RECOGNITION OF HANDWRITTEN MODI DIGITS AND CHARACTERS BY USING DEEP LEARNING ALGORITHM.

Jidnyasa Kondhare
Department of
E&TC, Smt.Kashibai Navale
College of Engg.Pune
kondharejidnyasa@gmail.com
radhikakaldate99@gmail.com

Prof.V.A.Yaduvanshi
Department of
E&TC, Smt.Kashibai Navale
College of Enggi. Pune.
Vijaya.yaduvanshi_skncoe@sinhgad.edu

Ruchira Patil
Department of
E&TC, Smt.Kashibai Navale
College of Enggi.Pune.
ruchirapatil22@gmail.com

Radhika Kaldate
Department of
E&TC,Smt.Kashibai Navale
College of Enggi.Pune

***Abstract*— Recognition of characters is simple for humans, but itis quite complex to build a software that can recognize characters. MODI is one of the ancient language. We intend to develop such a character recognition system capable to recognize printed and handwritten MODI characters at an electronic speed by scanning the documents. It is a challenging issue to develop a practical Character Recognition System which can maintain a high recognition accuracy without concerning the quality of input documents. We have developed MODI Script Character Recognizer System (MSCR) using Convolutional Neural Network (CNN) algorithm. We have considered 48 different characters of MODI script including vowels and consonants and 10 numerals to train the system by using CNN. The sample data set (training images) of the characters is maintained by taking handwritten samples from different peoples.**

Keywords- CNN, Vgg16 ,digit and char.recognization,MSCR.

INTRODUCTION

'MODI' is an ancient script as compared to other ancient Indian languages. MODI script was used for writing purposes only, which was a cursive type of writing in 'Marathi' (the primary language of Maharashtra state in western India); there are several theories about the origin of this script. One of them claims that in 12^{th} Century MODI was developed by 'Hemadpant' or 'Hemadri.' In Human Society, the dominant information medium documents. Hence document image analysis is an important research area in the field of image processing and pattern recognition. The world as we know it today is very fast-moving and highly automated. Technology has become synonymous with automation. All this is because we, humans, tend to do our job faster and in the most efficient way. Hence the more we automate, the easier and quicker our work becomes. The next trend in today's fast-changing world is digitalization. Since this is the age of computers, we want all information available to be digitized and stored in the computers since they have faster computing capabilities. But the problem in digitizing real-world information into the digital domain is we need to teach the computer specifically about our concerned real-world data.

The topic of our discussion here is the Handwritten Character Recognition (HCR) technique. HCR has been in quite a few research works in the last decade, and it is getting much more attention day by day. It is essential because we can make the computer learn and recognize the regional languages pretty well. If we do that, then it opens a whole new world of endless possibilities. Most of the works about HCR pertain to the recognition of handwritten characters, which can serve as the input for the computer directly. Research has been done on common languages such as English, French, Hindi, and other foreign languages such as Chinese, Japanese, etc.

For an offline character recognition system, the input image is first acquired from the scanner. As we obtained the scanned digital image, the second step is preprocessing. It essentially enhances the image rendering it suitable for segmentation. Erosion, dilation, opening, closing, smoothing operations are used for preprocessing. The binarization process converts a grayscale image into a binary image using the global thresholding technique. Detection of edges in the binarized image using Sobel technique, dilation of the image, and filling the holes present in it are the operations performed in the last two stages to produce the preprocessed image suitable for segmentation.

In the segmentation, the input image is segmented into individual characters, and then, each character is resized into m x n pixels towards the training network. The features will be extracted from Boundary tracing and their Fourier Descriptors. The character will be identified by analyzing its shape and comparing the characteristics that distinguish each character In the present scenario, more importance is given to the "paperless office," thereby more and more communication and storage of documents are performed digitally. Documents and files that were once stored physically on paper are now being converted into electronic form to facilitate quicker additions, searches, and modifications and prolong the life of such records. Because of this, there is a great demand for software, which automatically extracts, analyze, recognize and store information from physical documents for later retrieval. One of the essential steps of document processing is Textual processing through an Optical Character Recognizer (OCR).

LITERATURE SURVEY

I. Besekar D.N.[1] stated a morphological approach for recognition of five numerals from MODI script. Feature set was created by using blobs, vertical lines, horizontal lines, concavities present in the numerals. Recognition and classification was done by using decision tree and mathematical morphology. In this case 75% recognition rate was achieved for 0 and 1 and lower for 4,7 and 9.

II. Chain code and image centroid based recognition model was presented by Besekar D.N [2] for vowels in MODI script. In this work median filter was used for removing noise, global threshold for binarization, flood fill for avoiding boundary breaks and size normalization. Two layer feed forward Neural Network and SVM was used for classification and achieved 65.3% to 73.5% recognition rate.

III. Structural similarities was used by Ramteke A.S. & Katkar G.S [3] for recognition of MODI characters. Structural similarity approach was used for measuring image quality and image quality metrics. measured structure similarity (SSIM), KNN and back propagation NN was used for classification and achieved 91 to 97% recognition rate.

IV. Zone based approach was used by Besekar D.N. & Ramteke R.J. [4] for recognition of offline handwritten numerals. In this study preprocessing was done by using median filter for removing noise, global threshold for binarization, flood fill for avoiding boundary breaks and size normalization. Feature set was created with the help of four equal square zones of size 15×15 and their polar coordinate, Variance, Theta angle and Rh distance. Using variance table for classification this study achieved 93.5% recognition rate.

V. Theoretical analysis of MODI Script recognition was done by Besekar D.N. & Ramteke R.J. [5]. Devanagari, MODI and Roman scripts were compared in this work and found that structural features were difficult to extract for MODI script. In this work internal and external segmentations were discussed and advised Internal segmentation for MODI script. In this analysis structural as well as topological features were suggested. This study also explains that HOCR for MODI script was a difficult task as compared to other handwritten script due to the cursive nature, variations in character, handwriting habits and synonymous structure of characters.

VI. In [6], the implementation of CNN autoencoder as a feature extractor is proposed for the character recognition of the MODI script. The is feature set size was reduced from 3600 to 300 with CNN autoencoder. The extracted features were then subjected to classification using SVM. An accuracy of 99.3 % reported, which is better compared to any other reported accuracy of MODI script character recognition. The main contribution of this work is the achievement of high accuracy in MODI character recognition.

VII.

VIII. Parag A. Tamhankar [7] presented in this paper tackles segmentation of individual characters from ancient handwritten MODI Script documents. Vertical Projection Profile (VPP) method can segment characters from a given line accurately only when two adjacent characters are separated by a zero-pixel column. Since characters of one line in this script are written without lifting hand, output of VPP does not suffice. This paper presents a novel approach for isolating individual characters from the line obtained using authors' previous research work which uses dual thresholding criterion to minimize the character segmentation error. The techniques used in this paper are straightforward and reasonably economical in terms of execution time efficiency.

IX. Savitri Chandure & Vandana Inamdar [8] comprises the creation of an image dataset for MODI handwritten characters and the development of a supervised Transfer Learning (TL)-based classification framework. It makes use of Deep Convolutional Neural Network (DCNN) Alexnet as a pre-trained network to transfer weights to retrain the network. This network is used as a feature extractor to extract features from different layers of the network. A Support Vector Machine (SVM) is trained on activation features to obtain classifier models. These models are investigated further for recognition accuracy and feature analysis. Subjective and objective measures are used to select discriminant deep features. We achieved recognition accuracies of 92.32% and 97.25% for Handwritten MODI character recognition and handwritten Devnagari character recognition, respectively.

X. Manisha Deshmukh et al. [9] present a system for recognition of off-line handwritten Modi Numerals is presented. To extract the features of handwritten Modi numeral chain code feature extraction technique is used with non-overlapping blocking strategy. A correlation coefficient is used for Modi numeral recognition. Experimental results are evaluated using two strategies: different numeral image non-overlapping division and different sizes of data set. On experimentation the maximum recognition rate of 85.21% is achieved on a database of 30000 images. The recognition results shows better performance for 5X5 grids divisions.

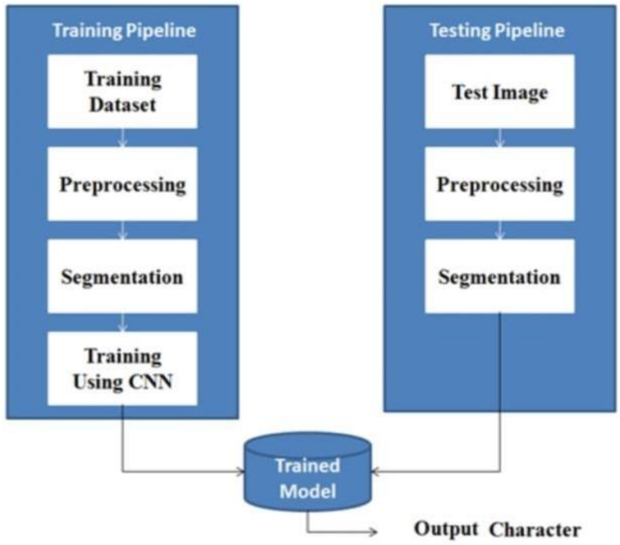
XI. Snehal R. Rathi et al [10] mentioned about the various steps in the image processing techniques in order to convert the Modi characters into English. Several important documents written in 'Modi' language still remain in vegetative state. These documents have priceless data and information. They can be of great help if they are successfully decoded. The problem of Modi OCR and handwriting recognition is a challenging job, and experts try hard to interpret these issues and fabricate potential answers to these issues. A large number of issues still remain to be solved and active research in this area is required to take this potential problem to useful levels, when product using the solution would become available to common man.

XII. Sanjay S. Gharde et al. [11] presents the Identification and Recognition of the handwritten MODI characters are performed. The database of handwritten samples is prepared by using ANESP program. The original manuscript of MODI script is collected and it is preprocessed. Moment Invariant and Affine Moment Invariant are the two techniques which are used for extracting the features from the handwritten segregated samples. Identification and recognition is done using machine learning technique. Among the variants of machine learning techniques, support vector machine is used as a classifier. Linear kernel function is applied in this support vector machine while performing classification. This system provides good recognition rate for the handwritten samples of MODI script. This research will be helpful to the researchers to enlighten the uncovered part of history.

XIII. B. Solanki et al [12] presents Performance Evaluation of Thresholding Techniques on Modi Script. They strive to increase image prominence, enhance contrast value, and distinguish foreground and background information using the thresholding technique. For many scripts, there are multiple popular thresholding strategies such as bernsen, wolf, sauvola, otsu, niblack, and bradley. This work proposes a threshold model for binarizing Modi character images efficiently. It employs a variety of global and local thresholding approaches to achieve greater contrast, better illumination, and other benefits. The impacts of various thresholding approaches are measured using two performance parameters: mean square error and peak signal to noise ratio. As a result, the Otsu thresholding technique successfully binarizes Modi vowels in a more acceptable form.

XIV. Sidra Anam et al. [1] Using Otsu's Binarization technique and the Kohonen neural network method, the Modi Script Character Recognizer System (MSCR) was created. We used a Kohonen neural network to train the system using22 different Modi script characters, including vowels and consonants. Handwritten samples from various persons are used to maintain the sample data set (training pictures) of the characters. These graphics are used to offer system training. The collected results show that the proposed recognition strategy is effective. We have lower recognition rates for characters that have similar shapes and structures. In the case of handwritten characters, an adequate character recognition rate of 72.6 percent was attained

- **Proposed System**
- **Modi character and digit dataset**
  self-created dataset
  75% of data used for training
  25% of the data for testing
- **Image Pre-processing:**
  Resize input image to 256*256 pixels
  Enhance the quality of image
- **Classification:**
  The deep learning algorithm is selected based on the training accuracy.



Block Diagram of MODI character recognition system

Figure 1. Detailed Block Diagram of face recognition system training

- Database

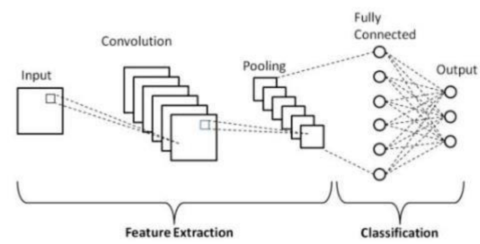
The database distribution of the database for each character is tabulated in Table I.

TABLE I. DATABASE DISTRIBUTION

Algorithm	Training		Validation	
	Accuracy	Loss	Accuracy	Loss
CNN	0.7393	0.9254	0.7646	0.8129
Vgg16	0.9973	0.0108	0.9248	0.3771

Dataset was divided into training and testing.

A. CNN



- In this system, the training data is trained with the convolutional neural network.

- The Convolutional layer performs the core building block of a Convolutional Network that does most of the computational heavy lifting. The primary purpose of the Convolution layer is to extract features from the input data, which is an image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of an input image. The input image is convoluted by employing a set of learnable neurons. This produces a feature map or activation map in the output image, and after that, the feature maps are fed as input data to the next Convolutional layer.

- **Pooling Layer**

The pooling layer reduces the dimensionality of each activation map but continues to have the essential information. The input images are divided into a set of non-overlapping rectangles. Each region is down-sampled by a non-linear operation such as average or maximum. This layer achieves better generalization, faster convergence, is robust to translation and distortion, and is usually placed between Convolutional layers.

- **ReLU Layer**

ReLU is a non-linear operation and includes units employing the rectifier. It is an element-wise operation that is applied per pixel and reconstitutes all negative values in the feature map by zero. To understand how the ReLU operates, we assume that there is a neuron input given as x, and from that, the rectifier is defined as f(x)= max(0,x) in the literature for neural networks image. The goal of employing the FCL is to utilize these features for classifying the input image into various classes based on the training dataset. FCL is the final pooling layer feeding the features to a classifier that uses the Softmax activation function. The sum of output probabilities from the Fully Connected Layer is 1. This is ensured by using Softmax as the activation function. The Softmax function takes a vector of arbitrary real-valued scores and squashes it to a vector of zero and one that sums to one.

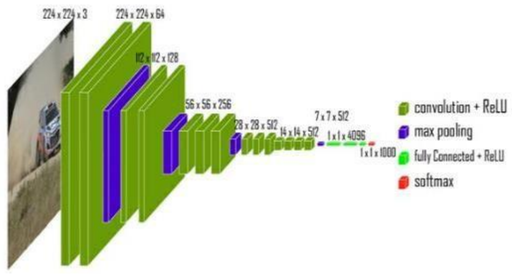
- **Flattening Layer**

After finishing the previous two steps, we're supposed to have a pooled feature map by now. As the name of this step implies, we will flatten our pooled feature map into a column.

- **Fully Connected Layer**

The goal of employing the FCL is to utilize these features for classifying the input image into various classes based on the training dataset. FCL is regarded as the final pooling layer given the features to a classifier that uses the Softmax activation function. The sum of all the output probabilities from the FC is 1. The Softmax function is used as the activation function. The Softmax function takes a vector of arbitrary real-valued scores and squashes it to a vector of zero and one that sums to one.

a. VGG16



VGG16 proved to be a significant milestone in the quest of mankind to make computers "see" the world. A lot of effort has been put into improving this ability under the discipline of Computer Vision (CV) for a number of decades. VGG16 is one of the significant innovations that paved the way for several innovations that followed in this field.

It is a Convolutional Neural Network (CNN) model proposed by Karen Simonyan and Andrew Zisserman at the University of Oxford. The idea of the model was proposed in 2013, but the actual model was submitted during the ILSVRC ImageNet Challenge in 2014. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was an annual competition that evaluated algorithms for image classification (and objectdetection) at a large scale.

A stack of multiple (usually 1, 2, or 3) convolution layers of filter size 3 x 3, stride one, and padding 1, followed by a max-pooling layer of size 2 x 2, is the basic building block for all of these configurations. Different configurations of this stack were repeated in the network configurations to achieve different depths. The number associated with each of the configurations is the number of layers with weight parameters in them.

The convolution stacks are followed by three fully connected layers, two with size 4,096 and the last one with size 1,000. The last one is the output layer with Softmax activation. The size of 1,000 refers to the total number of possible classes in ImageNet.

VGG16 refers to the configuration “D” in the table listed below. The configuration “C” also has 16 weight layers. However, it uses a 1 x 1 filter as the last convolution layer in stacks 3, 4, and 5. This layer was used to increase the non-linearity of the decision functions without affecting the receptive field of the layer.

Image is passed through the first stack of 2 convolution layers of the very small receptive size of 3 x 3, followed by ReLU activations. Each of these two layers contains 64 filters. The convolution stride is fixed at 1 pixel, and the padding is 1 pixel. This configuration preserves the spatial resolution, and the size of the output activation map is the same as the input image dimensions. The activation maps are then passed through spatial max pooling over a 2 x 2-pixel window, with a stride of 2 pixels. This halves the size of the activations. Thus the size of the activations at the end of the first stack is 112 x 112 x 64.

IMPLEMENTATION

Software Specification

Software Requirement

The system is developed using python language. The open-source anaconda distribution of the python libraries is used. The image processing algorithms were created using the open-source OpenCV library. The scikit-learn library is used to train the ML model and classify the images. A brief introduction of all software tools, libraries, and IDE is explained below.

- **Spyder**

Spyder is a robust environment built-in Python for Python and designed by and for scientists, engineers, and data analysts. It offers a unique amalgamation of a wide-ranging development tool's advanced editing, analysis, debugging, and profiling functionality with the data study, interactive implementation, deep examination, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended further via its plug-in system and API. Also, Spyder can be used as a PyQt5 extension library, allowing developers to develop upon its functionality and embed its mechanism, such as the interactive console, in their PyQt software.

- **Keras**

Keras is an open-source NN library written in Python. It can be executed on top of Tensorflow Microsoft Cognitive Toolkit, R-language, Theano, or PlaidML. It is developed to enable superior experimentation with DNN. It focuses on being user-friendly, modular, and extensible. It was developed as part of the study effort of project ONEIROS, and its primary author and maintainer are Francois Chollet, a Google engineer.

Keras properly doesn't do its low-level operations, such as tensor products and convolutions; it relies on a backend engine. Even though Keras supports numerous backend engines, its primary back end is TensorFlow, and its primary supporter is Google. The Keras API comes packaged in TensorFlow as tf. Keras, which, as mentioned previously, will become the direct TensorFlow API as of TensorFlow 2.0.

- **Image processing library: OpenCV**

Open-source Computer Vision (OpenCV) is an image processing and computer vision library mainlydeveloped for artificial vision. It has a BSD license (free for commercial or research use). OpenCV was originally written in C, but currently, it's a whole C++ interface, and there's additionally an entire Python interface to the library. Open-source computer Vision Library, also called OpenCV, is associated with a freeware software package for computer vision. It is used in this project because of its versatility and the fact that it has a C++ interface. OpenCV runs on most major Operating Systems (OS), making it worthwhile to use another computer to program or test.

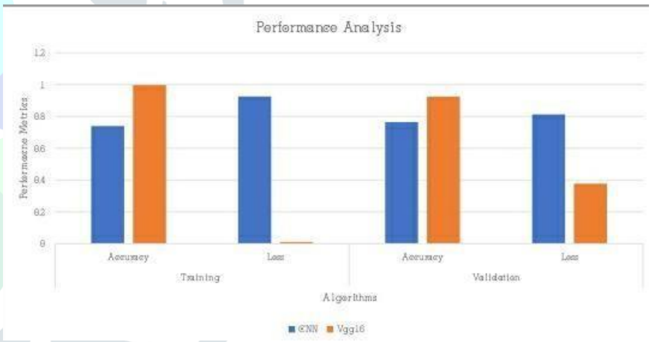
- ## Language: Python

Python is a high-level programming language extensively used for programming. Python, an interpreted language, supports several programming scripts and a syntax that allows you to use programs in most languages such as C ++ or Java. The language provides constructions designed to permit straightforward programs at each scale. Python is easy to know. The python code is way easier than alternative languages.

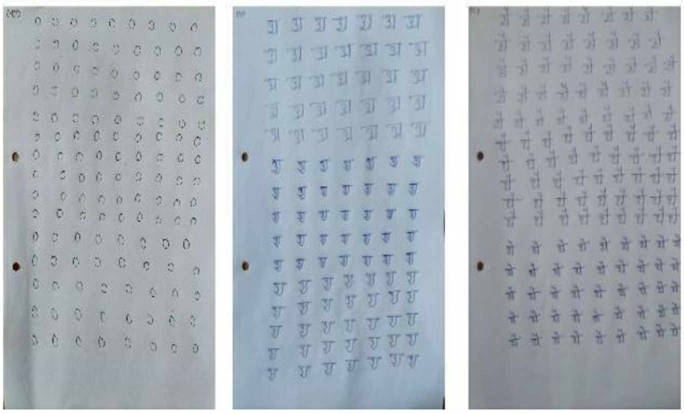
4.1.1. Hardware Requirement

- **Hard Disk:** 200 GB
- **RAM:** 8 GB
- **Processor:** Intel Pentium i5 and above

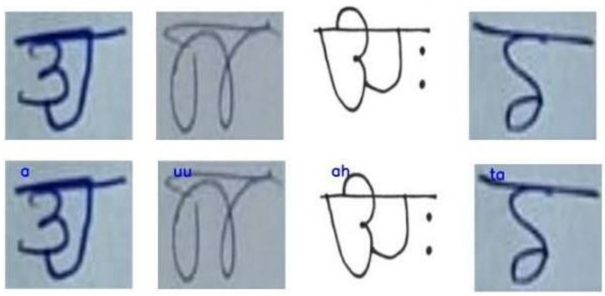
GRAPHICAL ANALYSIS OF CNN & VGG16 NETWORK

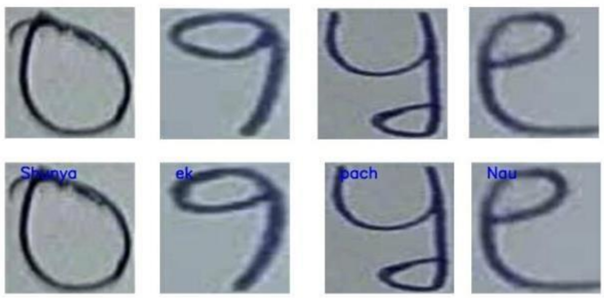


RESULT



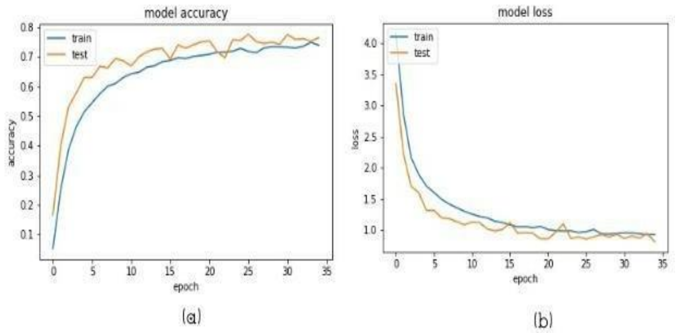
QUALITATIVE ANALYSIS





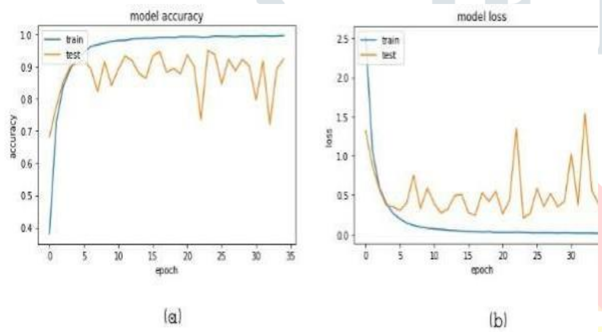
QUANTITATIVE ANALYSIS

CNN:



The training progress of CNN algorithm shows the training algorithm accuracy of 73.93%, loss of 0.9254. Validation accuracy of 76.46% and validation loss of 0.8129. The Training time for CNN is 9866 second.

VGG16:



The training progress of vgg16 algorithm shows the training accuracy of 99.73%, loss of 0.0108. Validation accuracy of 92.48% and validation loss of 0.3771. The Training time for CNN is 7267 second.

Conclusion

In this approach, the MODI character recognition using a convolutional neural network algorithm has been presented. The handwritten MODI character and the numeral dataset is created from different writers of different writing style. The dataset is split into training (80%) and testing (20%). In this system, CNN and Vgg16 algorithm is used to train and test the MODI character, and numerals and performance are evaluated using accuracy and loss parameters.

The proposed system shows that CNN algorithm achieved the training algorithm accuracy of 73.93%, loss of 0.9254. Validation accuracy of 76.46% and validation loss of 0.8129. The Training time for CNN is 9866 second. While the training progress of vgg16 algorithm shows the training accuracy of 99.73%, loss of 0.0108. Validation accuracy of 92.48% and validation loss of 0.3771. The Training time for CNN is 7267 second. Hence we can conclude that the accuracy of the vgg16 algorithm for proposed system is better than the CNN algorithm for training as well as validation. Also, loss of the vgg16 algorithm is less than the CNN algorithm for training and validation.

References

[1]. Besekar D.N., 2011, Recognition Of Numerals Of MODI Script Using Morphological Approach‖, International Referred Research Journal, ISSN 0974- 2832, vol. 3, Issue -27, pp. 63-66.

[2]. Besekar D.N., 2012, ―Special Approach for Recognition of Handwritten MODI Script‘s Vowels‖, International Journal of Computer Applications (IJCA), MAHA-2012, pp. 48-52.

[3]. Ramteke A.S., Katkar G.S., 2012, ―Recognition of Offline MODI Script‖, International Journal of Research in Engineering, IT and Social Science (IJREISS), Volume 2, Issue 11, ISSN 2250-0588, pp.102-109.

[4]. Besekar D.N., Ramteke R.J., 2012, ―Feature Extraction Algorithm for Handwritten Numerals Recognition of MODI Script using Zoning-based Approach‖, International Journal of Systems, Algorithms & Applications, Volume 2, Issue ICRASE12, ISSN 2277 2677, pp. 1-4.

[5]. Besekar D.N., Ramteke R.J, 2013, ―Study for Theoretical Analysis of Handwritten MODI Script – A Recognition Perspective‖, International Journal of Computer Applications, vol. 64, no. 3, ISSN 0975-8887, pp. 45-49.

[6]. S. Joseph and J. George, "Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier," 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP), 2020, pp. 32-36

[7]. Parag Tamhankar, Krishna Masalkar, and Satish kolhe. A novel approach for character segmentation of offline handwritten marathi documents written in modi script. Procedia Computer Science, 171:179–187, 01 2020.

[8]. Savitri Laxman Rao Chandure and Vandana Inamdar, "Performance Analysis of Handwritten Devnagari and MODI Character Recognition System'', 2016 International Conference on Computing, Analytics and Security Trends (CAST) College of Engineering Pune, India. Dec 19-21, 2016.

[9]. Manisha s. Deshmukh and Manoj Patil and Satish kolhe," Offline handwritten Modi numerals recognition using chain code''.

[10]. Prof. Mrs. Snehal R. Rathi, Rohini H. Jadhav, Rushikesh A. Ambildhok "Recognition and conversion of handwritten MODI characters," International Journal of Technical Research and Applications. Jan-Feb 2015.

[11]. Sanjay S. Gharde and Dr. R. J. Ramteke, "Indian Recognition of Characters in Modi Script'', 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication,978-1-5090-0467-6/16/$31.00 ©2016 IEEE.

[12]. B. Solanki, M. Ingle "A Comparative Study of Specific Phase based Character Recognition Techniques for Various Scripts." International Journal of Computer Sciences and Engineering 6.8 (2018): 496-502.

[13]. Sidra Anam and Saurabh Gupta, "An Approach for Recognizing Modi Lipi using Otsu's Binarization Algorithm and Kohenen Neural Network," International Journal of Computer Applications (0975 – 8887) Volume 111 – No 2, February 2015.