Journal of Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7688717

MACHINE LEARNING AND EASYOCR BASED LANGUAGE RECOGNITION FOR HANDWRITTEN INDIC EXTRACTION AND CLASSIFICATION

Sakuldeep Singh
Email - sakuldeep.singhnas@gmail.com

Dr R.B.Singh
Email - Hod.maths@monad.edu.in

Abstract - Handwritten character and number recognition remains challenging after decades of study of offline Indic recapitulations. This is because the Indic scripts share a similar structure and have characters that look very similar to one another. Similar to other computer vision tasks, state-of-the-art results have been achieved in handwritten Indic scripts recognition by employing deep learning-based methods. This is the case even though the problem is relatively new. However, developing a successful handcrafted Machine learning model for various Indian languages from scratch involves a large amount of trial and error and calls for a lot of expertise about the problem domain. By employing an evolutionary meta-heuristics approach, we were able to streamline the search process and find a solution. We were able to automatically improve our text-extraction and language-recognition capabilities by using this method, which relied on a combination of Machine learning and EasyOcr. We focused on Hindi, Malayalam, Kannada, and Tamil languages with Machine learning models to detect languages present in images using EasyOcr library, proposed five distinct models in which Naive Bayes and Random outperform with accuracy 98.70% and 98% with 100% detection and text extraction rate. This is in comparison to previous work that was focused on single languages such as Bengali, Gujrati, and Devnagari rather than Hindi and Dravidian languages. Take into consideration this fact.

Keyword - OCR, Indian Handwritten Scripts, Machine Learning, EasyOcr, Naïve Bayes, Random Forest.

INTRODUCTION

OCR systems exist for a long time, and with progressions, they can now read a variety of other scripts, including Catholic, Chinese, Japanese, and others, with only a fair degree of precision. There isn't currently any OCR technology available for learning to read Indian script. The prevalent use of technology in India for information dissemination and communication has increased demand for optical acknowledgment (OCR) brands that work of Indian scripts[1][2]. Since machine readable acknowledgement (OCR) systems have been in use for some time, they have become more adept at identifying numerous scripts, such as Catholic, Chinese, Japanese, as well as others. Currently, there is no available commercially OCR software that can read Indian script.[3]. There is a circumstance that is extremely comparable to this one with regard to Indian scripts. The usage of numerous scripts inside a single document is the most difficult part of the many difficulties that are related with the scripts that are used in Indian documents. There are many issues linked with the scripts that are used in Indian documents.

For example, in India, a single document may have text words written in not one, not two, but often even three different scripts[4]. This is because India uses a variety of scripts for its written language. It is well known that OCR engines require some kind of coding in order to function properly. To simplify and hasten the text recognition process, we must develop a well before device that can recognise the script being referenced.[5]. As a consequence of this, the primary focus of this research will be on identifying the scripts utilized in the writing of Indian manuscripts on a word level[6][7].

As a country, India employs a wide range of written languages and alphabets. Only one of the 10 official scripts, Urdu, is not a direct descendant of Brahmi, an ancient Indian script. The four most widely used scripts in India are Devanagari, Bangla, Telugu, and Tamil[8]. Devanagari has a larger user base than any other Indian script. Around 500 million people use this on a daily basis and do so in a variety of languages including Indian languages, Bhojpuri, Marathi, Nepali, and Sanskrit. More than 220 million people speak Bangla in the neighbouring nations of Bangladesh and India, making it the second most extensively spoken language here on Indian subcontinent. Assamese and Manipuri can also be written using this alphabet in addition to Bangla. More people use Devanagari more than any other Indian script. This is used on a daily basis by about 500 million people who speak a variety of languages, including Bhojpuri, Marathi, Nepali, and Sanskrit. Bangla is the second most widely spoken language on the Indian subcontinent, with more and over 220 million people using it in the neighbouring countries of Bangladesh and India. Along with Bangla, this alphabet can be used to write Assamese and Manipuri. [10].

Using the Deep Learning library EasyOcr to extract text from images & convert it to text, then feeding this text in and out of trained machine learning models to classify & detect the languages, we will enforce a hybrid machine learning and deep learning-based approach in this work to classify & identify written in cursive Indian languages such as Hindi, Tamil, Kannada, and Malayalam Then, using test data, we will run machine learning algorithms to determine how well trained ML models perform.

LITERATURE REVIEW

Due to its varied applicable environment, character recognition is the most difficult research topic. Devanagari fundamental characters have been the subject of extensive study, but the sample's handwritten qualities have been given far less attention. Problems are exacerbated by the authors' different writing styles or emotional moods. Feature extraction is given more weight in the classic machine learning approach to character recognition, while deep learning, a subfield of machine learning, makes extensive use of neural networks to learn. Early studies on [11] considered it to be produced our own dataset for manually written Devanagari characteristics of the sample for the research we are currently conducting. In our dataset, which we collected from writers among all ages, there are 5000 instances that represent 50 categories of sample characteristics. For realising Devanagari compound characters, Sachdeva also provides a model based on a convolutional neural network. For effective deep neural network training, we used the CNN ResNet template with ReLu as the activation function. We implemented three-, four-, & five-layer CNN models to our dataset and compared the results. We obtained 100% accuracy on our dataset, the highest level possible.

Because of their complicated structures and similar-looking letters, Because of the Indian script, it is challenging for handwritten documents to accurately identify both characters and numbers. Published by finger Indic note - taking identification has advanced significantly with the help of need plenty algorithms, much like other computer vision problems. Efficacious left hand Deep Convolution Net (DNN) construction, however, calls for thorough iteration & familiarity with particular problems. This procedure seems to require a significant amount of time and resources. We solved this problem and accelerated the search by applying a theoretical developmental technique to rapidly produce the excellent Continues to follow Neural (CNN) architecture.[12]. Additionally, a novel framework for automatically designing CNN architecture is proposed in this work., one that makes use of augmented and swiftly convergent responsive particle swarm optimization (APSO). The next stage is to run computer studies on eight datasets containing cursive digits and characters from three popular Indic scripts: Bangla, Devanagari, & Dogri. The experimental results demonstrate that, across all datasets, its suggested APSO-CNN method outperforms state-of-the-art methods.

In their groundbreaking paper, [13] offer a unique method for personality data-only sentence Indic script detection during training. Our method uses the dual neural CNN that can simultaneously examine data both from online and offline modalities in order to identify scripts. With the help of handwritten data from either modality, we create a different modality using intramodality conversion. Our network then receives this pair of offline-online modals. As a result, in addition to the benefit of only using data from both modalities, its same conceptual model can be used to determine a script's identity both offline and online, eliminating the need to create two different script identity modules with each modality. In order to combine the information from We present a new conditional intra nuclear fission methodology which also merges information iteratively based on its source or its context and that operates in both offline and online modes. A thorough new study has been done on a data set that includes six additional official Dharmic languages in addition to English (Roman). Our proposed approach outperforms conventional classifiers, manually created features, and deep learning methods. The experiment's results show that personality training set alone can compete with conventional training that is using word-level data.

The purpose of a spoken language identification system is to assign language tags to voice in an audio file (SLID). Our study applies a convolutional neural network–based approach to the issue of automatically recognizing Bengali, Gujarati, Tamil, and Telugu (CNN). Each of the four languages contributes five hours of audio data for the classifier's training process. The CNN typically uses raw audio insight with a duration of two to four seconds and then a highly variable voice quality but rather noise print to produce MFCC spectrogram images. Different sample sizes are used to gauge the SLID system's performance for both testing and training purposes.[14]. The suggested CNN model outperforms all competing machine learning models in terms of accuracy (88.82%).

Finding the [15] of a work of literature from a group of potential authors (suspects). In the ever-growing sea of document that is the Internet, there is a pressing need to properly credit published material as it appears. For this very reason, a lot of work has been put into in the English language. Comparatively speaking, regional Indian languages like Tamil, Telugu, Bengali, and Punjabi have had less research done on them than Marathi has had. A method for identifying the authors of documents written in Marathi is presented in this study. We also analysed the text using two separate models, each focusing on a distinct set of lexical and stylistic details. The significance of both the feature extraction technique was again verified to make sure it held true among all models used during the experiment. The efficacy of the suggested method has been evaluated using recall, precision, F-measure, & accuracy values.

Table 1. Reported Performance of Exiting Work with Indian Scripts

References	Scripts	Methods/Models	Metrics
[11]	Devanagari	MLP, SVM	Acc = 97%
[12]	Devnagari, Bangla, Dogri	APSO-CNN	Acc = 99%
[17]	Gujrati	CNN	Acc = 97%
[18]	Multi Indian Scripts	ELM	Acc = 92%
[19]	Gujrati	CNN	ACC = 97%

PROPOSED METHODOLOGY

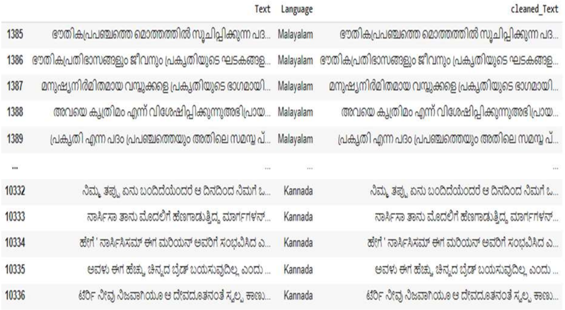
For the recognition and classification of written by hand Indian language text & script using images, this current topical a hybrid deep computing and machine learning library. This is achieved through the use of both handwritten text and image features. In this design for natural language processing based on text data from four Indian languages (Malayalam, Tamil, Kannada, and Hindi), after the data have been collected and cleaned, the next steps are to apply a label encoder for categorical features to convert them to numerical, apply a count vectorizer, and then implement machine learning algorithms. Finally, after all of these steps have been completed, the performance of trained models is evaluated over text data. Implement the deep learning-based EasyOcr library to detect the scrip-form image. EasyOcr is used for extracting text from images and passing it on to trained machine-learning algorithms for detection and classification[16].

A. Data Collection

Due to the lack of availability of different languages with a single data set, it was decided to collect data from a variety of sources in order to create a new one. This was done in order to train machine learning models that could detect and classify the languages that were designed. Data for four Indian languages were collected from open sources. Within the scope of this project, collect data from the Kaggle, UCI, and Data world websites. Narrow your choices down to just four languages, most likely Malayalam, Tamil, Kannada, and Hindi. Malayalam has 591 samples of text, Tamil has 464 samples, Kannada has 366, and Hindi has 62 samples of the text of varying lengths. These data can be converted into a pandas' data frame for analysis.

B. Preprocessing

Preprocessing task was started by finding null and nan values, removing duplicate values in text data using, and finally designing a clean text column in which use a regular expression library in python to cleanse the text to remove symbols, numbers, punctuation URLs, hashtags, and whitespaces were the first steps of the preprocessing task. The preprocessing task began with a data collection task that collected various data from different languages and selected only Indian languages as the root sources. Apply the label encoder in order to convert categorical data to numerical data, and then apply the count vectorizer to the bag of words in order to convert text to numerical data for the independent variable.



	Text	Language	cleaned_Text
1385	ഭൗതികപ്രപഞ്ചത്തെ മൊത്തത്തിൽ സൂചിപ്പിക്കുന്ന പദ...	Malayalam	ഭൗതികപ്രപഞ്ചത്തെ മൊത്തത്തിൽ സൂചിപ്പിക്കുന്ന പദ...
1386	ഭൗതികപ്രതിഭാസങ്ങളും ജീവനും പ്രകൃതിയുടെ ഘടകങ്ങള...	Malayalam	ഭൗതികപ്രതിഭാസങ്ങളും ജീവനും പ്രകൃതിയുടെ ഘടകങ്ങള...
1387	മനുഷ്യനിർമ്മിതമായ വസ്തുക്കളെ പ്രകൃതിയുടെ ഭാഗമായി...	Malayalam	മനുഷ്യനിർമ്മിതമായ വസ്തുക്കളെ പ്രകൃതിയുടെ ഭാഗമായി...
1388	അവയെ കൃത്രിമം എന്ന് വിശേഷിപ്പിക്കുന്നുഅഭിപ്രായ...	Malayalam	അവയെ കൃത്രിമം എന്ന് വിശേഷിപ്പിക്കുന്നുഅഭിപ്രായ...
1389	പ്രകൃതി എന്ന പദം പ്രപഞ്ചത്തെയും അതിലെ സമസ്ത പ്...	Malayalam	പ്രകൃതി എന്ന പദം പ്രപഞ്ചത്തെയും അതിലെ സമസ്ത പ്...
...	...	...	...
10332	ನಿಮ್ಮ ತಪ್ಪು ಏನು ಬಂದಿದೆಯೆಂದರೆ ಆ ದಿನದಿಂದ ನಿಮಗೆ ಒ...	Kannada	ನಿಮ್ಮ ತಪ್ಪು ಏನು ಬಂದಿದೆಯೆಂದರೆ ಆ ದಿನದಿಂದ ನಿಮಗೆ ಒ...
10333	ನಾರ್ಸಿಸಾ ತಾನು ಮೊದಲಿಗೆ ಹೆಣಗಾಡುತ್ತಿದ್ದ ಮಾರ್ಗಗಳನ್...	Kannada	ನಾರ್ಸಿಸಾ ತಾನು ಮೊದಲಿಗೆ ಹೆಣಗಾಡುತ್ತಿದ್ದ ಮಾರ್ಗಗಳನ್...
10334	ಹೇಗೆ ' ನಾರ್ಸಿಸಿಸಮ್ ಈಗ ಮರಿಯನ್ ಅವರಿಗೆ ಸಂಭವಿಸಿದ ಎ...	Kannada	ಹೇಗೆ ' ನಾರ್ಸಿಸಿಸಮ್ ಈಗ ಮರಿಯನ್ ಅವರಿಗೆ ಸಂಭವಿಸಿದ ಎ...
10335	ಅವಳು ಈಗ ಹೆಚ್ಚು ಚಿನ್ನದ ಬ್ರೆಡ್ ಬಯಸುವುದಿಲ್ಲ ಎಂದು ...	Kannada	ಅವಳು ಈಗ ಹೆಚ್ಚು ಚಿನ್ನದ ಬ್ರೆಡ್ ಬಯಸುವುದಿಲ್ಲ ಎಂದು ...
10336	ಟರ್ಕಿ ನೀವು ನಿಜವಾಗಿಯೂ ಆ ದೇವದೂತನಂತೆ ಸ್ವಲ್ಪ ಕಾಣು...	Kannada	ಟರ್ಕಿ ನೀವು ನಿಜವಾಗಿಯೂ ಆ ದೇವದೂತನಂತೆ ಸ್ವಲ್ಪ ಕಾಣು...

Figure 1. Pandas Data frame of Clean and processed data.

C. Exploratory Data Analysis

Python and the seaborn library should be used for the analysis and visualization of the data contained in this work. Exploratory data analysis, also known as EDA[20], is a method of analyzing data sets to highlight the most important aspects of those sets. This analysis method frequently makes use of statistical graphics and other methods for visualizing the data. Because its main goal is to ascertain what the info can tell us besides the formal modelling, EDA differs from the more traditional method of testing hypotheses. Use of a simulation-based strategy is an option.The below graphs represent the contribution of language in the data for training machine learning models.

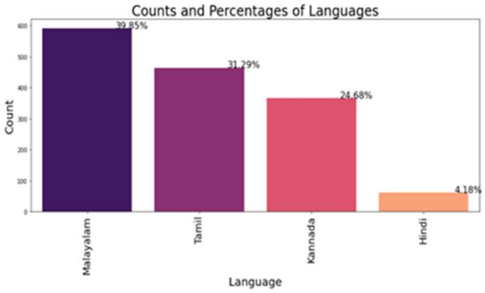


Figure 2. Bar Plot of Languages Distribution.

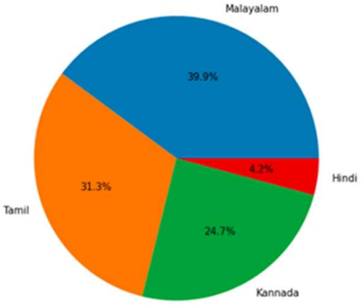


Figure 3. Pie Chart Distribution of Languages.

The counts and percentages of languages that are present in the data that are used to train machine learning models are depicted in Figure 2. Malayalam has the highest count of languages, with 39.85%, while Hindi has the lowest count of languages, with 4.18%. Tamil has 31.29%, and Kannada has 24.68%. Figure 3 shows the pie chart distribution of models.

D. Machine Learning Models

Five different artificial intelligence models were used in this study to detect various scripts and language types using the suggested method. the K Nearest Neighbor model [21], Rough Forest [22], Simple Bayes [23], Rational Regression [24], and Tree of decisions [15] were put to use.

1. k-Nearest Neighbor (kNN)

Since the early 1970s, the non-parametric k-nearest neighbour (kNN) technique has been used in statistical applications. The k-nearest neighbour (kNN) algorithm's central principle is that, given a calibration dataset, it should be able to identify a set of k samples within the dataset that are physically closest to unknown chemicals (e.g., based on distance functions). By averaging the response variables across these k samples, we might potentially determine the label (class) of the missing examples (i.e., the class attributes of the k nearest neighbor). For this reason, a kNN classifier's overall efficacy is highly sensitive to the value of k; more specifically, it is the most critical tuning parameter for the kNN classifier. A bootstrap method was utilised in order to arrive at k as the value for the parameter. In this investigation, we looked at k values ranging from 1 to 20 in an effort to find the one that worked best for all of the training sample sets[22].

2. Random Forest (RF)

Two parameters must be set in order to activate the RF: the number of trees, denoted by "ntree," and also the amount of features included within every split, respectively (mtry). According to research, using the settings frequently produces excellent results. On the opposite hand, Liaw and Wiener contend that a large number of trees will consistently produce the same outcome, though the relative importance of every tree's contribution will vary. Furthermore, it was said that using more trees than necessary could not be necessary but would not hurt the model either. Furthermore, mtry is typically left at its default value of mtry = p, where p is the number of predictors, in many research. This is a pre-used value. Here, however, we explore the effects of varying these parameters' values to identify the optimal RF model for classification. The range of values was from 100 to 1000 for ntree, and from 1:10 to 1:10 for mtry, with a step size of 1[25].

3. Naïve Bayes

Classifiers can be constructed with Naive Bayes, a simple technique. These models classify instances of an issue, which are presented as a vector of extracted features, into predetermined categories using a small set of features. There isn't just one method for training these classifiers; rather, there's an entire family of algorithms based on the idea that, considering the class variable, the importance of any given feature is independent of the importance of any other feature. If a fruit is red, has a diameter of around 10 centimetres, and is round, it might be classified as an apple. As an example, a naive Bayes categorization would treat the fruit's colour, roundness, or diameter as three independent factors in determining whether or not the object in question is an apple, regardless of any possible relationships among them. Because the maximum Logistic likelihood method is commonly used in modeling for naive Bayes models, it is possible to work with the naive Bayes classification model without adopting Bayesian probability or using any Bayesian procedures[23].

4. Regression (LR)

By converting the log-odds of an event into such a linear function of one or even more relationships among independent variables, logistic models, also referred to as logit models, are employed to forecast the likelihood of an occurrence. Regression analysis using logistic regression, also known as logit regression, is a technique for estimating a logistic model's parameters (the coefficients in the linear combination). In binary logistic regression, the control variables can be continuous or binary, and a single multiple regression is used statistically (two classes, encoded by a parameter) of values of 0 and 1. (any real value). Because the likelihood of a value marked "1" may range from 0 (definitely the value "0") to 1, the logistic function is utilized to transform file to probability. (definitely the value "1")[24].

5. Decision Tree

In a decision tree, each leaf node corresponds to a class label, each inner node to a "test" of a characteristic (such as whether a coin will land on its head or tail), and each branch to the outcome of that experiment (decision taken after computing all attributes). A tree's classification criteria are visually represented by the paths that connect its trunk to its leaves. In decision analysis, a decision tree and a heavily linked influence diagram are used to visually and analytically weigh the relative merits of several alternatives and determine their expected utilities[8].

Nodes in a decision tree can be one of three varieties:

Squares depicting decision nodes

Chance nodes, which are frequently depicted as circles

Triangles are frequently used to symbolize end nodes.

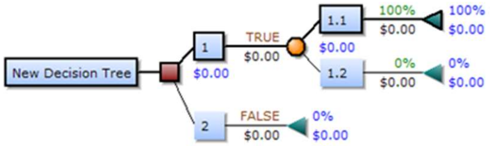


Figure 4. Decision Tree.

E. EasyOcr

The end nodes, which are generally depicted as triangles, The goal is to make it such that EasyOCR [26] can accommodate any model that is considered to be cutting edge. Character-level bounding boxes are used to generate the ground truth label again for region value as well as the affinity value for each training image. To each photograph belongs its own unique description. A pixel's region score indicates how likely it is that pixel is positioned in the centre of the character, whereas an affinity score tells how likely it is that pixel is located in the centre of the space between two surrounding characters. In contrast to a binary segmentation map, which assigns a label to each individual pixel, we use a Gaussian heatmap to express the probability that the pixel represents the character centre. This heatmap format has been employed for purposes outside from pose estimation due to its versatility in dealing with ground truth zones that are not tightly delineated. Pose estimation is one example of this kind of use. We first learn the region score using the heatmap representation, and then we learn the affinity score using that. The grey slots are designed to accommodate interchangeable modules in light blue[20].

F. Flowchart

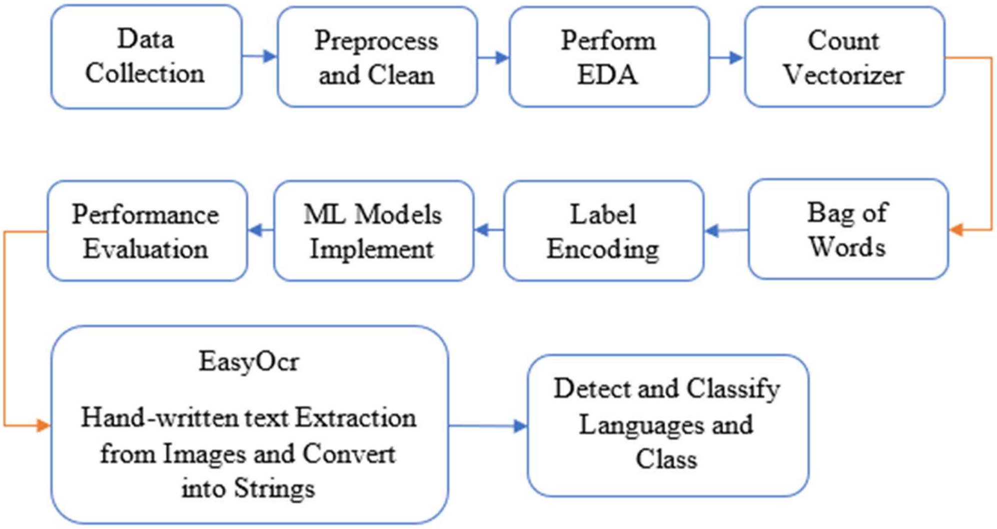


Figure 5. Flowchart of Proposed work.

Results and Discussion

The combination of EasyOcr and machine learning models is the work that has been implemented and that which is suggested. The primary reason for carrying out this work was to utilise pictures and ML models in order to detect handwritten scripts of a number of different Indian languages. Figure 5 depicts the flowchart of the proposed method, which includes the following steps: first, collect data from various sources because different languages are not available in one data set; second, preprocess text by using regular expressions; third, apply bag of words; and fourth, train machine learning models and evaluate their performance by using accuracy, precision, recall, and f score; and finally, implement EasyOcr library and use different languages' hand written images to extract text. Use Google Colab Notebook , which is built on Python 3.7, EasyOcr, and Sklearn, and has 16 Gigabytes of RAM and cloud-based GPUs, in order to put this work into action.

A. Performance Evaluation

Utilizing evaluation measures, the model's quality is determined. It is common practise to divide issues into two groups [20]when diagnosing them: those with sickness and those without. To accurately classify data samples and identify which ones correspond to which classes is the aim of such a system, keeping this in mind. Although there are numerous metrics for evaluating quality, Confusion matrix, f1-score, accuracy, recall, and precision are some of the most well-known[27].

Confusion matrix

The confusion matrix is a graphic representation that can be used to demonstrate categorization criteria (CM). Depending on the systems' predictions as well as the actual class of a certain instance of data, this matrix compares several classes. It is not an independent, measurable statistic but rather depends on the following four factors: FP, TN, FP, FP, and FP[25].

True Negative (TN): True negative results are those for which it can be shown that the model properly predicted the lack of the target class.

True Positives (TP): True positive results are those for which the model can determine that the target class exists.

False Positives (FP): A result is said to be measured as FP when the model incorrectly assesses the existence of a positive class.

False Negative (FN): A result is deemed false if the model incorrectly expects the presence of negative classes.

Accuracy

The proportion of appropriately classified information samples to all data samples, as well as projections, determine the accuracy limits in a classification problem. number of forecasts and samples of data combined. We did this by dividing the quantity of correctly identified samples by the sum of TP and TN (the main diagonal of the CM)[22].

$$Accuarcy = \frac{TP + FN}{TP + TF + FP + FN} \quad (1)$$

Precision

By comparing the true positive (TP) with every positive occurrence (TP + FP), one may determine the system precision. The task can be completed by simply dividing by the product of the two components (TP + FP)[15].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall

The sensitivity statistic shows how many supposedly "good" events actually were. TP is essentially divided by the sum of TP + FN in the formula.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F-Score

The likelihood that a favourable forecast will come true is depicted here. The appropriate mathematical processes are carried out in order to determine the harmonic mean between two values. Because of this, consideration is given to both the FN and FP points of view. To determine a user's F1 score, utilise the continuity formula[15].

$$F1 - score = \frac{2(precision \times recall)}{precision + recall} \quad (4)$$

Table 2. Performance of various proposed Models.

Models	Accuracy%	Precision%	Recall%	F Score%
KNN	83.20	83.20	83.20	83.20
Random Forest	98	98	98	98
Naïve bayes	98.70	98.70	98.70	98.70
Logistic Regression	97.30	97.30	97.30	97.30
Decision Tree	92.60	92.60	92.60	92.60

The final results of trained models over test data are represented in Table 2, where it can be seen that naive bayes and random forest outperform other models with an accuracy of 98.70% and 98% respectively, which is the highest among all models. On the other hand, k nearest neighbor performs the worst with an accuracy of 83.20%, while the decision tree has an accuracy of 92.60%. These models have an accuracy of one hundred percent when it comes to recognizing four different Indian languages, including Tamil, Kannada, Hindi, and Malayalam. For the purpose of prediction, EasyOCR data was input into trained models, and the resulting models were able to detect handwritten text and correctly forecast its class with a hundred percent success rate.

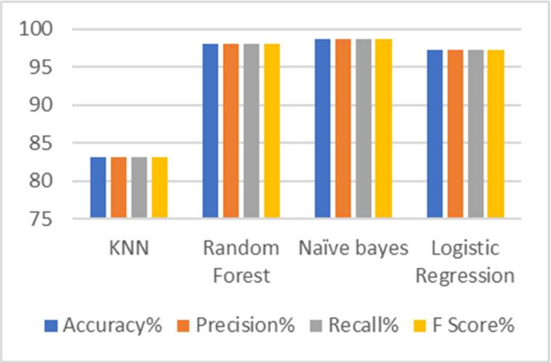


Figure 6. Performance of Models.

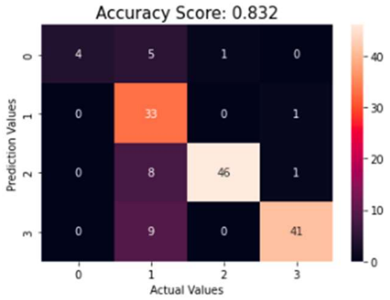


Figure 7. Confusion Matrix of KNN.

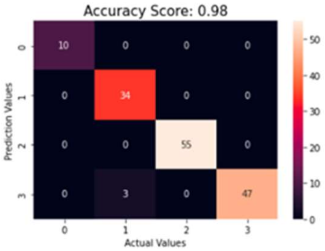


Figure 8. Matrix of Stochastic Forest Confusion.

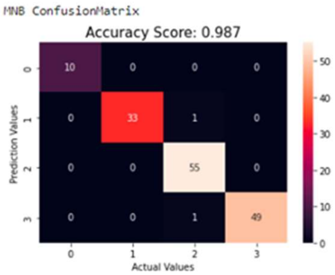
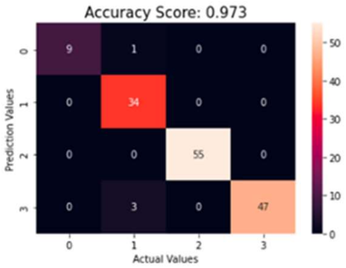
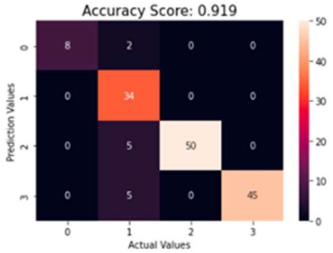


Figure 9. Matrix of Confusion for Naive Bayes.



	0	1	2	3
0	9	1	0	0
1	0	34	0	0
2	0	0	55	0
3	0	3	0	47

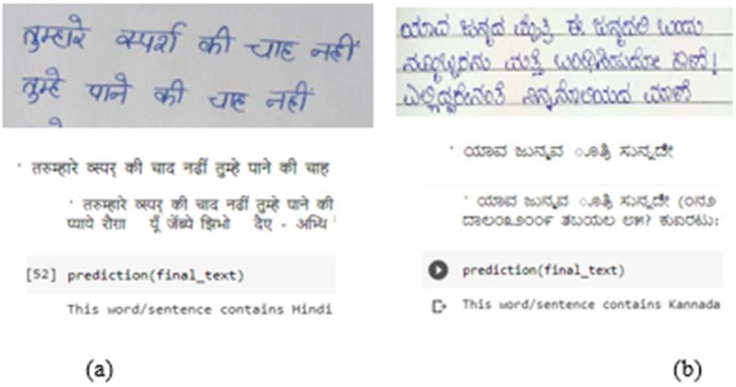
Figure 10. Matrix of Confusion in Logistic Regression.



	0	1	2	3
0	8	2	0	0
1	0	34	0	0
2	0	5	50	0
3	0	5	0	45

Figure 11. Decision Tree Confusion Matrix.

From figure 7 to11 shows the models confusion matrix with accuracy score in which naïve bayes and random forest has highest accuracy.



(a)

तुम्हारे स्पर्श की चाह नहीं
तुम्हे पाने की चाह नहीं

```
' तरुम्हारे व्स्पर् की चाद नहीं तुम्हे पाने की चाह

' तरुम्हारे व्स्पर् की चाद नहीं तुम्हे पाने की
प्याये रौरा  यूँ जेंब्ये श्रिभे  दैए - अभ्यि

[52] prediction(final_text)

This word/sentence contains Hindi
```

(b)

```
' ಯಾವ ಜುನ್ನವ ಂಣಕ್ತಿ ಸುನ್ನದೇ

' ಯಾವ ಜುನ್ನವ ಂಣಕ್ತಿ ಸುನ್ನದೇ (ಂನಾ
ದಾಲಂಬಾಂಂF ತಬುಯಲ ಲಾ? ಕುವರಟು:

prediction(final_text)

This word/sentence contains Kannada
```

Figure 12. Extracted and Predicted Scripts.

Conclusion and Future Work

In order to recognize and extract hand written text that is present in photographs, this work concentrated on hand written text based on four Indian languages including Hindi, Kannada, Tamil, and Malayalam. These languages were employed. In order to train models on digital text data, a new benchmark data set that is based on natural language processing has been presented. The goal of this data set is to recognize and predict the languages. The dataset will unquestionably be very helpful toward the scientific community as it is a conventional dataset for use in this field. In this research, machine learning models like auto encoders, random forest, decision trees, logistic regression, & k nearest fore detection were used. In the process of extracting text from images, the EasyOcr collection, which is centered on Python & uses deep-learning algorithms, was also used. The data have undergone some preliminary processing, and model tuning has been done with the aid of hyper parameters. The Knn Algorithm and Random Forest models outperform all others in this investigation, with accuracy rates for speech extraction and prediction rates of 98%, 98.70%, and 100%, respectively. The suggested approach can be expanded upon and trained over both shallow and deep text-based neural networks, including recurrent neural networks.[13], Hybrid Approaches (LSTM-GRU)[16], and BERT[8] models.

References

[1] H. Cecotti, "Cascade of Distances Approach".

[2] A. I. M, K. P. Hebbar, and N. S. K, "Machine Learning for Handwriting Recognition," Int. J. Comput., pp. 93–101, [Online]. Available: http://ijcjournal.org/

[3] U. Garain, B. B. Chaudhuri, and T. T. Pal, "Online handwritten Indian script recognition: A human motor function based framework," Proc. - Int. Conf. Pattern Recognit., vol. 16, no. 3, pp. 164–167, 2002, doi: 10.1109/icpr.2002.1047820.

[4] Institute of Electrical and Electronics Engineers. Kharagpur Section., Institute of Electrical and Electronics Engineers. India Council., and Hughes Software Systems (Firm), "Proceedings of the IEEE INDICON 2004 : first India Annual Conference, December 20-22, 2004, Indian Institute of Technology Kharagpur," p. 608, 2004.

[5] B. B. Chaudhuri and S. Bera, "Handwritten text line identification in Indian scripts," Proc. Int. Conf. Doc. Anal. Recognition, ICDAR, pp. 636–640, 2009, doi: 10.1109/ICDAR.2009.69.

[6] T. Mondal, U. Bhattacharya, S. K. Parui, K. Das, and D. Mandalapu, "On-line handwriting recognition of Indian scripts - The first benchmark," Proc. - 12th Int. Conf. Front. Handwrit. Recognition, ICFHR 2010, pp. 200–205, 2010, doi: 10.1109/ICFHR.2010.39.

[7] K. S. Varghese, A. James, and S. Chandran, "A Novel Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition," Procedia Technol., vol. 24, pp. 1333–1340, 2016, doi: 10.1016/j.protcy.2016.05.137.

[8] R. Pardeshi, B. B. Chaudhuri, M. Hangarge, and K. C. Santosh, "Automatic Handwritten Indian Scripts Identification," Proc. Int. Conf. Front. Handwrit. Recognition, ICFHR, vol. 2014-Decem, pp. 375–380, 2014, doi: 10.1109/ICFHR.2014.69.

[9] P. M. Kamble and R. S. Hegadi, "Handwritten Marathi character recognition using R-HOG feature," Procedia Comput. Sci., vol. 45, no. C, pp. 266–274, 2015, doi: 10.1016/j.procs.2015.03.137.

[10] S. M. Obaidullah, C. Halder, N. Das, and K. Roy, "Numeral Script Identification from Handwritten Document Images," Procedia Comput. Sci., vol. 54, pp. 585–594, 2015, doi: 10.1016/j.procs.2015.06.067.

[11] J. Sachdeva and S. Mittal, "Handwritten Offline Devanagari Compound Character Recognition Using CNN," Lect. Notes Data Eng. Commun. Technol., vol. 90, pp. 211–220, 2022, doi: 10.1007/978-981-16-6289-8_18.

[12] R. Sharma and B. Kaushik, "Handwritten Indic scripts recognition using neuro-evolutionary adaptive PSO based convolutional neural networks," Sadhana - Acad. Proc. Eng. Sci., vol. 47, no. 1, 2022, doi: 10.1007/s12046-021-01787-x.

[13] A. K. Bhunia, S. Mukherjee, A. Sain, A. K. Bhunia, P. P. Roy, and U. Pal, "Indic handwritten script identification using offline-online multi-modal deep network," Inf. Fusion, vol. 57, pp. 1–14, 2020, doi: 10.1016/j.inffus.2019.10.010.

[14] L. R. Arla, S. Bonthu, and A. Dayal, “Multiclass spoken language identification for indian languages using deep learning,” 2020 IEEE Bombay Sect. Signat. Conf. IBSSC 2020, pp. 42–45, 2020, doi: 10.1109/IBSSC51096.2020.9332161.

[15] K. S. Digamberrao and R. S. Prasad, “Author Identification using Sequential Minimal Optimization with rule-based Decision Tree on Indian Literature in Marathi,” Procedia Comput. Sci., vol. 132, pp. 1086–1101, 2018, doi: 10.1016/j.procs.2018.05.024.

[16] A. Trivedi, S. Srivastava, A. Mishra, A. Shukla, and R. Tiwari, “Hybrid evolutionary approach for Devanagari handwritten numeral recognition using Convolutional Neural Network,” Procedia Comput. Sci., vol. 125, pp. 525–532, 2018, doi: 10.1016/j.procs.2017.12.068.

[17] K. Limbachiya, A. Sharma, P. Thakkar, and D. Adhyaru, “Identification of handwritten Gujarati alphanumeric script by integrating transfer learning and convolutional neural networks,” Sadhana - Acad. Proc. Eng. Sci., vol. 47, no. 2, 2022, doi: 10.1007/s12046-022-01864-9.

[18] M. Ghosh, H. Mukherjee, S. M. Obaidullah, K. C. Santosh, N. Das, and K. Roy, "Extreme learning machine for artistic number of co identification at character level," Procedia Computer Science, volume 167, issue 2019, pages 496–505, 2020, idt: 10.1016/j.procs.2020.03.268.

[19] J. Pareek, D. Singhania, R. R. Kumari, and S. Purohit, “Gujarati Handwritten Character Recognition from Text Images,” Procedia Comput. Sci., vol. 171, no. 2019, pp. 514–523, 2020, doi: 10.1016/j.procs.2020.04.055.

[20] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, “Character region awareness for text detection,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 9357–9366, 2019, doi: 10.1109/CVPR.2019.00959.

[21] G. G. Rajput and S. B. Ummapure, “Script Identification from Handwritten document Images Using LBP Technique at Block level,” 2019 Int. Conf. Data Sci. Commun. IconDSC 2019, pp. 1–6, 2019, doi: 10.1109/IconDSC.2019.8816944.

[22] Sensors (Basel), vol. 18, ref. 1, 2017, doi: 10.3390/s18010018; P. Thanh Noi and M. Kappas, "Comparison of Random Wilderness, k-Nearest Neighbor, & Svm Classifier Classification methods for Land Cover Classification. Using Sentinel-2 Imagery."

[23] F. J. Yang, “An implementation of naive bayes classifier,” Proc. - 2018 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2018, pp. 301–306, 2018, doi: 10.1109/CSCI46756.2018.00065.

[24] X. Zou, Y. Hu, Z. Tian, and K. Shen, “Logistic Regression Model Optimization and Case Analysis,” Proc. IEEE 7th Int. Conf. Comput. Sci. Netw. Technol. ICCSNT 2019, pp. 135–139, 2019, doi: 10.1109/ICCSNT47585.2019.8962457.

[25] K. O. Mohammed Aarif and S. Poruran, “OCR-Nets: Variants of Pre-trained CNN for Urdu Handwritten Character Recognition via Transfer Learning,” Procedia Comput. Sci., vol. 171, no. 2019, pp. 2294–2301, 2020, doi: 10.1016/j.procs.2020.04.248.

[26] “Jaided AI: EasyOCR demo.” https://www.jaided.ai/easyocr/ (accessed Jan. 02, 2023).

[27] M. Karthi, R. Priscilla, and K. J. Syed, “A novel content detection approach for handwritten english letters,” Procedia Comput. Sci., vol. 172, no. 2012, pp. 1016–1025, 2020, doi: 10.1016/j.procs.2020.05.149.