OCR of Devnagari Characters

Final Project Report for CS 6350

Yogesh Mishra   Nishant Agarwal

1 Introduction

Optical character recognition (**OCR**) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. It is widely used as a form of data entry from printed paper data records, whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation. It is a common method for digital exploration as well as updation of data and used in machine processes such as machine translation, text-to-speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

OCR is a widely researched field and it spans over a lot of languages. English has been the widely researched language for OCR, but with time researchers have worked on different languages in order to digitize text. Each language has unique characteristics and vary widely from each other is many aspects. The task is very challenging, but the outcome of being able to digitize text and making it accessible to the world is a huge achievement.

We have taken up OCR of Devnagari Script Characters as our problem. This script is used in Hindi language. It is interesting to us as even today there are millions of documents that are waiting to be digitized and some of them are in pretty bad conditions. We found the task very exciting and challenging as OCR would help the world get access to rare manuscripts and documents which would become easily and always accessible.

In this report we present a comparison of performance of different classifiers on 3 forms of feature extraction methods: Binarization, Histogram of Oriented Gradients and Zoning.

2 Work

Our dataset [4] contains 110 unique characters, including complex characters and possible segmented characters.The first step towards our project was extracting features from our images. We used computer vision techniques using **OpenCV**. We began by converting images to grayscale and binarizing the pixel vales. We worked with two sets of image sizes, 32×32 and 64×64, to get a push forward. We got 1024 and 4096 features respectively.

Using the **KNN** and **SVM** code from our assignments we modified the code to perform Multilable clasification. We used a One-vs-All approach for multi-classification instead of All-vs-All as for 110 lables we would have to train $\sim$ 10000 classifiers.It would have made classification very slow. The accuracies received on the extracted features for binarized images were 3% and 8% respectively for $k = 1$ and $C = 15$. We read upon on research papers [1, 2, 3] about the ways of extracting features for handwritten characters and found that simple binarization usually gives very bad accuracies and mostly is used as a base steps for other complicated feature extractions. We selected two methods: **HoG**(Histogram of Oriented Gradients) and **Zoning** from the papers. HoG gives the pixel density values of the image based on the number of bins and block sizes. Zoning is the concept of finding the number of intersections, horizontal and vertical lines, distinct markers such as curves etc as features.

We started with small number of features,
324 and 85 features respectively for each feature extraction method. At this point we modified our **Perceptron** code to perform multi-label classification. We performed experiments using our modified KNN, SVM and Perceptron. We also used scikit library to also run test using Naive Bayes and KMeans.

We faced a lot of difficult issues during experiments as we were not getting very good accuracies and our code was taking a lot of time to run. We used cross validation for Perceptron and SVM to find the best parameters. We used $[0.0001, 0.001, 0.01, 0.1, 1.0, 2.0]$ as our rates, $[0.1, 1.0, 0.25, 3.5, 4.0, 0.05, 0.2, 0.4, 0.3, 0.5]$ was used for margins and $[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$ was used as values for C. Even after performing cross validation our accuracies were not satisfactory. So we realized that we were not able to extract enough features to distinguish between the labels. So we used the concept of feature expansion. We performed zoning in two different ways and concatanated the output of both the methods to make a single feature vector which increased our accuracy. We also tried quadratic feature expansion for varied experiments. We increased the dimensions of our feature vectors too in the case of HoG in order to get better accuracies.

All these concepts from class finally led us to an accuracy jump from 3% for binarization to 78% for HoG features.

3 Results

The below graphs provide comparisons of classifier accuracy on varied feature extraction methods.
Classifiers are plotted on the x-axis and their respective accuracy on y-axis. Different colors are used to represent feature-extraction method. For eg: Zoning-85 means Zoning is Extraction method and 85 Features were extracted. In General HoG Feature extraction

method has fared better than the Zoning. We got the best accuracy of $\sim 79\%$ with SVM for HoG-3780. Zoning accuracy decreased a lot with 10010 features due to curse of dimensionality. As we can see from the Figure 2, using Binarization

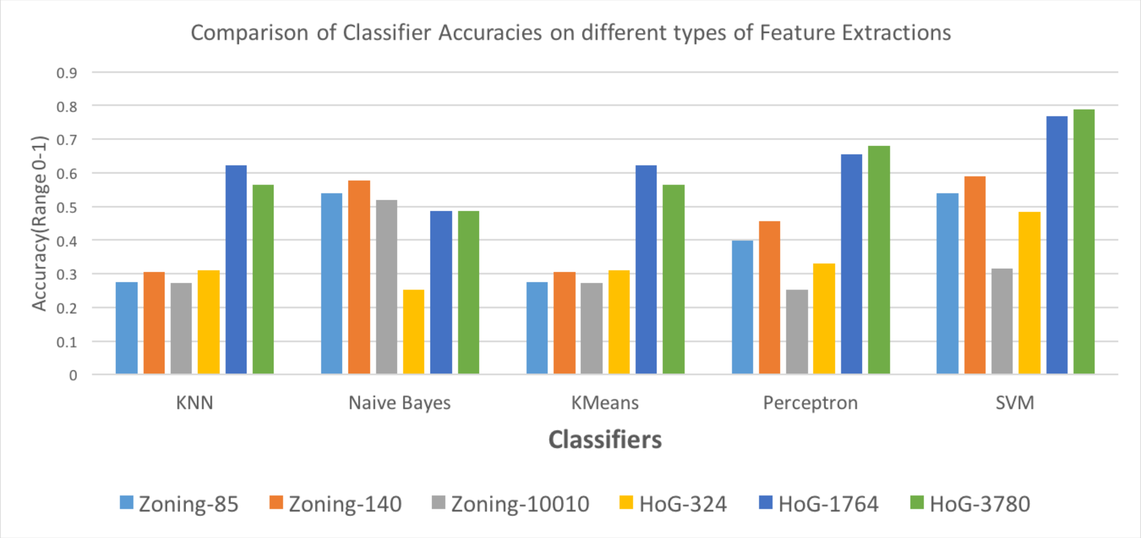


Figure 1: Comparison graph of classifiers on HoG and Zoning Feature Extraction

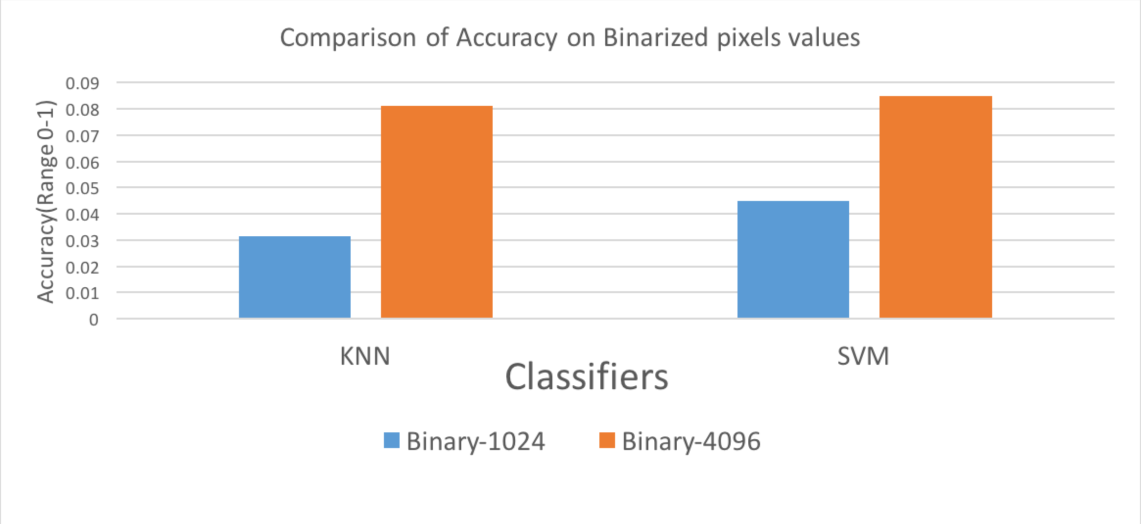
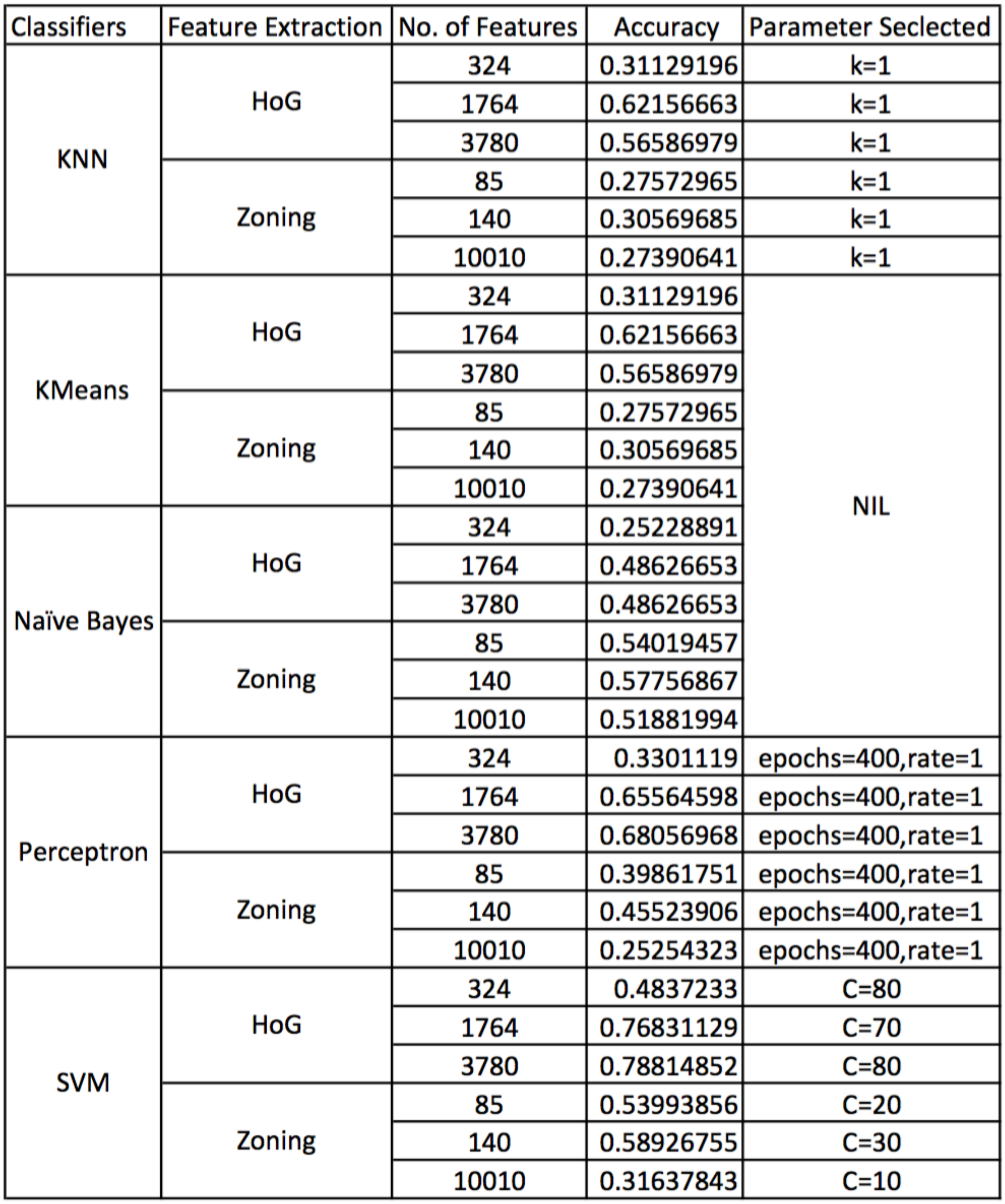


Figure 2: Comparison graph of classifiers on binarized pixel value features of images



Classifiers	Feature Extraction	No. of Features	Accuracy	Parameter Seclected
KNN	HoG	324	0.31129196	k=1
		1764	0.62156663	k=1
		3780	0.56586979	k=1
	Zoning	85	0.27572965	k=1
		140	0.30569685	k=1
		10010	0.27390641	k=1
KMeans	HoG	324	0.31129196	NIL
		1764	0.62156663	
		3780	0.56586979	
	Zoning	85	0.27572965	
		140	0.30569685	
		10010	0.27390641	
Naïve Bayes	HoG	324	0.25228891	
		1764	0.48626653	
		3780	0.48626653	
	Zoning	85	0.54019457	
		140	0.57756867	
		10010	0.51881994	
Perceptron	HoG	324	0.3301119	epochs=400,rate=1
		1764	0.65564598	epochs=400,rate=1
		3780	0.68056968	epochs=400,rate=1
	Zoning	85	0.39861751	epochs=400,rate=1
		140	0.45523906	epochs=400,rate=1
		10010	0.25254323	epochs=400,rate=1
SVM	HoG	324	0.4837233	C=80
		1764	0.76831129	C=70
		3780	0.78814852	C=80
	Zoning	85	0.53993856	C=20
		140	0.58926755	C=30
		10010	0.31637843	C=10

Figure 3: Table of experiment results

4 Conclusion

The project helps us to understand the aspects of machine learning and its practical applications. As noticed, machine learning is a small module while trying to solve a real world problem. We noticed during our implementation that feature extraction plays a very big role and for that domain knowledge is very necessary. Proper selection of features is very important. As seen in the results, we had tried 3 type of feature extraction methods and all of them performed very differently. Still, the feature extraction and selection was not accurate enough to get the desired accuracy of higher 90's.

One of the fascinating observations was that KNN and KMeans gave the same accuracy. The machine learning course was a big help in understanding the problem. We have tried working on OCR previously,about a year back, but due to lack of knowledge on the subject we wary and did not know how to move ahead. After the course, we were able to understand the problems better and also move ahead with a proper plan of action.

5 Future Work

The future path for this project is:

1. Preprocessing of documents
2. Sentence and Word segmentation
3. Character Segmentation
4. Try more feature extraction methods
5. Apply Deep Learning for classification
6. Understanding word meanings

References

[1] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Dipak Kumar Basu, and Mahantapas Kundu. Combining multiple feature extraction techniques for handwritten devnagari character recognition. In *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*, pages 1–6. IEEE, 2008.

[2] Deepali R Birajdar and Manasi M Patil. Recognition of off-line handwritten devanagari characters using combinational feature extraction. *International Journal of Computer Applications*, 120(3), 2015.

[3] Aditi Goyal, Kartikay Khandelwal, and Piyush Keshri. Optical character recognition for handwritten hindi, 2010.

[4] HP Labs. Hp labs india indic handwriting datasets. `http://lipitk.sourceforge.net/datasets/dvngchardata.htm`.