Automated Handwritten Devanagari Text Recognition using OCR

Sharwari Ambade
School of Computer Engineering and Technology
MIT Academy of Engineering
Pune, India
sbambade@mitaoe.ac.in

Tushar Gavkhare
School of Computer Engineering and Technology
MIT Academy of Engineering
Pune, India
tpgavkhare@mitaoe.ac.in

Yeeshant Dahikar
School of Computer Engineering and Technology
MIT Academy of Engineering
Pune, India
ysdahikar@mitaoe.ac.in

Tanisha Nazare
School of Computer Engineering and Technology
MIT Academy of Engineering
Pune, India
ttnazare@mitaoe.ac.in

Diptee Ghusse
School of Computer Engineering and Technology
MIT Academy of Engineering
Pune, India
diptee.chikmurge@mitaoe.ac.in

***Abstract*— The variety of handwritten scripts and absence of a corresponding linguistic dictionary make it exceedingly challenging to manually process the handwritten records. The majority of big businesses and small businesses desire to automate the script recognition process. Making machines identify the hand-printed scripts is a major difficulty. Words that have been penned or printed by a human can be recognized after they have been exposed to a particular language. The handwritten scripts should be recognized by machines in the same way. To automate the transfer of human knowledge to computers we use Machine Learning. The suggested research project makes use of multiple machine learning classifiers to automate the character recognition system for Devanagari script. A manually produced database made up by numerous people is being used. Pre-processing, or segmentation, is carried out using the bounding box method. It was attempted to extract features using HOG, LBP, and Gabor. LBP is chosen since it provides the highest level of precision. SVM is used for classification with parameters. The performance of all the classifiers is evaluated using accuracy parameter as success criteria.**

Keywords— Artificial Intelligence (AI), Machine Learning, OCR, Handwritten Text, Devanagari Script, Mathematical Expression

I. Introduction

Handwritten recognition is one of the most interesting areas of research in computer science and difficult because of variability in writing styles of person. Artificial Intelligence (AI) is a specialized field aimed at making computers mimic human intelligence. Important point in AI is to train a machine such that it can read, translate and see the documents. This can be achieved with the help of OCR (Optical Character Recognition) system. OCR basically processes images of text and converts it into system readable form. It has its disadvantages because of correctness. But OCR with Artificial intelligence (AI) and Machine learning (ML) techniques, machines are able to convert physical text into digital more accurately and check errors that may occur during conversion. OCR in the context of education includes printed and handwritten characters as well as text, mathematical expressions, chemical equations, and more. The Devanagari script, used for more than 120 languages, is the fourth most extensively used writing system in the world. It consists of 47 main letters, comprising 14 vowels and 33 consonants.

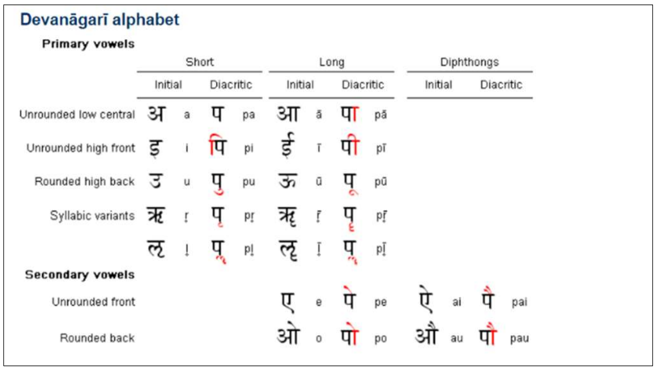


Fig. 1 Devanagari Alphabet

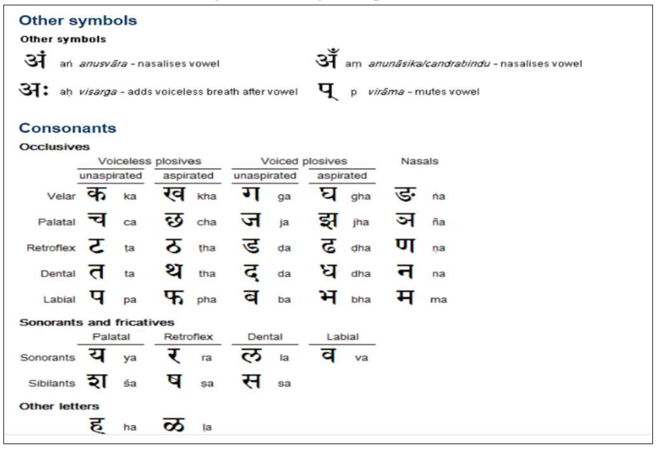


Fig. 2 Other symbols of Devanagari

In comparison to scripts of other languages like English, there are many complexities in Marathi or Hindi language. Therefore, creating an OCR system for Devanagari script is troublesome. Devanagari script contains a wide character set with a lot of vowels and modifiers [2], makingthe OCR process difficult. This is shown above in Fig. 1 and Fig. 2 Human-to-human communication and data transfer are incredibly simple and clear. However, communication between humans and machines is extremely challenging because, despite the fact that a machine is aware of every language that humans are capable of speaking and understanding, it is unable to do so.

Our team is working to create a methodical strategy for recognizing handwritten Devanagari script utilizing different AI and ML techniques.

II. RELATED WORK

An automatic text recognition system has four stages [1]: segmentation, detection, recognition, and classification. Two domains are distinguished from the perspective of theinput image: scene text recognition [2] and recognition of scanned documents. Text detection is not too difficult to handle in former recognition, and various classic pre- processing techniques, including noise removal, binarization, skew detection, text line segmentation, character segmentation, and word segmentation achieved great performance. The X-Y cut algorithm [3], Run-length smearing [4], Hough transform [5], and connected component labelling [6] are all text-line detection techniques. Other text detection techniques locate text in images by using statistical characteristics of nearby local image neighborhoods and texture segmentation. Despitethe fact that the aforementioned approaches had great accuracy, they must include several hand-crafted features and experienced weak robustness [7]. The dataset is restricted to vowel-concatenated characters like ka, ki, kai, and ke. When a dataset contains this collection of characteristics, accuracy improves and it is easier to use the optimal model [8].

Due to a variety of factors, it is difficult to recognize handwritten text. Words are broken up into smaller pieces using the vertical projection for character identification, and a unique horizontal histogram technique is employed for line segmentation [9]. The words are divided up using the bounding box approach [10], which gives accurate segmentation.[11].

In order to identify the pixels in a picture, the Local BinaryPattern (LBP) approach [12] thresholds the region surrounding each pixel and uses the result as a binary number. It is used to define surface texture features. LBP can be used to summarize texture pattern probability into ahistogram. Each and every image pixel needs to have its LBP value determined. LBP examines the points that surround a centre point and determines if they are more or less than the focal point. For the purposes of object detection and picture recognition, the Histogram of Oriented Gradients [20] approach, sometimes known as HOG [20], is employed. Based on feature descriptors, HOG only retains the relevant information while discarding the rest.

HOG may detect shifts in the data by calculating the gradient's amplitude and direction for each pixel and grouping the findings into a 9-bin histogram. The performance and bias of the model can be enhanced by block normalization. Several industries, including as driverless automobiles, augmented reality, and virtual reality, can use HOGs (mostly anything involving image detection). The local structure features encoded in text images using multiscale and multi-orientation Gabor filters [13], which have been widely and successfully employed in word recognition. Two filters: Local filters targeted at accuracy and global filters targeted at processing speed are employed to extract the text's features. Classification [11] is the next step after feature extraction. To recognize characters, statistical and structural aspects are used. Effective precision was achieved by the multi-layer perceptron classifier. A recognition system using the k-NNclassifier and curve let transform has been developed [12].Different methods for reading handwritten and printed scripts have been examined. Convolutional and pooling layers often make up a CNN [14] model. CNN performs effectively for picture classification issues because it performs better for data that are represented as grid structures. The dropout layer is used to deactivate some neurons, and while the model is being trained, it reduces overfitting. The experimental data are converted into a classifier using Support Vector Machines (SVM). By using cross validation accuracy of 94%, the experimental result reveals great performance of these characteristics [15]. The earlier method entails segmenting text into lines, words, and characters before using feature extraction techniques like CNNs or SVMs [16] to run character classifiers. The lattermethod just breaks down sentences and words into individual words before attempting to recognize the complete word at once. In the current scenario, CNN-RNN hybrid networks [17], which use CNNs for feature extraction and RNNs for Connectionist Temporal Classification [18], are used to recognize word labels from word pictures.

This stage works on finding the constraints of the word to be segmented and helps to correct the word even if found error while merging which is done by pre-trained models. These recognition steps recognize the word from a pre-existing words dataset. This helps in working on increased model accuracy and efficiency as per the constraints being handled [19]. This stage is the final solution which helps the merging part to be corrected when error occurs. In order to train this recognition, we use a recognition model.

III. PROPOSED SYSTEM

A. Dataset

We created our own dataset of Devanagari characters for training the model. We used 37*12 characters those are consonants merged with vowels which we called vyanjans. These vynajans total counts to be 37*12. We took handwritings of 150 people and trained our model likely so that we could get the max accuracy. The expression for HOG, LBP and GABOR are given in Eq. (1), (2), (3) respectively. Our dataset was manually fed to model and was then segmented by specific characters which further provided the classification set of data. This data consisted of almost every character from theDevanagari script. The challenges we faced with the dataset were that few writing styles were confusing while some were similar and few were having similar strokes for two different vyanjans as shown in below Fig 3. We faced this challenge by working on large dataset that helps on creatingthe perfect model.

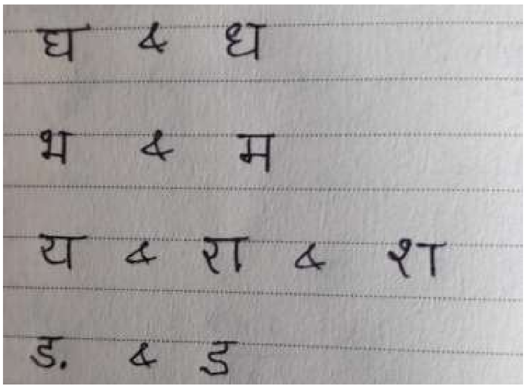


Fig. 3 Alphabets that caused challenges

B. Recognize the character

In this paper, we will use four-stage pipeline as shown in Fig 4, Fig 5 and Fig 6. The first step is segmentation [11], which uses the Bound Box Algorithm [10] method todivide the input image into words and then characters. Theresult is a Devanagari character for each character. The detection and classification stage uses this output as an input before classifying the particular character and returning the letter as an output. A recognition stage is employed to turn these output letters into particular, meaningful words once they have been ordered and sorted. This word output is additionally given to the merge component stage, where it is changed into a particular word[11]. The overall system strives to deliver the most accurate and effective automatic text recognition output possible.

The First stage is segmentation process for conversion of word to characters as shown below in Fig 4. We took words into consideration and performed segmentation by tracing the character boxes using Bound Box Algorithm which in turn provided us with the Characters specific at that location. These characters were then printed as outputs and concat to form a word of specific characters. These characters come from the classification models we used.

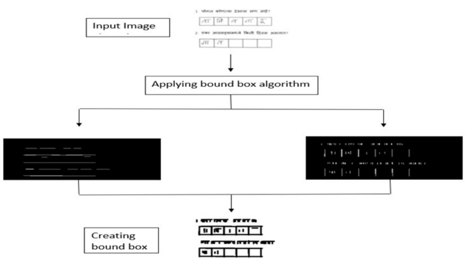


Fig. 4 Segmentation using bound box method

The second stage is taking features from segmented characters. We use Feature extraction to convertcharacter image to specific featured vectors so that we canperform classification [11] based on features. We usealmost 150 handwritten text of specific Devanagari characters. The LBP [12], HOG [20] and GABOR [13]which are Local Binary Pattern, Histogram OrientedGraphs and Texture based feature extraction respectively. Here we compare every algorithm for feature extraction. Our outcomes were quite clear and we accepted LBP for further implementation as HOG being unable to addressminute strokes and creating less accuracy. Similarly,GABOR being texture based gave error as shadows werebeing considered of scanned documents. The LBP gave thebest result as it was able to classify minute details as wellas ignoring texture differences.

HOG: $$f_{bi} \to \frac{f_{bi}}{\sqrt{\|f_{bi}\|^2 + \varepsilon}} \quad (1)$$

LBP: $$\sum_{i=0}^{P-1} s(n_i - G_c)2^i \quad (2)$$

GABOR: $$g(x, y; \alpha, \theta, \varphi, \sigma, \gamma)$$

$$\exp\left(-\frac{x\prime^2 + \gamma^2 y\prime^2}{2\sigma^2}\right) \exp(\mathrm{i}(2\pi \frac{x'}{\alpha} + \varphi)) \quad (3)$$

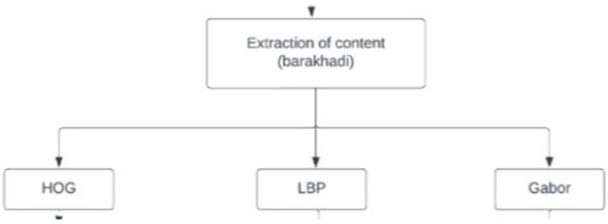


Fig. 5 Feature extraction

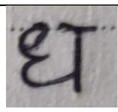
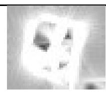
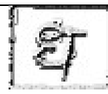
Input	HOG	GABOR	LBP
			

Table 1. Showing feature extraction

The features extracted as shown in Table 1 are used furtherto create models with specific attributes in next stage of classification which classify specific character for letters inword. These classification models are important for automating the classification and minimizing the time consumption. This model is available to use once created. This model helps the system note the character being inputto be classified as a specific label as mentioned in the model [11].

a) Naive Bayes: The Algorithm is good for classification but it was too time consuming the model was taking lots of time. The accuracy of the model was 67%. Eq. (4) demonstrate the expression for Naive Bayes.

$$P\left(\frac{x_i}{y}\right) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp(-\left(x_i - \mu_y\right)^2 / 2\sigma_y^2) \quad (4)$$

b) Decision Tree/ RandomForest/ Extra Tree Regressor: These algorithms work on Feature selection properties so give more accuracy than other classifications. These algorithms were time consuming so were not best suited with accuracy of 70-83%. Eq. (5) demonstrate the expression for Decision Tree/ Random Forest/ Extra Tree Regressor.

Information Gain= Entropy(S) - [(Weighted Avg) *Entropy] (5)

c) MLP Classifier: The MLP Classifier being Multi layered was used with Decision tree and Logistic regression it was bit faster to create models but the accuracy was not acceptable it reached 51%. Eq. (6) demonstrate the expression for MLP Classifier.

$$u(x) = \sum_{i=1}^{n} w_i x_i \quad (6)$$

d) SVM: This model gives best accuracy with 91.66% and model created faster. We used parameters as C and gamma as well as kernel rbf/poly. This helped us to trace specific strokes and classify rather than taking redundant results.

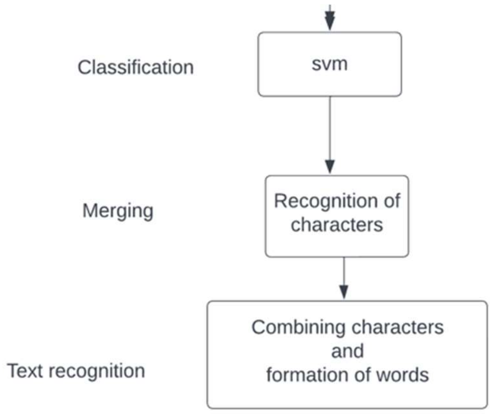


Fig. 6 Classification using SVM

The final stage is recognition where the classified character is merged for further word suggestions to complete the automatic text recognition. The pretrained models are thusused for this automatic text recognition.

IV. RESULT AND CLASSIFICATION

The results consist of various evaluation metrics formed after running various models that works on classification and helps the efficiency of recognition. These models are further compared by their training and test evaluations which results that SVM and LBP provides best accuracy of 91.67% test and 100% train.

Feature Extraction	HOG							
Model	Train				Test			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
SVM	100	100	100	100	75	66	90.91	76.92
MLP	100	100	100	100	91.68	84.68	100	91.67
Decision Tree	100	100	100	100	45.83	41.67	45.46	43.48
Random Forest	100	100	100	100	79.17	71.43	90.91	80
Extra Tree	100	100	100	100	75	69.23	81.82	75
Gradient Boosting	100	100	100	100	62.5	60	54.55	57.14
Ada Boosting	100	100	100	100	79.17	75	81.82	78.26
K Neighbors	89.58	86.79	93.88	90.2	83.33	76.92	90.91	83.33
Naive Bayes	95.83	100	91.84	95.75	70.83	64.29	81.82	72

Table 2. Accuracy, Precision, Recall and F1 using HOG for extraction

Feature Extraction	GABOR							
Model	Train				Test			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
SVM	91.66	91.84	91.84	91.84	75	66.67	90.91	76.92
MLP	51.04	51.04	100	67.59	45.83	45.83	100	62.86
Decision Tree	100	100	100	100	37.5	35.71	45.45	40
Random Forest	100	100	100	100	75	64.71	100	78.57
Extra Tree	100	100	100	100	70.83	62.5	90.91	74.07
Gradient Boosting	100	100	100	100	58.33	52.94	81.81	64.29
Ada Boosting	100	100	100	100	70.83	66.67	72.73	69.57
K Neighbors	81.25	81.63	81.63	81.63	70.83	66.67	72.73	69.6
Naive Bayes	88.54	85.19	93.88	89.32	66.67	57.81	100	73.33

Table 3. Accuracy, Precision, Recall and F1 using GABOR for extraction

Feature Extraction	LBP							
Model	Train				Test			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
SVM	100	100	100	100	91.67	84.62	100	91.67
MLP	51.04	51.04	100	67.59	45.83	45.83	100	62.86
Decision Tree	100	100	100	100	37.5	38.89	63.64	48.28
Random Forest	100	100	100	100	83.33	73.33	100	84.62
Extra Tree	100	100	100	100	87.5	83.33	90.91	86.96
Gradient Boosting	100	100	100	100	67	58,82	90.91	71.43
Ada Boosting	100	100	100	100	66.67	60	81.82	69.23
K Neighbors	88.54	93.18	83.67	88.17	79.17	80	72.72	76.69
Naive Bayes	91.67	95.56	87.8	91.5	70.83	64.3	81.82	72

Table 4. Accuracy, Precision, Recall and F1 using LBP for extraction

a) Accuracy. The accuracy of the model
b) Precision: The precision is quality of positive prediction of model
c) Recall: The recall is model's ability to detect positivesamples
d) F1 score: The harmonic mean of precision and recallwhich depicts predictive performance of model.

We performed the model training using various dataset and feature extractions models. This model work efficiently with LBP and provides maximum accuracy of 91.67% on test data. This result is proved based on Table 2, Table 3 and Table 4.

V. CONCLUSION

This study highlighted the drawbacks of manually processing handwritten data. Next, move on to the suggested model, which implemented a character recognition system for Devanagari script using a number of Machine learning classifiers, including Decision Tree classifier, Nearest Centroid classifier, K Nearest Neighbors classifier, Extra Trees classifiers, and Random Forest classifier. According to the results for all classifiers, the SVM was unquestionably the best classifier among those used. The statistical measuring method used for comparison was classifier accuracy. The precision, recall and f1score supported in distinguishing the best model.

Notations-

Notation	Description
Acc	Accuracy
Pre	Precision
Rec	Recall
F1	F1 score
HOG	Histogram Oriented Graphs
LBP	Local Binary Pattern

VI. REFERENCES

[1] Xingqian Xu, Zhifei Zhang, Zhaowen Wang, Brian Price, Zhonghao Wang, Humphrey Shi (2021). "Rethinking Text Segmentation: A Novel Dataset and a Text-Specific Refinement Approach." Publishes in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 12045-12055

[2] Guanjin Li (2022). "CSNet-PGNet: Algorithm for Scene Text Detection and Recognition in IEEE conference". Published in: 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA).

[3] Weijie Zhou, Xiaoyu Du, Senhao Wang (2021). "Techniques for Image Segmentation Based on Edge Detection in IEEE conference". Published in: 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)..

[4] Rupinder Pal Kaur, M. K. Jindal & Munish Kumar (2020). "Text and graphics segmentation of newspapers printed in Gurmukhi script: a hybrid approach". DOI : 10.1007/s00371-020-01927-0.

[5] Subhranil Das, Rashmi Kumari (2021). "Application of Extended Hough Transform Technique for Stationary Images in Vehicle License Plate". Published in: 2021 6th International Conference for Convergence in Technology (I2CT). ISBN:978-1-7281-8876-8.

[6] Robert, P.; Kavida, A. C. (2021). "Morphological Based Medical Image Processing on Cervical Cytology Cancer Images Using Connected Component Techniques". DOI: 10.2174/1573405617666210707161945.

[7] "A Survey of Models for Constructing Text Features to Classify Texts in Natural Language". IEEE Conference Publication. IEEE Xplore. Published in: 2021 29th Conference of Open Innovations Association (FRUCT). ISBN:978-952-69244-5-8.

[8] "DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition". SpringerLink. DOI:10.1007/s11042-021-10775-6.

[9] K.K.D. Ramesh, G. Kiran Kumar, K. Swapna, Debabrata Datta, S. Suman Rajest (2021). "A Review of Medical Image Segmentation Algorithms". DOI:10.4108/eai.12-4-2021.169184.

[10] MinLiang, Jie-BoHou, XiaobinZhu, ChunYang, JingyanQin, Xu-ChengYin, "Multi-orientation scene text detection with scale-guided regression", Neurocomputing, Volume 461, 21 October 2021, Pages 310-318, j.neucom.2021.07.026

[11] Shilpa Mangesh Pande and Bineet Kumar Jha (2021). "Character Recognition System for Devanagari Script Using Machine Learning Approach". ISBN:978-1-6654-0360-3. DOI: 10.1109/ICCMC51019.2021.9418028.

[12] Cheng-Lin Liu, Masashi Koga and Hiromichi Fugisawa (2005). "Gradient Feature Extraction for Character Recognition". DOI:10.1109/ICDAR.2005.119.

[13] S. Berisha (2009). "Image Classification using Gabor Filter and Machine Learning. M. Tech Thesis, Wake Forest University".

[14] Sandeep Dwarkanath Pande, Pramod Pandurang Jadhav, Rahul Joshi, Amol Dattatray Sawant, Vaibhav Muddebihalkar, Suresh Rathod, Madhuri Navnath urav, Soumitra Das (2021). "Digitization of handwritten Devanagari text using CNN transfer learning – A better customer service support".j.neuri.2021.100016

[15] Warkhede S E,Yadav S K,Thakare V M,Ajmire P El (2021). "Handwritten Recognition of Rajasthani Characters by Classifier SVM", DOI:10.1109/MAHC.2009.1

[16] Amani Ali, Ahmed Ali, Suresh Mallaiah (2022). "Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout". DOI:10.1007/s42979-020-00187-y

[17] R. Geetha, T. Thilagam, T. Padmavathy (2021)."Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks". DOI:10.1007/s00521-020-05556-5

[18] Gowtham Senthil, Nandhakumar, K. Gorthi Rama Krishna, Sai Subrahmanyam (2020). "Handwritten Hindi Word Generation to enable Few Instances Learning of Hindi Documents" – IEEE, DOI: 10.1109/SPCOM50965.2020.9179634

[19] Boya Wang , Jianqing Xu , Junbao Li , Cong Hu , Jeng-Shyang Pan (2017) "Scene text recognition algorithm based on faster RCNN", ISBN:978-1-5386-0844-9

[20] Vharkte, M. N., & Musande, V. B. (2018, December). A Novel Method for Retrieval of Remote Sensing Image Using Wavelet Transform and HOG. In *International Conference on Intelligent Systems Design and Applications* (pp. 540-549). Springer, Cham.