An Automated Ancient Tamil Script Classification System Using Machine Learning Techniques

Suganya T.S.[1]* and Murugavalli S.[2]
1. Sathyabama University Chennai, Tamilnadu, INDIA
2. CSE Department, Panimalar Engineering College, Chennai, INDIA
*tssuganya@yahoo.com

Abstract
Information pertaining to history can be acquired through inscriptions that are found worldwide. The regional languages of a particular origin were used in writing inscriptions. One of the most ancient language in the world is 'Tamil' with rich heritage and literature. The writings were encrypted using various materials like stones, metals, palm leaf, conch shells, and copper plate. These inscriptions are rich in information pertaining to astronomy, history, culture, religious, economic tax, administrative and educational conditions. This paper uses Shape and Hough transform for feature extraction and neural network to recognize the ancient Tamil script.

Keywords: Tamil Inscription, Neural Network, J48, k Nearest Neighbor, Naïve Bayes, Feature extraction, Shape and Hough transform.

Introduction

The historical places in India house majority of the inscriptions. The archaeological department plays a significant role in preserving and translating them. The surface on which these inscriptions were written are stones, metals, rocks, copper plate as well as on palm leaf. However, the mediums utilized for writing and the toughest challenge for the epigraphists is the forms of graphemes as their style is vast with inscriptions bearing scratches as well as cracks because of ageing [1]. The Archaeological department focuses on preserving the inscriptions as well as has taken several initiatives to preserve and develop epigraphy. The era of computers play a vital role in the movement of computerization or digitization of inscriptions, removing disturbances as well as breakages by using image processing methods. Besides preserving the epigraphy it is also important to find epigraphers who can read them.

During the prehistoric period, Tamil Nadu had Lower Paleolithic settlements. From the estimations it could be deduced that the settlements had existed from about 1,510,000 BCE until around 3000 BCE. During most of the lower Paleolithic stage, the human settlements were found close to river valleys where there was thin forest cover orin grassland surroundings. The inhabitants were quite sparse as well as only two localities of lower Paleolithic civilization are found in south India until today. The fossils of animals as well as primitive stone tools were unearthed by the Archaeological department near the northern Tamil Nadu region. The findings are believed to be a part of 3,000,000 BCE. The population in Southern India belonged to the species, Homo erectus, and survived in Paleolithic 'old stone age' for a considerable period. Barely crude implements like hand axes and choppers were used by them to subsist as hunter gatherers.

The advent of Neolithic period was around 2500 BCE in Tamil Nadu. During Neolithic period stone tools were grinded and polished to make them into fine tools. A Neolithic axe head bearing the ancient writing was found in Tamil Nadu. The Neolithic people were found to be settled either on small flat hills or on foothills. Mostly, they had temporary settlements as they had migrate in search of grazing lands. The followed the practice of burying the dead within urns or pits. The Neolithic people started to use copper to make tools and weapons.

During Iron Age the humans used iron for making tools and weapons. In peninsular India many places have the Megalithic burial sites which mark the Iron Age culture. From the evidences obtained through excavations as well as typology of the burial monuments, it is believed that Iron Age had gradually moved from northern regions to the regions in south. Similarly through comparative excavations that were performed in Adichanallur in Thirunelveli district as well as in North India, it is evident that Megalithic culture had migrated towards southern regions.

Many literature works were referred to continue the funerary and burial practices of the sangam period in the post sangam centuries. Among them Manimekalai (5th century A.D), the well-known Buddhist epic refers in chapter 6 (66-67) to the several types of burials namely cremation (cuṭuvōr), post excarnation burial (iṭuvōr), burying the deceased in a pit (toṭukuḻippaṭuvōr), rock chamber or cist burial (tāḻvāyiṉaṭaippōr), urn burial encapped with lid (tāḻiyiṟkavippōr) are some of them. The above burial methods were found to exist even in the Sangam age where kingship as well as an ordained society had emerged The practice of construction of temples with the locally available stones must have taken its inspiration from the above discussed burial methods.] A literature form 7th century namely the Thevaram (7731) refers burial ground as muthukadu, which is found in Purananuru (356:4.According to it, the society gets wiped out as people perish but only the burials remain as a permanent memorial[2].

Many Indo-Aryan and Dravidian languages like Hindi, Bengali, Gujarati, Tamil and Kannada that are spoken in present-day India have their scripts derived from Brahmi. Brahmi, besides being the mother of scripts in India, it is also

used in neighboring countries like Sri Lanka, Myanmar, Thailand, and Tibet, amongothers. Additionally, even the kanas, the minimal units in Japanese writing system is believed to be derived from Brahmi. Aksara is the minimal unit in Brahmi and is a theoretically interesting component of the syllable, that is, (consonant) (consonant) (consonant) vowel (vowel) or only the final consonant. This unit is interpreted in terms of current models of syllable structure in generative phonology: The ancient linguistic concept of the unit aksara has modern currency [3].

A tradition in the Brahmi inscriptions and Brahmi potsherds was to inscribe Graffiti Marks either in the end or in the middle. It is no wonder that the same trend is seen on the TisamaragamaBrahmipotsherd. Typically, the Brahmi inscriptions written on caves are from left to the right with some exceptions where a few inscriptions in Tamil Nāṭu and Sri Lanka were written from the right to the left. The right to left inscriptions could have been due to tall and unreachable upper edges so they could have started writing from the upper part to the lower part of the cave [4].

No such difficulties would have occurred when writing on pottery as there are no reasons to write some inscription from left to right and to write new inscription from right to left. The inscriptions on pottery must have been written in the usual form of left to right as there are no evidences to prove dual trends of writing inscriptions on pottery. Consequently, the three letters which are read as "Tiṛaḷi" (jpusp) from right to left can be read as"Puḻaiti" (Giojp) from left to right. The first letter which gives the sound "Pu" (G), there is small bend to the right on the straight line which appears on the right side. This is slightly different from the letter "Pu" which appears on Brahmi cave inscriptions. These type of inscriptions are found on the Sri Lankan and Tamil Nāṭu pottery which bear Brahmiscripts.

From the Sangam collections it is evident that such inscriptions had prevailed. The poems of the Sangam collections are the kuruntokai, ainkurunrru, Purananuru, patirruppattu and in the grammar of tolk¹ppiyam, which are dated between the 2nd B.C. and 3rd CenturyA.D5. This fact can be considered to illustrate that Tamil must have adopted a well refined writing system during this time period. It is also evident from the fact that the grammar of Tolkappiyam defines the word eruttu to be consisting of thirty letters in a sequence from a to n along with the three other secondary forms.

எழுத்தெனப்படுப
eṛuttenappaṭupa
அகரமுதல் னகர இறுவாய் முப்பஃதென்ப
akaramutal nkara iṛuvāy muppaxtenpa
சார்ந்துவரன் மரபின் மூன்றலங் கடையே
cārntuvaran marapin mūṇṛalaṅ kaṭaiyē

Mahadevan (1966: 58), who presents an alphabet system of Tamil- Brahmi based on a corpus of inscriptions excavated in Tamil Nadu in various periods (See Figure 1),states that the Tamil- Br¹hm» script used during the first Millennium B.C. neither uses the dot, nor does it distinguish between short and long e and o by way of distinct symbols.

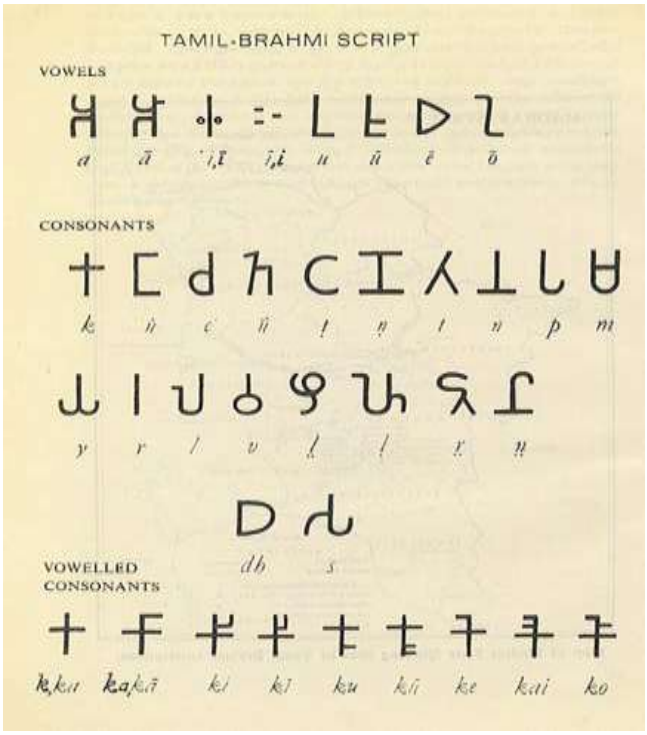

Figure 1: **Tamil- Brahmi Alphabet reconstructed based on various inscriptions found in Tamil Nadu (Mahadevan 1968: 56)**

The most important aspect in achieving high character recognition for ancient Tamil is identifying the suitable feature extraction method. Recently this factor has gained significance in research field in the domain of image processing as well as patterns recognition. By achieving this, the automation process can be advanced along with the improvement in the interface among man as well as machine in a number of applications. Recent studies are concentrating on new methods as well as approaches, which are capable of providing high accuracy with less computational time. The general classification of character recognition is off-line as well as on-line recognition methods. In the former recognition method, a high resolution digital camera is used in capturing the input image but whereas in latter, two dimensional coordinates of consecutive points are denoted as a function of time as well as the writer's order of strokes is also made available. The procedure of identifying letters as well as words that exist on ancient stone inscriptions is called Offline ancient Tamil text recognition. Then again, to achieve high recognition accuracy in the off-line methods, the neural networks are been employed

In recent times, the active area of research is finding new techniques for off-line ancient Tamil Character recognition to enhance recognition accuracy. The initial and the most important step in the system of recognizing ancient Tamil is pre-processing. Following this is the process of

segmentation as well as feature extraction. In Pre-processing, a number of stages are included to mold the input image into an adequate form for segmentation. The procedure of segmentation includes splitting of the input image into individual characters as well as resizing each of them into m x n pixels in order to train the network. The selection of recognition performance. A number of feature extraction techniques are now accessible to identify ancient Tamil characters. The task of classification and recognition is performed by the artificial neural networks at the backend. Since, neural networks helps in achieving high recognition precision for the offline ancient Tamil character recognition system, it is considered as the most reliable tool with high-speed [5].

This paper uses shape and hough transformation for feature extraction and classifies the features using neural network, j48, naïve bayes and knn classifiers. The other sections included are: Section 2 performs an overview of the related literature. Section 3 describes method and section 4 gives a discussion of the experiment outcomes. Section 5 gives the conclusion to the proposed work.

Related Work

Rajakumar and Bharathi[6] presented contour-let transform, a technique to recognize Tamil characters from stone inscriptions. From earlier studies, it could be deduced that Wavelet transforms are not suitable for perfectly restructuring curved images. This drawback can be rectified by using the contour-let transform. Contour-let transform signifies a threedimensional method while wavelet transform signifies a two-dimensional approach. Clustering mechanism is used to recognize characters from the input image and fuzzy median filters are used to eliminate the noise present. To obtain a precise recognition of Ancient Tamil characters, neural networks are used where it trains and compares the information with current century characters.

Mahalakshmi and Sharavanan[7] introduced a simple method to recognize and translate Tamil inscriptions. Besides this a detailed study was carried out on ancient Tamil inscriptions along with the presentation of current century characters. Using LabVIEW, the Tamil stone inscriptions were recognized and translated. Segmentation technique is used in segmenting the images of the ancient script. A few of the segmentation techniques used are Particle Swarm Optimization (PSO), Discrete PSO (DPSO) and Fuzzy PSO (FPSO). The images are enhanced using contour let transform and fuzzy median filters are utilized to eliminate the noise present in the image. Simulation of the suggested method was done under Matlab and LabVIEW.

Rajakumar and Bharathi[8] suggested the recognizing ancient Tamil character is a significant area of research and finds application in pattern recognition theory. It is also important to realize different methods to automate the process of inputting character at all instances. The author proposes an ancient Tamil character recognition protocol derived from artificial immune. This algorithm refers to immune biological principle and according to this; the character recognition rate is improved with decreased recognition training time. The proposed method was simulated and the results prove that this method has better speed as well as accuracy when compared to the traditional method of ancient Tamil character recognition derived from neural network. The protocol has features like self-adaptive learning as well as immune memory similar to the biology immune system. This is applied to detect abnormality as well as in recognizing patterns.

Rajakumar and Bharathi[9] suggested an innovative feature extraction technique that is capable of enhancing the results when two similar shaped ancient Tamil characters are taken for recognition. In this work, the author has used instances of Tamil characters from 6 distinct centuries. The approach has its basis in F-ratio, a statistic specified by the ratio between-class as well as within-class variance. F-ratio alters the features vector of 2 analogous shape characters by weighing the features components. Through this weighing strategy similar shaped characters are easily determined. This scheme works by improving the feature elements belonging to apparent parts of the similarly shaped characters as well as by minimizing features components of typical portions of characters. The suggested method uses gradient features and template matching method and a maximum recognition accuracy of 94% is achieved.

Rajakumar and Bharathi[10] investigated a few of the structural features that are helpful in offline Ancient Tamil character recognition .It is not possible to classify all the characters by only using Structural features. So it necessitates the use of some other features along with artificial neural networks for enhancing the performance of the system. The suggested protocol is capable of achieving precision of 97.9% for some letters at an average of 80% and also with respect to time consumption.

Kumar and Poornima[11] presented epigraphically inscriptions. This plays a significant role in discovering the civilized past and in classifying the characters belonging to various periods. The proposed system can read the ancient Tamil characters belonging to various periods by testing a small amount of characters called as examined characters in Tamil. Through automated means, the examined characters are extracted from the script as well as coordinated with the characters that belong to distinct periods utilizing machine intelligence. Therefore the suggested system is made of different modules like images acquisition, binarization, pre-processing, features extraction, segmentation as well as lastly the classifications as well as predictions of period using Transductive Support Vector Machine (TSVM). The simulation results shows higher accuracy when compared to the accuracy of Support Vector Machine (SVM).

Tomar et al[12] suggested a variety of technique to automatically identify the character and recognize the scripts. This manuscript was a concise survey on image prior-processing techniques, segmentation techniques and feature extraction and classification via dimensionality reduction techniques. A wide research was previously carried out in this domain but yet the ancient inscription character recognition was a challenging task which requires more effective methods. This review can be considered as the basis for the preliminary level of image preprocessing and the dimensionality reduction approaches in feature and classification.

Methodology

The framework for the proposed work is shown in figure 2.

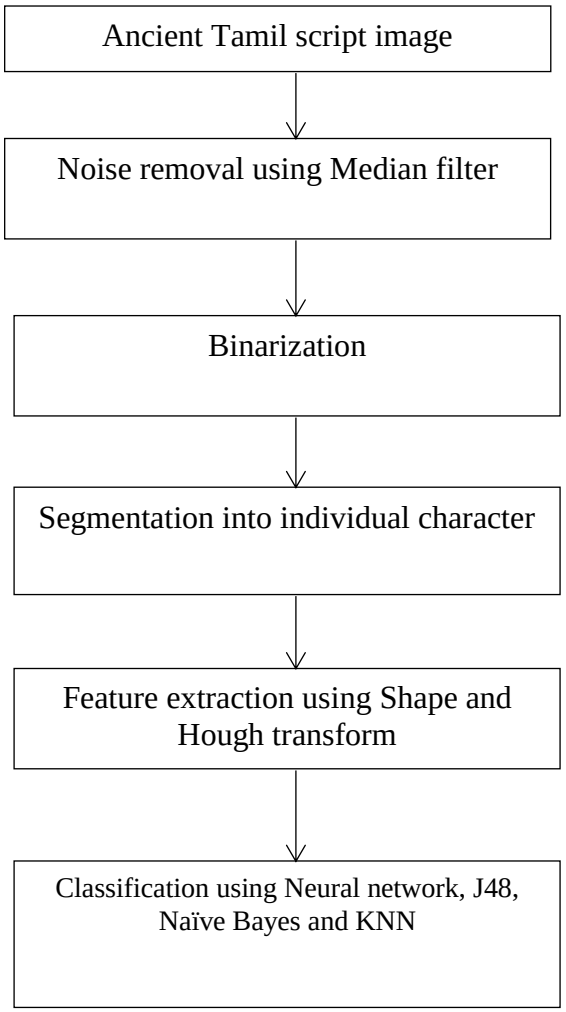

Figure 2: **Flowchart of the proposed work**

Noise removal using median filter: Noise present in an image is visible as grains and leads to arbitrary variation of image intensity. Noise generally occurs during image acquisition. Several algorithms are used to eliminate noise in an image and also, filters are employed to reduce and remove noise [13]. There are various kinds of noise present in document images and some of them are Salt and Pepper noise, Gaussian noise, Gamma noise, Uniform noise etc. Several type of filtering methods like Gaussian filtering method, Min-max filtering method etc. are applied to remove noise. Median filter is employed to eliminate salt and pepper noise.

The widely used filtering method in digital image processing is Median filtering as under certain conditions it can preserve the image edges while eliminating noise. It is a nonlinear local filter wherein output magnitude represents the middle component of an arranged array of pixel values from filter windows. Median filters works by filtering all pixels in the image in turns and its close neighbours determine if they are representative of surrounding. Median filter substitutes the magnitude of the pixel with median of the values. Initially, the neighboring values are arranged in numerical order while the value of the pixel under attention is substituted with the middle (median) pixel value.

The neighbourhood is called the window. The window is of different shapes centred on target pixel, a standard square shape is selected for window defined for two dimensional image. The middle value of the neighboring pixels is mostly the value of a pixel in the neighbourhood within the window. So the median filter will rarely create novel unrealistic pixel values. Since the sharp edges of the image are preserved, the median filtering method is considered to be superior to the mean filtering method. A major drawback in median filter is that it always has a constrained output, as median value in the window.

All the available filtering techniques are well suited for eliminating noise in smooth patches or areas of a signal but fails when removing noise from the edges as the image gets affected during the process. More often, in the process of text extraction as well as recognition, it is vital to remove noise from image as well as to protect the edges. Since edges are very essential in the visual appearance of images. These aspects make median filtering as the most sought technique in digital image processing.

Binarization: The process of transforming grayscale images into binary images is called Binarization. It helps in identifying the objects of interest from the image by separating the foreground pixels from the background pixels. In any grayscale captured image, the value of pixels vary between 0 and 255. During binarization procedure, grey scale value is thresholded as well as transformed into black (‘1’) for foreground or white (‘0’) for background pixel. Thresholding is a technique utilized to binarize the image. In this method, the greyscale image is transformed into binary using a finite threshold value. Binarization process can be achieved through local or global thresholding [14].

In Local thresholding techniques, different threshold values are applied to distinct regions of the image. Conversely, in global thresholding methods, one value is applied o the entire image. Local thresholding methods are applicable for images having varying intensities, for example, images from

satellite cameras. On the other hand, global thresholding is applicable for simple images.

Segmentation: Through segmentation the individual characters are isolated from the prehistoric text. It is performed by employing projection profile analysis as well as connected component labeling. The below stages are included in segmentation [15]:

- Line Segmentation
- Word Segmentation
- Character Segmentation

Segmentation breaks single text line from scanned documents, single word from single line as well as single character from the single word. There are two major classes in segmentation:

a) External segmentation is carried out to isolate paragraphs, single lines or words.
b) Internal segmentation is carried out to isolate single characters.

A number of methods are obtainable to segment individual characters derived from projection profiles, connected component labeling or white space and pitch.

Features Extraction: The performance of a character recognitions system is evaluated depending on the attributes used in the process. The possibility of achieving high recognition lies in selecting the most suitable feature selection method. Unique identification of character set must be possible with the extracted features and also there should be large variations among the features that belong to distinct character set.

Shape Transform Recognition: The approach simply has its basis in the concept that the position of each individual pixel is a attribute and it is not able to be discarded. The Shape Transformation protocol is somewhat similar to the dynamic programming method, utilized for lexicographical correction which is a postprocessing stage in OCR. The binary character image x may be converted into image y through [16]:

A) Abandoning certain pixels that are undamaged (replacement)
b) Shifting a few of the pixels of x into positions which relate to their corresponding nearest pixels in y as well as
c) Eliminating few pixels if required.

Insertion, deletion or substitution cost matrices is absent due to differences in context. So it is essential to invent our own. There is lack of cost attribute to substitutions, as it is the same to retaining them in the place as they are common for both characters. The costs involved in shifting a pixel toward a different position is established equal to Euclidean distance among source as well as destination locations. As every pixel in x is either replaced or translated, the respective pixels in x as well as y are related. The related pixels are not reused.

Hough transform: Hough transform are used for feature extraction in image processing. The statistical method has gained importance in the recent years in the field of classification and is very essential in OCR. Further ,by combining the statistical method and Hough transform there has been good results for font recognition and facial recognition[17]. The Hough Transform was originally used in the domain of computer vision to identify lines in images. Later it was used for various other geometrical attributes like circle as well as ellipse. In the early eighties, it was enhanced such that it is suitable for detecting standard shapes that yields GHT.

The fundamental concept of HT is to map edge point from image to variable space; this denotes every cases of potential attributes existing in image. Every edge points polls for the instance to which it is a part of. Consequently, the instance with maximal polling specifies the feature available in the image.

The reason for HT being successful is given by its global aspect. There is no compulsion to acquire a basic knowledge on point distributions however, the voting procedure of all points lead to rise of peak in accumulators. Voting procedure ensures robustness to HT towards missing edge point. All points that is individually taken is unimportant however every point polls for a specific shape.

A series of image points (x, y) that rest upon a straight line may be stated through a relationship, f, as in equation (1) such that:

$$f((\hat{m},\hat{c}),(x,y)) = y - \hat{m}x - \hat{c} = 0 \quad (1)$$

wherein m as well as c are 2 variables, slope as well as intercept, that characterizes the line. Eq (1) map all values of variable combinations $(\hat{m},\hat{c})$ to a series of image points.

Neural Network Classifier: An ANN is an information processing paradigm. It is based on the biological nervous systems, i.e. the ways in which the brain processes the information. The most important component of the domain is the new configuration of the information processing systems.

This system is made of numerous extremely inter-connected processing components called neurons, which works together for solving particular problems. The ANN is analogous to human as they also realize by sample. ANN can be formulated for certain applications namely for patterns recognition, data classifications, via a learning procedure [18].

In Neural Network, every individual node performs certain computation while each connection communicates a signal from one node to another labeled by a number known as the

"connection strength" or weight signifying the extent to which signal is magnified or weakened by the connection. The network examines the distinct function that results due to distinct selection of weight. If for a given network, there are randomly values for weights and if the task to be achieved is known, then a learning protocol is required to find the value of weights that will end up in achieving the task.

A computing system is called Artificial Neural Network depending upon the Learning Algorithm used. The function of a node is predetermined to apply a particular function on inputs such that a basic limitation is imposed on the abilities of the network.

1. The jth neuron in hidden layer that computes incoming data (x_i) by: (i) measuring weighted sum as well as appending a "bias" term θ_j as per Eq. (2):

$$net_j = \sum_{i=1}^{m} x_i \times w_{ij} + \theta_j (j = 1, 2, ..., n) \quad (2)$$

2. Altering net_i through an appropriate mathematical "transfer function", as well as

3. Moving the output to neurons in the subsequent layer. Several transfer functions are present. But, sigmoid one is generally used as in equation (3):

$$f(\mathrm{x}) = \frac{1}{1+e^{-x}} \quad (3)$$

J48 Classifier: J48 is a decision tree induction algorithm which is the enhanced version of the C4.5algorithm. This algorithm uses an information theoretic methodology to generate decision trees. The fundamental protocol for decision tree induction is a greedy protocol which builds decision tree in top-down recursive divide & conquer fashion.

The main objective is the construction of decision trees with minimal quantity of nodes which can give minimal quantity of mis-classifications on training dataset [19]. According to J48 decision trees algorithm is a predictive machine-learning framework, which can decide target values (dependent variables) of novel samples on the basis of several features values of present data. It can be applied on discrete data, continuous or categorical data.

There are two pruning methods employed in J48 namely sub tree replacement as well as raising. In sub-tree substitution, the nodes in decision trees can be substituted with leaves – in order to minimize the quantity of tests alongside a particular route. This procedure begins with the leaves of the completely grown tree as well as works backward towards the root.

Algorithm
Input D
Output T
DTBUILD (*D)
{
T=φ;
T=Build root node as well as label with splitting feature;
T=Append are to root node for every split predicate as well as label;
For every arc do
D=Database Built by applting splitting predicate to D;
If terminating condition achieved for the route for this path, then
T′=DTBUILD(D);
T=add T′ to arc;
}

k Nearest Neighbor (kNN): The k-nearest neighbor algorithm (kNN) refers to a technique to classify objects on the basis of nearest training samples in the features space. KNN is a kind of sample-based or lazy learning. In this algorithm, functions are solely estimated locally as well as every computations are deferred till classifications. KNN protocol is considered as the easiest compared to other machine learning algorithm [20]: the object is categorized on the basis of the number of votes by its neighbours. The object is allocated to the class, typical among k closest neighbours (k refers to a positive integer, generally small). If k = 1, then object is merely allocated to the class of its closest available neighbour.

In this algorithm, the input test sample is classified into members of a class based on the training samples where features are matched using Euclidian distance or any distance method. (Check the meaning). The test sample that closely matches with the training sample is considered as detected, recognized and classified class of membership.
The similarity of x as well as each neighbour document signifies the score of the group of the neighbour document. If many of the K closest neighbour documents are part of the same group, then the sum of the score of the group is the similitude score of the group with respect to test document x. By arranging the scores of the candidate groups, system designates the group with the biggest score to the test document x. The decision rule of KNN may be stated as in equation (4):

$$f(x) = \arg\max score(x, C_j) = \sum_{d_j \in KNN} sim(x, \mathrm{d}_i) y(\mathrm{d}_i, C_j) \quad (4)$$

where f(x) is the label designated to the test document x; $score(x, C_j)$ denotes the score of the candidate category, C_j with regard to x; $sim(x, \mathrm{d}_i)$ represents the similitude

between x as well as the training document d_i $y(d_i, C_j) \in \{1, 0\}$ represents the binary category value of the training document d_i in regard to C_j y=1 denotes document d_i is part of category C_j or y=o).

Naïve Bayes Classifier: A kind of statistical classifier is the Naïve Bayes classifier. This classifier has its basis in Bayesian principle, which evaluates the maximum posterior hypothesis. This classifier measures the probability of a character for a particular class and then maximizes the posterior hypothesis for that character belonging to that particular class. Naive Bayes classifier is an easy probabilistic classifier derived by application of Bayesian principle (from Bayesian statistics) with robust (naive) independence assumption. In Naïve Bayes classifier, it is assumed that the impact of a variant value on a particular class is not dependent of the values of other parameter. The NB inducer processes the conditional probability for the classes for a provided sample as well as the class with the largest posterior is chosen [21]. Considering the accurate nature of the probability model, NB may be effectively trained in supervised learning environment.

By Bayesian principle (Liu et al 2014), posterior probability $P(\omega \mid x)$ is defined as in equation (5):

$$P(\omega \mid x) = \frac{P(x \mid \omega)P(\omega)}{P(x)} \tag{5}$$

Here $P(\omega)$ is approximated by calculating the proportion of class ω in the training dataset and $P(x)$ may be discarded as it is compared with different 'x's on the same x. Thus considered $P(x \mid \omega)$. If a precisie estimate of $P(\omega \mid x)$ is performed, then the best classifier is obtained theoretically from the provided training dataset, i.e., bayes optimum classifier with Bayes error rate, minimal error rate theoretically. But, approximating $P(x \mid \omega)$ is indirect, as it includes the approximation of exponential numbers of joint-probabilities of the attributes. For making the estimations tractable, certain presumptions are essential. NB classifier presumes that provided the class label, n attributes do not depend on one another in the class. Hence, equation (6) shows:

$$P(x \mid \omega) = \prod_{i=1}^{n} P(x_i \mid \omega) \tag{6}$$

Signifies the requirement to approximate every attribute value in all classes for estimating the condition probabilities, so the computation of joint probability can be discarded. In the training phase, NB classifier approximates the probabilities $P(\omega)$ for every class $\omega \in \omega$ while $P(x_i \mid \omega)$ for every feature $i = 1, 2, ..., n$ as well as every features value x_i from the training dataset. In the testing phase, a test case would be forecasted with label ω if ω results in the biggest value of every class label as in equation (7):

$$P(\omega \mid x) \propto P(\omega) \prod_{i=1}^{n} P(x_i \mid \omega) \tag{7}$$

Concatenation of Features: The objective of Concatenation operations is to compile novel units of vocabulary, like words, from a small pool of basic instances, like characters [22]. Concatenation operations are executed either by connecting or not the aligned units. Fusion is performed at the feature level to select features and as well feature as combined to eliminate redundant and irrelevant features. Based on feature fusion reports given by a few of the researches, it is evident that feature fusion results in dimensionality problems due to large dimensions of the fused feature vectors.

In this technique, training instances for a particular term are created from a character attribute list that accumulates the sample feature that occurs for every character. If an input term W has series of characters C_1, C_2, C_3,...,C_m, wherein m signifies the overall quantity of characters that make up the term. Then, example attributes of the term are produced by combining the sample features of each character thereby yielding w sample features for the word computed as:

$$w = \prod_{i=1}^{m} n(C_i)$$

Wherein $n(C_i)$ signifies the total quantity of samples for character C_i.

Experiment Results

Experimental setup: We use 9 ancient characters with each character containing 35 samples each for the experiments.

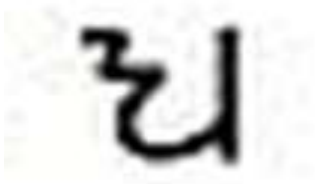

Figure 3: **Sample image 1**

Figure 4: **Sample image 2**

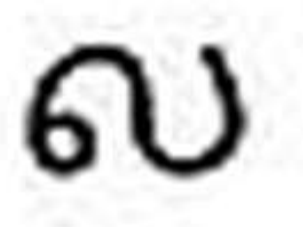

Figure 5: **Sample image 3**

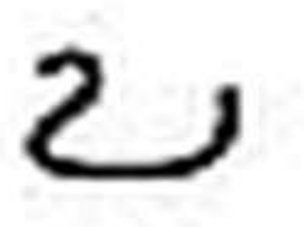

Figure 6: **Sample image 4**

Table 1
Classification Accuracy

Classification Accuracy	J48	KNN	NN
Shape Features	88.57	86.03	90.48
Hough Features	89.84	87.94	91.75
Concatenation of features	91.11	88.89	93.02

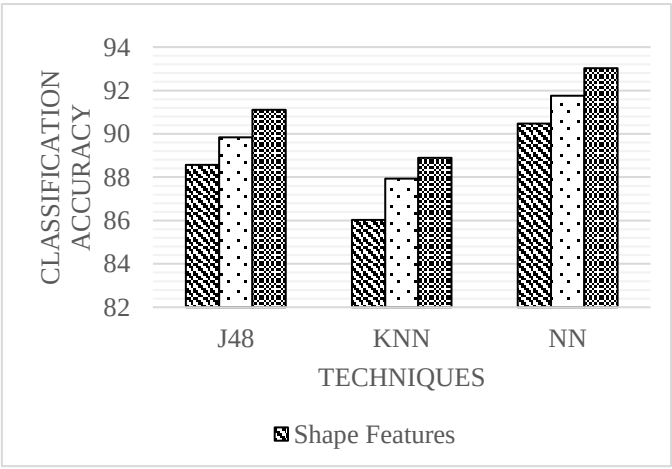

Figure 7: **Classification Accuracy**

Table 2
Precision

Precision	J48	KNN	NN
Shape Features	0.885722	0.860322	0.904767
Hough Features	0.898411	0.879378	0.917478
Concatenation of features	0.911122	0.8889	0.930189

From the figure 7, it can be observed that the neural network classifier improved accuracy than the other methods. Neural network classifier improved accuracy for concatenated features by 4.54% than KNN and 2.07% than J48 classifier

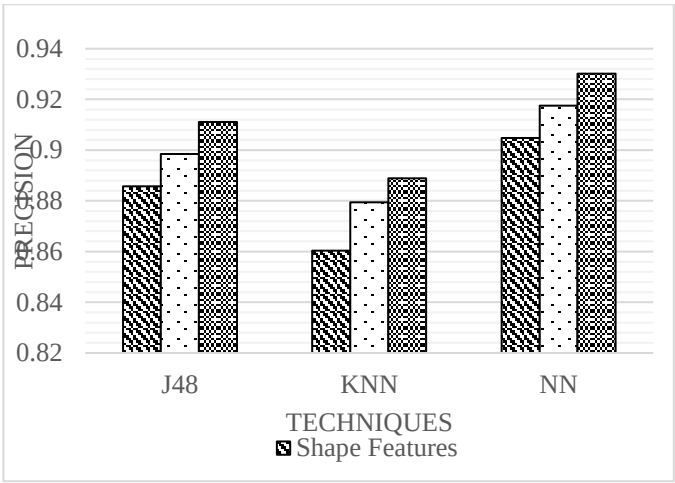

Figure 8: **Precision**

From the figure 8, it can be observed that the neural network classifier improved precision for concatenated features by 4.54% than KNN and 2.07% than J48 classifier.

Table 3
Recall

Recall	J48	KNN	NN
Shape Features	0.8881	0.861667	0.906033
Hough Features	0.899678	0.879767	0.918089
Concatenation of features	0.9116	0.889822	0.930933

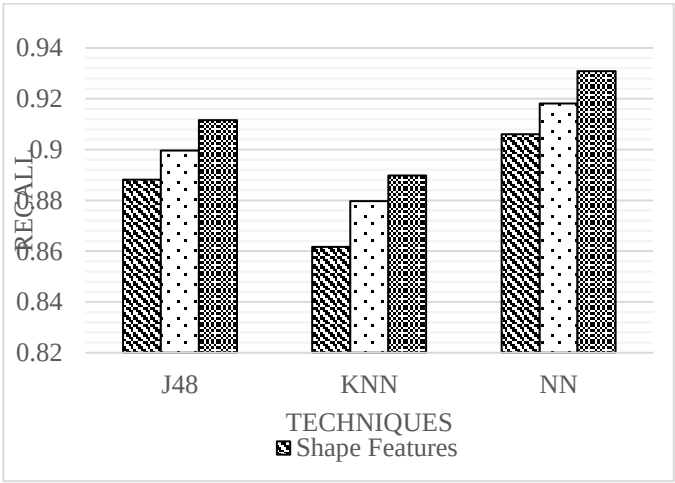

Figure 9: **Recall**

From the figure 9, it can be observed that the neural network classifier improved recall for concatenated features by 4.52% than KNN and 2.09% than J48 classifier.

Table 4
F Measure

F measure	J48	KNN	NN
Shape Features	0.8857	0.8600 89	0.905
Hough Features	0.8986 44	0.879	0.9174 89
Concatenation of features	0.9111 89	0.8887 11	0.9303

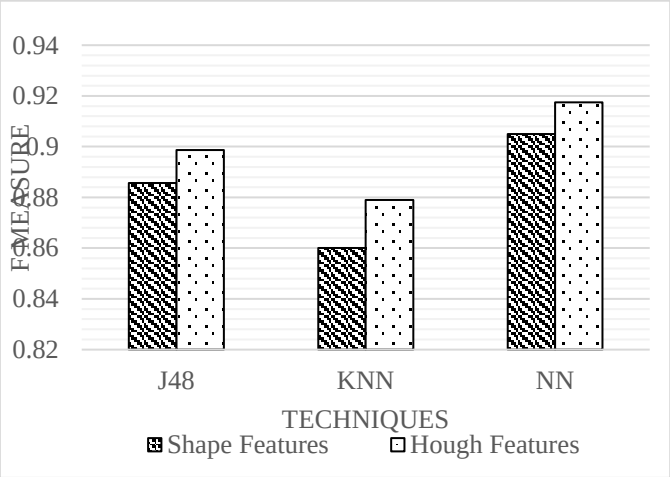

Figure 10: **F Measure**

It can be observed from figure 10, that the neural network classifier improved f measure for concatenated features by 4.57% than KNN and 2.09% than J48 classifier.

Conclusion

This paper presents feature extraction and classification schemes for optical character recognition of Tamil script. Median filter is used for eliminating noise and segmented each individual character for feature extraction using shape and Hough transformation. The suggested work uses three classifiers like K-Nearest Neighbor (KNN), NN, J48 classifier. To validate the result, the outputs are compared with other methods and the improvements are noted. A maximum recognition accuracy of 93.02% can be achieved for concatenation of features. From the results it is evident that the neural network classifier is improved than other classifiers. The work can be expanded to maximize the recognition accuracy by appending extra relevant features.

References

1. Preethi P. and Mamatha H.R., A Review on Automation of Ancient Epigraphical Images, *International Journal of Database Theory and Application*, **9(4)**, 143-150 **(2016)**

2. Iniyan E., Burial and Fuunerary Culture of Ancient Tamils (During 1000 BC-250/300 AD), *International Journal of Social Science and Humanity*, **5(12)**, 1068 **(2015)**

3. Patel P.G., Constructing a Framework for theories of the Brahmi Writing System

4. Pushparatnam P., Tamil Brahmi Inscription Belonging to 2200 years ago, Discovered by German Archaeological Team in Southern Sri Lanka, Jaffna University International Research Conference **(2014)**

5. Rajakumar S. and Bharathi V.S., 12th Century Ancient Tamil Character Recognition from Temple Wall Inscriptions. *i-Manager's Journal on Embedded Systems*, **1(2)**, 27 **(2012)**

6. Rajakumar S. and Bharathi V.S., Century Identification and Recognition of Ancient Tamil Character Recognition, *International Journal of Computer Applications*, doi: 10.5120/3090-4237 **(2011)**

7. Mahalakshmi M. and Sharavanan M., Ancient Tamil script and recognition and translation using LabVIEW. In Communications and Signal Processing (ICCSP), 2013 International Conference, IEEE, 1021-1026 **(2013)**

8. Rajakumar S. and Bharathi V.S., Eighth century Tamil consonants recognition from stone inscriptions, In Recent Trends In Information Technology (ICRTIT), 2012 International Conference, IEEE, 40-43 **(2012**)

9. RajaKumar S. and Bharathi V.S., Similar Shaped Ancient Tamil Character Recognition using Gradient Feature and Template Matching, *Digital Image Processing*, **4(6)**, 337-340 **(2012)**

10. Rajakumar S. and Bharathi V.S., Offline Ancient Tamil Character Recognition System Based On Structural Features, *i-Manager's Journal on Communication Engineering and Systems*, **1(3)**, 17 **(2012)**

11. Kumar S.V.K. and Poornima T.V., An Efficient Period Prediction System for Tamil Epigraphical Scripts Using Transductive Support Vector Machine, *International Journal of Advanced Research in Computer and Communication Engineering*, **3(9)**, 7999-8002 **(2014)**

12. Tomar A., Choudhary M. and Yerpude A., Ancient Indian Scripts Image Pre-Processing and Dimensionality Reduction for Feature Extraction and Classification: A Survey, *International Journal of Computer Trends and Technology (IJCTT)*, **21(2) (2015)**

13. Gaikwad M.R. and Pardeshi N.G., Text Extraction and Recognition Using Median Filter, *International Research Journal of Engineering and Technology*, **3(1)**, 717-721 **(2016)**

14. Chacko A.M.M. and Dhanya P.M., Handwritten Character Recognition, In Malayalam Scripts-A Review, National Conference on Indian Language Computing **(2014)**

15. Ardeshana M., Sharma A.K., Dipak M.A. and Tanish H.Z., Handwritten gujarati character recognition basedon discrete cosine transform, Proceedings of IRF-ieeeforum International Conference, 23-26 **(2016)**

16. Liolios N., Kavallieratou E., Fakotakis N. and Kokkinakis G., A new shape transformation approach to handwritten character recognition, In Pattern Recognition, Proceedings, 16th International Conference, IEEE, **1**, 584-587 **(2002)**

17. Kanimozhi V.M. and Muthumani I., Optical Character Recognition for English and Tamil Script, International *Journal of Computer Science and Information Technologies*, **5(2)**, 1008-1010 **(2014)**

18. Venkateswara Rao N., Srikrishna A., Raveendra Babu B. and Babu G.R.M., An efficient feature extraction and classification of handwritten digits using neural networks, *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, **1(5)**, 47-56 **(2011)**

19. Tadesse S., Feature Extraction and Classification Schemes for Enhancing Amharic Braille Recognition System (Doctoral dissertation, aau) **(2011)**

20. Ansari S. and Sutar U., Devanagari Handwritten Character Recognition using Hybrid Features Extraction and Feed Forward Neural Network Classifier (FFNN), *International Journal of Computer Applications*, **129(7)**, 22-27 **(2015)**

21. Liu Q., Lu J., Chen S. and Zhao K., Multiple Naïve Bayes Classifiers Ensemble for Traffic Incident Detection, *Mathematical Problems in Engineering*, **2014**, 1-16 **(2014)**

22. Elarian Y., Ahmad I., Awaida S., Al-Khatib W.G. and Zidouri A., An Arabic handwriting synthesis system, *Pattern Recognition*, **48(3)**, 849-861 **(2015)**.