Handwritten Gujarati Character Recognition Using Machine Learning and Deep Learning

Yogiraj Zala[1](✉), Krishn Limbachiya[1], Ankit Sharma[1], and Pooja Shah[2]

[1] Department of Electronics and Instrumentation Engineering, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India
{19bic071,20ftphde43,ankit.sharma}@nirmauni.ac.in

[2] Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar, Gujarat, India
pooja.shah@sot.pdpu.ac.in

Abstract. Gujarati character recognition has always been a field of study in need of improved techniques to reach a satisfactory degree of accuracy. This study developed a technique for predicting handwritten Gujarati numerals using various feature extraction and classification techniques. Deep learning models, including MobileNet, DenseNet, and NasNetMobile, are components of the feature extraction technique. XGBoost, Random Forest, Logistic Regression, Naive Bayes, Stochastic Gradient Descent, Decision Tree, and K-Nearest Neighbor are some of the techniques used to classify Gujarati numerals. This study used over 14,000 images of Gujarati numerals for an experiment and found significant outcomes.

Keywords: Handwritten Gujarati numerals · Numerals classification · Deep Learning · Machine Learning techniques

1 Introduction

A handwritten numeral recognition system can also help with the automatic processing of tax and admission applications, bank checks, and postal codes. Over 55 million people speak Gujarati, a language that is a constituent of the Devanagari family. The Gujarati script resembles Indo-Aryan writing, consisting of 34 consonants and 12 vowels, with 10 numerals, making up the Gujarati alphabet.

Machine learning and deep learning techniques are used to solve the challenges of handwritten Gujarati numeral recognition. Handwritten numerals include a variety of writing styles, thicknesses, and curves, rendering interpretation difficult. Gujarati handwritten character recognition has reportedly received relatively little effort. OCR on Gujarati script is quite difficult since even a small amount of noise from changes in writing style might result in incorrect character classification. For a growing nation like India, the research in this field has the potential to positively affect the human-machine interaction system.

S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 863–873, 2023.
https://doi.org/10.2991/978-94-6463-136-4_76

2 Related Work

Research in the field of Gujarati character recognition is limited. In this section, A brief of all the previous research will be described in this section. In [1], the Authors proposed KNN and Minimum humming distance classifier for Gujarati character recognition and got 67% accuracy while the authors of [2] used KNN and SOM for the same and got 82.36% accuracy. In [3], the authors used ANN and multilayer feed-forward neural network with Back Propagation for Gujarati Numerals recognition and got 88.7% accuracy. The method for Gujarati character recognition using Back propagation on the neural network has been proposed in [4]. In the proposed method, 280 samples were tested and 925 samples were trained which gained 98.5% accuracy. In [5], the method for recognition of English and Gujarati characters used by the authors was KNN and it achieved an accuracy of 99.23%, where a total of 3500 samples were used. Authors in [6] proposed the KNN classification method to recognize Gujarati Characters with 88% accuracy but this method was not able to perform symbol classification. In [7], 7960 Gujarati characters were recognized using KNN and SVM getting 86.6% accuracy. The authors of [8] used the Modified chain code method, Discrete Fourier Transform, Discrete Cosine Transform for feature extraction, and KNN, SVM, and ANN for classification and compared them with one another in which Discrete Cosine Transform gave the highest accuracy of 93%. Gujarati character recognition using Transform Domain (DWT, DCT, and DFT), Structural and Statistical Method, and Geometric Method feature extraction methods was done in [9] along with SVM classifier which got 87.29% accuracy when trained and tested with 7800 samples. [10] is consisting of CNN as a feature extractor and classifier with 90.10% accuracy for the recognition of Gujarati characters. The authors used 4500 samples for the model. The authors of [11] used KNN, SVM, and ANN for Gujarati character recognition using 3000 samples and got 91.6% accuracy. In [12], dynamic time warping (DTW) and Gray Level Co-occurrence Matrix (GLCM) were used by the authors for classification and feature extraction respectively for Gujarati character recognition with 99.4% accuracy. Filtering, edge detection, and morphological transformation were used as feature extractors, and KNN and NNC were used as classifiers by the authors in [13] and an accuracy of 82.0% was achieved. In [14], the authors proposed the Daubechies D-4 wavelets and principal component analysis (PCA) for feature extraction and SVM and KNN for the classification, for Gujarati character recognition, and got 88.4% accuracy. Authors in [15] used histogram of oriented gradients (HOG) for feature extraction, and KNN and SVM for classification, for Gujarati character recognition with 85.8% accuracy. DNN-based recognition was proposed in [16]. In [17], OTSU thresholding algorithm was used. CNN-based Gujarati character recognition was done in [18] with 98.6% accuracy. Authors in [19] trained 8000 samples and then tested 2000 samples for recognition of Gujarati characters using MLP and CNN. Gujarati numeral recognition was done using ANN in [20] with 81.6% accuracy. In [21], the authors used Low-level stroke features for feature extraction and KNN for classification. Naive Bayes classifier and multilayer feed-forward neural network are used in [22] by the authors as classifiers along with the Zoning method as a feature extractor with 11200 training samples and 2800 testing samples for Gujarati numeral recognition. Gujarati numeral recognition was done using ANN, SVM, and naive Bayes on 14000 samples by the authors which was 99.92% accurate. SVM was used in [24] by the authors and chain code detection as a feature extractor for Gujarati numeral recognition with 95% accuracy. Gujarati character recognition was done in [25] using SVM and RBF on 13000 samples and a combination of freeman chain code and zoning-based chain code as a feature extractor. Authors of [26] used Feature Vector through Structural Decomposition (FVSD), Zone Pattern Matching, and Normalised Cross-Correlation to extract features and SVM and Naïve Bayes as classifiers for Gujarati character recognition. Recognition of 2000 training and 9429 testing samples of Gujarati language was done using SVM by the authors of [27]. In [29], the authors used CNN as a feature extractor and SVM as a classifier for Gujarati numeral recognition which got 96.2% accuracy.

3 Proposed Method

In this study, we proposed the classification of Gujarati numerals using a hybrid approach of ML and DL methods, where CNN models such as MobileNet, DenseNet121, and NASNetMobile were combined and used to extract distinctive features. Further, numerals were identified by the usage of ML classifiers such as Random Forest, Logistic Regression, Naive Byaes, XGBoost classifier, Stochastic Gradient Descent, Decision Tree, and K Nearest Neighbours as shown in Fig. 1. Separately the network performance is calculated by many performance indicators including accuracy, precision, and F1 score.

Image Pre-processing: Image Pre-Processing: In order to provide uniformity, resizing each image in the proposed dataset is the first and most crucial step in OCR for Gujarati numerals. Image rescaling, RGB to Grayscale image conversion, noise removing by image filters, are some of the image pre-processing methods we explored. Pre-processing

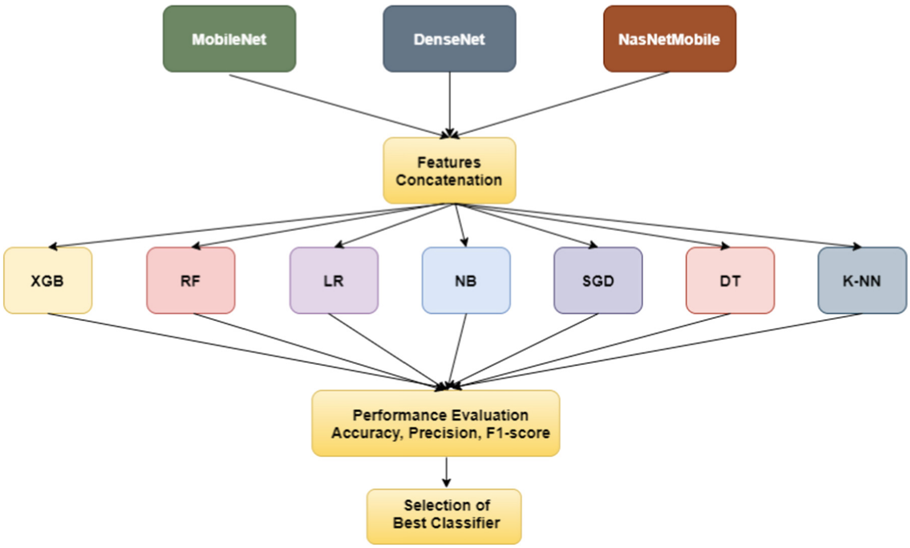

Fig. 1. Proposed Methodology of Fusion Network

improves the quality of a picture by removing unnecessary information, which then facilitates in better feature extraction and, eventually, results in the accurate prediction of Gujarati characters.

Image Feature Extraction: It is one of the most crucial aspects of OCR. The efficiency of the model classifier depends on the developed feature set because the feature set is a substantial and comprehensive set that depicts every aspect of the source image. It is a smaller set of data than the original image data, which makes the training process more rapid than usual. There is a requirement for a straightforward method that can produce the most important collection of features because variations in handwriting styles need to extract those features that can distinguish sharp holes, curves, and strokes in Gujarati numerals. Feature extraction is carried out by utilising imagenet weights with the MobileNet, DenseNet, and NasNetMobile architectures. The associated features will be assigned weights based on how important they are. A compact neural architecture called MobileNet uses both pointwise and depthwise convolutions. To restrict the number of parameters, it employs a width multiplier. In a similar way, neural architecture search uses reinforcement learning to enable the identification of excellent architectures over a small dataset and transfer them to a large dataset for classification tasks. In DenseNet topologies, the layers are physically connected to one another to maximise information to minimise the vanishing gradient issue and protect against overfitting problems.

4 Image Classification

Data classification is the process of organising raw data into categories and labelling those categories for future use. Gujarati OCR depends on this stage since identifying the data will make Gujarati character OCR very easy and simple. The handwritten Gujarati numeral dataset used in this work consists of over 14,000 images divided into 10 different classes in an 80:20 ratio to create a training and testing set. A trial-and-error strategy is performed for the predictions to select the best parameters for a number of ML classifiers.

XGBoost (XGB): It is an ensemble learning technique, which means it makes a prediction by combining the outcomes of multiple models, known as base learners. XGBoost uses Decision Trees as base learners, just like Random Forests. XGBoost includes frameworks for a variety of languages, including Python, and it works well with Python data scientists' favorite machine learning framework, scikit-learn. It is ideal for the great majority of common data science difficulties since it can be used to address classification and regression problems. On classification and regression predictive modeling challenges, XGBoost dominates structured or tabular datasets. It is the go-to method for competition winners on the Kaggle competitive data science platform, according to the evidence. We first used the XGBoost classifier and then printed the summary report with accuracy, precision, and F1 score.

Random Forest (RF): It is the classifier that contains multiple decision trees on different subsets of any dataset and takes out the average of it to increase the overall accuracy of the dataset. Instead of using one decision tree, it takes the use of multiple decision trees. This classification algorithm is used mainly for classification and regression problems. The reason we used the Random Forest classifier is that it takes less training time as compared to other algorithms and gives highly accurate results. It can maintain the accuracy of the model even when a large proportion of the data form the dataset is missing. Random forest prevents the overfitting issue. We used the RF classifier and printed the summary containing accuracy, F1Score, Precision, and Recall.

Logistic Regression (LR): It is used of calculating the probability of the target variable or variable under test. It is used for predicting the categorical dependent variable using a set of an independent variable. When a test data is used for Logistic Regression, It gives probabilistic answer between 0 and 1. The architecture of Logistic Regression is very similar to Linear regression but one demerit of Linear regression is that it can't be used for classification. We trained Logistic Regression model and then we predicted the test data using it and finally we printed the model summary including accuracy, Precision, F1 Score and Recall.

Naive Bayes (NB): This model is based on Bayes' theorem which is about finding the probability of an event given probability of another event that has already occurred. Naïve Bayes classification is mostly used for text dataset but can be used for image data as well. Naïve Bayes is used for quick prediction of test data. It can be used for binary as well as multi-class predictions. Naïve Bayes' biggest application is that it can be used for real time classification. We used Naïve Bayes classification on our dataset and then displayed the summary report with accuracy, Precision, Recall, and F1 Score.

Stochastic Gradient Descent (SGD): The class SGD Classifier implements a simple stochastic gradient descent learning method that supports a variety of classification loss functions and penalties. A SGD Classifier trained with the hinge loss, which is identical to a linear SVM, has the decision boundary. SGD is faster as compared to GD. After witnessing only a single or a few training instances, Stochastic Gradient Descent (SGD) addresses both of these concerns by following the negative gradient of the objective. The application of SGD The high cost of executing back propagation through the entire training set motivates it in the neural network context. We implemented SGD and printed the summary of it containing accuracy, Recall, F1 Score and Precision.

Decision Tree (DT): It creates the classification model by building a decision tree. Each node in the tree represents a test on an attribute, and each branch descending from that node represents one of the property's possible values. The Decision Node and the Leaf Node are the two nodes of a Decision tree. Leaf nodes are the output of those decisions and do not contain any more branches, whereas Decision nodes are used to make any decision and have several branches. The decision tree algorithm is easy to understand because its logic/decision-making is very similar to human thinking. The decision making in Decision Tree is based on root nodes and branches. One of the main advantages of Decision Tree is that it requires less data cleaning as compared to other algorithms. We implemented the Decision Tree algorithm and then took out the summary with accuracy, precision, F1 Score and Recall.

K-Nearest Neighbour (K-NN): The algorithm classifies test data into a particular category from multiple categories based on the closeness of the data with the categories. KNN is a non-parametric algorithm which means it does not make any assumptions about data. The first step of the KNN algorithm is to pre-process data and make it free of errors or repeating values. The next step is to fit the KNN algorithm in the training set and prediction of the training dataset. Once the model gives satisfactory results for the training data, then the model is implemented on test data. The accuracy, Precision, etc. of the result is then summarised in the summary report and printed.

5 Results and Discussion

Classification of handwritten Gujarati numerals is performed through the recent techniques available in machine learning and deep learning. The proposed method tried to achieve desired accuracy by the use of popular machine learning classifiers and deep learning architectures. Three different deep learning architectures are Mobile Net, DenseNet, and NasNetMobile used to extract low to high level unique properties from the numerals' image whereas Boost, Random forest, Logistic regression, Naïve bayes, Stochastic gradient descent, Decision tree, K-nearest neighbor are selected as top layer for image classification.

The proposed method utilized handwritten Gujarati numerals dataset having 14000 images among 10 different classes and divided by 80:20 ratio to training and testing dataset. So total 11200 images are considered for training and the remaining 2800 images used for testing of the network. The results obtained over test dataset is evaluated through accuracy, precision and f-score metrics as well as confusion metrix were developed for the better understanding of classwise prediction. The most logical performance metric is

Table 1. Accuracy score obtained through different classifiers

	Accuracy						
Class	**XGB**	**RF**	**LR**	**NB**	**SGD**	**DT**	**K-NN**
0	0.5	0.86	0.61	0.52	0.64	0.31	0.94
1	0.97	0.97	0.99	0.85	0.98	0.83	0.99
2	0.98	0.99	1	0.95	1	0.88	1
3	0.96	0.97	0.99	0.79	0.99	0.84	0.99
4	0.98	1	1	0.97	1	0.92	1
5	1	0.99	1	0.99	1	0.87	0.99
6	0.97	0.98	1	0.93	0.99	0.75	0.99
7	0.99	0.99	1	0.96	1	0.84	1
8	0.99	0.99	1	0.98	1	0.84	1
9	0.99	0.99	0.99	0.99	0.99	0.95	0.99

Table 2. Precision score obtained through different classifiers

	Precision						
Class	**XGB**	**RF**	**LR**	**NB**	**SGD**	**DT**	**K-NN**
0	0.99	1	1	0.99	1	0.8	1
1	0.97	0.99	1	0.94	1	0.79	0.99
2	0.97	1	1	0.82	1	0.89	1
3	0.97	0.99	1	0.97	0.99	0.79	0.99
4	0.99	0.99	1	1	1	0.94	1
5	0.97	0.98	1	0.94	1	0.89	1
6	0.98	0.99	0.99	0.91	1	0.75	1
7	0.7	0.85	0.72	0.61	0.74	0.57	0.94
8	0.98	0.99	0.99	0.98	0.97	0.57	0.99
9	0.95	1	1	1	1	0.86	1

Table 3. F1-score obtained through different classifiers

	F1 Score						
Class	**XGB**	**RF**	**LR**	**NB**	**SGD**	**DT**	**K-NN**
0	0.67	0.92	0.76	0.68	0.78	0.45	0.97
1	0.97	0.98	0.99	0.89	0.99	0.81	0.99
2	0.98	0.99	1	0.88	1	0.89	1
3	0.97	0.98	1	0.87	0.99	0.82	0.99
4	0.98	0.99	1	0.99	1	0.93	1
5	0.99	0.98	1	0.96	1	0.88	1
6	0.98	0.99	0.99	0.92	0.99	0.75	0.99
7	0.82	0.92	0.83	0.75	0.85	0.68	0.97
8	0.99	0.99	0.99	0.98	0.98	0.68	0.99
9	0.97	1	1	0.99	0.99	0.9	1

accuracy, which is just the proportion of properly predicted observations to all observations shown in Table 1. Whereas the ratio of accurately predicted positive observations to all expected positive observations is known as precision. High precision leads to a low false positive rate listed in Table 2. Similarly, F1 score is the weighted average of precision and recall. Therefore, both false positives and false negatives are considered while calculating it. When false positives and false negatives cost about the same, accuracy performs best. The result of F1-score are listed in the Table 3.

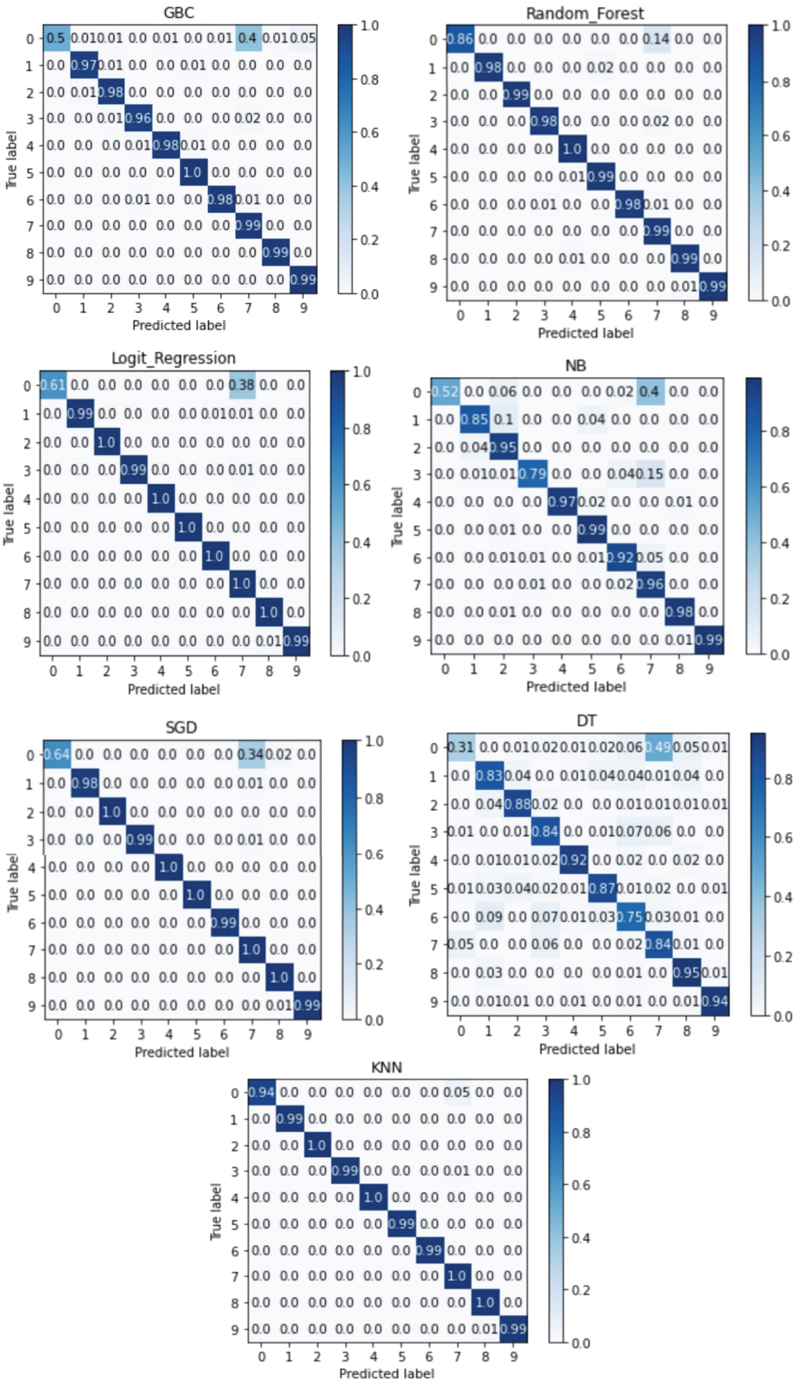

Fig. 2. Confusion Matrix developed through ML classifiers

Table 4. Performance of Model in terms of accuracy(%)

No	Top Layer Classifier	Model Accuracy (%)
1	XGBoost	93.50
2	Random Forest	97.42
3	Logistic Regression	95.80
4	Naïve Bayes	89.25
5	Stochastic Gradient Descent	95.80
6	Decision Tree	81.20
7	K-Nearest Neighbors	99.00

A classification problem's predicted outcomes are compiled in a confusion matrix. The confusion matrix demonstrates the manner in which your classification model makes predictions while being confused. Count values are used to describe the number of accurate and inaccurate predictions for each class. It gives you information on the types of errors being produced, which is more relevant than just the faults our classifier is making. Results obtained through the proposed method are represented in the form of confusion metrix shown in the Fig. 2.

Overall, the methodology received remarkable results over the handwritten numeral dataset of Gujarati language listed in the Table 4. Where, Naive Bayes classifier achieved 89.25%, Stochastic Gradient Descent obtained 95.8% accuracy, Decision Tree produced 81.2%, KNN classification achieved the highest 99% accuracy.

6 Conclusion

In this research, OCR for Gujarati numerals was successfully accomplished with the help of various classification and feature extraction techniques. A summary of the accuracy, precision, recall, and F1 scores for each classification model used in this study can be seen below. According to our investigation, the K-Nearest Neighbor algorithm has a maximum accuracy of 99%. There is a scope for development by combining Gujarati consonants and vowels in the present dataset to expand the limitations of the proposed approach.

Acknowledgment. The authors are thankful to the Institute of Technology, Nirma University for providing the research environment and resources for conducting this research.

References

1. Antani, S. and Agnihotri, L., 1999, September. Gujarati character recognition. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318) (pp. 418–421). IEEE.
2. Goswami, M.M., Prajapati, H.B. and Dabhi, V.K., 2011, November. Classification of printed Gujarati characters using SOM based k-Nearest Neighbor Classifier. In 2011 International conference on image information processing (pp. 1–5). IEEE.
3. Vasant, A.R., Vasant, S.R. and Kulkarni, G.R., 2012, May. Performance evaluation of different image sizes for recognizing offline handwritten gujarati digits using neural network approach. In 2012 international conference on communication systems and network technologies (pp. 270–273). IEEE.
4. Chaudhary, M., Shikkenawis, G., Mitra, S.K. and Goswami, M., 2012, December. Similar looking Gujarati printed character recognition using Locality Preserving Projection and artificial neural networks. In 2012 Third International Conference on Emerging Applications of Information Technology (pp. 153–156). IEEE.
5. Chaudhari, S.A. and Gulati, R.M., 2013, March. An OCR for separation and identification of mixed English—Gujarati digits using kNN classifier. In 2013 International Conference on Intelligent Systems and Signal Processing (ISSP) (pp. 190–193). IEEE.
6. Mendapara, M.B. and Goswami, M.M., 2014, March. Stroke identification in Gujarati text using directional feature. In 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE) (pp. 1–5). IEEE.
7. Desai, A.A., 2015. Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space. CSI transactions on ICT, 2(4), pp.235-241.
8. Vyas, A.N. and Goswami, M.M., 2015, August. Classification of handwritten Gujarati numerals. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1231–1237). IEEE.
9. Macwan, S.J. and Vyas, A.N., 2015, August. Classification of offline Gujarati handwritten characters. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1535–1541). IEEE.
10. Gohel, C.C., Goswami, M.M. and Prajapati, V.K., 2015, December. On-line handwritten Gujarati character recognition using low level stroke. In 2015 Third International Conference on Image Information Processing (ICIIP) (pp. 130–134). IEEE.
11. Naik, V.A. and Desai, A.A., 2017, July. Online handwritten Gujarati character recognition using SVM, MLP, and K-NN. In 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–6). IEEE.
12. Sunilkumar, S.B., Arvind, R.Y., Mihir, M.P. and Mitul, M.P., 2017, August. Gujarati handwritten character recognition using grey level co-occurrence matrix and dynamic time warping technique. In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon) (pp. 340–346). IEEE.
13. Joshi, D.S. and Risodkar, Y.R., 2018, February. Deep learning based Gujarati handwritten character recognition. In 2018 International Conference On Advances in Communication and Computing Technology (ICACCT) (pp. 563–566). IEEE.
14. Mehta, A. and Gor, A., 2017, August. Multifont multisize Gujarati OCR with style identification. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 275–281). IEEE.
15. Paneri, P.R., Narang, R. and Goswami, M.M., 2017, December. Offline handwritten Gujarati word recognition. In 2017 Fourth International Conference on Image Information Processing (ICIIP) (pp. 1–5). IEEE.

16. Aniket, S., Atharva, R., Prabha, C., Rupali, D. and Shubham, P., 2019, January. Handwritten Gujarati script recognition with image processing and deep learning. In 2019 International Conference on Nascent Technologies in Engineering (ICNTE) (pp. 1–4). IEEE.
17. Parikh, M. and Desai, A., 2019, September. Segmentation of frequently used handwritten gujarati conjunctive alphabet. In 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA) (pp. 1–6). IEEE.
18. Borad, P., Dethaliya, P. and Mehta, A., 2020, November. Augmentation based Convolutional Neural Network for recognition of Handwritten Gujarati Characters. In 2020 IEEE International Conference for Innovation in Technology (INOCON) (pp. 1–4). IEEE.
19. Pareek, J., Singhania, D., Kumari, R.R. and Purohit, S., 2020. Gujarati handwritten character recognition from text images. Procedia Computer Science, 171, pp.514-523.
20. Desai, A.A., 2010. Gujarati handwritten numeral optical character reorganization through neural network. Pattern recognition, 43(7), pp.2582-2589.
21. Goswami, M.M. and Mitra, S.K., A Database of Offline Handwritten Gujarati Numerals and Mixed Numeral Recognition.
22. Sharma, A.K., Adhyaru, D.M., Zaveri, T.H. and Thakkar, P.B., 2015, November. Comparative analysis of zoning based methods for Gujarati handwritten numeral recognition. In 2015 5th Nirma University International Conference on Engineering (NUiCONE) (pp. 1–5). IEEE.
23. Sharma, A.K., Thakkar, P., Adhyaru, D.M. and Zaveri, T.H., 2017. Gujarati handwritten numeral recognition through fusion of features and machine learning techniques. International Journal of Computational Systems Engineering, 3(1-2), pp.35-47.
24. Naik, V.A. and Desai, A.A., 2018. Online handwritten gujarati numeral recognition using support vector machine. Int. J. Comput. Sci. Eng., 6(9), pp.416-421.
25. Naik, V.A. and Desai, A.A., 2019. Online handwritten Gujarati word recognition. International Journal of Computer Vision and Image Processing (IJCVIP), 9(1), pp.35-50.
26. Sharma, A.K., Thakkar, P., Adhyaru, D.M. and Zaveri, T.H., 2019. Handwritten Gujarati Character Recognition Using Structural Decomposition Technique. Pattern Recognition and Image Analysis, 29(2), pp.325-338.
27. Naik, V.A. and Desai, A.A., 2019. Multi-layer classification approach for online handwritten Gujarati character recognition. In Computational Intelligence: Theories, Applications and Future Directions-Volume II (pp. 595–606). Springer, Singapore.
28. Gupta, D. and Bag, S., 2021. CNN-based multilingual handwritten numeral recognition: A fusion-free approach. Expert Systems with Applications, 165, p.113784.