CS 725 : Foundations of Machine Learning

Optical Character Recognition for Telugu Script

Viswa Virinchi(130070038), Sai Santhosh(130070042), Sashank Reddy (130070046), Ravi Kishore(130070048)

ABSTRACT

Optical Character Recognition (OCR) is the electronic conversion of scanned images of handwritten text into machine encoded text. In this project various image pre-processing, features extraction and classification algorithms have been explored and compared, to design high performance OCR for Telugu script. The best performance obtained with numerical data is **95.88%** using **CNN** technique and for character data is **80.57%** with CNN technique.

1. Motivation

Optical Character Recognition (OCR) is a field of research in pattern recognition, artificial intelligence and machine learning. OCR is a mechanism to convert machine printed or handwritten characters to digitized script.This field is broadly divided into two parts, Online and offline character recognition. Off-line Character recognition further divided into two parts, machine printed and handwritten character recognition. In handwritten Character Recognition, there are lots of problems as compared to machine printed document because different people have different writing styles, the size of pen-tip and some people have skewness in their writing etc. All these challenges make OCR a challenging problem to solve.

The problem of Telugu script is even more challenging due the complexity of the script. A single character can be formed by a single vowel or a single consonant or it can be a compound character consisting of vowels and consonants. Telugu script consists of 14 vowels and and 37 consonants , different combinations of these vowels and consonants leads to around 500 unique glyphs.

2. Training and Validation data

Handwritten character data-set was obtained from “HPL Isolated Handwritten Telugu Character Dataset”. Numerical dataset was obtained from ”CMATERdb: The pattern recognition database repository”. The character data consists of handwritten data by 146 users, each user data has around 320 images. The total number of samples are around 47,000. From this 60% is used for training, 20% is used for the testing and the remaining 20% for cross-validation. The input data from all the users was randomly shuffled and was then split according to the description above.

The numerical dataset consists of consists of 3000 samples i.e 300 samples representing each digit.In this case around 60% is used for training , 20% is used for testing and the remaining 20%

is used for cross-validation. Even in this case the input data was shuffled and the split accordingly.

3. Methodology

3.1. Pre-processing and Feature Extraction

The images in the dataset were large and of different sizes. A 7x7 min filter followed by resizing to maximum length of 64 was performed on the images. The images were then padded with white to get 64x64 images followed by dilation.

The Images are of size 64x64. Divide images into 64 zones of size 8x8. The feature vector is sum of all pixel intensities in the zone.

3.2. Classification and Architecture

3.2.1. Numerical data :

After the above feature extraction technique, while choosing the classifier algorithm for OCR the following three classifiers were analyzed.

3.2.1.1. Artificial Neural Network

The ANN consists of 1 hidden layer an input layer and an output layer:

- Input Layer
- Dense 1024 “relu”
- Output (Dense 10 “softmax”)
- Loss Function : Cross-entropy Loss

3.2.1.2. Convolutional Neural Network

The CNN consists of of 4 hidden layers an input layer and an output layer :

- Convolution 2D 5x5 ,30,’relu’ then MaxPooling 2D 2x2
- Convolution 2D 3x3 15 ‘relu’ then MaxPooling 2D 2x2
- Dense 128 ‘relu’
- Dense 50 with ‘Relu’
- Output with ‘softmax’
- Loss function = cross entropy loss

3.2.1.3. Support Vector Classifier

A support vector classifier was implemented using SciKit learn’s SVM library , with “RBF” kernal and “linear” giving better results .The rest of the parameters were set to default. The accuracy was **92.83%**

3.2.2. Character Set data

After the above feature extraction technique, while choosing the classifier algorithm for OCR the following three classifiers were analyzed.

3.2.2.1. Artificial Neural Network

The ANN consists of 1 hidden layer an input layer and an output layer:

- Input Layer
- Dense 1024 “relu”
- Output (Dense 10 “softmax”)
- Loss Function : Cross-entropy Loss

3.2.2.2. Convolutional Neural Network

The CNN consists of of 5 hidden layers an input layer and an output layer :

- Convolution 2D 7x7 ,1,’relu’ then batch normalization followed MaxPooling 2D 2x2
- Convolution 2D 5x5 64 ‘relu’ then batch normalization followed MaxPooling 2D 2x2
- Convolution 2D 3x3 128 ‘relu’ then batch normalization followed MaxPooling 2D 2x2
- Convolution 2D 3x3 256 ‘relu’ then batch normalization followed MaxPooling 2D 2x2
- Dense 1024 with ‘Relu’
- Output with ‘softmax’
- Loss function : cross entropy loss

3.2.2.3. Support Vector Classifier

A support vector classifier was implemented using SciKit learn’s SVM library , with “RBF” kernel and “linear” giving better results

- Linear kernel with C = 1 and gamma = 0.1 gave an accuracy of 62%
- A “Polynomial” kernel gave an accuracy of around 58%
- “Linear” kernel with a very high value of C gave an accuracy of 50% only.

3.2.2.4. Using Zoning Features

We find the Euclidean distance between column matrix of test image and each of training image and then apply K-Nearest Neighbour with k=7

4. Results and Conclusions

Numerical Data:

Classifier	Accuracy
Artificial Neural Network	91.33%
Convolutional Neural Network	95.88%
Support Vector Classifier	92.83%

Character Data:

Classifier	Accuracy
Artificial Neural Network	72.74%
Convolutional Neural Network	80.57%
Support Vector Classifier	62.8%
Using Zoning Features	64.48%

Installed CUDA on an NVIDIA machine to speed up the learning process.The speed improved by almost 4 times.

5. Future Work

- Character Recognition of FONTED text
- Segmentation can be done to implement a proper script recognition scheme from characters
- Improving Telugu character recognition rate

6. References and Acknowledgements

- http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7021817
- http://deeplearning.net/tutorial/lenet.html

- http://cs229.stanford.edu/proj2011/Moparti%20OCR%20for%20telugu%20script.pdf
- https://medium.com/towards-data-science/devanagari-script-character-recognition-using-machine-learning-6006b40fa6a9
- http://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

7. Individual contributions

1. Viswa Virinchi:

   CNN Implementation for Numerical and Character Datasets

2. Sai Santhosh Kota:

   CNN and ANN Implementation for Numerical Datasets

   Image Pre-processing

3. Sashank Reddy:

   Image Pre-processing

   Zoning Feature Methods

4. Ravi Kishore:

   Improving accuracy by cross validation

   SVM for numerical dataset and character datasets

All of us tried to implement OCR with devnagri dataset as a better dataset was available compared to the OCR.