Performance Analysis of Random Forests with SVM and KNN in Classification of Ancient Kannada Scripts

Soumya A[1], G Hemantha Kumar[2]
[1] Department of Computer Science & Engineering, R V College of Engineering, India
[2] Department of Studies in Computer Science, University of Mysore, India

ABSTRACT

Ancient inscriptions which reveal the details of yester years are difficult to interpret by modern readers and efforts are being made in automating such tasks of deciphering historical records. The Kannada script which is used to write in Kannada language has gradually evolved from the ancient script known as Brahmi. Kannada script has traveled a long way from the earlier Brahmi model and has undergone a number of changes during the regimes of Ashoka, Shatavahana, Kadamba, Ganga, Rashtrakuta, Chalukya, Hoysala , Vijayanagara and Wodeyar dynasties. In this paper we discuss on Classification of ancient Kannada Scripts during three different periods Ashoka, Kadamba and Satavahana. A reconstructed grayscale ancient Kannada epigraph image is input, which is binarized using Otsu's method. Normalized Central and Zernike Moment features are extracted for classification. The RF Classifier designed is tested on handwritten base characters belonging to Ashoka, Satavahana and Kadamba dynasties. For each dynasty, 105 handwritten samples with 35 base characters are considered. The classification rates for the training and testing base characters from Satavahana period, for varying number of trees and thresholds of RF are determined. Finally a Comparative analysis of the Classification rates is made for the designed RF with SVM and k-NN classifiers, for the ancient Kannada base characters from 3 different eras Ashoka, Kadamba and Satavahana period.

Indexing terms/Keywords:

Classifier; Random Forest (RF); Support Vector Machine (SVM); K-Nearest Neighbor (k-NN); Classification Rate; Optical Character Recognition(OCR).

Council for Innovative Research
Peer Review Research Publishing System
Journal: INTERNATION JOURNAL OF COMPUTERS AND TECHNOLOGY
Vol. 13, No. 9
editorijctonline@gmail.com
www.ijctonline.com, www.cirworld.com

1. INTRODUCTION

Epigraphy is the study of ancient inscriptions, and is a primary tool of archaeology when dealing with literate cultures. The discipline of ancient history and archeology, play an important role in preserving these historical records and thus contribute in upholding the culture and heritage of the past. India is a multilingual country, possessing a rich collection of written ancient scripts. Three important varieties of scripts that were prevalent in ancient India are: Indus valley script, Brahmi Script and Kharosti script. The scripts of modern Indian languages have evolved from one of these scripts over the centuries. In India currently there are 13 Scripts and 23 official languages for communicating at state level. Apart from these, there are many languages & dialects used by number of people. The Kannada script has been used to write in Kannada language which is the official language of Karnataka state.

1.1 Properties of Kannada Script and Language

Kannada is one of the most enriched languages in India, with its long historical heritage.. and the modern script has gradually evolved from this ancient script known as Brahmi. Kannada language has evolved to present form, from earliest written records about the third century B.C. In fact, the Indian linguists have divided the whole of this evolutionary process in to four broad phases:

- Poorvada Halegannada or Pre-ancient Kannada
- Halegannada or Ancient Kannada
- Nadugannada or Middle Kannada
- Hosagannada or Modern Kannada

Kannada script has traveled a long way from the earlier Brahmi model, has undergone a number of changes during the regimes of Shatavahana, Kadamba, Ganga, Rashtrakuta, Chalukya, Hoysala , Vijayanagara and Wodeyar dynasties as shown in Figure 1. The modern readers find difficulty in interpreting these ancient scripts and need much time for deciphering the ancient written records. Many researchers have been working on Indian script recognition for more than three decades and there are many conventional OCR systems found for modern Kannada Script, but very few work on ancient Kannada epigraphical scripts are reported. Hence, there is a need for the automation of deciphering ancient epigraphical scripts, which is much relevance in knowing the past.

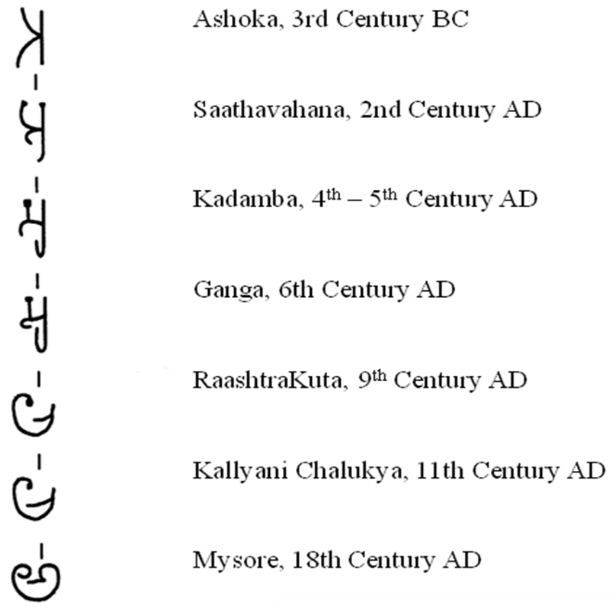

Ashoka, 3rd Century BC
Saathavahana, 2nd Century AD
Kadamba, 4th – 5th Century AD
Ganga, 6th Century AD
RaashtraKuta, 9th Century AD
Kallyani Chalukya, 11th Century AD
Mysore, 18th Century AD

Figure 1: Changes In Character Shape Since 3rd Century B.C To 18th Century A.D

The results of our earlier works [1][2] on dating a given epigraph into the corresponding period are found to be satisfactory. The work on dating of inscriptions using Support Vector Machine for ancient Kannada scripts [1] was carried out which used only few unique characters from character bank of a script for classification. Later in [2] a Random Forest Classifier is designed, and system dates an input ancient Kannada epigraph, considering complete character set pertaining to a specific era. The Classifier reads reconstructed images of epigraphs and categorizes it into one of the six periods: Ashoka, Satavahana, Kadamba, Chalukya, Rastrakuta and Hoysala.

The proposed work performs a comparative study of the performance characteristics of the designed RF with the Classifiers SVM and k-NN supported by OpenCV libraries. This paper is organized as follows: few works on OCR of Indian/non-Indian scripts is reported in Section 2. The System architecture of the proposed work and the description of the approaches are covered in Section 3. The structure chart and methodology of proposed system is detailed in Section 4. Experimental results and analysis are demonstrated in Section 5, and concluding remarks is provided in Section 6.

2. LITERATURE SURVEY

In this section, some of the contributions in Optical Character Recognition of ancient and modern documents, and performance analysis is covered:

RF Classifier has been used on Persian language [3] to classify handwritten Persian characters with Loci features. A classification rate of up to 87% has been achieved. RF Classifier's performance for Handwritten Digit recognition has been accounted in [4]. A feature extraction technique based on a grayscale multi-resolution pyramid was chosen to explore how the RF parameters affect the recognition accuracy. Classification rates of 85%-93% is reported in it. [5] describes a method for recognizing ancient tamil scripts from temple wall inscriptions. It uses fourier and wavelet features for describing the features and k-means algorithm for character recognition. It claims to achieve a maximum recognition accuracy of 84%. An effective system for the classification of ancient handwritten documents according to the writing style has been reported in [6]. It employs a set of features that are extracted from the contours of the handwritten images. These features are based on the direction and curvature histograms that are extracted at a global level from local contour observations. Two writings are then compared by computing the distance between their respective histograms. An identification rate of 94% is obtained in this. A method for dating of the Greek inscription's content[7] uses "platonic" realization of alphabet symbols for the specific inscription and various Geometric characteristics for the features, and classifies the period according to some statistical criteria. A study for the recognition of ancient middle Persian documents [8] chooses a set of invariant moments as the features and the classifiers used are minimum mean distance, k-Nearest Neighbours (KNN) and Parzen. A classification rate of 90.5%-95% was achieved in that. Characterization of the Arabic and Latin ancient document images is explained in [9]. Regions of images having the same size are extracted from the heterogeneous base and fractal dimension method is used to discriminate between ancient Arabic and Latin scripts. It achieves 95.87% accuracy on the discrimination between Arabic and Latin ancient document collections. It claims that the advantage of the approach is that it can be easily adapted for the identification of other ancient document collections. [10] proposes a texture-based approach for text recognition in ancient documents. It copes with the challenges such as degradation, staining, fluctuating text lines, superimposition of text etc. The approach is applied to three different manuscripts, namely to Glagolitic manuscripts of the 11th century, a Latin and a composite Latin-German manuscript, both originating from the 14th century. [11] presents an approach for the detection of elements like initials, headlines and text regions, focused on ancient manuscripts. SIFT descriptors are used to detect the regions of interest, and the scale of the interest points is used for localization. It gives a detection rate of 57% for initials and headlines, and 74% for regular text. An approach for transcribing historical documents[12] divides a text-line image into frames and a graph is constructed using the framed image. Dijkstra algorithm is applied later to find the line transcription. A character accuracy of 79.3% is found in its experiments.An efficient technique for multi-script identification at connected component level using convolutional neural network is described in [13]. Suitable script identification features are automatically extracted and learned as convolution kernels from the raw input. It is tested on a dataset of ancient Greek-Latin mix script document images and an accuracy of 96.37% is achieved on a test dataset at connected component level and improved to 98.40% by using class majority in the left-right neighboring area. A description of a method to efficiently create the ground truth to train and test the different classifiers is given in [14]. Since the manual labelling of the data is a tedious process, the data is represented in different abstraction levels, which is clustered in unsupervised manner. The different clusters are labeled by the human experts. In this method, less than 0.5% of the data is manually labeled and achieves a recognition rate of 86.21% and 94.81% for two different sets of scripts. [15] describes an age identification of ancient Kannada scripts. A hybrid neural network classifier is used where the first phase incorporates an Artificial Neural Network for identifying the base character. The second phase uses a Probabilistic Neural Network model designed for the identification of age pertaining to the base character. A system to classify both printed and handwritten Kannada numerals are discussed in [16]. An average recognition rate of 97% is got by using SVM with Fourier descriptors and chain codes.

The problem of recognizing accented and non-accented characters in French handwriting [17] is reported. The performances of SVM are declined by the presence of accents. An accented character is segmented into two parts: the root character or letter and the accent. These two parts are recognized separately, and the results are combined to rebuild the accented character. This approach avoids the combination of characters and accents that causes an increase in the number of classes to be considered with higher recognition accuracy. An OCR system for handwritten text documents in Kannada using Support Vector Machine and Zernike Moments features is described [18]. The recognition is independent of the size of the handwritten text and the system has achieved the recognition rate around 94 %. The paper [19] proposes a new technique of OCR using Gabor filters and Support Vector machines (SVM). The model proposed is trained and validated for two languages – English and Tamil and works for the entire character set in both the languages including symbols and numerals. In addition , the model can recognise the characetrs of six different fonts in English and Twelve different fonts in Tamil. The average accuracy of recognition for English is 97% and for Tamil it is 84%, which is achieved in just three iterations of training. The work [20] deals about reconstructing handwritten scanned images into text. Using supervised learning algorithm SVM for classification, classes are mapped onto Unicode for recognition. Finally the text is reconstructed using Unicode fonts which are subjected to readable and editable documents. A simple method for converting ancient Tamil handwritten scripts into text format is proposed [21]. There are thousands of Tamil palm manuscripts that are yet to be digitalized. The aim of this paper is to convert the palm manuscript image into digitized text format. In the paper [22], the research objective for recognizing Ancient Tamil handwritten characters , is fulfilled by applying the genetic algorithm technique based on the basic features of handwritten characters namely: loop, line, and location of loop and line connection. The system generates 66-bit string chromosome to represent a handwritten character. Then the system uses the 66-bit string chromosome to identify each handwritten character.

An Automatic License Plate Recognition (ALPR) system using Python and OpenCV [23], addresses the complex characteristics due to light and speed. ALPR systems is implemented using Open source tools and software including Python and the Open Computer Vision Library. Character Recognition of 94.3% is achieved.

3. PROPOSED METHOD AND BACKGROUND OF THE APPROACHES

The proposed system classifies the characters of ancient Kannada Scripts from three different periods Ashoka, Kadamba and Satavahana. The methods and techniques used in the proposed work are described in this section:

3.1 The System Architecture

Figure 2 depicts the System Architecture for Classification of Ancient Kannada Epigraphical characters.

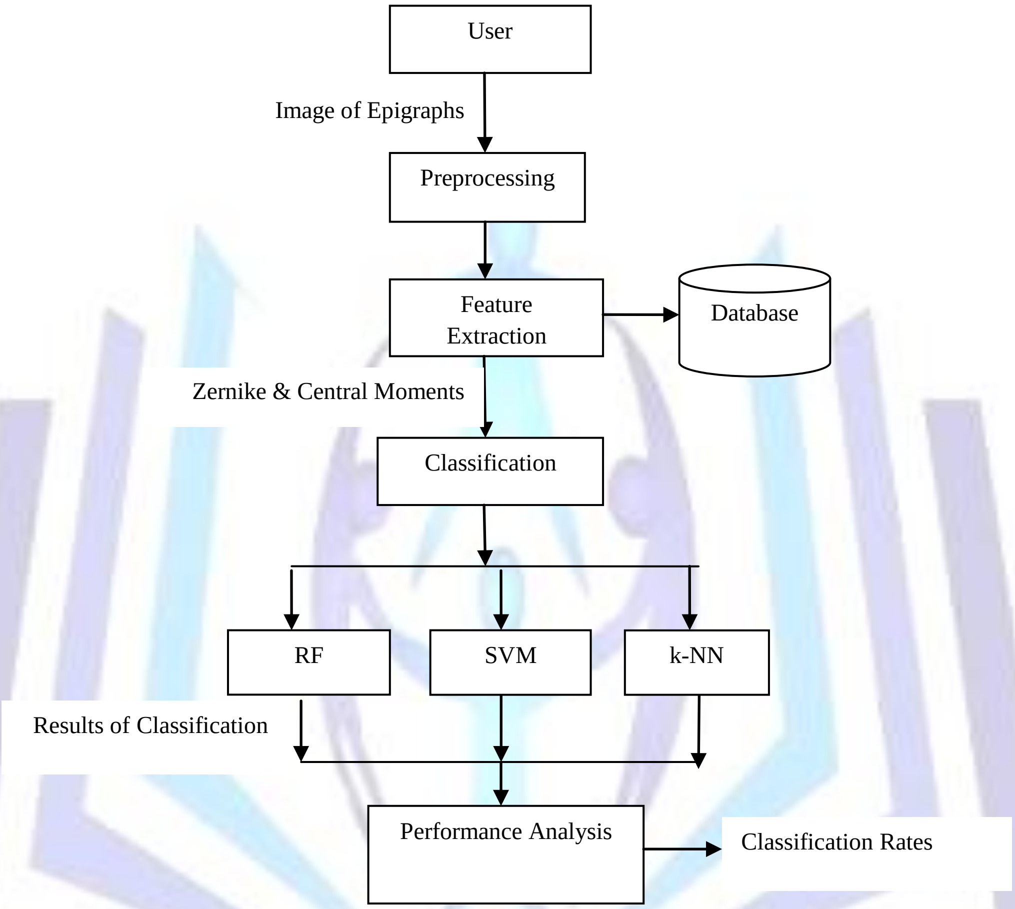


Figure 2: Classification of Ancient Kannada Epigraphical Characters

- User: A reconstructed ancient Kannada epigraph image, from one of the period- Ashoka, Kadamba and Satavahana is provided as input by the user .
- Preprocessing: The input image is binarized and segmented
- Feature Extraction: The Moment based features are extracted for the segmented characters and saved in a text file.
- Database: The database here represents the file used to save the extracted features in comma-delimited format. The feature vectors can be used for training the classifier and later classification of test data .
- Classification: The RF classifier is designed as described in [2]. The designed RF is trained using the features stored in the file. The trained Classifier is used to classify the ancient Kannada characters.

  Similarly the SVM and k-NN classifier provided in OpenCV are used in classification of ancient charcters.
- Performance Analysis: The performance characteristics of the designed RF are measured w.r.t. Time for training and testing ancient characters, and also accuracy of classifying the trained data and test characters.

  Finally, the Classification accuracy of the RF is compared with that of SVM and k-NN for the characters of Ashoka, Kadamba and Satavahana.

3.2 Classification Methods Used In The Proposed Work

Many text classifiers have been proposed in the literature using machine learning techniques, probabilistic models, etc. They often differ in the approach adopted: decision trees, naıve-Bayes, rule induction, neural networks, nearest neighbors, and support vector machines. Although many approaches have been proposed, automated text classification is still a challenging area of research. Hence in the current work a comparative study of the designed RF classifier with that of SVM and k-NN supported in OpenCV libraries, is made in classification of characters of ancient Kannada Script.

3.2.1 *Random Forest (RF)*

In this work, we use the random forest (RF) classifier for document image classification. The RF is an ensemble-based learning algorithm which constructs a set of tree-based classifiers, and then classifies new data points by taking a vote of the predictions of each classifier. In RF, the classifiers are the decision trees and it constructs a series of Classification Trees which will be used to classify a new example. For reducing the variance of an estimated prediction function of a Forest, a technique known as bagging or bootstrap aggregation is used. The idea used to create a classifier model is constructing multiple decision trees, each of which uses a subset of attributes randomly selected from the whole original set of attributes. There are multiple reasons to select the RF over other classifiers for this problem. The RF has been shown to work well when many features (on the order of thousands) are available. It does not over-fit with the increase in number of features and increases diversity among the classifiers by resampling the data, and by changing the feature sets over the different classifiers (trees). Random selection of features to split each node makes it more robust to noisy data.

3.2.2 *Support Vector Machine (SVM)*

The objective of any machine capable of learning is to achieve good generalization performance, given a finite amount of training data, by striking a balance between the goodness of fit attained on a given training dataset and the ability of the machine to achieve error-free recognition on other datasets. With this concept as the basis, support vector machines have proved to achieve good generalization performance with no prior knowledge of the data [25]. The principle of an SVM is to map the input data onto a higher dimensional feature space nonlinearly related to the input space and determine a separating hyperplane with maximum margin between the two classes in the feature space. A support vector machine is a maximal margin hyperplane in feature space built by using a kernel function in gene space. This results in a nonlinear boundary in the input space. The optimal separating hyperplane can be determined without any computations in the higher dimensional feature space by using kernel functions in the input space.

An SVM in its elementary form can be used for binary classification. It may, however, be extended to multiclass problems using the one-against-the-rest approach or by using the one-against-one approach. We begin our experiment with SVM's that use the Linear Kernel because they are simple and can be computed quickly. There are no kernel parameter choices needed to create a linear SVM, but it is necessary to choose a value for the soft margin (C) in advance.

A classification task usually involves with training and testing data which consist of some data instances. Each instance in the training set contains one "target value" (class labels) and several "attributes" (features). The goal of SVM is to produce a model which predicts target value of data instances in the testing set which are given only the attributes. Given a training set of instance-label pairs (xi, yi), i = 1,2. . . , where xi ∊ Rn and y ∊ 2 {1,−1} l, the support vector machines(SVM) require the solution of the following optimization problem:

$$\min_{w,b,\xi} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{subject to} \quad y_i(w^T\phi(x_i)+b) \geq 1 - \xi_i ,$$
$$\xi_i \geq 0$$

where ξ is an l-dimensional vector, and ω is a vector in the same feature space as the xi . The values ω and b determine a hyper plane in the original feature space, giving a linear classifier. Here training vectors xi is mapped into a higher (may be infinite) dimensional space by the function φ. Then SVM finds a linear separating hyper plane with the maximal margin in this higher dimensional space. C > 0 is the penalty parameter of the error term. Furthermore, K(xi, xj) ≡ φ(xi)^T φ(xj) is called the kernel function.

Commonly used kernels apply linear classification techniques to non-linear classification problems, based on the concept of decision planes that define decision boundaries. Commonly used kernels include:

1. Linear Kernel :

$$K(x, y) = x.y$$

2. Radial Basis Function (Gaussian) Kernel :

$$K(x,y) = \exp(-||x - y||^2 / 2\sigma^2)$$

3. Polynomial Kernel :

$$K(x, y) = (x.y + 1)^d$$

3.2.3 *k-Nearest Neighbors (k-NN)*

Another simple classifier is a K-nearest neighbor classifier with a Euclidean distance measure between input images [26]. The algorithm caches all of the training samples, and predicts the response for a new sample by analyzing a certain number (K) of the nearest neighbors of the sample (using voting, calculating weighted sum etc.) The method is sometimes referred to as “learning by example”, because for prediction it looks for the feature vector with a known response that is closest to the given vector. This classifier has the advantage that no training time, and no complex processing on the part of the designer, are required. However, the memory requirement and recognition time are large: the complete training images must be available at run time.

4. SYSTEM STRUCTURE CHART AND METHODOLOGY

The structure chart of the system designed for classification of ancient Kannada characters using RF is shown in Figure 3

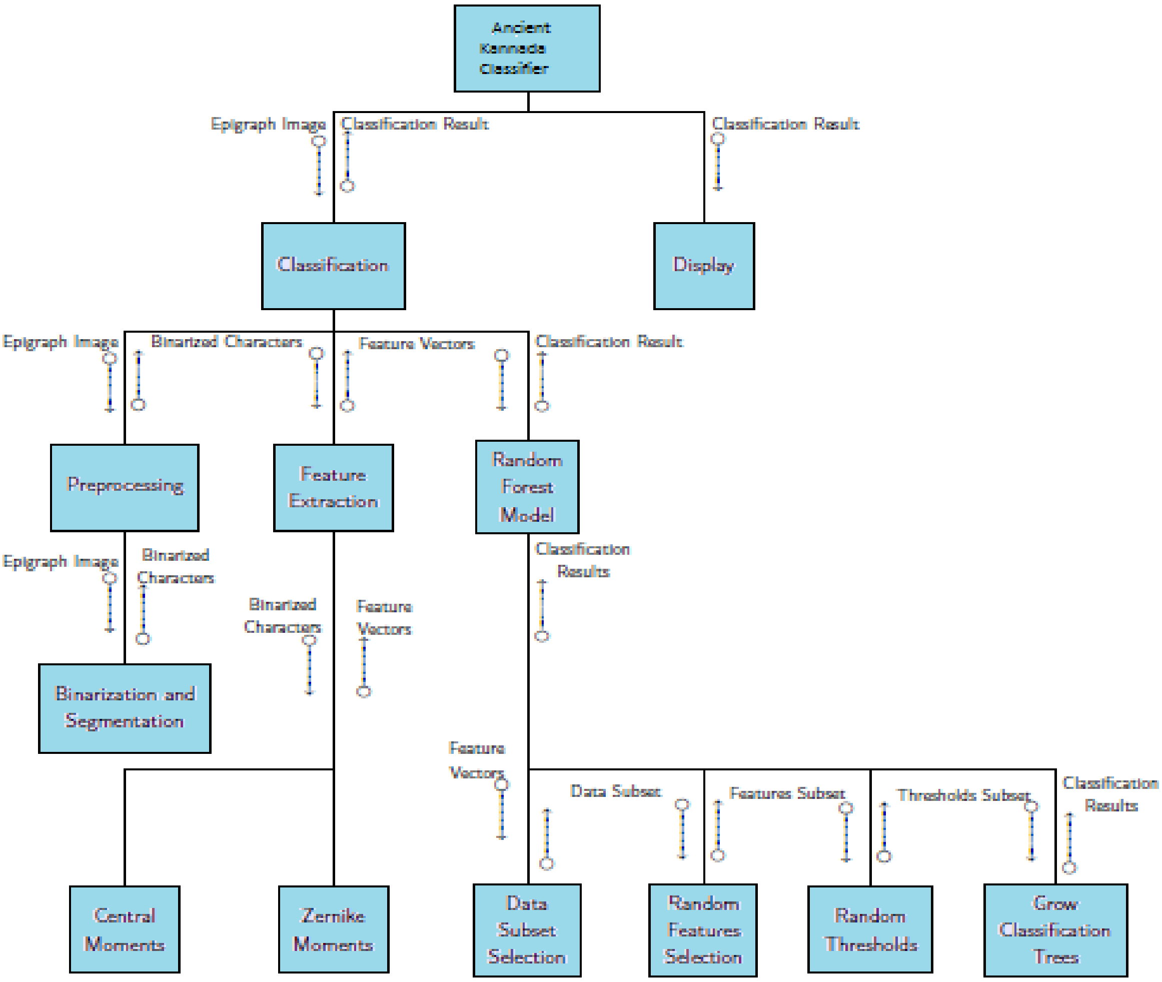


Figure 3: Structure Chart For The Classification Of Ancient Kannada Epigraphical Characters Using RF

Input: The image of a Kannada epigraph in PNG format.

Output: Classification of characters from Ashoka, Satavahana, Kadamba using the trained RF Classifier.

The steps towards the classification are as follows:

The steps towards Feature Extraction are:

Step 1: [Preprocessing]: A reconstructed image of an epigraph is read in grayscale format, converted to binary image using Otsu’s approach and is segmented to characters using connected components labeling.

Step 2: [Feature Extraction]: The Normalized Central Moments and Normalized Zernike Moments are computed, and write the computed feature vectors to a text file.

Step 3: [Random Forest Classification]

Step 3a:* [*Load Text*] *: Gets the feature vectors from the text and saves it in two arrays, one consisting of the classes and the other consisting of feature vectors of the corresponding classes.

Step 3b:* [*Fit Forest*] *: Train the trees in the RF which can be used to classify the ancient Kannada characters.

Step 3c:* [*Fit Tree*]*: A random subset of the training data from the sub-module Fit Forest is taken as input and a single Classification Tree for the given subset of data is made.

Step 3d:* [*Get Gini Impurity*] *: Determines the impurity index of a subset of classes and corresponding data for the node so that it can determine the best split and the best threshold value for that feature.

Step 3e: [*Predict*] *:* Considering the data consisting of feature vectors, predicts the class of the the given test characters using the trained RF Classifier.

5. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results and analysis of the designed RF [2] for classifying ancient Kannada Epigraphical characters is discussed.

The SVM and KNN classifiers provided by the OpenCV library are used to analyze the performance characteristics. The parameters used for the classifiers are as follows:

SVM classifier: Kernel type – Linear; SVM type – C-SVM (type 1)

KNN classifier: k = 5

5.1 Experimental Results and Evaluation Metrics

The dataset of the epigraphs considered are reconstructed images of ancient Kannada Scripts. The RF Classifier is tested on handwritten base characters belonging to Ashoka, Satavahana and Kadamba dynasties. For each dynasty, 105 handwritten samples with 35 base characters are considered. Two-thirds of the data is used for training and the remaining one-third is taken for testing the Classifiers.

Figure 4 shows the GUI of the system with the epigraph image selected and Figure 5 depicts the feature vectors provided for training the classifiers.

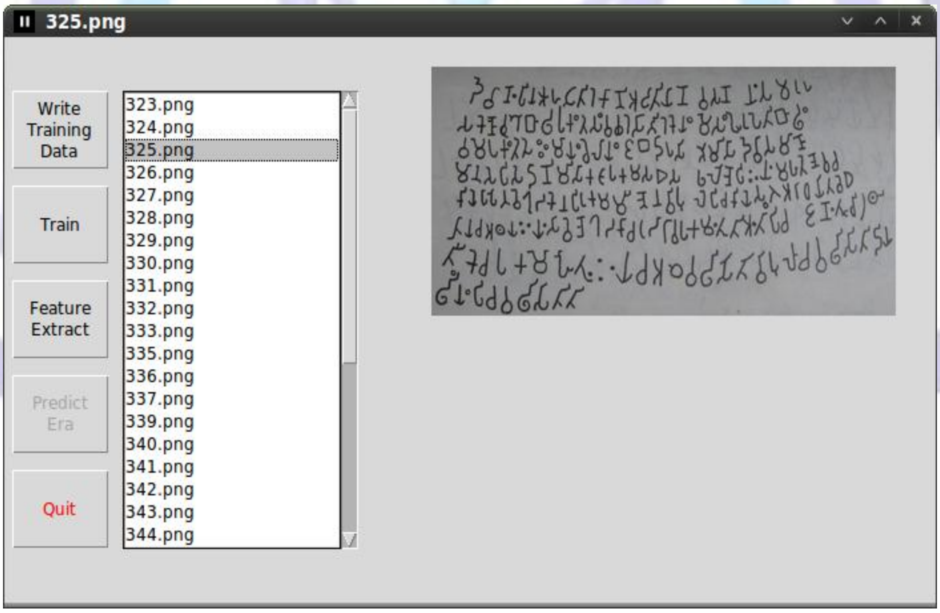
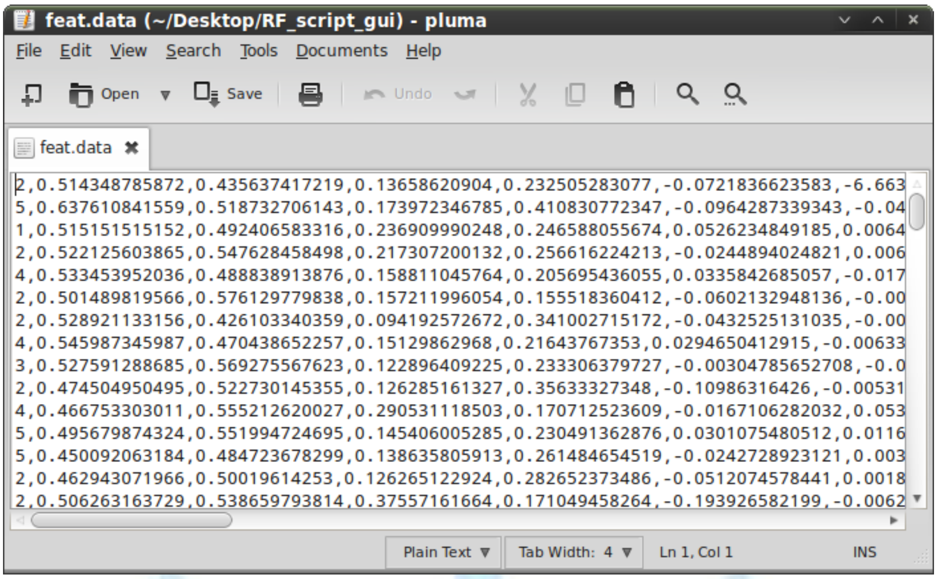


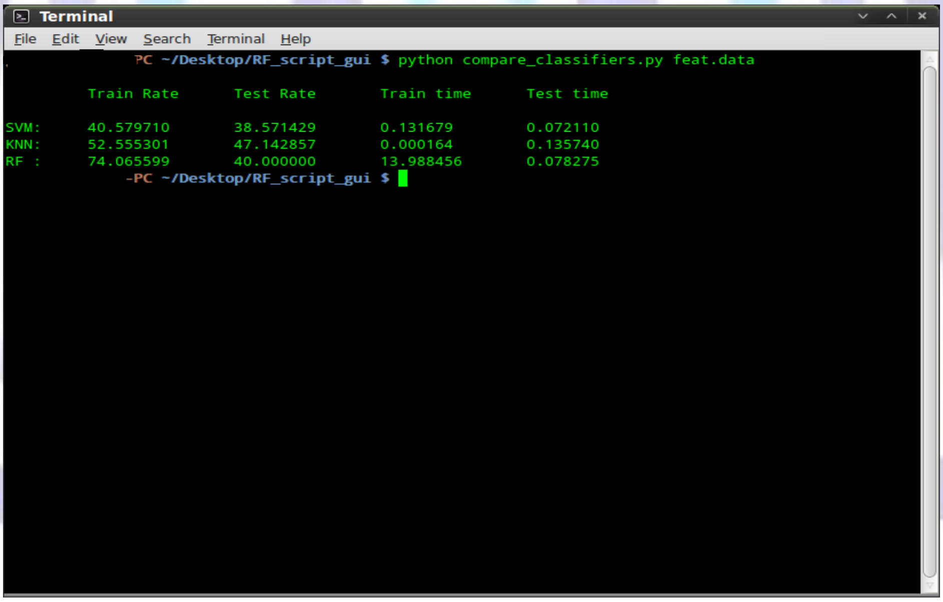
Figure 4: Epigraph Image Selected for Testing


feat.data (~/Desktop/RF_script_gui) - pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
feat.data
Plain Text
Tab Width: 4
Ln 1, Col 1
INS

```
2,0.514348785872,0.435637417219,0.13658620904,0.232505283077,-0.0721836623583,-6.663
5,0.637610841559,0.518732706143,0.173972346785,0.410830772347,-0.0964287339343,-0.04
1,0.515151515152,0.492406583316,0.236909990248,0.246588055674,0.0526234849185,0.0064
2,0.522125603865,0.547628458498,0.217307200132,0.256616224213,-0.0244894024821,0.006
4,0.533453952036,0.488838913876,0.158811045764,0.205695436055,0.0335842685057,-0.017
2,0.501489819566,0.576129779838,0.157211996054,0.155518360412,-0.0602132948136,-0.00
2,0.528921133156,0.426103340359,0.094192572672,0.341002715172,-0.0432525131035,-0.00
4,0.545987345987,0.470438652257,0.15129862968,0.21643767353,0.0294650412915,-0.00633
3,0.527591288685,0.569275567623,0.122896409225,0.233306379727,-0.00304785652708,-0.0
2,0.474504950495,0.522730145355,0.126285161327,0.35633327348,-0.10986316426,-0.00531
4,0.466753303011,0.555212620027,0.290531118503,0.170712523609,-0.0167106282032,0.053
5,0.495679874324,0.551994724695,0.145406005285,0.230491362876,0.0301075480512,0.0116
5,0.450092063184,0.484723678299,0.138635805913,0.261484654519,-0.0242728923121,0.003
2,0.462943071966,0.50019614253,0.126265122924,0.282652373486,-0.0512074578441,0.0018
2,0.50626316372 9,0.538659793814,0.37557161664,0.171049458264,-0.193926582199,-0.0062
```

Figure 5: The Feature Vectors Provided For Training The RF Classifier

Figure 6 demonstrates the screen shot of the performance characteristics of RF, SVM and k-NN w.r.t classification accuracy of trained characters (Train Rate), classification accuracy of new characters (Test Rate), the times for training (Train time) and the time for testing (Test time).



```
              PC ~/Desktop/RF_script_gui $ python compare_classifiers.py feat.data

         Train Rate       Test Rate        Train time       Test time

SVM:     40.579710        38.571429        0.131679         0.072110
KNN:     52.555301        47.142857        0.000164         0.135740
RF :     74.065599        40.000000        13.988456        0.078275
             -PC ~/Desktop/RF_script_gui $ 
```

Figure 6: Comparison Of Performance Characteristics of RF, SVM And K-NN Classifiers

5.1.1 *Evaluation Metrics*

Metrics are the various measures which facilitate the quantification of some particular characteristics. The metrics used to evaluate the proposed system are:

- ***Classification rate:*** This metric is used to determine the accuracy of the Classifier, which is given by the number of correct classifications out of the total number of samples considered.

$$\text{Classification rate} = \frac{\text{Number of correctly classified characters}}{\text{Total number of characters in the data}}$$

- ***Training time:*** This metric measures the time taken to train the Classifier.
- ***Classification time:*** The classification time is the time taken to predict the class labels for the given set of inputs.

5.2 Performance Characteristics of RF Classifier

5.2.1 *Classification Rate of RF on the Characters from Satavahana Period*

The accuracy of RF in classifying characters from trained data set of Satavahana period for the threshold value 10 and varying number of trees are tabulated in Table 1. The plot in Figure 7 illustrate the results of the same on trained data.

Table 1: Classification Rates In Percentage For Trained Data

		Number of Trees		
Thresholds		**10**	**20**	**30**
	10	47.692308	81.538462	90.769231
	20	50.769231	69.230769	84.615385
	30	49.230769	73.846154	87.692308

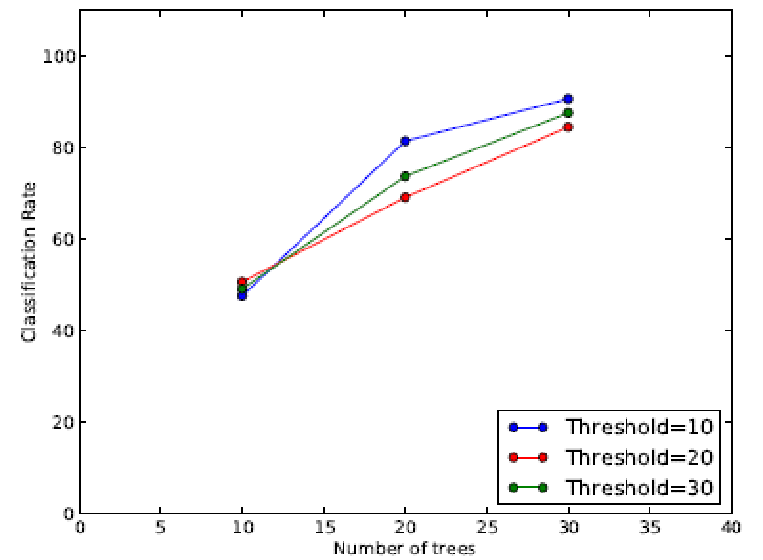


Figure 7: Classification Rates Of RF With Different Parameters For Trained Data

The Classification rate for new characters are tabulated in Table 2 and shown in Figure 8.

Table 2: Classification Rates In Percentage For Test Data

		Number of Trees		
Thresholds		**10**	**20**	**30**
	10	43.529412	52.352941	67.058824
	20	44.529412	52.352941	70.000000
	30	58.235294	53.352941	61.176471

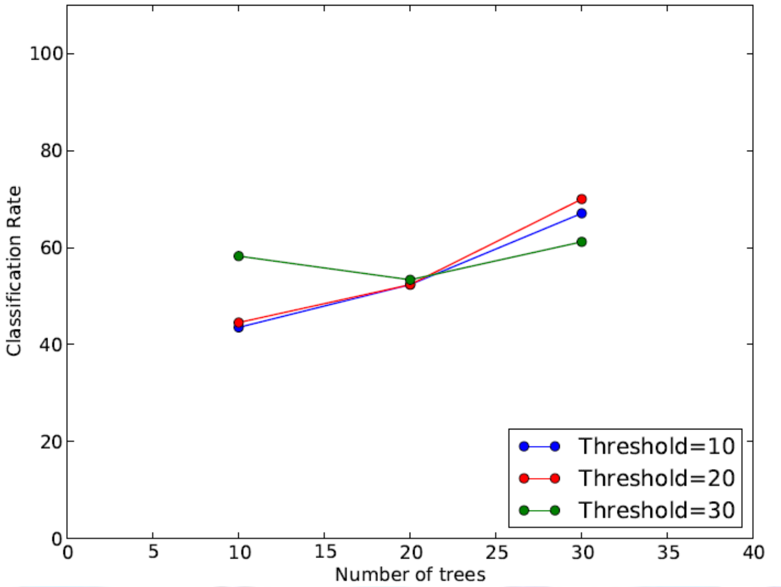


Figure 8: Classification Rates Of RF With Different Parameters For Test Data

5.2.2 *Times for training and testing of RF on the characters from Satavahana period*

The time (in sec) for training characters from Satavahana period are tabulated in Table 3 and plotted in Figure 9 respectively. As the number of trees in the forest increases, the time taken for training also increases proportionately.

Table 3: Training Time Taken In Seconds By Random Forest

		Number of Trees		
Thresholds		**10**	**20**	**30**
	10	3.529771	7.415497	11.617379
	20	6.624592	11.990089	19.072764
	30	7.786249	15.838297	22.640056

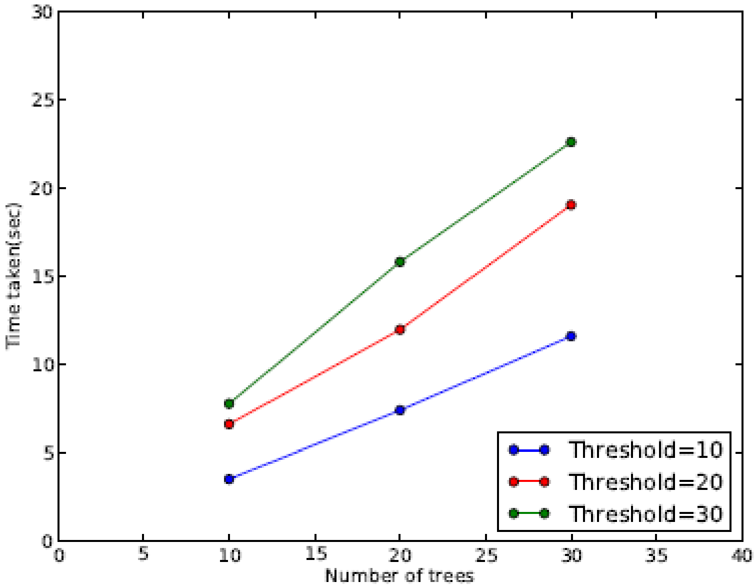


Figure 9: Training Time For RF With Different Parameters

The classification times (in seconds) for new characters are tabulated in Table 4. Each column indicates the number of trees used and the rows indicate the threshold values.

Table 4: Classification Time Taken In Seconds For RF

		Number of Trees		
		10	**20**	**30**
Thresholds	**10**	0.018280	0.036135	0.54362
	20	0.018697	0.036706	0.055347
	30	0.018132	0.036499	0.53434

The plot for the classification times for different parameters is shown in Figure 10.

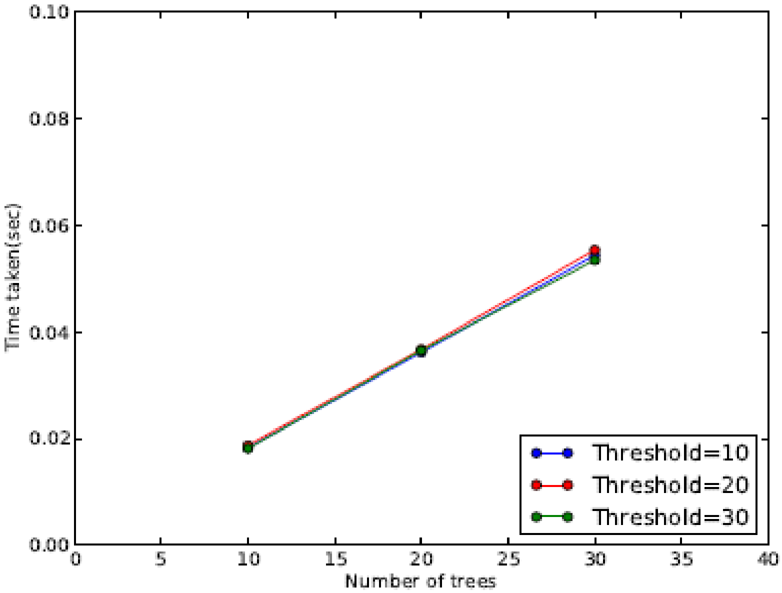


Figure 10: Classification Time For RF For Different Values Of Threshold And Number Of Trees

5.3 Comparison of the Classification accuracy of the designed RF with SVM and k-NN on ancient Kannada Characters:

The designed RF Classifier is tested on 105 handwritten samples with 35 base characters, belonging to each of the dynasties: Ashoka, Kadamba and Satavahana dynasty.

These handwritten characters are also tested with the SVM and KNN classifiers provided by the OpenCV library. The parameters used for the classifiers are as follows:

SVM classifier: Kernel type – Linear; SVM type – C-SVM (type 1)

KNN classifier: k = 5

RF classifier: Number of trees = 30; number of thresholds = 10

5.3.1 *Classification of characters from Ashoka period*

The classification rates for handwritten ancient Kannada characters from Ashoka period are tabulated in Table 5. The number of samples taken for training are 70 and 35 samples are taken for testing. The accuracy in the classification of characters is compared using the designed RF with SVM and k-NN

Table 5: Classification Rates In Percentage For Scripts From Ashoka Period

Classifier		**Training**	**Testing**
	SVM	88.405797	77.125432
	KNN	56.521739	44.444444
	RF	94.202899	69.222222

The Figure 11 shows the corresponding plot representing the classification rates using RF,SVM and k-NN, for characters from Ashoka period .

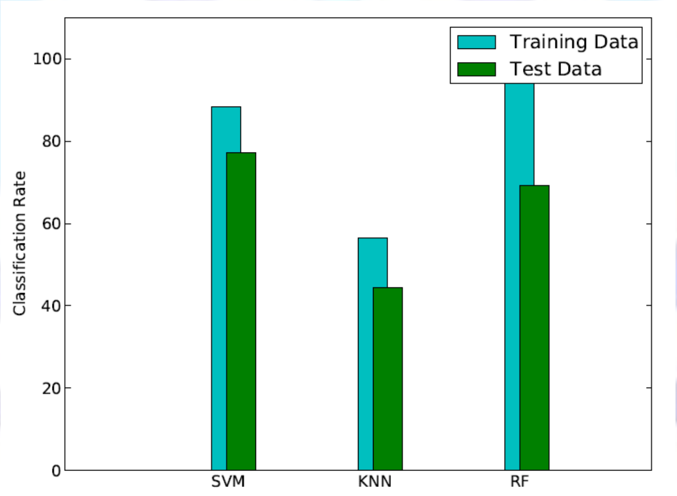


Figure 11: The Classification Accuracy Of Handwritten Characters From Ashoka Period

5.3.2 *Classification of characters from Kadamba period*

The classification accuracy, of the designed RF is compared with SVM and k-NN, for handwritten ancient Kannada characters from Kadamba period and are tabulated as in Table 6. The number of samples taken for training is 70 and 35 for testing.

Table 6: Classification Rates In Percentage For Scripts From Kadamba Period

Classifier		Training	Testing
	SVM	77.615385	69.941176
	KNN	52.230769	48.176471
	RF	87.153846	65.117647

The Figure 12 shows the corresponding plot representing the classification rates using RF, SVM and k-NN, for characters from Kadamba period.

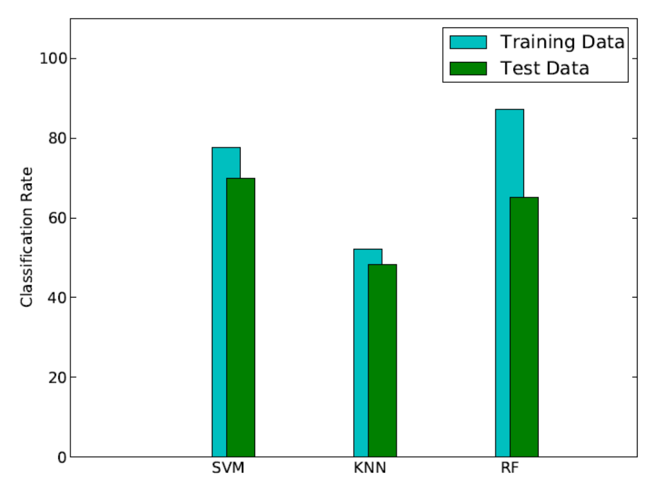


Figure 12: The Classification Rates Of Handwritten Characters From Kadamba Period

5.3.3 *Classification of characters from Satavahana period*

The classification rates for handwritten ancient Kannada characters from Satavahana period are tabulated in Table 7. The number of samples taken for training are 70 and 35 samples are taken for testing. The accuracy in the classification of characters using the designed RF with SVM and k-NN is being compared.

Table 7: Classification Rates In Percentage For Scripts From Satavahana Period

Classifier		Training	Testing
	SVM	81.538462	74.117647
	KNN	58.461538	52.941176
	RF	90.769231	67.058824

The Figure 13 represents the corresponding plot of accuracy, in classification of characters from Satavahana period, using RF, SVM and k-NN

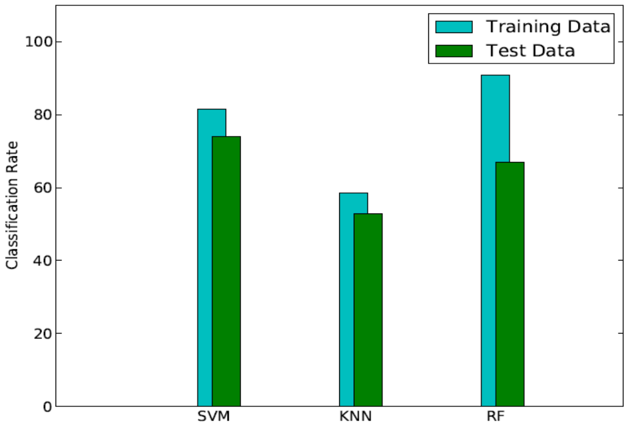


Figure 13. Classification Rates Of Handwritten Characters From Satavahana Period

The following inferences are drawn from the performance analysis:

- The accuracy in classification of the trained data is at least 1.08 times greater than the classification rate of new characters for any classifier.
- There is a linear increase of classification rate as the number of trees in the forest is increased, but no significant changes when the number of thresholds is increased.
- The training time is directly proportional to the number of classification trees and the number of thresholds.
- The classification time is directly proportional to the number of classification trees. It is not dependant on the number of thresholds since it is used only when growing the trees.
- The training time of the RF classifier is about 200 times more than the classification time. This is because most of the time is spent for the calculation of Gini index during training. Classification involves only a comparison at each node till it reaches the leaf.
- RF Classifier works better in classifying the trained characters. Hence a well trained RF classifier will perform comparatively better than SVM and k-NN in classification of new characters.
- SVM classifier works well in classification of new characters. Hence SVM performs comparatively better than RF and k-NN, when the classifier is not trained with sufficient samples.
- KNN Classifier's capability lies in between RF and SVM in classification of trained and test data.

6. CONCLUSION

A RF classifier is designed and tested in classification of ancient Kannada epigraphical characters, prevailing during the regime of Ashoka, Satavahana and Kadamba dynasties. The performance characteristics of RF, in terms of accuracy and time, is observed for classification of trained and test characters from Satavahana dynasty. The Performance analysis of RF illustrates that, fixing the number of thresholds at 10 would be a good tradeoff between training time and classification rate. Finally a comparative study on the performance of designed RF with SVM and k-NN is carried out, in the classification of ancient characters from Ashoka, Satavahana and Kadamba dynasties. The strengths and weaknesses of these classification methods is also discussed. The current work can be further extended on recognition of characters from ancient scripts. Thus it finds scope in development of an automatic recognition system, for deciphering ancient epigraphical records, which is of greater relevance to archeologists in exploring the details of past.

REFERENCES

[1] Alirezaee, S., Aghaeinia, H., Ahmadi, M., Faez, K., "Recognition of middle age Persian characters using a set of invariant moments," in Proc. 33rd Appl. Imag. Pattern Recognit. Workshop, Washington, DC, 2004, pp. 196-201.

[2] Bernard, S., "Using Random Forests for Handwritten Digit Recognition," in Int. Conf. Document Anal. and Recognit., vol. 2, Parana, 2007, pp. 1043-1047.

[3] De Cao Tran, Patrick Franco, Jean-Marc Ogier, "Accented Handwritten Character Recognition Using SVM – Application to French" in 12th International Conference on Frontiers in Handwriting Recognition, 2010.

[4] E. K. Vellingiriraj, P. Balasubramanie , “Recognition of Ancient Tamil Handwritten Characters in Palm Manuscripts Using Genetic Algorithm”, International Journal of Scientific Engineering and Technology (ISSN : 2277-1581) ,Volume 2 Issue 5, pp : 342-346 1 May 2013.

[5] E.K.Vellingiriraj, Dr.P.Balasubramanie, “A Multimodal Framework for the Recognition of Ancient Tamil Handwritten Characters in Palm Manuscript Using Boolean Bitmap Pattern of Image Zoning”.

[6] Garz, Angelika, and Robert Sablatnig, “Multi-scale texture-based text recognition in ancient manuscripts,” in Int. Conf. Virtual Syst. and Multimed., IEEE, Seoul, , 2010, pp. 336- 339.

[7] Garz, Angelika, Markus Diem, and Robert Sablatnig, “Detecting Text Areas and Decorative Elements in Ancient Manuscripts,” in Int. Conf. Front. in Handwrit. Recognit., IEEE, Kolkata, , 2010, pp. 176-181.

[8] G. G. Rajput, R. Horakeri, S. Chandrakant, “Printed and handwritten mixed Kannada numerals recognition using SVM,” in Int. J. Comput. Sci. and Eng., vol. 2, no. 5, 2010, pp.1622-1626.

[9] Kumar, S. Raja, and V. Subbiah Bharathi, “An Off Line Ancient Tamil Script Recognition from Temple Wall Inscription using Fourier and Wavelet Features,” in Eur. J. Sci. Res., vol. 80, no. 4, 2012, pp. 457-464.

[10] Kashyap, K. Harish, and P. A. Koushik, “Hybrid neural network architecture for age identification of ancient kannada scripts,” in Proc. 2003 Int. Symp. Circuits and Syst., IEEE, vol. 5, 2003, pp. 661-664.

[11] K.M. Sajjad, “Automatic License Plate Recognition using Python and OpenCV”.

[12] L. Breiman. Random forests. Machine Learning, 45:5–32, 2001.

[13] M. Zahedi, S. Eslami, “Improvement of Random Forest Classifier through Localization of Persian Handwritten OCR,” in ACEEE Int. J. Inf. Technol., vol. 1, no. 2, 2012, pp. 31-36.

[14] Meza-Lovn, Graciela Lecireth, “A Graph-Based Approach for Transcribing Ancient Documents,” in Proc. 13th Ibero-Am. Conf. AI, Cartagena de Indias, Colombia, 2012, pp. 210-220.

[15] Papaodysseus, Constantin, Panayiotis Rousopoulos et al. “Handwriting automatic classification: Application to ancient Greek Varzim, 2010, pp. 1-6.

[16] Rashid, Sheikh Faisal, Faisal Shafait, and Thomas M. Breuel, “Connected componen tlevel multiscript identification from ancient document images,” in IAPR Workshop Document Anal. Syst., 2010, pp. 1-4.

[17] R.Ramanathan, S.Ponmathavan, N.Valliappan, Dr. K.P.Soman “Optical Character Recognition for English and Tamil Using Support Vector Machines” in International Conference on Advances in Computing, Control, and Telecommunication Technologies, 2009.

[18] R. Dinesh Kumar and J. Suganithi, “Offline Handwritten Sanskrit Character Recognition using Support Vector Machines”, in Journal of Environmental Science, Computer Science and Engineering & Technology, JECET; June – August-2013; Vol.2.No.3, 769-775.

[19] Soumya A and G Hemantha Kumar, “SVM Classifier For The Prediction Of Era Of An Epigraphical Script,” in Int. J. Peer to Peer Netw., vol.2, no.2, 2011, pp. 12-22.

[20] Soumya A and G Hemantha Kumar, “Dating of Ancient Epigraphs using Random Forest Classifier,” in International Conference on Emerging Computation and Information Technologies (ICECIT-2013), Elsevier publications, 2013, pp. 331-339.

[21] Sandhya Arora, Debotosh Bhattacharjee , Mita Nasipuri , L. Malik , M. Kundu and D. K. Basu, “Performance Comparison of SVM and ANN for Handwritten Devnagari Character Recognition”, International Journal of Computer Science Issues (IJCSI), Vol. 7, Issue 3, May 2010”.

[22] Siddiqi, Imran, Florence Cloppet, and Nicole Vincent, “Contour Based Features for the Classification of Ancient Manuscripts,” in Int. Conf. Graphonomics Society, Dijon, France, 2009.

[23] Thungamani.M, Dr Ramakhanth Kumar P, Keshava Prasanna, Shravani Krishna Rau , “Off-line Handwritten Kannada Text Recognition using Support Vector Machine using Zernike Moments”, in International Journal of Computer Science and Network Security, VOL.11 No.7, July 2011.

[24] Vajda, Szilrd, Akmal Junaidi, and Gernot A. Fink, “A semi-supervised ensemble learning approach for character labeling with minimal human effort,” in Int. Conf. Document Anal. and Recognit., Beijing, IEEE, 2011, pp. 259-263.

[25] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, “Comparison of learning algorithms for handwritten digit recognition”.

[26] Zaghden, Nizar, Remy Mullot, and Adel M. Alimi, “Characterization of ancient document images composed by Arabic and Latin scripts,” in Int. Conf. Innov. Inf. Technol., IEEE, Abu Dhabi, 2011, pp. 124-127.