Optical Character Recognition for Devanagari script

CS 6384: Computer Vision Project Final Report

Prabhleen Kaur Saini[1]
University of Texas, Dallas

Ujwal Shah Sandeep[2]
University of Texas, Dallas

P Sai Pratyush Varma[3]
University of Texas, Dallas

1. Abstract

Developing optical character recognition (OCR) algorithms for printed and handwritten characters is a challenging area of study. While OCR techniques have been widely studied for scripts like Roman, Japanese, Korean, and Chinese, there has been little research done for Indian scripts. The Devanagari script, which is used by around 600 million people in World and serves as the primary script for several Indian languages, presents particular challenges due to its wide variety of conjuncts. To contribute to the recognition of this ancient language and help make it more accessible, we have used the "Devanagari Handwritten Character Dataset" from UCI, which contains 46 classes of characters with 2000 examples each. We have split the dataset into a training set (85%) and a testing set (15%) from the UCI machine learning repository, and our aim is to study previous research in the field, build our own model using the best algorithm and tuning the parameters, and evaluate the results using various metrics such as precision, recall, and F1 score.

2. Introduction

Devanagari OCR is particularly challenging due to the large number of conjuncts in the script, as well as the fact that it is an ancient language. The current project aims to address this issue by adding technology to recognize the ancient language and facilitate greater understanding of our history. For example, archaeologists could benefit from this work, as it would enable them to more easily comprehend this ancient language. Our project focuses on the practical applications of Devanagari OCR. Our method includes a comprehensive examination of OCR and literature review of Devanagari script recognition research to date. We analysed the data to better understand the features we are dealing with and will evaluate deep learning models such as KNN, SVM, and deep CNN from the scikit learn library, as well as construct our own CNN model. Once the model has been trained, we focus on testing and evaluating its analysis, retraining it based on the results until a satisfactory outcome is achieved. Realtime images will be used to test the model's performance, and a graphical user interface will be created once the model is complete. In conclusion, this research paper will explore the challenging area of building OCR algorithms for printed and handwritten Devanagari script. Given the wide use of the script and the fact that it serves as the primary script for multiple Indian languages, including Hindi and Sanskrit, the development of effective OCR technology for this script is of great importance. The current project contributes to this area by studying existing research, experimenting with various deep learning models, and building a CNN model to recognize Devanagari characters. The resulting model could have practical applications for archaeologists and other scholars interested in understanding ancient Indian languages.

3. Related-Work

Optical Character Recognition (OCR) is a field of computer vision and machine learning concerned with the automatic recognition of printed or handwritten text characters. Devanagari script is an important script used in India for various purposes, including literature, education, and official documentation. OCR of Devanagari script has gained significant attention in recent years due to the increasing demand for automated character recognition in various applications. In this literature survey, we will review some of the recent developments and state-of-the-art techniques for OCR of Devanagari script.

In 2013, Kale, K.V., Deshmukh, P.D., Chavan, S.V. proposed a system that used Zernike moment feature extraction technique, which has a rotation invariance property, to generate feature descriptors for each zone of the pre-classified character image. The system further implements SVM and k-NN based classification for recognition. The authors collected a large dataset of basic and compound characters from various age groups of writers, which was used for database creation and named as KVKPR2013. The proposed system achieved an overall recognition rate of 98.37

In 2015, S. Acharya, A. K. Pant and P. K. Gyawali in their research paper titled "Deep Learning Based Large Scale Handwritten Devanagari Character Recognition" introduced a new public image dataset for Devanagari script, called the Devanagari Handwritten Character Dataset (DHCD). The dataset consisted of 92,000 images of 46 different classes of characters segmented from handwritten documents. The paper also highlighted the challenges associated with recognition of Devanagari characters, including the visual similarities between characters and the ability to combine consonant characters with vowels to form additional characters. The authors proposed a deep learning architecture for recognition of Devanagari characters, using a Deep Convolutional Neural Network (CNN). To improve the accuracy of the model, the authors implemented the use of Dropout and dataset increment techniques. The proposed architecture achieves the highest test accuracy of 98.47% on the DHCD dataset. In 2019, Ankit K. Sharma, Dipak M. Adhyaru and Tanish H. Zaveri in their research paper - "A Survey on Devanagari Character Recognition" thoroughly studied and summarised the OCR research activities related to Indian script and the major obstacles. The survey paper presents a comparison of various methods in terms of feature extraction techniques, classifiers, datasets, and accuracy values. The authors acknowledge the remarkable research done in recent years on Devanagari character recognition. However, they also note that there is still a huge scope for new researchers to work in this area. The lack of standard datasets, a global platform, and coordination between researchers are identified as some of the major issues for the slow progress in the area of Indian script recognition. In 2020, "Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset" proposed a fine-tuning approach and analysis of state-of-the-art Deep Convolutional Neural Network (DCNN) designed for Devanagari Handwritten characters classification. The authors generated a new dataset of 5800 isolated images of 58 unique character classes of Devanagari script, including vowels, consonants, and numerals, which is publicly available. They implemented a two-stage VGG16 deep learning model to recognize those characters using two advanced adaptive gradient methods. The first model achieves 94.84% testing accuracy, and the second fine-tuned model achieves 96.55% testing accuracy with a training loss of 0.12, which is a state-of-the-art performance on a small dataset. The paper titled "On the recognition of Devanagari ancient handwritten characters using SIFT and Gabor features' , presents a particularly important use case by recognizing devanagari script engraved on ancient sculptures. They used a novel approach to recognize Devanagari ancient handwritten characters using feature extraction techniques and Support Vector Machine (SVM) classification. The paper highlights the importance of recognizing ancient Devanagari characters for the exploitation of valuable information contained in them. The authors have collected a database of 5484 Devanagari characters from various ancient manuscripts and used SIFT and Gabor filter-based feature extraction techniques to recognize the handwritten characters. Principal Component Analysis (PCA) has been used to reduce the length of the feature vector for reducing the training time of the model and to improve recognition accuracy. The authors have used SVM classifiers with different kernels for classification tasks. The proposed system achieves a recognition accuracy of 91.39% using tenfold cross-validation technique on Gabor filters and poly-SVM classifier.

Table 1. Literature Review Table

Author(s)	Model Used	Year	Dataset	Accuracy
S. Acharya, A. K. Pant and P. K. Gyawali [3]	Deep Convolutional	2015	92,000 images of 46 different classes	98.47%
SHALAKA PRASAD DEORE and ALBERT PRAVIN [4]	two-stage VGG16 deep learning model	2020	5800 isolated images of 58 unique character classes	96.55%
Narang, Sonika Rani, Manish Kumar Jindal, Shruti Ahuja, and Munish Kumar [5]	Gabor filters and poly-SVM classifier	2020	5484 Devanagari characters from various ancient manuscripts	91.39%
Kale, Karbhari and, Deshmukh, Prapti and Chavan, Shriniwas and Kazi, Majharoddin and Rode, Yogesh [6]	SVM and k-NN classifier	2021	27000 handwritten character	SVM - 98.37%, KNN - 95.82%

4. Methods

4.1. K-Nearest Neighbours

The machine learning technique known as KNN, or k-Nearest Neighbors, is used for classification and regression applications. A new data point's class or value can be predicted using a KNN model by comparing it to previously labeled data points in the training set and finding the k-nearest neighbors.Choosing a suitable value for k (the number of neighbors to consider), picking an appropriate distance measure, and ensuring that the data is correctly scaled and normalized to minimize any bias towards any single feature or characteristic are some essential concerns when utilizing the KNN model.

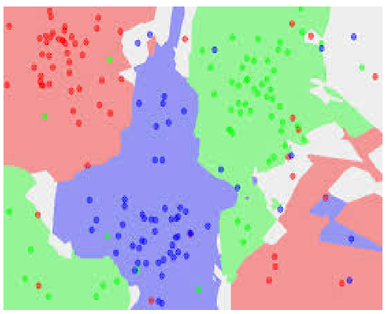


Figure 1. Example of KNN multi-class

4.2. Support-Vector-Machine

4.2.1 Linear Support-Vector-Machine

SVM, or Support Vector Machine, is a well-known machine learning technique used for both classification and regression applications. KNN, or k-Nearest Neighbors, is a form of machine learning. The algorithm is used in the context of linear SVM to identify the ideal linear boundary (or hyperplane) that categorizes the input data points.Linear SVM has several advantages over other linear classification algorithms, such as logistic regression, including better generalization performance and robustness to noise and outliers. However, one limitation of linear SVM is that it can only be used to classify linearly separable data, i.e., data that can be separated by a linear boundary. If the data is not linearly separable, non-linear SVM techniques such as kernel SVM can be used.

4.2.2 Sigmoid Support-Vector-Machine

A variant of the standard Support Vector Machine (SVM) technique known as Sigmoid SVM performs binary classification using a sigmoidal activation function in its decision function. Each input is translated by the sigmoid function into a number between 0 and 1, which can be thought of as the expected likelihood of falling into a particular class. Although sigmoid SVM is effective for dealing with non-linearly separable data, it is less popular than other kernel-based SVM models, like the radial basis function (RBF) kernel. While testing a model, fresh data points are classified according to their anticipated likelihood of falling into a certain class. During training, the model seeks to identify the best parameters that minimize the classification error on the training data.

4.2.3 RBF Support-Vector-Machine

The RBF SVM (Radial Basis Function SVM) technique performs nonlinear classification by means of a kernel function known as the Radial Basis Function. The distance between each data point is used by the RBF kernel function to calculate how similar two data points are by mapping the input data to a high-dimensional feature space. The RBF SVM is frequently used in image recognition, bioinformatics, and finance because it can handle complex data with nonlinear decision boundaries. The RBF SVM learns from training data to determine the hyperparameters' ideal values to reduce classification error. New data points are categorized during testing according to how close they are to the support vectors, which are the main data points that specify the decision boundary.

4.2.4 Polynomial Support-Vector-Machine

The Support Vector Machine algorithm has a variation known as Polynomial SVM that does nonlinear classification using a polynomial kernel function. A nonlinear decision boundary can be fitted by the SVM thanks to the polynomial kernel function, which translates the input data to a higher-dimensional feature space. The degree of the polynomial kernel function is a hyperparameter that controls the decision boundary's complexity. The polynomial SVM is frequently used in image classification, text classification, and bioinformatics and is effective at handling non-linearly separable data. The SVM develops during training the ideal values for the hyperparameters that minimize the classification error on the training set of data.

4.3. Convolutional-Neural-Network

Convolutional neural networks, or CNNs, are a typical type of neural network used for processing images and videos. In order to extract features from the input data, the network is made up of numerous layers of convolutional and pooling operations. With the help

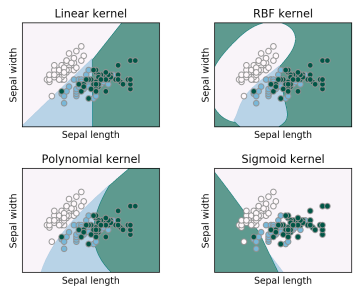


Figure 2. Examples of SVM multi-class

of a series of filters applied by the convolutional layers to the input data, the network can learn patterns and features that are helpful for classification or regression tasks. The pooling layers decrease the dimensionality of the feature maps, preventing overfitting and enhancing network performance. CNNs have attained state-of-the-art performance on numerous benchmark datasets and are very efficient for tasks like object identification, picture segmentation, and classification. In this project we made our own CNN model for classifying 46 classes .Our Model is shown in the below figure 3.

5. Experiments

5.1. Data-set

We have used the "Devanagari Handwritten Character Dataset" from UCI, which contains 46 classes of characters with 2000 examples each. We have split the dataset into a training set (85%) and a testing set (15%) from the UCI machine learning repository [2].Figure 4.The Evaluation matrix for this work used is accuracy .

Table 2. Experiments Table

Models	Accuracy (test)	Train-time (secs)	Test-time (secs)
SVM(RBF)	3.25%	5259.2	1354.7
SVM(sigmoid)	46.4%	1098.2	769.23
SVM(linear)	81.%	508.45	425.46
KNN	89.30%	0.56	82.65
SVM(polynomial=2)	93.29%	1408.7	593.4
CNN(our_model)	98.19%	1688	6

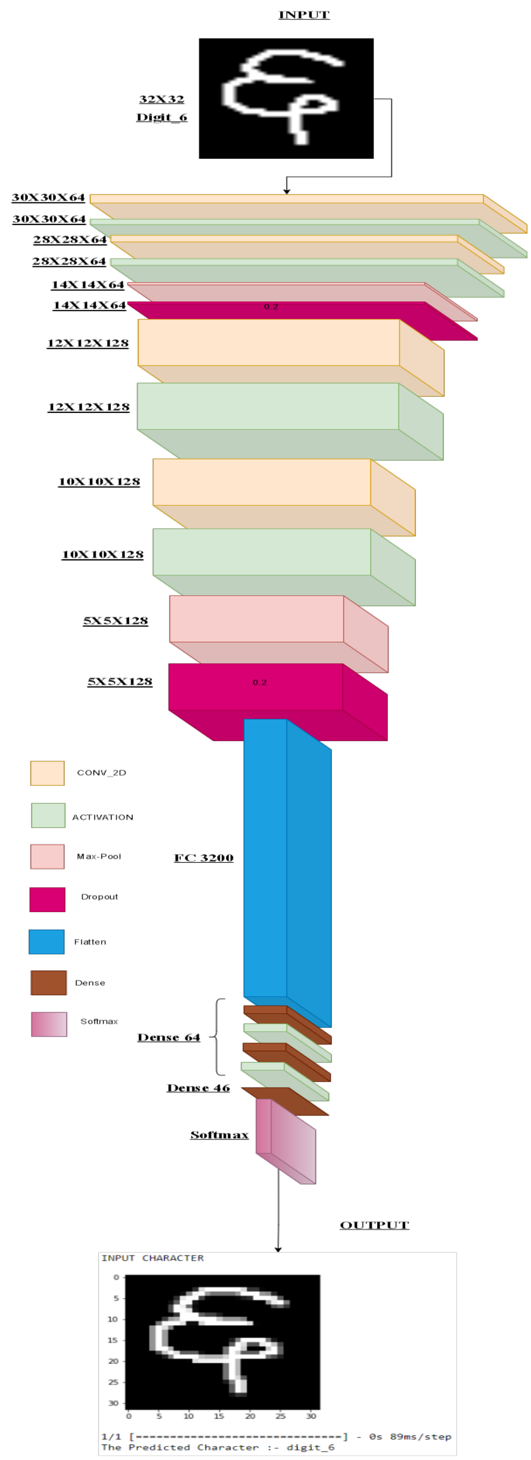


Figure 3. Our CNN MODEL

5.2. Results

We developed our own CNN model, with Trainable parameters: 471,022. The above figure shows the model structure defined with various layers. The result shows 3 graphs: The Model Train accuracy wrt epochs, the Model Loss wrt Epochs then both the accuracy the loss in one graph for better visualization.

Figure 4. Dataset and classes [1]

We could achieved a model accuracy from 0 to 98% in just 15 epochs. The test accuracy was 98.19% with only in 6 seconds for predicting 13,800 examples. The Resulting graphs are show in Figure 5 ,Figure 6 , Figure 7.

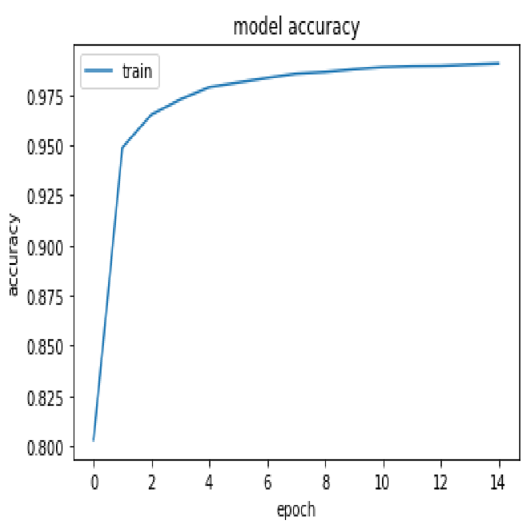


Figure 5. Accuracy graph

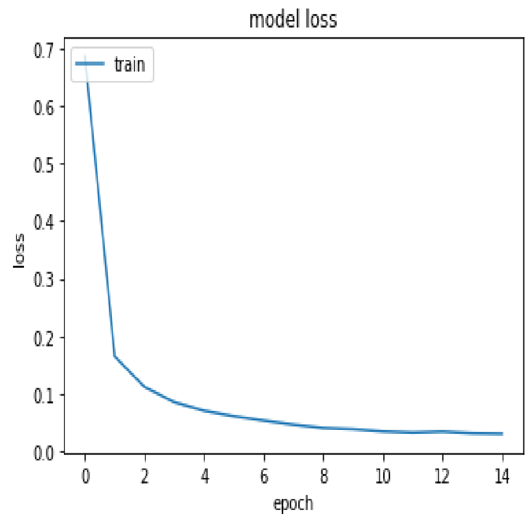


Figure 6. Loss graph

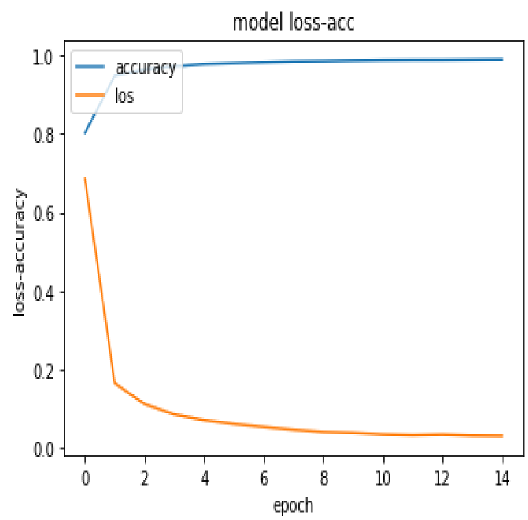
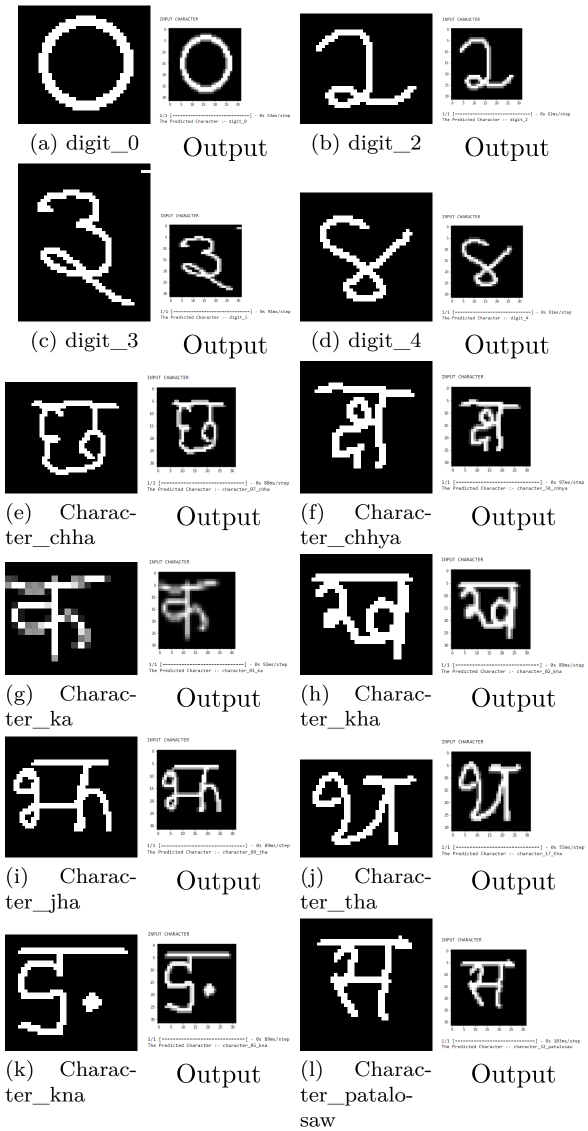


Figure 7. Accuarcy-Loss graph

6. Experimental results

The Left part of the image is the input image right part of the image is the prediction result. Figure No. 8 shows the correct images of Devanagari Letters Digits done by the model. Out of the all the consonants, we took a sample of them, applied our model got the prediction. Figure No. 9 shows the incorrect images of Devanagari Letters Digits done by the model. In figure 9, we used words which came from Machine written Devanagari Keyboard. It can be seen that these words were not classified correctly using our model. Hence, our model faces a little bit difficulty in predicting them.

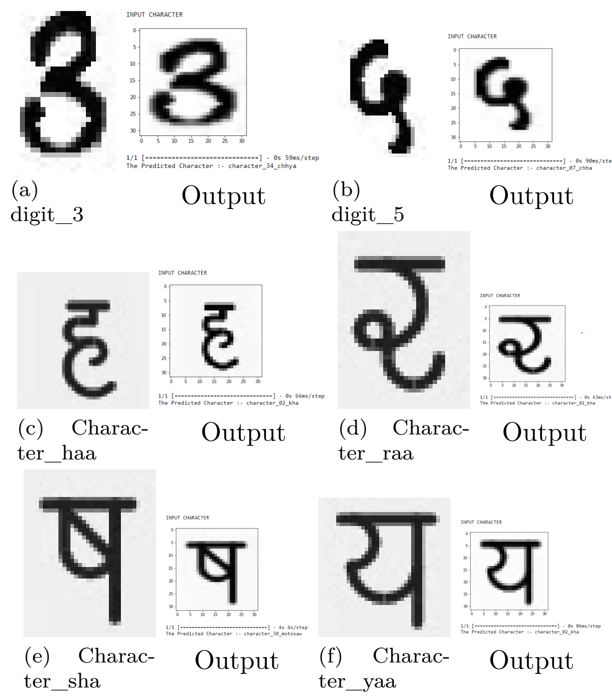


(a) digit__0 Output (b) digit__2 Output (c) digit__3 Output (d) digit__4 Output (e) Character__chha Output (f) Character__chhya Output (g) Character__ka Output (h) Character__kha Output (i) Character__jha Output (j) Character__tha Output (k) Character__kna Output (l) Character__patalosaw Output

Figure 8. Correct Predictions

7. Conclusion

OCR, is one of the most researched problem but there is no one solution to the OCR model. As every language is unique in each way, each language has its own way of defining elements, digits, and consonants. Therefore, a model which helps in detecting Russian language may not be good for Chinese Language, vice versa. etc. Hence, we can infer that Devanagari OCR is tough problem but not impossible. Each specific language OCR may have a different algorithm to solve it it is absolutely fine.Using the table no.2 we can infer that among all the different complex algorithm such as SVM, KNN CNN , CNN provided with one of the best Training accuracy . The testing time in CNN



(a) digit__3 Output (b) digit__5 Output (c) Character__haa Output (d) Character__raa Output (e) Character__sha Output (f) Character__yaa Output

Figure 9. Wrong Predictions

model was the least the training time was comparable to the other models.We also infer that Handwritten Dataset is a double edged sword. Our model has been extensively trained on only Handwritten Devanagari Script characters. Hence, when we included Machine written Devanagari Scripts, it failed to classify them as seen in Figure 9. So, Handwritten dataset behave as a double edged sword. Training a model only of them will helps us classify the handwritten characters correctly but not on Machine written characters.

8. Future Scope

At present, we have accomplished the analysis of data and developed our CNN model. Our off-the-shelf models, namely SVM, KNN, and CNN, have provided, among them we got the best acuuracy in CNN model that is 98.19%. Moving ahead, we plan to refine and optimise our self-built model by hyper-tuning it, considering the data intricacies, to attain even better accuracy. Moreover, we aim to create a graphical user interface (GUI) to facilitate the testing of the model with real-time data.

References

[1] “”Devanagari Handwritten Character Dataset” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Devanagari+Handwritten+Character+Dataset.

[2] S. Acharya, A.K. Pant and P.K. Gyawali â€œDeep Learning Based Large Scale Handwritten Devanagari Character Recognitionâ€ ,In Proceedings of the 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 121-126, 2015.

[3] S. Acharya, A. K. Pant and P. K. Gyawali, ”Deep learning based large scale handwritten Devanagari character recognition,” 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Kathmandu, Nepal, 2015, pp. 1-6, doi: 10.1109/SKIMA.2015.7400041.

[4] Deore, Shalaka Prasad, and Albert Pravin. ”Devanagari handwritten character recognition using fine-tuned deep convolutional neural network on trivial dataset.” Sādhanā 45 (2020): 1-13.

[5] Narang, Sonika Rani, Manish Kumar Jindal, Shruti Ahuja, and Munish Kumar. ”On the recognition of Devanagari ancient handwritten characters using SIFT and Gabor features.” Soft Computing 24 (2020): 17279-17289.

[6] Kale, Karbhari Deshmukh, Prapti Chavan, Shriniwas Kazi, Majharoddin Rode, Yogesh. (2013). Zernike Moment Feature Extraction for Handwritten Devanagari Compound Character Recognition.

5

4

2

2

2

2