling. As a result, while max pooling returns the most prominent feature in a feature map patch, average pooling returns the average of all features present in that patch.

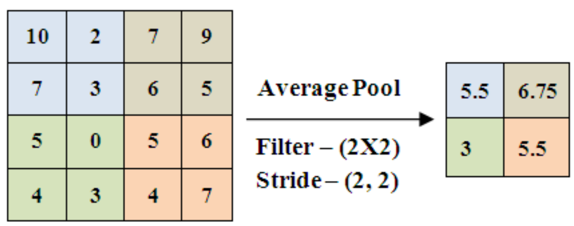

Figure 4. Average Pooling

Dense (fully connected) layers, this uses the features retrieved by the convolutional layers and down sampled by the pooling layers to achieve classification. Every node in a dense layer is connected to every other node in the preceding layer.The dense layer operation is a linear operation in which each input is coupled to each output by a weight, resulting in a weight of n inputs * n outputs generally followed by a non-linear activation function.

CNN is made up of a series of convolutional modules, each of which does feature extraction and includes a convolutional layer and a pooling layer. One or more dense layers conduct categorization after the last convolutional module.

A single node for each target class in the model is found in the final dense layer of a CNN, which uses a softmax activation function to generate a value between 0 and 1 for each node.

IV. OUR APPROACH

Proposed system architecture of deep learning based convolutional neural network model is shown in Figure 5. For training MNIST dataset is used, it consists of 70000 handwritten English numeral samples out of which 60000 samples are used for training and 10000 for testing.Similarly, Chars74K dataset consists of 25 samples for each 657 classes of Kannada character set. For testing handwritten document is used.

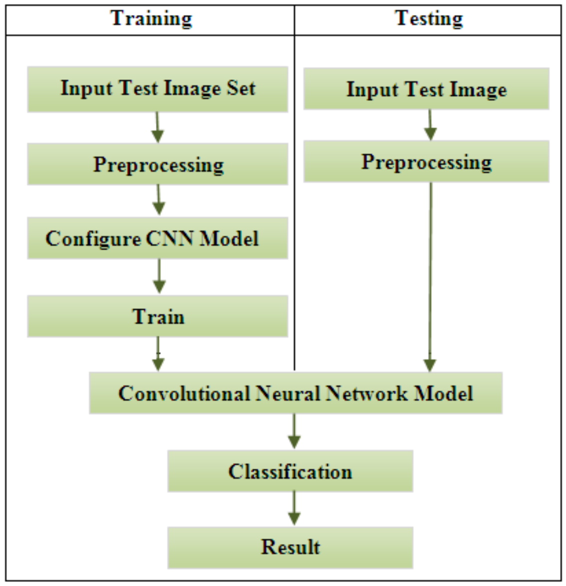

Figure 5. System Architecture

In pre-processing size normalization and segmentation is not needed for MNIST and Chars74K dataset images, they are normalized isolated characters which are used for training. But for testing, image of the handwritten document is given as input, here preprocessing is needed in order to remove the noise from the input test image. Preprocessing will increase the recognition accuracy.

The color image of the handwritten document is transformed to a black and white image called a binary image. This conversion is called threshold segmentation, in which pixels with grey levels below a certain value are converted to white and those above that value are converted to black. Threshold segmentation is given in equation (2), and Figure 6 shows the procedure for choosing the threshold values.

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \leq T \end{cases} \quad (2)$$

.

Where

- g(x, y) at some global threshold T is a thresholded variant of f(x, y).
- If T can change over time in the image, variable thresholding can be used.
- If T is dependent on a community, local or regional thresholding can be used
- If T is a function of (x, y), adaptive thresholding is used

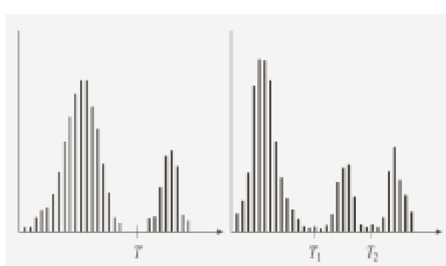

Figure 6. Choosing the Threshold values

The image histogram's peaks and valleys will aid in determining the acceptable threshold values

The suitability of the histogram for directing the selection of the threshold is influenced by a number of factors:

- the distance between peaks;
- the amount of noise in the image;
- the items' and background's relative sizes;
- the illumination's consistency;
- the reflectance's uniformity.

Configuration of CNN model

Following is the Configuration of CNN model used to classify the images in MNIST dataset.

Convolutional Layer #1: Applies 32 5x5 filters (extracting 5x5-pixel sub regions), with ReLU activation function.

Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 2.

Convolutional Layer #2: Applies 64 5x5 filters, with ReLU activation function.

Pooling Layer #2: Again, performs max pooling with a 2x2 filter and stride of 2.

Dense Layer #1: 1,024 neurons with drop out regularization rate of 0.4

Dense Layer #2 (Logits Layer): 10 neurons, one for each digit target class (0–9).

Table 1 shows the Configuration of CNN model used to classify the images in Chars74K dataset.

The input are 64x64 grey scale images 4 convolution layers with filter size 3x3 and ReLU activations. Max pooling layers after every other convolution layer2 hidden layers with drop out and Softmax output.

Table1.CNNConfiguration

LayerType	Parameters
Input	size:64x64,channel:1
convolution	kernel:3x3,channel:128
ReLU	
convolution	kernel:3x3,channel:128
ReLU	
maxpool	kernel:2x2
convolution	kernel:3x3,channel:256
ReLU	
convolution	kernel:3x3,channel:256
ReLU	
maxpool	kernel:2x2
convolution	kernel:3x3,channel:512
ReLU	
convolution	kernel:3x3,channel:512
ReLU	
maxpool	kernel:2x2
fullyconnected	units:2048
ReLU	
dropout	0.5
fullyconnected	units:2048
ReLU	
dropout	0.5
softmax	units:62

V. EXPERIMENTANDRESULT

The experiment is conducted on Colaboratory. Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to useand runs entirely in the cloud. Colaboratory notebooks are stored in Google Drive and can be shared just as you would with Google Docs or Sheets. Colaboratory is free to use.

Colaboratory supports both Python2 and Python3 for codeexecution.

- When creating an ewnotebook, you'll have the choice between Python2 andPython3.
- We can also change the language associated with a notebook; this information will be written into the .ipynb file itself, and thus will be preserved forfuturesessions.

Colab also supports connecting to a Jupyter runtime on your local machine.

GRID K520 GPU was used in the cloud instance. Figure 7 shows the home page of Colaboratory. Figure 8 and Figure 9 shows the Jupyter notebook and result obtained respectively.

Figure 7. Colaboratory Home Page

Figure8. JupyterNotebook

Figure9. Result obtained

Figure 10, Figure 11 and Figure 12 shows the output obtained for image of the handwritten document.

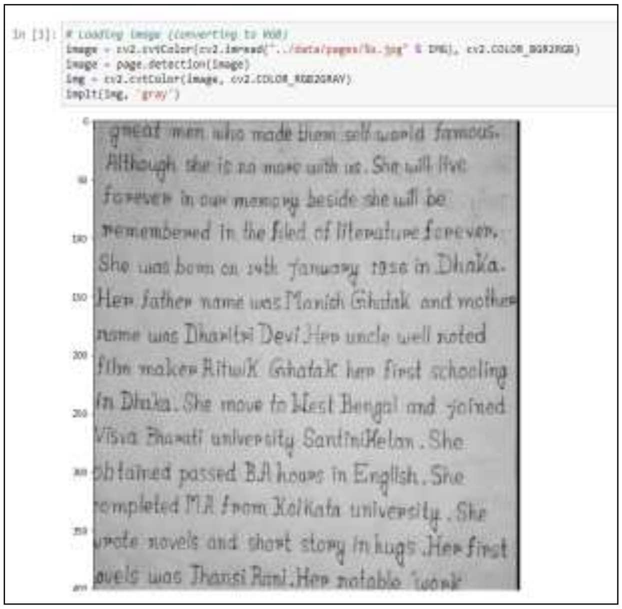

Figure10. Scanned copy of the Image

Figure 11. Binary Inverted Image

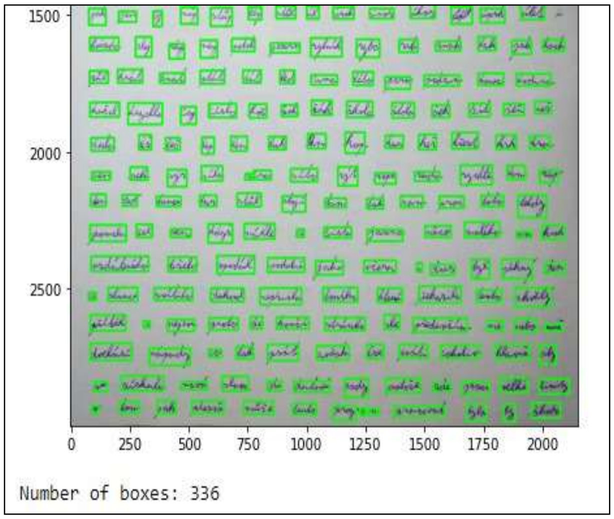

Figure12. Extracting Words.

Table 2 and Table 3 show the performance evaluation of CNN model and Performance evaluation of feature selection algorithm on MNIST dataset respectively.

Table 2.Performance evaluation of CNN Model in character recognition on MNIST Dataset

	Accuracy (%)	**Sensitivity (%)**	**Specificity (%)**	**F-score (%)**
Kannada	73.24	75.49	73.51	68.96
English	46.48	50.39	44.72	45.34

Table 3.Performance evaluation of feature selection algorithm in character recognition on MNIST Dataset

	Accuracy (%)	**Sensitivity (%)**	**Specificity (%)**	**F-score (%)**
Kannada	68.53	68.1	76.57	68.53
English	69.45	69.45	69.45	69.45

VI. CONCLUSION

Handwriting recognition system for South Indian languages like Kannada, Tamil, Telugu and Malayalam are gaining much attraction in the field of research. Deep learning approach discussed, obtained 75% accuracy for English numerals and can be applied to entire Kannada character set of Chars74K with change in CNN configuration accordingly. This paper has more technical details which is very useful for implementation of both offline and online handwriting recognition system for South Indian languages. We hope this work encourages the research in the field of Indian scripts.

REFERENCES

[1] Asha K, Krishnappa H K, “Segmentation of Characters in a Handwritten Kannada Document” International Journal of Sensors, Wireless Communications and Control, Volume 11, Issue 3, 2021

[2] Asha K, Kiran Kumar Gowda R, Diksha M M, Soniya B S, “Design of handwriting recognition system for Kannada characters using tensorflow” International Journal Of Current Engineering And Scientific Research (IJCESR), ISSN (PRINT): 2393-8374, (ONLINE): 2394-0697, VOLUME-8, ISSUE-3, 2021

[3] Keerthi Prasad G, Vinay Hegde “Recognition of Online Handwritten Isolated Kannada Characters using PCA and DTW” International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019

[4] Asha K, Krishnappa H K, ” Handwriting Recognition using Deep Learning based Convolutional Neural

Network” International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019

[5] Asha K, Krishnappa H.K. “Kannada Handwritten Document Recognition using Convolutional Neural Network” 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS)

[6] Anirudh Ganesh, Ashwin R. Jadhav, K.A. CibiPragadeesh, “Deep Learning Approach for Recognition of Handwritten Kannada Numerals”, Springer International Publishing AG 2018, Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition.

[7] Pardeep Kumar, Sai Akhi, Nandanavanam, M Mounika, Ch Goutham Reddy, P Nikhlender Reddy, “Handwriting Recognition using TensorFlow and Convolutional Neural Networks”, IJRASET, Vol. 6,issue IV, Apr 2018, pp 901-903.

[8] Salma ShofiaRosyda and Tito WaluyoPurboyo, “A Review of Various Handwriting Recognition Methods” International Journal of Applied Engineering Research, Vol. 13, 2018, pp 1155-1164.

[9] Darmatasia and Mohamad Ivan Fanany, “Handwriting Recognition on Form Document Using Convolutional Neural Network and Support Vector Machines”, CoICT.2017.

[10] BatuhanBalci, Dan Saadati and Dan Shiferaw, “Handwritten Text Recognition using Deep Learning”.Yu Hu, Li Chen, Jun Cheng, “A CAPTCHA recognition technology based on deep learning”, ICIEA 2018.

[11] Yoshihiro Shima, Yumi Nakashima, Michio Yasuda, “Classifying for a Mixture of Object Images and Character Patterns by Using CNN Pre- trained for Large-scale Object Image Dataset”, ICIEA 2018.

[12] Bappaditya Chakraborty. Bikash Shaw, Jayanta Aich, “Does Deeper Network Lead to Better Accuracy: A Case Study on Handwritten Devanagari Characters” IAPR International Workshop on Document Analysis Systems, 2018.

[13] Keerthi Prasad G. Vinay Hegde, Asha K, Krishnappa H.K. “Handwriting Recognition System for South Indian Languages - A Technical Review” 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)

[14] Bappaditya Chakraborty. Bikash Shaw, Jayanta Aich, “Does Deeper Network Lead to Better Accuracy: A Case Study on Handwritten Devanagari Characters” IAPR International Workshop on Document Analysis Systems, 2018.

[15] Naveen R Chanukotimath, Feroz Khan, Keerthi Prasad G, Imran Khan, Deepak D J, Nasreen Taj M B, “Dvadasham (Dodeca) Edge Filter for Impulse Noise, Gaussian Noise Quantum Noise Reduction in Images”, Compusoft, An International Journal of advanced computer technology, July2013,Volume-I