Recognition of Teeline Shorthand using Deep Learning.

Shivaprakash, Vishwanath C Burkpalli B. S. Anami

Abstract: Recognition of the Teeline shorthand is the interesting research problem. The accurate recognition of the Teeline patterns which helps the journalists to summarize the VIPs talk during the press meets which reduces the writing time. In order to address the above issues, the proposed method addresses the automatic recognition of the Teeline symbols which is used to provide alphabets as output. The framework consists of obtaining the input image, pre-processing it, perform the background subtraction on grayscale image and prepare the dataset for recognition. To prepare the dataset, the letters are written using the digital pen, it generates the pixelsand it is recorded in the Android application. Later these pixels are used to convert them into the image dataset. The prepared database is fed to the manually designed eleven layered Convolutional Neural Network (CNN). CNN is designed in such a way that it accepts the input Teeline alphabet and it able to say the equivalent English alphabet based on the training process. It has been carried out on the MATLAB 2018b and achieved the accuracy of 95%.

Keywords: Teeline shorthand, CNN, Matlab, accuracy.

I. INTRODUCTION

a) Teeline Shorthand Language.

The Teeline shorthanded system has been developed by the professor James Hill in the year 1968. The challenges in detecting the short hand languages are pattern recognition problem, as it includes various shapes, pattern and orientations while constructing the language. This Teeline shorthanded work has been accepted by the National Council for the Training of Journalists accepts this, which certifies the training of journalists in the United Kingdom.

Basically, there are twenty-six symbols for every English alphabet. All the symbols are represented using the simple geometric forms as shown in the below figure.
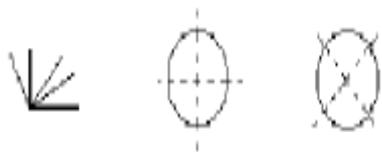
Figure 1: Geometrical Forms for Teeline.

Mr. Shivaprakash*, Department of Computer Science and Engineering Government Engineering College, Devagiri, Haveri, India.
Dr. Vishwanath C Burkpalli, Department of Information Science and Engineering P.D.A. College of Engineering Gulbarga, India.
Dr. B. S. Anami, Principal, K.L.E. Institute of Technology Hubli, India.

The symbols can be drawn at above, below, middle or at bottom of the primitive (writing space). Eight curved primitives (arcs) and Five straight-line primitives fundamental set of primitives are made. Four hook-structured primitives are used as consonants. The twenty-six symbols are shown in the below figure,

Figure 2: Teeline primitives used for shorthand writing.

There are up-to fifty shorthand languages out of which the Pitman and Teeline shorthand language has been globally accepted in the year 1837 and 1968. The literature survey is provided in this section. Very little information is found in the literature concerned to Pitman shorthand recognition (Leedhan&Qiao 1992; Hemanth Kumar 1998). For recognising primitives by the Hough transformation (Nagabhushan& Murali 2000) a lot of computational time and requires a large amount of memory. Basic patterns (lines, arcs) (Chen & Lee 1992; Anin et al 1997; Foggia et al 1997) individually there are several techniques to recognise. Hence from the survey it is clear that the

Ma Yang et. Al., investigates the potential of fast real-time handwritten text entry (>120wpm) on hand-held devices of Pitman shorthand as a mean. The input combined by the online system for recognition and transcription of Pitman shorthand is proposed. In the demonstration system the main recognition algorithms taken are discussed and evaluated. Based on the implementation experience of the system, future research directions are discussed. [2]

Proposed systemis capable of taking Teeline alphabet image as the input, it circulates through the eleven layers and predicts the English alphabet.

Section 2 describes about the working of the network, Section 3 explains about the experimental procedure, section 4 describes about the accuracy of the system (result analysis) and next section is about the conclusion and future scope.

II. NETWORK WORKING

Working of the network is divided into two sections. Section2.1 defines about the theory related to CNN in brief manner. Section 2.2 deals with the properties of CNN and the function of layers.

1.1 Theoretical Background

Computational models of neural networks have been around for a long time, first model proposed was by McCulloch andPitts as in [3].

Neural networks are made up of a number of layers with each layer connected to the other layers forming the network. Afeed-forward neural network or FFNN can be thought of in terms of neural activation and the strength of the connections between each pair of neurons [4].

In FFNN, the neurons are connected in a directed way having clear start and stop place i.e., the input layer and the output layer. The layer between these two layers, are called as the hidden layers. Learning occurs through adjustment of weights and the aim is to try and minimize error between the output obtained from the output layer and the input that goes into the input layer. The weights are adjusted by process of back propagation (in which the partial derivative of the error with respect to last layer of weights is calculated). The process of weight adjustment is repeated in a recursive manner until weight layer connected to input layer is updated.
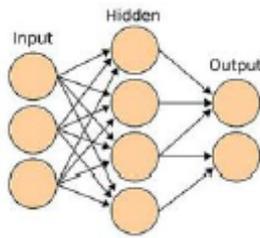

Figure 3: Typical Network [5].

Convolutional Neural Networks (CNN) is variants of Multi-Layer Perceptron(MLPs)which are inspired from biology. To exploit the strong spatially local correlation present in natural images these filters are local in input space and are thus better suited

[5]. To process two-dimensional (2-D) image convolutional neural networks are designed

[6]. A CNN architecture used in this project is that defined in

[7]. The three types of layers are constituted by the network namely convolution layer, hidden layers and output layer. Hidden layers includerelu, maxpooling, batch normalization layer totally eleven layers excluding input and output layer.

1.2 Working of CNN algorithm

This section explains the algorithm working in brief. In detailed version is available in [7].

The input to the network is a 2D image. The image is taken as input by the network, output layer from where we get the intermediate layers and the trained output called as the hidden layers. As stated earlier, a series of convolutional is present in the network, activation function, soft max layer and pooling layer.

The input to the network is a 2D image. The image is taken as input by the input layer, output layer from the trained

output is got from the intermediate layer called as the hidden layers. As stated earlier, the net-work has a sub-sampling layers and series of convolutional. Together the layers produce an approximation of input image data. A CNN's exploit local correlation by enforcing them between the neurons of adjacent layers a local connectivity is checked.

From neuron layers of (m-1) we consider that the subset of neurons connected are 'm', whereas shown in the below figure the neurons of the (m-1) layer have continuous receptive fields.
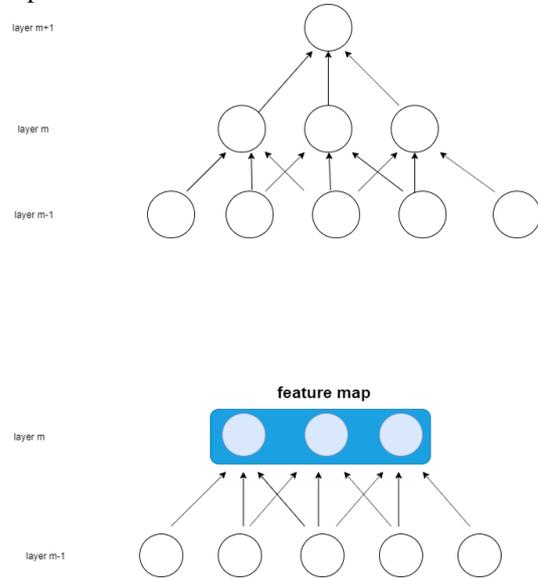

Figure 4: Graphical flow of layers showing connection

In the CNN algorithm, the entire visual field is replicated by each sparse filter. Feature maps are formed by these units, weight vector and bias are shared by them. Three hidden units of same feature map is represented by Figure (4). These are identical and are constrained with the weights of same colours are shared.

The parameters of gradients being shared is the sum of gradients of shared weights. In visual field regardless of the position such replication allows feature to be detected. Weight sharing in addition to reduce the number of free learning parameters. Better generalization on vision is achieved by CNNproblems due to this controls.

Max-pooling is made use in the concept of CNN, down-sampling which is a form of non-linear.The inputted image is separated into the non-overlapping rectangles. The maximum value for each sub-region is the output.
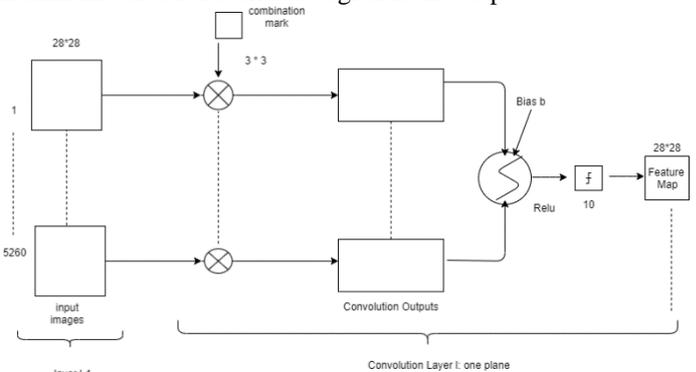

Figure 5: Convolution Layer Working

The first layer of the CNN network is the convolution network.

In the figure 5 the structure of the layer is shown. It consists of a function expression, bias terms and a convolution mask. Together, the output of these layers is generated.

The figure 5. shows a 3x3 mask of 28x28 input feature map which performs convolution over it. Then bias is added and Reluactivation function is applied on the matrix [7].

1.2.1   Layer of Sub sampling

The convolution layer comes before the sub sampling layer. As the convolutional layer the number of planes is the same. The size of the feature map is reduced by doing so. Averaging is performed after dividing the image into blocks of 2x2. In the Sub sampling not the exact relation but the information of features between layers is preserved.
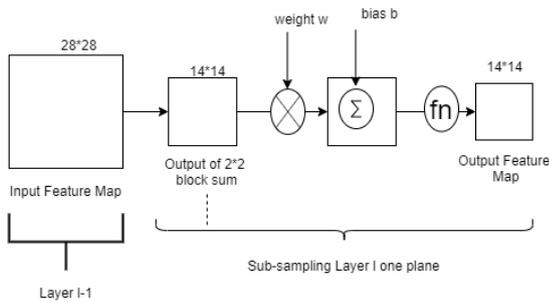

Figure 6: Sub-Sampling layer working [7]

III.   Experimental Setup

1.3 Preparing Dataset

The input to the network is the images. The letters are written using the digital pen later those pixels are extracted and images of them are generated from the Android application as shown in the Figure 7, the original image dimension is 2000*2100. First stage of the preparation is to reduce the dimension from 2000*2100 to 28*28. The main reason for reduction is for reduce the computation. For ex: 2000*2100 = 4200000 pixels need to be processed per image and 5240 images each for alphabets are considered, hence 4200000 * 5240 it needs high configuration system, on the other hand for 28*28 image,only 784 (24*24) pixels per image need to be processed hence computation cost reduces.

Figure 7: Front end of the Android Application

Second stage is to convert the using the MATLAB inbuild command the colour image is converted to greyscale image.

In the third stage the grayscale image is converted to binary image. The last stage of data preparation is to invert the binary image for obtaining the clear boundary of the symbols or alphabets. Through reduced image representation the image pyramid is a data structure to support efficient scaled convolution. Sample density and resolution are decreased in regular steps in which consists of a sequence of copies of an image [9].

1.4 Testing and Network training

The purpose of the training the network is to reduce the error between the actual output and the predicted machine output. The error function is same for weights as well as bias terms and is as defined by the equation below.

$$E(w) = \frac{1}{K * N} \sum_{K=1}^{K} \sum_{n-1}^{N_L} \left(y_n^k - d_n^k\right)^2$$

In the above equation y_n^k is the actual output of the network, K is the number of input image and the output vectors desired. x^K be the kth training image and dk corresponding output vector. The error sensitivities compute the error gradients, which are defined as the weighted sum input to a neuron to the error function partial derivative.

Once E(t) the error gradient is derived, the energy function for minimizing in training the network can be applied. Here SGDM (stochastic gradient descent with momentum) is used. Based on local gradient information a direct adaptation that is performed in the learning scheme [10].

The setup of the network is as shown in figure.8.
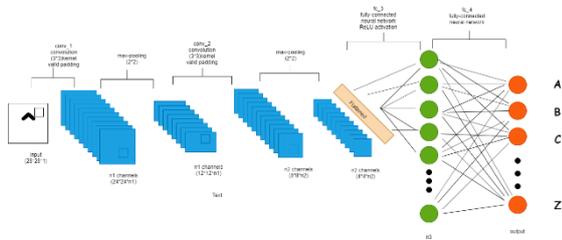
Figure 8: CNN layer of the implementation

IV. RESULT AND DISCUSSION

This section presents the result of the classification accuracy obtained using the CNN algorithm on the prepared dataset using online android application. The dataset is divided into training, testing and validation in 60%, 20% and 20% respectively for conducting the experiment. The results are presented in the form of graphs which includes accuracy, Mean Square error and confusion matrix. The Graphs show the change of MSE with respect to the training epochs. MSE metric is the widely used and simplest metric quality used. Between original image the mean of squared difference and trained approximation. A better trained image will result in lower MSE with the original image. As the value for Mean Squared Error (MSE) tends to decrease, the variation in the final reconstructed output and the original image is very less. MSE indicates the close proximity between underlying true image and the final reconstructed output. The idea here is to use enough number of epochs that would result in low MSE, high classification accuracyand with least duration for training the network.

The network is tested on various new images, in turn for each new image is tested for different epochs, different iterations. In the section IV presents the results of the algorithm.

1.5 Image employed in classification

The data used in the experiment are generated from the android application which has been shown in Figure 7. The 200 different images for each character of Teeline is collected and further these are divided into testing and training images for network. The images have the dimension 28*28*3. Some of the sample images are shown in the Figure 9.
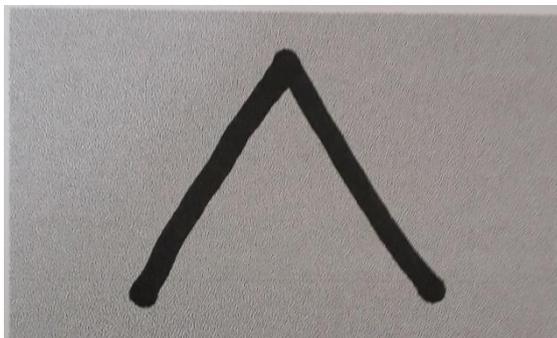
Figure 9: Samples of Database

The Figure 10 a) and 10 b) shows the variations of the Teeline losses for epochs 8 and 21 respectively. From the graphs it is observed that the losses go on decreasing once the epochs go on increasing from 8 to 21. The epochs can be varied in the range in between 8 to 21.
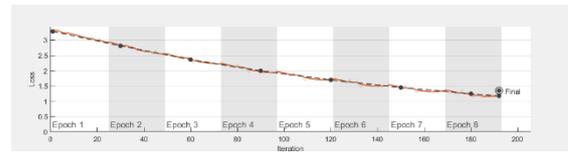

Figure 10 a): Losses for the eight epochs
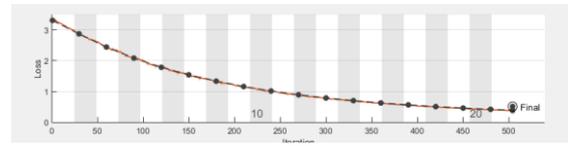

Figure 10 b): Losses for the twenty-one epochs

The same observation is also obtained from the classification accuracy values where; accuracy improves as it moves from 8 to 21 epochs. The computation time goes on increasing once it reaches the 21 hence the best suited results can be obtained from epochs 8.

The confusion matrix of epochs 8 and 21 is shown in the Figure 11 a) and Figure 11 b) respectively.
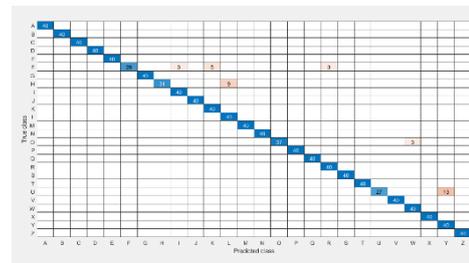
Figure 11 a): Confusion Matrix for eight epochs
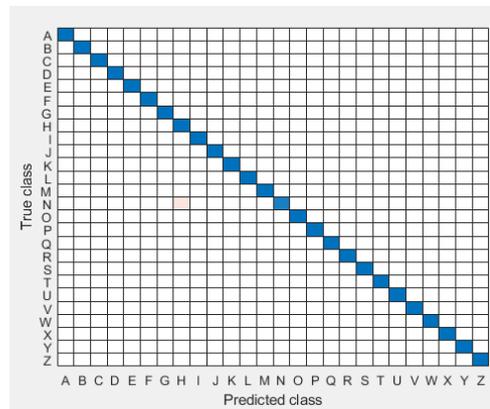
Figure 11 b): Confusion Matrix for twenty-one epochs

The training phase of the different epochs in the Figure 12 a) and 12 b) are shown
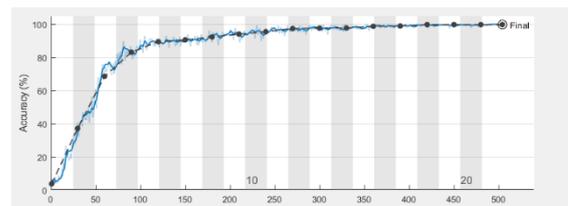
**Figure 12 a): Training Iterations for the epochs 21**
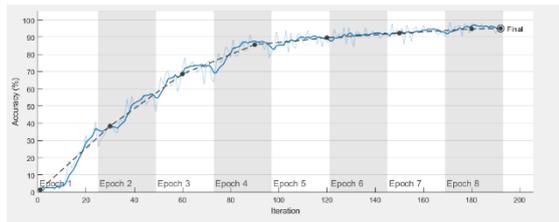
Figure 13 b): Training Iterations for the epochs 8

V. CONCLUSION

The proposed method using the deep learning (CNN) is used to carry out the task for recognising the text written in Teeline short hand language. It is able to recognise most of the online generated Teeline alphabets. Recognition rate has been increased by varying the number of layers in CNN and epochs during the training process. Compared to the accuracy of the PITMAN [2] recognition, the proposed method achieved 10% more. Hence this method can be employed for real time recognition of the Teeline shorthand alphabets.

REFERENCE

1. The image downloaded from this link
   https://en.wikipedia.org/wiki/Teeline_Shorthand
2. Ma Yang, Graham Leedham, Colin Higgins &SweMyoHtwe, "An On-line Recognition System for Handwritten Pitman Shorthand", TENCON 2005 - 2005 IEEE.
3. P NAGABHUSHAN and S MURALI, "Recognition of Pitman shorthand text using tangent featurevalues at word level" Sadhana Vol. 28, Part 6, December 2003.
4. Amin A, Kim S, Sammut C 1997 Hand printed Chinese character recognition via machine learning.4th Int. Conf. on ICDAR, vol. 1, pp 123–156
5. Babaguchi N, Aibara T 1987 Curvedness of a line picture. J. Pattern Recogn. 20: 273–280
6. Chen B, Lee J H 1992 Recognition of handwritten Chinese characters via short line segments. J.PatternRecogn. 25: 543–552
7. Foggia P, Sansone C, Tortorella F 1997 Character Recognition by Geometrical Moments on StructuralDecomposition. 4th Int. Conf. on ICDAR, vol. 1, pp 6–10
8. Hemanth K G 1998 On automation of text production from Pitman shorthand notes. Ph D thesisUniversity of Mysore, MysoreLeedham C G, Nair A 1992 Evaluation of dynamic programming algorithms for the recognition ofshort forms in Pitman's shorthand. J. Pattern Recogn. Lett. 13: 343–350
9. Leedham C G, Qiao 1993 Segmentation and recognition of handwritten Pitman shorthand outlinesusing an interactive heurtistic search. J. Pattern Recogn. 26: 433–441
10. Nagabhushan P, Anami B S 2000 A knowledge-based approach for recognition of grammaloguesand punctuation symbols useful in automata. National Conference on Recent trends in Advanced Computing, Tirunelveli, pp 173–185
11. Nagabhushan P, Murali S 2000 Linear Hough transformation for recognition of stroke primitives inc. Pitman shorthand text, National Conference on Recent trends in Advanced Computing, Tirunelveli,pp 126–128
12. Rosenfield A 1995 Geometric properties of set of lines. J. Pattern Recogn. Lett. 16: 643–648
13. Tappert C C, Suen C Y,Wadhara T 1990 The state of the art in on-line handwriting recognition. IEEETrans. Pattern Anal. Mach. Intell. 12: 1029–1058