Optical Character Recognition for Devanagari Script (Interim Report)

Anand
2017218
IIIT-Delhi, India
anand17218@iiitd.ac.in

Akash Singh
2017013
IIIT-Delhi, India
akash17013@iiitd.ac.in

Atul Anand
2017284
IIIT-Delhi, India
atul17284@iiitd.ac.in

1. Introduction

We aim to develop a Machine Learning model which identifies any handwritten character as one of the 46 characters available in Devanagari Script which is used by around 120 Indo-Aryan Languages and thus such work is helpful for a huge population.

Unlike Latin characters used in English, all Devanagari characters of a word are written in a hanging baseline without inter-character separation. So, developing OCR for such script can be challenging. This OCR system can help editing and economic sharing of existing printed texts of this script, which is otherwise not possible with their scanned counterparts.

2. Related Work

Machine Learning has been explored for character recognition and researchers have used various algorithms and feature extraction methods to improve the performance of the model.

The table shows the state of art work on using machine learning models for Devanagari character recognition dataset.

Source	Model	Accuracy
S. Acharya [1]	CNN	98.47%
R.R. Kabra [2]	CAE+SVM	96%

3. Data set and Evaluation

The total data has 92000 samples with 1024 (32x32) features. The data is divided into training and testing in the ratio 85:15. We used randomization for better validation and uniformity of data samples used to train

Figure 1. Sample image containing 32x32 gray scale pixels including 2 pixels of padding from each side

data. The table gives a summary about the information of the data.

3.1. Feature Extraction

Correlation between feature and result was enquired for all features.It was found that around 23% of data had almost no correlation with results. Another 6% features were having low correlation (less than 0.01). This was mostly because already there was a padding of 2 pixels from each side(240 pixels used in padding) and anyhow, letter is mostly placed in centre of image, hence only *central features* contribute high.

Total Size	92000
Training Size	78200
Testing Size	13800
Features	1024
After Extraction	722

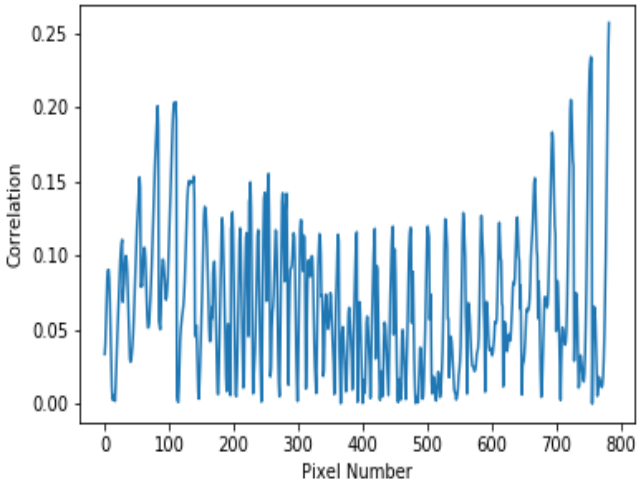

Figure 2. Correlation of features with output variable (*after dropping padding*)

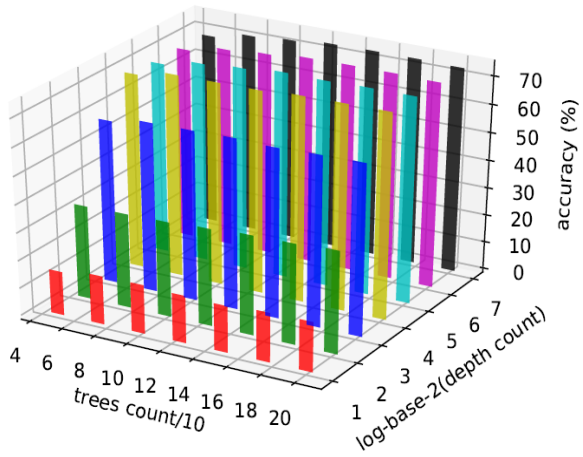

Figure 3. Random Forest Classifier with different tree counts and depth

3.2. Evaluation Metrics

For our dataset, accuracy is a sufficient measure of the performance of the model as the dataset is perfectly balanced. Each class has equal number of samples hence we can rely on accuracy as an evaluation metric. Since the number of classes are very high, we are not going to use other evaluation metrics like confusion matrix and ROC-AUC curve as it will result in high computational complexity and would be harder to visualize in our case.

4. Analysis and Progress

The main objective was to get a good accuracy. We initially used logistic regression, SVM, random forest and K-nearest neighbour classifiers.

4.1. Challenges Faced

Training 92000 samples was computationally very heavy. For tuning we used grid search to get the best parameters and used the best parameters to further train the model. We then evaluated various metrics and compared them before and after the tuning.

4.2. Design Choices and Progress

We trained our data on different models and

5. Results

The table gives a summary of the results obtained.

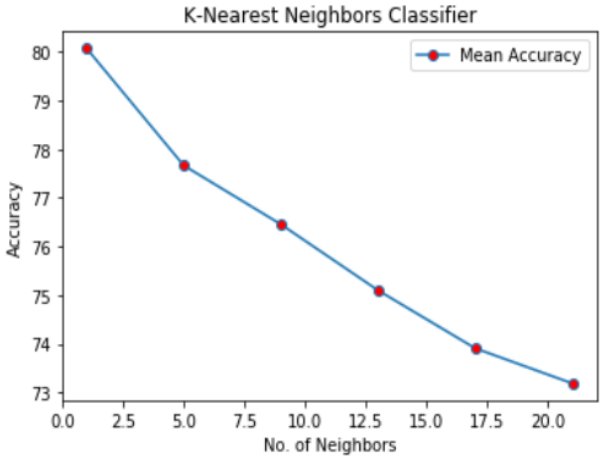

Figure 4. K-Nearest Neighbour classifier with different K values

Model	Accuracy
Logistic Regression	65.9%
Random Forest	83.1%
SVM with Linear Kernel	74.44%
SVM with RBF Kernel	82.44%
SVM with degree 2 Polynomial Kernel	82.52%
KNN	83%

We used logistic regression and due to its average performance, we ruled out Logistic and other linear classifiers as they may not be optimal for this data set. Then, decision trees were used which again gave average results on Test Set due to over-fitting on training set.

Later, Random Forests were used which use Ensemble Learning, hence giving better results.AUC scores of random forest and gradient boosted trees tell that both the trained models are equally good but the false positive rate is lower for gradient boosted trees and accuracy is higher.

6. Future Work

We wish to explore more machine learning algorithms including Neural Networks, specifically Convolutional Neural Networks(CNN) which mainly includes following parts :

- Convolutional Layer
- Pooling Layer
- Fully Connected Layer

Following are the roles each member have followed and will follow for the rest of the project duration and evaluation:

- Anand: Data visualization , pre-processing and outlier removal.
- Akash Singh: Model and parameter selection, feature Extraction, training and analysis
- Atul Anand: Parameter tuning, model improvement and analysis

References

[1] Deep learning based large scale handwritten Devanagari character recognition: https://ieeexplore.ieee.org/document/7400041

[2] Contractive autoencoder and SVM for recognition of handwritten Devanagari numerals https://ieeexplore.ieee.org/abstract/document/8122142

[3] UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset#