Script Identification In Multilingual Documents Using Artificial Neural Networks

Nikita Reddy T, Rahul Ch, Yashwanth Reddy, Ruchitha Chinthala Sri Ram TK
Student, Student, Student, Student, Student
Computer science,
Chaitanya Bharathi Institute of Technology, India

***Abstract*—** In India, where we see a multi script environment, majority of the documents may contain text information printed in more than one script/language. For automatic processing of such documents through Optical Character Recognition (OCR), it is necessary to identify different script regions of the document. Script identification is the process of identifying scripts in any multi-scripzt environment so that the recognized scripts can be sent to their corresponding OCR software for recognition purpose. Identifications aims to extract information presented in digital documents namely articles, newspapers, magazines and e-books. This has given rise to many language identification systems. The aim of the project is to propose visual clues based procedure to identify different text portions of a document. It has been observed that the three scripts - Telugu, Hindi and English possess their own distinct features. These distinct features could be used as supporting features in the process of script identification system. To identify the type of the language, we use visual clues or features, like top holes, bottom holes, top max row, bottom max row, bottom up curves, top down curves, vertical lines, slant lines, top and bottom component density, coefficient profile, top horizontal lines, without reading the contents of the document. The classification of a particular script is done using Artificial Neural Networks. Once the features are extracted, we can go ahead and train a neural network using the training data for which we already know the true classes. The inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

***Index Terms*—Optical Character Recognition, Artificial Neural Networks.**

I. INTRODUCTION

Objective

As the world is moving electronically, there is a growing tendency of converting the physical documents into electronic forms for easier access and purposes of privacy and security. According to the three-language policy adopted by most of the Indian states, the documents produced in an Indian state, are composed of texts in the regional language, the National language-Hindi and the world wide commonly used language-English. In addition, majority of the documents found in most of the private and Government sectors of Indian states, are tri-lingual type (a document having text in three languages). So, there is a growing demand to automatically process these tri-lingual documents in every state in India.

So, there is a great demand for software that can automatically extract information from these physical documents. But, the currently available OCR systems recognize only a specific language. The solution can be building a system that can identify and separate different languages and feed them into their appropriate OCR systems.

Problem Statement

Script identification is the process of identifying scripts in any multi-script environment so that the recognized scripts can be sent to their corresponding OCR software for recognition purpose.

Identifications aims to extract information presented in digital documents namely articles, newspapers, magazines and e-books. This has given rise to many language identification systems. The aim of the project is to propose visual clues based procedure to identify different text portions of a document.

Motivation

Multi lingual document segmentation has strong direct application potential, especially in a multilingual country like India. In the context of Indian languages, some amount of research work has been reported. Further there is a growing demand for automatically processing the documents in every state in India. Under the three language formulae, adopted by most of the Indian states, the document in a state may be printed in its respective official regional language, the national language Hindi and also in English. Accordingly, a document produced in Telangana, a state in India, may be printed in its official regional language Telugu, national language Hindi and also in English. With this context, in this project, an attempt has been made to simulate the human visual system, to identify the type of the language based on visual clues, without reading the contents of the document.

II.SYSTEM DESIGN

Project Flow

TRAINING PHASE

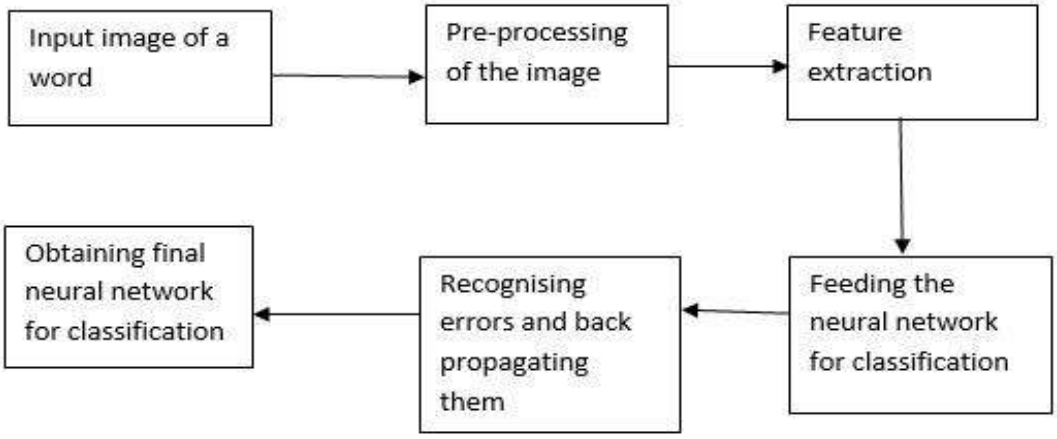

Figure 1. Training Phase

Fig 1. Shows the steps in the training phase for the Script Identification using Artificial Neural Network classifier

TESTING PHASE

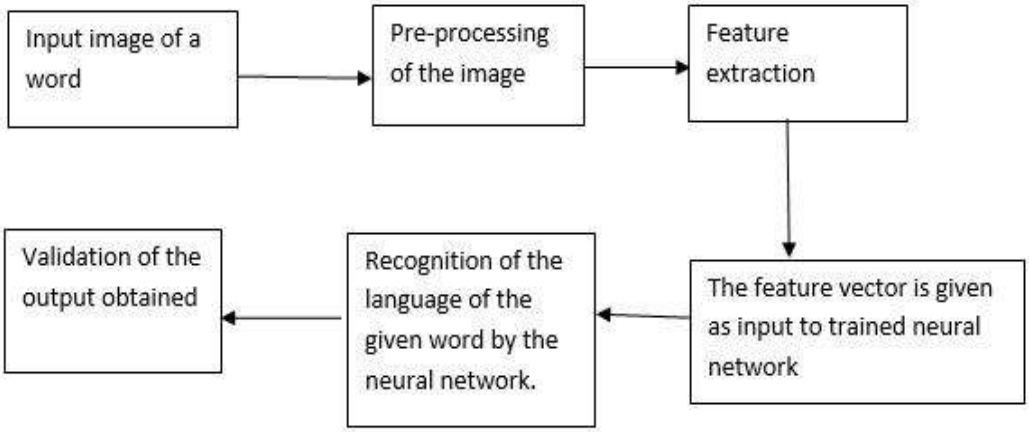

Figure 2. Testing Phase

Fig. 2. Shows the steps in the testing phase for the Script Identification using Artificial Neural Network classifier.

III.IMPLEMENTATION

INPUT TO THE SYSTEM

TRAINING STAGE

During the training stage, the dataset containing images belonging to the three scripts i.e., Telugu, Hindi and English are given as input to the classifier. Three sets of images are given as options to be trained to the classifier. These sets have 900 images, 1200 images, 1500 images respectively which have equal number of images belonging to each script/language. Comparisons on the performance can be made when tested with these three datasets.

The methodology section outline the plan and method that how the study is conducted. This includes Universe of the study, sample of the study,Data and Sources of Data, study's variables and analytical framework. The detailsare as follows;

TESTING STAGE

During the testing stage, the user has two options to test the classifier. First, the classifier can be tested using a single word belonging to either Telugu or Hindi or English. Second, a set of images containing equal number of images belonging to Telugu, Hindi and English is used to test the classifier.

OUTPUT

The trained neural network is tested either by using a single word or by using a set of images. The image and its corresponding class is taken as input when we are testing with a single image. A set of images and the corresponding classes are taken as input when we are testing the network using a set of images. The output is generally evaluated using the Confusion matrix, Receiver Operating Characteristic (ROC) curve and Neural Network Training Performance.

plotconfusion(targets,outputs) function in MATLAB returns a confusion matrix plot for the target and output data in targets and outputs, respectively. On the confusion matrix plot, the rows correspond to the predicted class (Output Class), and the columns show the true class (Target Class). The diagonal cells show for how many (and what percentage) of the examples the trained network correctly estimates the classes of observations. That is, it shows what percentage of the true and predicted classes match. The off diagonal cells show where the classifier has made mistakes. The column on the far right of the plot shows the accuracy for each predicted class, while the row at the bottom of the plot shows the accuracy for each true class. The cell in the bottom right of the plot shows the overall accuracy.

The receiver operating characteristic is a metric used to check the quality of classifiers. For each class of a classifier, roc applies threshold values across the interval [0,1] to outputs. For each threshold, two values are calculated, the True Positive Ratio (TPR) and the False Positive Ratio (FPR). For a particular class *i*, TPR is the number of outputs whose actual and predicted class is class *i*, divided by the number of outputs whose predicted class is class *i*. FPR is the number of outputs whose actual class is not class *i*, but predicted class is class *i*, divided by the number of outputs whose predicted class is not class *i*.

We can visualize the results of this function with **plotroc**. plotroc(targets,outputs) in MATLAB plots the receiver operating characteristic for each output class. The more each curve hugs the left and top edges of the plot, the better the classification.

plotperform tool in MATLAB plots error versus epoch for the training, validation, and test performances of the training record returned by the training function. Generally, the error reduces after more epochs of training, but might start to increase on the validation data set as the network starts overfitting the training data. In the default setup, the training stops after six consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error.

The neural network performance is calculated using crossentropy.

perf = crossentropy(net,targets,outputs,perfWeights) calculates a network performance given targets and outputs, with optional performance weights and other parameters. The function returns a result that heavily penalizes outputs that are extremely inaccurate (y near 1-t), with very little penalty for fairly correct classifications (y near t). Minimizing cross-entropy leads to good classifiers.

The cross-entropy for each pair of output-target elements is calculated as: ce = -t .* log(y).

The aggregate cross-entropy performance is the mean of the individual values:

perf = sum(ce(:))/numel(ce).

IV. RESULTS AND DISCUSSIONS

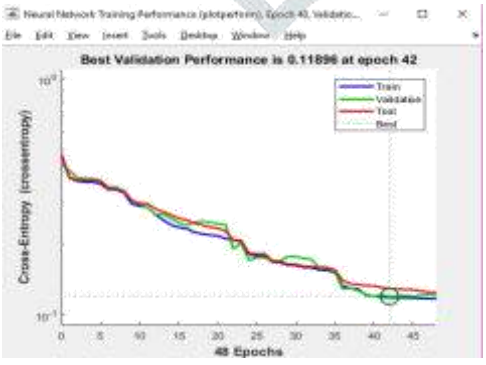

Figure 3. Neural Network Training Performance

Fig 3. Shows plotperform which plots error vs. epoch for the training, validation, and test performances of the training record returned by the function train. Generally, the error reduces after more epochs of training, but might start to increase on the validation

data set as the network starts overfitting the training data. In the default setup, the training stops after six consecutive increases in validation error, and the best performance is taken from the epoch with the lowest validation error.

Figure 4. Confusion matrix after testing

The above fig 4. Shows us the confusion matrices for training, testing and validating the input data. It also shows us the overall confusion matrix. These confusion matrices help us in evaluating the performance of the neural network when tested upon the input data.

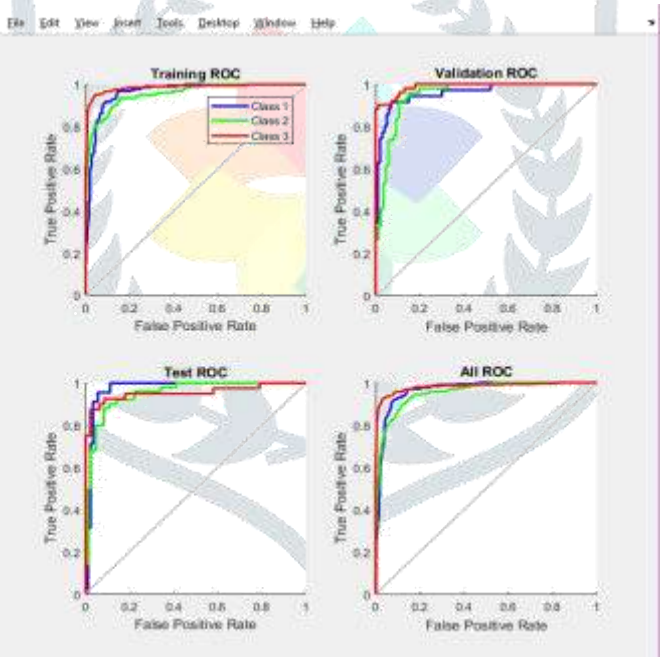

Figure 5. ROC curves after testing

The ROC curve is a graph drawn True Positive Rate (TPR) and False Positive Rate (FPR). The above figure shows us the ROC curves for the training, testing and validation of the input data. It also shows us the overall ROC curve. These ROC curves help us in evaluating the performance of the neural network for a give

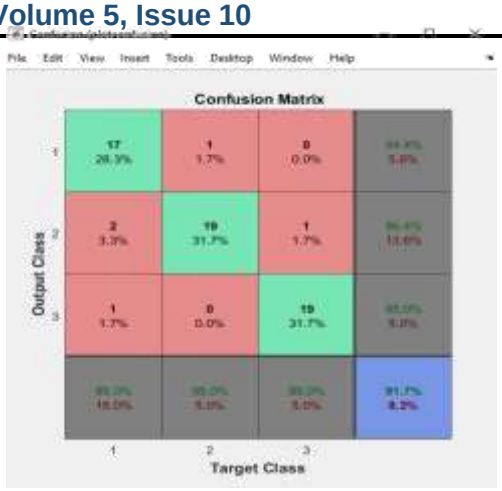

Figure 6. The Confusion matrix for test data

In this figure, the first three diagonal cells show the number and percentage of correct classifications by the trained network. For example 17 words are correctly classified as English. This corresponds to 28.3% of all 60 words. Similarly, 19 words are correctly classified as Telugu. This corresponds to 31.7% of all words. And, 19 words are correctly classified as Hindi. This corresponds to 31.7% of all words.

3 of the English words are incorrectly classified and this corresponds to 4% of all 60 words in the data. Similarly, 1 of the Telugu words is incorrectly classified as English and this corresponds to 1.7% of all data. And, 1 of the Hindi words is incorrectly classified as Telugu and this corresponds to 1.7% of all data

Out of 20 English word predictions, 85.0% are correct and 15% are wrong. Out of 20 Telugu word predictions, 95% are correct and 5% are wrong. Out of 20 Hindi word predictions, 95% are correct and 5% are wrong.
Overall, 91.7% of the predictions are correct and 8.3% are wrong classifications.

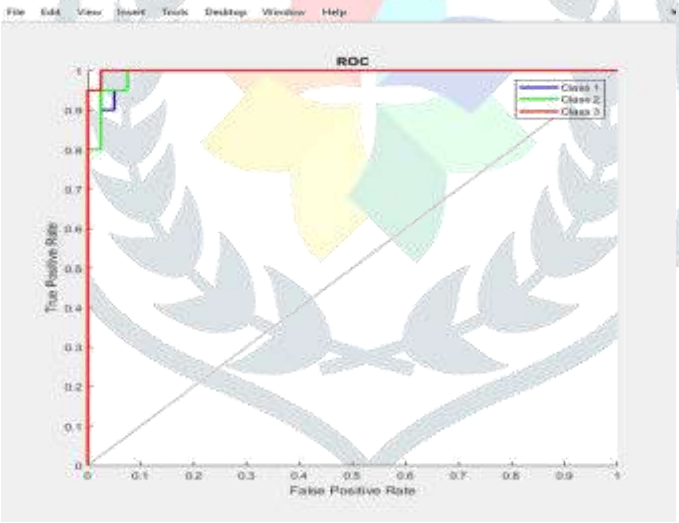

Figure 7. Receiver Operating Characteristic Curve (ROC)

The above figure shows us the ROC curve. The *receiver operating characteristic* is a metric used to check the quality of classifiers. For each class of a classifier, roc applies threshold values across the interval [0, 1] to outputs. For each threshold, two values are calculated, the True Positive Ratio (TPR) and the False Positive Ratio (FPR). For a particular class *i*, TPR is the number of outputs whose actual and predicted class is class *i*, divided by the number of outputs whose predicted class is class *i*. FPR is the number of outputs whose actual class is not class *i*, but predicted class is class *i*, divided by the number of outputs whose predicted class is not class *i*.

DISCUSSIONS

The neural network is trained with three sets of data i.e., 900, 1200, 1500 images with different number of neurons in the hidden layer. The results obtained are compared below

	10	20	30	40	50
900	1.06E-01	7.23E-02	5.55E-02	5.66E-02	7.62E-02
1200	7.21E-02	6.90E-02	7.76E-02	8.17E-02	6.88E-02
1500	1.20E-01	9.50E-02	1.07E-01	9.87E-02	8.19E-02

Table 1. Comparing the performance of the neural network.

Minimizing the performance or Cross-Entropy results in good classification. Lower values are better. Zero means no error.

As we see in the above table 1. the Cross-Entropy tends to decrease from left to right and tends to increase from top to bottom. This implies that when we increase the number of neurons in the hidden layer, the classification gets better and with the increase in the training set images, there is an observation of over classification.

Table 2. Comparing the Output of confusion matrix.

	10	20	30	40	50
900	88.3	91.7	93.3	96.7	93.3
1200	91.7	90.7	90.7	91.7	93.3
1500	88.3	86.7	88.3	88.3	90

As we see in the above table 2. the percentage of the correctly classified words tends to increase from left to right. The percentage also increases when the training set is increased to 1200 images from 900 images but decreases when the training set is increased from 1200 to 1500 images. This implies that as we increase the number of neurons in the hidden layers, the classifications gets better. And if increase the training data images, the classification gets better to some extent and then there is over classification observed.

The maximum percentage of the correctly classified words was 96.7 from the observed result

V. CONCLUSION AND FUTURE WORK

The results of the implementation of this application identifies the words of the three languages i.e., English, Hindi, Telugu.

CONCLUSION

The languages have been identified with different levels of accuracy with different training and testing datasets. In this project, a method to identify and separate text lines of Telugu, Hindi and English scripts from a trilingual document is presented. The approach is based on the analysis of the top and bottom profiles of individual text lines and hence does not require any character or word segmentation. The experimental results show that the algorithm used is effective and good enough to identify and separate the three language portions of the document.

FUTURE WORK

Future work is to develop script identification model at line level or paragraphs for the text with the words printed in different scripts. These features can be tested on other algorithms and features. Scripts of different fonts, font size can be identified. This project is limited only to identifying words. The project can be extended to identify the script type at line level by segmenting the text line into words. Our future work is to consider identification of the script type at word level. Further this project extends to identify combination of different Fonts with different Font sizes and for identification of hand written documents also. The experimental results show that the algorithm used is effective and good enough to identify and separate the three language portions of the document, which further helps to feed individual language regions to specific OCR system. Our future work is to develop a system that can identify other Indian and foreign languages. Also testing the features with various other algorithms and comparing the results can be done in the future. In this project we have extracted 15 features, future work can be done on increasing the number of features and including other features.

VI. REFERENCES

[1] M. C. Padma, P. A. Vijaya, "Script Identification from Trilingual documents using Profile based features", International Journal of Computer Science and Applications, Techno mathematics Research Foundation Vol. 7 No. 4, pp. 16 - 33 , 2010.

[2] C.R.K Reddy, M. Swamy Das, D. Sandhya Rani, "Heuristic Based Script Identification From multi Lingual text documents", 2012.

[3] Priyanka P. Yeotikar, P. R. Deshmukh, "Script Identification of Text Words from Multilingual Indian Document", International Journal of Computer Applications (0975 – 8887) National Level Technical Conference, X-PLORE 13.

[4] Vitaly Ablavsky, Mark R. Stevens, "Automatic Feature Selection with Applications to Script Identification of Degraded Documents", Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003).

[5] G. Srinivas Rao, Mohammed Imanuddin, B. Hari Kumar, " Script Identification of Telugu, English and Hindi Document Image", International Journal of Advanced Engineering and Global Technology Vol-2, Issue-2, February 2014.

[6] D S Guru, M Ravikumar, B S Harish, " A Review on Offline Handwritten Script Identification", International Journal of Computer Applications (0975 – 8878) on National Conference on Advanced Computing and Communications - NCACC, April 2012

[7] Padma M.C. and Nagabhushan P., (2002), Horizontal and Vertical Linear Edge Features as Useful Clues in the Discrimination of Multilingual (Kannada, Hindi and English) Machine Printed Documents, proc. of National Workshop on Computer Vision, Graphics and Image Processing (WVGIP), 204-209.