Text Recognition from Complex Colored Images using Neural Network with Discriminative Feature Extraction

G. Gayathri Devi, PhD
Department of Computer Science,
SDNB Vaishnav College for Women,
Chennai, India

G. Sathyanarayanan
Senior Professional Project Management,
DXC Technology

C. P. Sumathi, PhD
Department of Computer Science,
SDNB Vaishnav College for Women,
Chennai, India

ABSTRACT
The objective of this paper is to project a new methodology for text recognition from the features of segmented text component of images. Text classification algorithm is the main decision making stage of text recognition system. Artificial neural network approach has been used to train and test the character based on the extracted features. Finally, the identified texts are converted in to readable/editable version of text file.

Keywords
Text extraction, Feature extraction,Text Recognition, Neural Network, Back Propagation.

1 . INTRODUCTION
The aim of text recognition is to recognize and covert human readable text image characters to machine readable characters. Classification stage is the main decision making stage of text recognition system and uses the features extracted in the previous stage to identify the text component according to the features extracted.

2 . EXISTING TEXT RECOGNITION METHODS
Text recognition stage is the main decision making stage of text recognition system. Various classifiers techniques are proposed in the literature and are used for the recognition of text. Some of them are multi-level slice classifier, minimum distance classifier, maximum likelihood classifier, fuzzy measure, artificial neural network, support vector machines, decision tree etc.

A robust method [1] that uses convolutional co-occurrence histogram of oriented gradient (ConvCoHOG) and discriminative than both the histogram of oriented gradient (HOG) and the co-occurrence histogram of oriented gradients (CoHOG).An image was first divided into smaller patches and feature extraction procedure was applied in every patch separately to extract features. The orientation of gradient of each pixel within a patch is then quantized into histogram bins and then, normalized histogram was concatenated together to form a feature vector ant it was trained by al linear SVM classifier.

In end-to-end method [2] individual characters were detected as Extremal Regions. The regions were first agglomerated into text lines by an efficient pruned exhaustive search that estimates the text direction on each triplet of regions and the constraints induced by the text direction contribute to the similarity measure used for clustering. In the next stage, each region in the text line was labeled by the character recognition module, which was trained on synthetic fonts. Regions with low confidence were rejected, which eliminates clutter regions that were included in the text line formation stage. In the last step, a directed graph was constructed with corresponding scores assigned to each node and edge, the scores were normalized by width of the area that they represent and a standard dynamic programming algorithm was used to select the path with the highest score. The sequence of regions and their labels induced by the optimal path was the output of the method.

Gokhan Yildirim et.al [3] proposed a technique to detect and recognize text in a unified manner by searching for words directly without reducing the image into text regions or individual characters. In first stage, an object detection framework called Hough Forests was modified by introducing cross-scale binary features to compare the information between the same image patch at different scales. This modified technique was used to produce likelihood maps for every text character. In second stage, word-formation cost function and computed likelihood maps were used to detect and recognize the text in natural images.

A system for open-vocabulary text recognition in images of natural scenes was presented in [4]. Bilateral regression segmentation was introduced to segment images into foreground text and background. The algorithm considered connected component in the binary image as a character and used a linear-chain conditional random field to represent the sequence of those characters in a word. A recognition component was introduced based on a conditional random field with histogram of oriented gradients descriptors and incorporate language information from a lexicon to improve recognition performance.

A novel approach for end-to-end text recognition was introduced in [5]. The proposed system combined the large representational power multilayer neural networks with unsupervised feature learning that allowed using a common framework to train highly-accurate text detector and character recognizer modules. As the system use large representational power, it was able to use simple non-maximal suppression and beam search techniques to construct a complete system.

A framework [6] that exploits both bottom-up and top-down cues from the street images. The bottom-up cues were derived from individual character detections from the image. The method built a Conditional Random Field model on the detected character to identify the strength of the detections and the interactions between them. The top-down cues were obtained from a lexicon-based prior on the model. The optimal word represented by the text image was obtained by minimizing the energy function corresponding to the random field model.

An easy and fast method to recognize individual characters in images of natural scene was presented in [7]. The feature vector was computed based on a gradient direction feature and was classified to a class using K- Nearest Neighbors. The e proposed method had been evaluated on the ICDAR test dataset and showed 85.4% result.

The architecture [8] was described to learn the feature representations and train the classifiers used for detection and character recognition systems. In the first stage, an unsupervised feature learning algorithm was applied to a set of image taken from the training data to extract image features. The number of features extracted was reduced by spatial pooling and it was trained by a linear classifier.

3 . BACK PROPAGATION ALGORITHM

Back Propagation is the standard way of training neural networks (Fig 1). The input pattern on which the network had to be trained is presented at the input layer of the net and the net is run normally to see what output it actually produces (Jacek M. Zurada 1992). The actual output is compared to the desired output for that input pattern. The differences between actual and desired output form an error pattern. This is then used to adjust the weights on the output layer so that the error pattern would be reduced the next time if the same pattern is presented at the inputs. The network gets trained by presenting each input pattern in turn at the inputs and propagating forward and backward, followed by the next subsequent input patterns. Then this cycle is repeated several times until the overall error value drops below some pre-determined threshold and at this point the network is said to be fully trained.

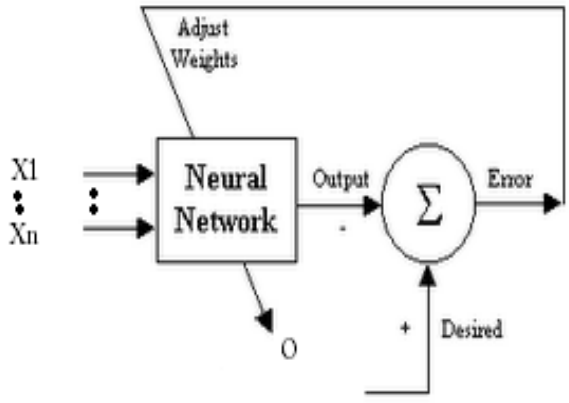


Fig 1: Neural Network

- The weight of a connection is adjusted by an amount proportional to the product of an error signal δ, on the unit k receiving the input and the output of the unit j sending this signal along the connection:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p$$

- If the unit is an output unit, the error signal is given by

$$\delta_0^p = (d_0^p - y_0^p) F'(s_0^p)$$

- taken as the activation function F the 'sigmoid' function

$$y^p = F(s^p) = 1/(1 + e^{-s^p})$$

There are many successful applications of Back propagation (BP) for training multilayer neural networks and character recognition, is one of the most widely used applications of Back Propagation Neural Networks (BPNN). Properly trained back propagation networks have proved to give reasonable answers when presented with inputs that they have never seen In many cases, it is possible to train a network on a representative data set and get good results without training the network on all possible combinations.

4 . PROPOSED METHODOLOGY

The classifier chosen in the proposed text recognition algorithm is a Feed-forward Multilayer Perceptron Neural Network trained with back propagation algorithm. In text recognition it is most important to recognize if a character is uppercase or lowercase. Some character shapes are very similar, such as an uppercase 'C' and a lowercase 'c', uppercase 'K' and a lowercase 'k', uppercase 'O' and a lowercase 'o', , uppercase 'P' and a lowercase 'p', an uppercase 'S' and a lowercase 's', uppercase 'U' and a lowercase 'u', uppercase 'V' and a lowercase 'v', uppercase 'W' and a lowercase 'w', uppercase 'X' and a lowercase 'x', an uppercase Z' and a lowercase "z. The text features vector for similar character will be mostly same. So, the idea of having a single network by combining all uppercase and lowercase texts for recognition will make the recognition process difficult.

An unconstraint character recognition using different classification strategies was propped by [11]. Two networks consisting of 26 outputs and one middle layer of 100 neurons, one for uppercase and one for lowercase characters was designed and the outputs were combined by three combination rules: average, maximum and product. The recognition rate when tested on NIST test data of 26+26- class classification problem are 82.60, 82.92 and 72.86 for average, maximum and product respectively. Same strategy of combining two networks but with different combination rule is used in this research.

Two neural networks with 26 outputs named as NNUC and NNLC are designed. One for recognizing the uppercase text and one for recognizing lowercase characters, but cannot restrict that input image should contain only uppercase or lowercase. To allow the input image of mixed case texts, the features vector of the text component is given as input to both the networks NNUC and NNLC and the output of NNUC and NNLC are analyzed to select the output node. Two MLP networks NNUC and NNLC (Fig 2) of each size 42-80-75-26 structure have been used.

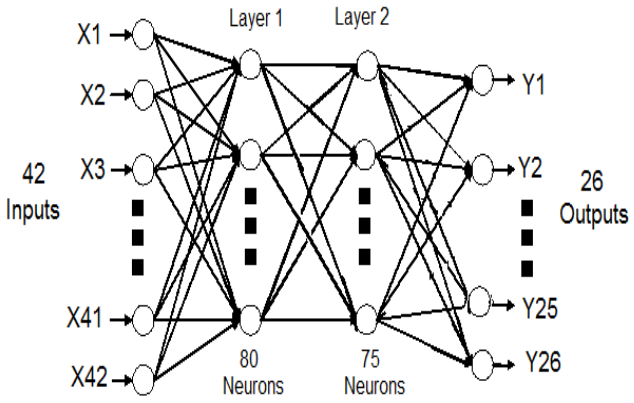


Fig 2: Architecture of NNLC / NNUC

The number of inputs to each network is associated with the size of the feature vector which is 42 (**BH1**, **BH2**, **BH3, BV1**, **BV2**, **BV3, SH1**, **SH2**, **SH3 SV1**, **SV2**, **SV3, LQ1**, **LQ2**, **LQ3**, **LQ4, BWQ1, BWQ2**, **BWQ3**, **BWQ4, FlipBWLR , FlipBWUD, Loop1**, **Loop2, SLANTHL20, SLANTHL40, SLANTHL60, SLANTHL80, SLANTHL100, SLANTHL120, SLANTHL140, SLANTHL160, SLANTVL20, SLANTVL40, SLANTVL60, SLANTVL80, SLANTVL100, SLANTVL120, SLANTVL140, SLANTVL160, HWRATIO, PRATIO)** for a text component image.

The output vector is a 26 element vector with a 1 in the position of the character it represents, and 0"s elsewhere. The output layer of NNUC and NNLC consists of 26 output units matching to 26 uppercase and 26 uppercase English characters respectively. There are two hidden layer in the network. The number of hidden neurons in the first hidden layer and the second hidden layer are set to 80 and 75 respectively. The network is trained by batch learning using a learning rate of 0.05, error tolerance of 0.01, 5000 learning cycles and a momentum factor of 0.8. The activation function for the hidden layer and output layer of neural network are based on tan sigmoid functions. The range of output for the neural network ranges between -1 and 1.

4.1 Training Data Set

The training dataset is taken from ICDAR and some of the training files needed are manually prepared (TRAINDATAUC consists of 13000 images of uppercase characters (A–Z), TRAINDATALC consists of 13000 lowercase characters (a–z)) with different font style and size. Two different feature vectors (FVUC, FVLC) are generated using the training dataset. The training is carried out using MATLAB .The reduction of error in the network is based on Sum Squared Error (SSE) till a goal of 0.01 is reached first or 5000 iterations is reached, whichever comes first.

4.2 Testing Data Set

The dataset is taken from ICDAR and some of the test files needed are manually prepared train dataset (TESTDATAUC consists of 10400 images of uppercase characters (A–Z), TESTDATALC consists of 10400 lowercase characters (a–z)) with different font style and size.

4.3 Experimentation

The dataset is taken from ICDAR and some of the test files needed are manually prepared train dataset (TESTDATAUC consists of 10400 images of uppercase characters (A–Z), TESTDATALC consists of 10400 lowercase characters (a–z)) with different font style and size.

Two different feature vectors (FVUC, FVLC) are generated and tested on NNUC and NNLC. The maximum value generated by all the output nodes of the network is considered as the output and also that value should be near to 1. For example if the 7th output node is maximum of all the output nodes and near to 1 ,the recognized text for the input image is the 7th character in English Letter set i.e. 'G' if the network is NNUC or 'g' if the network is NNLC. If any node does not give a value 1 or near to 1, then no output node is selected and reported that input as noise.

To allow the input image of mixed case texts for recognition, the feature vector of the text component is given as the input to the two networks NNUC and NNLC. The range of output for the neural network ranges between -1 and 1. Any one of the network will respond with a value 1 or near to 1 for the characters with non similar appearance like 'A' ,'a', b, 'B', 'D', d' etc. NNLC and NNUC will respond with the output 1 or near to 1 for the character with similar appearance. The network which responds with the maximum value is taken in to consideration. For example , if NNLC produces 0.9941 in output node 3 and NNUC produces 0.9599 in the output node 3 , then NNUC is considered and the character 'c' is reported as the output. If both the network produces 1 as the output then NNLC is taken into consideration.

In **post processing stage**, the details of the text position produced by the Text Position Details (TPD) algorithm is used to organize the characters produced by the neural network into editable text file. The output produced by the neural networks NNUC + NNLC for the sample ICDAR Dataset after post processing is shown in the Table 1.

Table 1: Output for the sample ICDAR images

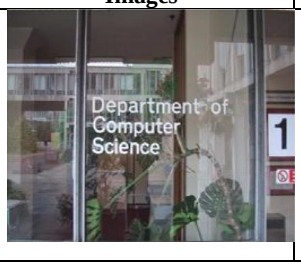
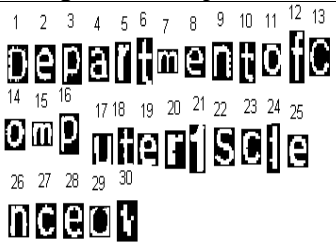
Image No	Images	Segmented Components	Output
Image 1			Department of Computer l Science

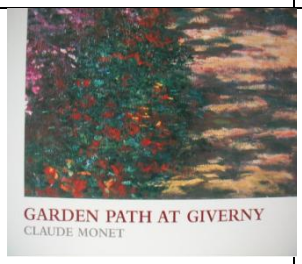
Image 2		Tesco Dlets lose a stone in B weeks GO
Image 3		Join our facebook group
Image 4		GARDEN PATH AT GIVERNY CLAUDE MONET
Image 5		EMERGENCY DOOR CONTROL
Image 6		FORUMS ENTER
Image 7		Example Image

Image 8		Plan your journey check the traffic join the Drlver s centre
Image 9		TESCO RACE FOE LIF CANCER RESEARCH UK
Image 10		Special TShirt
Image 11		EMERGENCY STOP
Image 12		WASINGLON

It can be observed from the output column of the Table 1; the noise in the Image 1, 2, 4, 9 is recognized as noise by the neural network and removed. This shows the noise which is not been removed by the text extraction and segmentation algorithm is successfully removed by the devised text recognition algorithm and improves the accuracy of the output. '22' in Image 12 is not recognized by the neural network as the system is trained only for English character set. In image 8, the 'i' in 'Driver' is recognized as 'I'.

5 . EXPERIMENTAL RESULTS

The experimentation of the algorithms was carried out on the ICDAR data set consisting of 500 different images and as well as from TESTDATAUC, TESTDATALC. Some of the experimental results are shown in the Section 4. Recognition rates achieved by the proposed NNLC, NNUC for each Class are shown in the Table 2.

Table 2: Recognition rates (in %) for each class on the ICDAR + TESTDATAUC + TESTDATALC test dataset

Class	NNUC	NNUC+NNLC	Class	NNLC	NNUC+NNLC
A	98.1	98.1	a	97.3	97.3
B	99.3	99.3	b	100	100
C	93.3	92	c	98.1	96
D	98.2	98.2	d	99.3	99.3
E	100	100	e	98.2	98.2
F	100	100	f	97.4	97.4
G	94.5	94.5	g	98.3	98.3
H	100	100	h	96.2	96.2

I	100	95.2	i	94.3	93
J	99.2	96.1	j	94.5	92.1
K	98	92	k	98.5	97
L	100	100	l	96.1	96.1
M	97	97	m	100	100
N	98.1	98.1	n	99.6	99.6
O	100	95.5	o	100	98
P	100	96.5	p	100	96
Q	96.5	96.5	q	100	100
R	97.2	97.2	r	97.4	97.4
S	94.4	93	s	95.2	94
T	100	100	t	96.4	96.4
U	100	97	u	99.8	97
V	99.1	96.3	v	99.4	96.8
W	97.5	95.5	w	99.5	97
X	99.1	95.6	x	99	94
Y	95	95	y	94.2	94.2
Z	96	93.4	z	96.4	95

6 . COMPARATIVE STUDY

The result of the proposed text recognition algorithm is compared with the experimental results of other text extraction algorithms that use ICDAR dataset for their experiment. The performance of each technique was evaluated based on recognition rate. The comparative study of the proposed method with the existing text extraction algorithms on ICDAR data set is shown in the Table 3. By comparing the proposed system with the existing text recognition algorithms, the algorithm achieves the high recognition rate and this shows the proposed method outperforms the existing methods.

Table 3: Comparative study of proposed Text Recognition method with existing method on ICDAR data set

S.No	Algorithm	Recognition Rate in %
1	Proposed Method	**96.7**
2	Anurag Bhardwaj et.al [10]	81
3	Alvaro Gonzal ez et.al [7]	81.4
4	Adam Coates et.al [8])	81.4
5	Bolan su et.al [1]	85.4
6	Lukas Neumann et.al [2]	85.4
7	Deepak Kumar et.al [9]	92
8	Tao Wang et.al [5]	84
9	Anand Mishra et.al [6]	81.78
10	Gokhan Yildirim et.al [3]	85.7
11	Jacqueline et.al [4]	62.96

7 . CONCLUSION

This paper presented a text recognition algorithm based on Feed-forward Multilayer Perceptron Neural Network trained with back propagation algorithm. The most important step involved in text recognition is the selection of the feature information of the text component. Recognition rate achieved is 96.7% and this shows that the feature extraction algorithm explained in [12] produced good set of features. The numbers are not recognized correctly as proposed NN is trained to recognize alpha characters. The noise which is not been removed by the text extraction and segmentation algorithm is successfully removed by the proposed text recognition algorithm and improves the accuracy of the result. As a future work, it is easy to extend the algorithm with an open-source spell checkers to achieve more accuracy.

8 . REFERENCES

[1] Bolan Su, Shijian Lu, Shangxuan Tian, Joo-Hwee Lim, Chew-Lim Tan. Character Recognition in Natural Scenes using Convolutional Co-occurrence HOG. Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.

[2] Lukas Neumann and J. Matas, "On combining multiple segmentations in scene text recognition," in Proc. ICDAR, 2013.

[3] Gokhan Yildirim, R. Achanta, and S. Ssstrunk, "Text Recognition in Natural Images Using Multiclass Hough Forests," in VISAPP, vol. 1, pp. 737-741, 2013.

[4] Jacqueline L. Feild and E. G. Learned-Miller, “Improving open-vocabulary scene text recognition,” in IEEE International Conference on. Document Analysis and Recognition, pp. 604–608, 2013.

[5] Tao. Wang, D. Wu, A. Coates, and A. Ng, "End-to-end text recognition with convolutional neural networks," in International Conference on Pattern Recognition, 2012

[6] Anand Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2012

[7] Alvaro Gonz´alez, Luis M. Bergasa, J. Javier Yebes, Sebasti´an Bronte, “Text Location in Complex Images ”, 21st International Conference on Pattern Recognition (ICPR 2012) , NOV 11-15, Tsukuba, Japan,2012.

[8] Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In ICDAR, pages 440-445, 2011.

[9] Deepak Kumar, M N Anil Prasad,A G Ramakrishnan,"Multi-script robust reading competition in ICDAR 2013" , MOCR ’13, Washington, DC, USA,August 24 2013.

[10] Anurag Bhardwaj, Chen-Yu Lee Wei Di, Vignesh Jagadeesh, and Robinson Piramuthu, “Region-based Discriminative Feature Pooling for Scene Text Recognition”, The 27th Annual Conference of the Japanese Society for Artificial Intelligence, 2013.

[11] Alessandro L. Koerich, Alceu de Souza Britto Jr., Luiz E. Soares de Oliveira,”Verification of Unconstrained Handwritten Words at Character Level” ,ICFHR,pp 39-44, 2010.

[12] G. Sathyanarayanan, G. Gayathri Devi and C. P. Sumathi,, International Journal of Computer Engineering and Applications,"Discriminative Feature Extraction Of Text Components From Complex Colored Images", Volume XI, Issue XI, Nov. 17,ISSN 2321-3469.