ISSN 2278-3091

Volume 9, No.4, July – August 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse290942020.pdf

https://doi.org/10.30534/ijatcse/2020/290942020

Computational Approach for Character Identification and Retrieval of Grantha Inscription

Dhivya.S[1], Usha Devi G[2], S. Deepanjali[3], Rijo Jackson Tom[4], Sathish Kumar[5]

[1]Department of Information Technology, School of Information Technology & Engineering, VIT, Vellore

[2]Department of Information Technology, School of Information Technology & Engineering, VIT, Vellore

[3]Department of Information Technology, SRM IST,India,

[4]Department of Computer Science Engineering, CMR Institute of Technology, Bengaluru.

[5]Department of Information and Communication Engineering, Sunchon National University, Suncheon.

ABSTRACT

Computational Epigraphy plays a significant role to extract the information about the cultural heritage of various civilizations of the dynasty which includes rich political thoughts, literature, mythology, architecture, medicine, script evolution, etc. Traditionally the study of the epigraph is performed manually and hence the quantitative analysis of script is required. The proposed methodology recognizes the ancient Grantha script. Our approach, builds a K-Nearest Neighbor predictive model, replacing the data to function as a classification basis. Automatically the value of K is calculated as it varies for several data, and is optimized in terms of efficiency. The advancement in computer vision techniques has supported various types of real-time application tasks. One such task is the ability to recognize a character or normally referred to as OCR. There are also several types of algorithms that can be transposed into an OCR. This work aims at figuring out the process behind the OCR model by using K-Nearest Neighbor algorithm and a basic OCR system is developed to identify Grantha script for generating and comparing actual results. The initial step is to train the machine to recognize the ancient character by using the classification algorithm K-nearest neighbor with Laplacian of Gaussian (LoG) filter. This framework obtained the prediction rate of 91% with 800 samples per character for recognizing the ancient script

Key words: Epigraphs; Character Recognition (OCR); K-nearest Neighbor; Laplacian of Gaussian (LoG)

1. INTRODUCTION

Historical period of south India is larger than 5000years with the powerful era which was filled with rich political thoughts, literature, mythology, temples and monuments surrounded with mysterious complex geometric structures and prediction made where the main idol receives the fall of sunlight, scripts evolution, Varmaetc.ofchera, chola, Pandya dynasty.Whereby their civilization extended to south east and west Asia. South Indian civilization has influenced more than ten countries that is today, there are plenty of metaphoric collection as a proof which is shown through their arts, culture, architect, sculptors and protohistoric inscription.

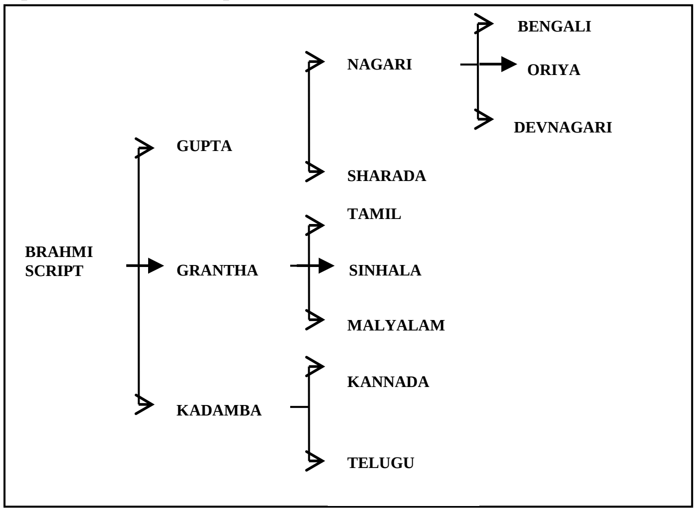


Figure 1: Evolution of ancient script

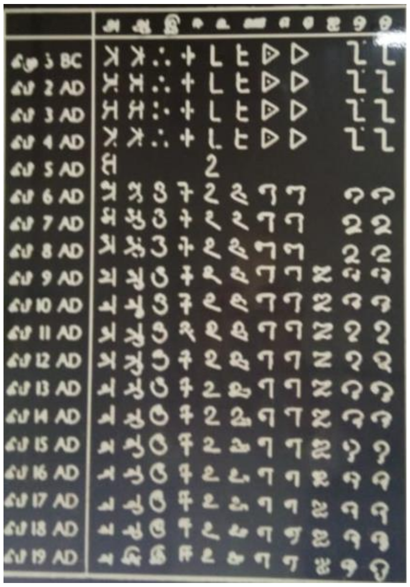


Figure 2: Palaeography of Manuscript

Evolution of Tamil literature and writing has its various versions which have been named as Tamil Brahmi, Dravidian or tamili in the period of 3rd century. Ancient Tamil script from 3rd century B.C to 5th century A. D is said as PallavaGrantha Script and it was refined to Vattezhuthu (601 A. D -800 A. D).At 9th century and 10th century A.D glory was added to Tamil language. The civilization of kings and their way of life, medicine culture has been already carried by our ancestors from the scripts of palm and temples. There are various stages of development in scripts as shown in the [Fig: 1] above and from the historical writing system to current modern writing and the scripts were distributed geographically. Brahmi script is said to be the mother of all scripts of Indic languages and various scholar's instrumental in formatting the structure of the script with some addition of character and named it by region or dynasty.

Archaeological data contains large quantity of ancient manuscripts through excavation, geological survey. The practice of writing reflects to the precise and categorical evidence of regional script. In Epigraphy the study of the ancient script is a significant task which will be the source of the quantitative linguistics particularly in the job of generating and converting to the digital format on contemporary inscription. This computed digital format possibly provide the descriptive notations for the ancient scripts which will the source of analyzing the scripts. Recognizing the contemporary script [Fig 2]is complex due to the multifaceted and compound structure. The proposed methodology is to recognize the historical script and decipher to contemporary script. The script from 11th to 13th century is given as the input and the first phase is to train the system with the character set of grantha script and the second phase is to recognize the script from the image and convert to the current script.

OCR is a method that is used to extract the data from any form of documents, the scripts that are in various formats like palm, stone and it is optically extracted as an image. The recognition of this pattern is complex which contains various curves for a single character and the style of witting overlap with curves and lines. Images which contain various paragraph, text determined through different steps to acquire the accuracy of character.

1.1 Manuscript Acquisition

Manuscript that is collected for the process is of the image format and for any interpretation or analysis of various algorithms and methodologies has been developed to extract more valuable and useful information. Scanners have been used for the conversion of digital formats with the resolutions and there are chances where the image orientation may not be properly aligned or skewed and issues with brightness etc.

1.2 Preprocessing

To increase the reliability and improve the quality of the image for further stages of processing like RGB to gray conversion, binarization, noise reduction, skew detection and correction, thinning, slant normalization, size normalization. The first phase of the methodology for character recognition is preprocessing to improve the readability of images. Preprocessing is a series of techniques applied to the source image to minimize noise and evaluating the respective phases of the character recognition technique.

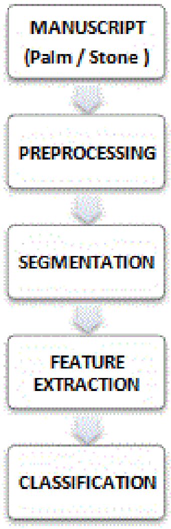


Figure 3: Basic process flow

1.3 Segmentation

The immediate step of preprocessing is segmentation where the image data is modeled into the grid structure and determine the essential parameters of the model with the entire possible optimized path by line segmentation, word segmentation, character segmentation. Various techniquesare applied for feature extraction like Stroke extraction, Loop Detection, Text Independent Features, Generalized Discriminate Analysis, Discrete Wavelet Transform (DWT).Several algorithms that are used for segmentation are Pixel counting approach, Histogram Approach, Smearing Approach, Stochastic Approach, Water Flow approach

1.4 Feature Extraction

Is a technique that is used to extract character features and those extracted features are used for machine training. The aim of extracting the feature is to quantify certain pattern characteristics most important to a particular classification function. Several features were identified and used for pattern recognition. A large scale of the feature is considered with the implementation of an identification task to optimize the recognition efficiency or maximize the output by reducing the feature count. Generally used methods of extracting features are based on geometric characteristics, structural features, spatial transformation

1.5 Classification

Classification models which have been effective or probably efficient in character recognition include Statistical methods (Nonparametric Methods, Bayes Decision Theory), Structural (string and graph matching) and ANN techniques.A classification model was built with algorithms like decision trees and Artificial Neural network but the simple and efficient learning system is KNN

2. FRAMEWORK OF PROPOSED METHODOLOGY

2.1 Overview of KNN

Researchers have used many algorithms for Classification of Text like KNN, NaïveBayes, SVM, Genetic Algorithm [17]. One of the most successful text categorization algorithms is KNN.In this paper, the model is built by Multi-class Classification and Neural Networks for prediction and logistic regression for implementing vectorization. This algorithm is a non-parametric system of classification, which is in many cases simple but effective. Its nearest neighbors are collected for a data set to be categorized, and this forms a dataset neighborhood. The data set in the neighborhood are typically used to assess classification both with and without weights concerning distance. To implement KNN, however, the appropriate K value should be selected and classification performance depends significantly on this value. The KNN procedure is always in a way biased by the value of k. Several ways are there to choose the k value, but the most common pattern is to execute the algorithm with several k values numerous times to select one of the best outputs. To allow kNN less dependent on k, Wang[2] suggested looking at various collections of nearest neighbors rather than just one set of non-nearest neighbors. In its simplest form, however, the model is generally slow, requiring O(n2) to generate a given instance, although it is less dependent on k and is capable of achieving output nearly to the best value of k. The approach builds a data model and uses the model to recognize the character. The k is automatically created in the model build process, Therefore, the list of selected members as a classification model not only decreases the number of classification data but besides greatly enhance the efficiency.

The basic working model of KNN depends on how the distance is calculated the commonly used method to calculate the distance between data.The Euclidean distance for n-dimentional space is given as:

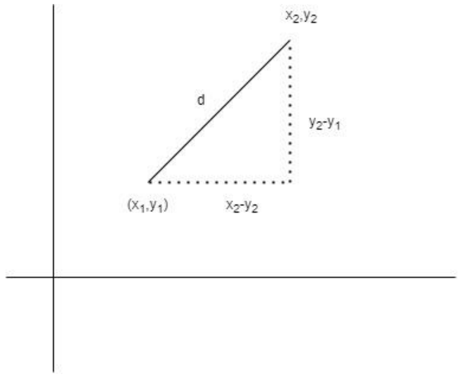


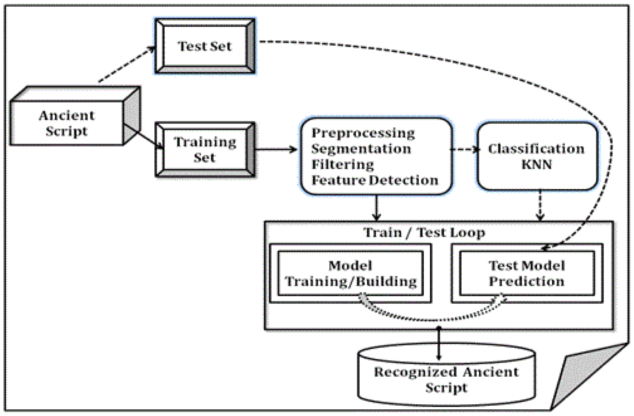
Figure.4: Illustration of Euclidean metric

$$d(x,y) = d(y,x) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + ... + (y_n - x_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

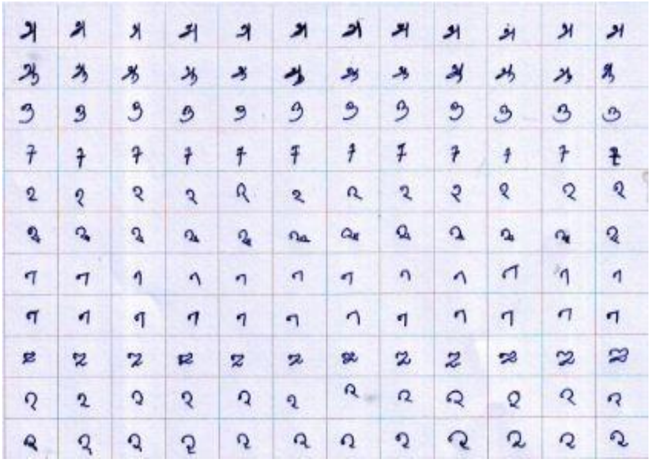
3. PROPOSED ARCHITECTURE

Split the dataset into train and test ,then distribute the train data into sets of trains and validation.KNN Classifier, the hyper parameters are the number of neighbour parameter k and the measure of distance. In this case, we use Euclidean distance as the distance measure. The parameter which is tuneable is k.



We will use the validation data set to change the optimal value of k. This newly trained model will then be used to estimate the class for all the samples in the test dataset to calculate the accuracy.

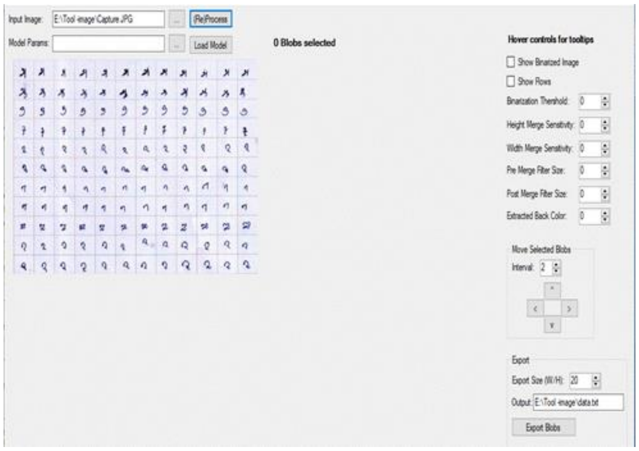
Figure.5: Architecture of the Proposed System



3.1 Preprocessing

Database of scripts are used for training and testing the system. First step to train the algorithm with a convention dataset subsequently carry out character recognition with the existing images. To denote the classes of character -give the input image [Fig 4]

Figure 6: Input - Image



Input Image: E:\Tool image\Capture.JPG

Model Params:

0 Blobs selected

Hover controls for tooltips

□ Show Binarized Image

□ Show Rows

Binarization Threshold: 0

Height Merge Sensitivity: 0

Width Merge Sensitivity: 0

Pre Merge Filter Size: 0

Post Merge Filter Size: 0

Extracted Back Color: 0

Move Selected Blobs

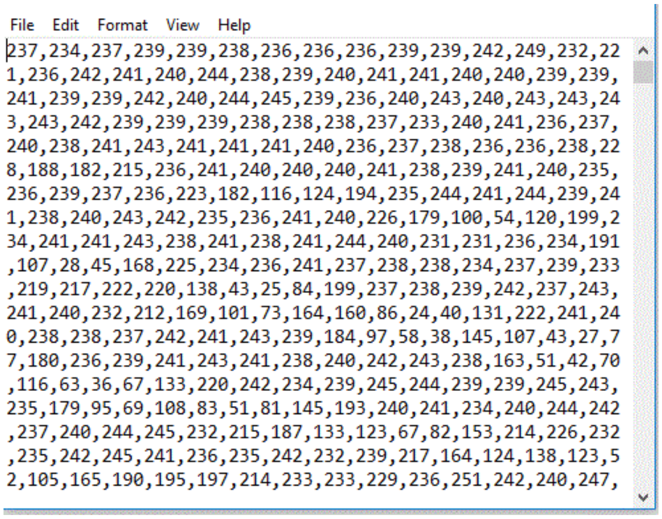
Interval: 2

Export

Export Size (W/H): 20

Output: E:\Tool image\data.txt

Figure 7: Preprocessing



```
237,234,237,239,239,238,236,236,236,239,239,242,249,232,22
1,236,242,241,240,244,238,239,240,241,241,240,240,239,239,
241,239,239,242,240,244,245,239,236,240,243,240,243,243,24
3,243,242,239,239,239,238,238,238,237,233,240,241,236,237,
240,238,241,243,241,241,241,240,236,237,238,236,236,238,22
8,188,182,215,236,241,240,240,240,241,238,239,241,240,235,
236,239,237,236,223,182,116,124,194,235,244,241,244,239,24
1,238,240,243,242,235,236,241,240,226,179,100,54,120,199,2
34,241,241,243,238,241,238,241,244,240,231,231,236,234,191
,107,28,45,168,225,234,236,241,237,238,238,234,237,239,233
,219,217,222,220,138,43,25,84,199,237,238,239,242,237,243,
241,240,232,212,169,101,73,164,160,86,24,40,131,222,241,24
0,238,238,237,242,241,243,239,184,97,58,38,145,107,43,27,7
7,180,236,239,241,243,241,238,240,242,243,238,163,51,42,70
,116,63,36,67,133,220,242,234,239,245,244,239,239,245,243,
235,179,95,69,108,83,51,81,145,193,240,241,234,240,244,242
,237,240,244,245,232,215,187,133,123,67,82,153,214,226,232
,235,242,245,241,236,235,242,232,239,217,164,124,138,123,5
2,105,165,190,195,197,214,233,233,229,236,251,242,240,247,
```

Figure 8: Feature vector is generated

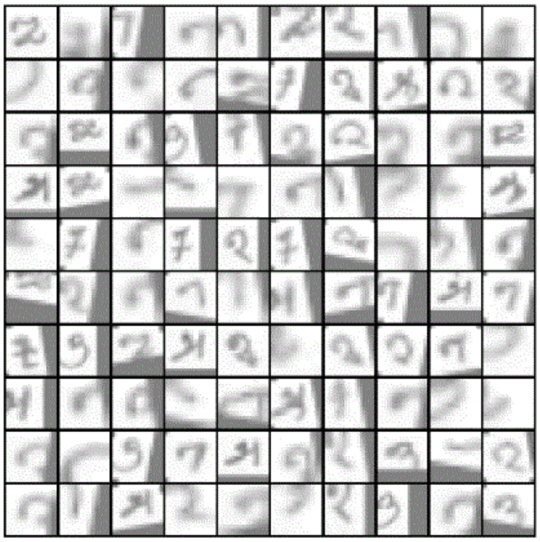


Figure 9: Feature vector is generated

Feature vector is generated by taking the average of the intensity for each character with the standard size Model for this system is produced using K-nearestneighbor's algorithm which possibly produce the optimal results with the focus on edge detection, Gaussian, Laplacefilters. After system is trained the next task is to predict the labels of the training set as shown in [Fig 7]

3.2 Filtering

Laplacian of Gaussian (LoG)

To determine the intensity of the image rapidly laplacian selects a particular region for edge detection. Most frequently laplacian is adopted for reducing the sensitivity to noise which has been taken as a first step of smoothing with the resemblance of Gaussian smoothing filter [6]. It includes the Input as a grayscale image and the generates another gray level image as an output

$$L(x,y) = \Delta^2 f(x,y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (1)$$

$$L(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 \mathrm{f(x,y)}}{\partial y^2} \ldots\ldots\ldots \qquad (2)$$

The combination of the Laplacian and Gaussian functions is used for smoothing Gaussian Filter

$$LoG(x,y) = -\frac{1}{\pi\partial^4}\left[1 - \frac{x^2+y^2}{2\partial^2}\right] e^{\frac{-x^2+y^2}{2\partial^2}} \ldots\ldots\ldots(3)$$

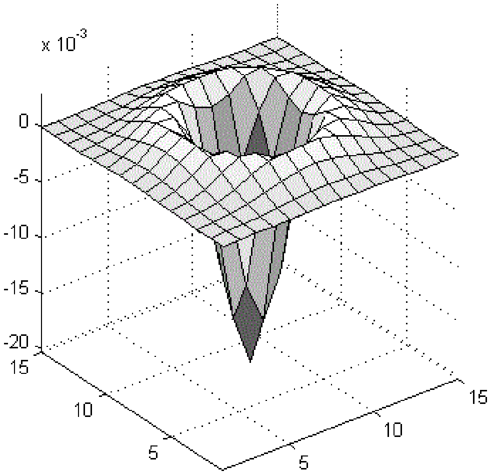
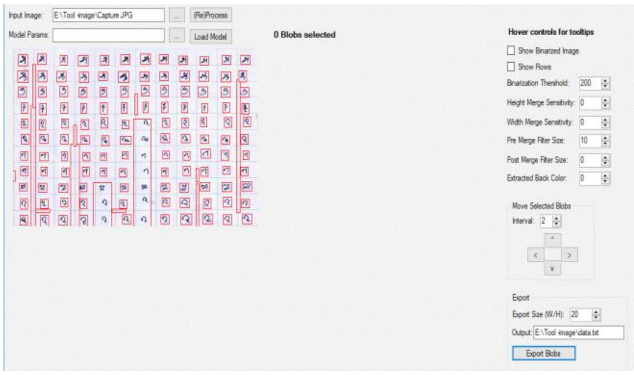


Figure 10: Laplacian of Gaussian

Laplacian and Gaussian functions determine the second derivative of the image from the equation (2). Between the regions and outcome will be zero, positive, negative and zero at the sides of the image [20]. Hence the scaling should range from negative to positive with the scaling factor to limit the range of values.

3.3 Feature Detection

Blobs create a complete portray of the image with the reference to the selected region. A center of gravity is determined as local maximum which acts as the expected point by the blob descriptors or said as the point operators as shown in the [fig 9]. The feature vector is extracted by detecting the edges of the image with the edge orientation and gradient magnitude

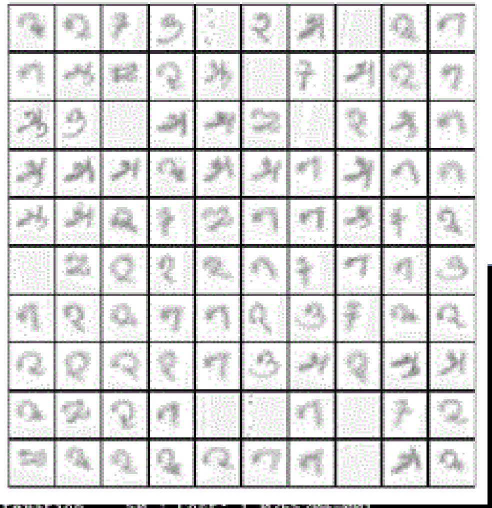


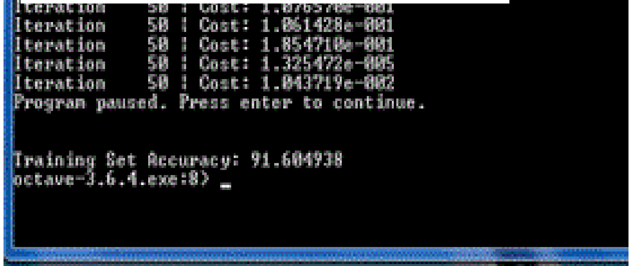
Input Image: E:\Tool -image\Capture.JPG
Model Params:
0 Blobs selected
Hover controls for tooltips
□ Show Binarized Image
□ Show Rows
Binarization Threshold: 200
Height Merge Sensitivity: 0
Width Merge Sensitivity: 0
Pre Merge Filter Size: 10
Post Merge Filter Size: 0
Extracted Back Color: 0
Move Selected Blobs
Interval: 2
Export
Export Size (W/H): 20
Output: E:\Tool -image\data.txt

Figure 11: Laplacian of Gaussian

3.4 Classification

To have the comparison of the characters and classify a database is required for which machine learning is used to train the data and we provide the code to teach the system to predict the character. When the process is completed training data is accommodated with the vectors linked to the character[18]. We retrieve the feature vector and applying the K-Nearest Algorithm we conclude which is the most appropriate character to be compared and the vector meets the threshold and return the actual character. By the implementation of the method stated above and no errors have been thrown while testing the training data while testing the we have attained with about 91% accuracy from the [Fig 13]





```
Iteration    50 | Cost: 1.076570e-001
Iteration    50 | Cost: 1.061428e-001
Iteration    50 | Cost: 1.054710e-001
Iteration    50 | Cost: 1.325472e-005
Iteration    50 | Cost: 1.043719e-002
Program paused. Press enter to continue.


Training Set Accuracy: 91.604938
octave-3.6.4.exe:8> _
```

Figure 12: Training Data

Figure 13: Training Set Accuracy

The script is recognized as the class with the largest frequency of instances close to K.Every character is taken as the prediction in effect with the votes for its class and the class with the most votes.[19]For a new data instance, class probabilities can be calculated as the normalized frequency of samples belonging to each class in the set of most similar instances K.The sigmoid function is used at the output layer to classify the 12 class with the input image size of 10x10.

```
Algorithm 1 Recognition Ancient Character
    Input : Grantha Characters Image IG[$i, j$] ; i= 1 . . . n;j=1 . . . m
    Output:Recognised Labels → $l_i[l_1 ... l_n]$ ; where n =12
procedure LOADING_DATA(IG[$i, j$])
    $sel\_array_{\mathrm{random}}$= random($IG_i$ [1 . . .n])
    rand_data_pts = $sel\_array_{\mathrm{random}}$[(1:100),n]
procedure INTIALIALIZE_NN_PARAMETERS
    Intialize:
        Input_layer_size = 400
        Hidden_layer = 25
        No_of_Labels = 12
        $\lambda = 12$
        load ( matrix_file )
procedure ONE_VS_ALL_CLASSIFIER
    for each label $li$
            Train $h_\lambda$ ($IG_i[1...n]$,$IG_j$[1 . . . m], No_of_Labels )
procedure PREDICTION(rand_data_pts,$\theta_1$,$\theta_1$)
    m = size_of( rand_data_pts,1)
    n = [ones(m,1),X]
    $X_2$= Sigmoid(n * $\theta_1$)
    Update →$X_2$ = [ones(m,1),$X_2$]
    $\theta_h$ = Sigmoid($X_2$ * $\theta_2$)
    Pred$_v alue = max(\theta_h)$
procedure SIGMOID(z)
```

$$sigval = \frac{1}{1+e^{-z}} \tag{1}$$

Since validation accuracy is greater than Training accuracy it shows that the model is performing well in classifying unseen data and the is no overfitting fig[14]

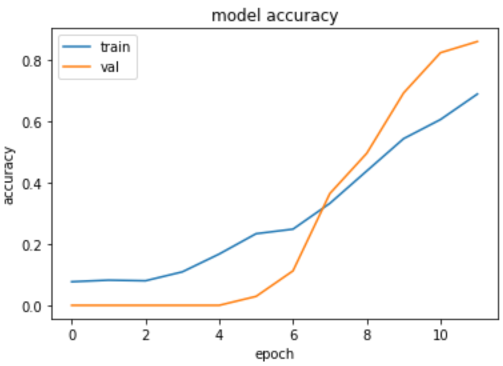


Figure 14: Grantha script - KNN Model Accuracy

4. PERFORMANCE MEASURE

The proposed methodology has been validated and result is retrieved with the accuracy of 91.6% .The below chart make obvious with performance of stated algorithm with its input script.

Table 1: Performance measure for various script

Title of the paper	**Name of the script**	**Algorithm applied**	**Prediction rate**
A : Ariyaka: A PALI alphabet Recognition Script(14)	**A^{S} :** PALI	Classification Algorithm	85.3
B : Recognition of Ancient Kannada Epigraphs using Fuzzy-Based Approach(13)	**B^{S} :** Ancient Kannada Script	Fuzzy Classifier	85%
C: Developing a commercial grade Tamil OCR for recognizing font and size independent text[12]	**C^{S} :** Tamil	Tesseract.	81 %.
D : Feature Selection For An Automated Ancient Tamil Script Classification System Using Machine Learning Techniques	**D^{S} :** Ancient Tamil Script Classification	Group Search Optimization	80%
E : Information Extraction and Text Mining of Ancient Vattezhuthu Characters in Historical Documents Using Image Zoning(11)	**E^{S} :** Ancient Vattezhuthu	Image zoning method.	89.75%.
F : Classification of Ancient Epigraphs into different periods using Random Forests[16]	**F^{S} :** Ancient Kannada	epigraph using a Random Forest (RF) Classifier	85%
G : Akkhara-Muni: An instance for classifying PALI characters[8]	**G^{S} :**Akkhara-Muni	Classification algorithms	85.66%.
H : A Combined Method of Segmentation for Connected Handwritten on Palm Leaf Manuscripts[7]	**H^{S} :** Ancient Thai language	Contour tracing algorithm	83.83%
I : Impact of Zoning on Zernike Moments for Handwritten MODI Character Recognition[5]	**I^{S} :** MODI	Zoning	82.61%

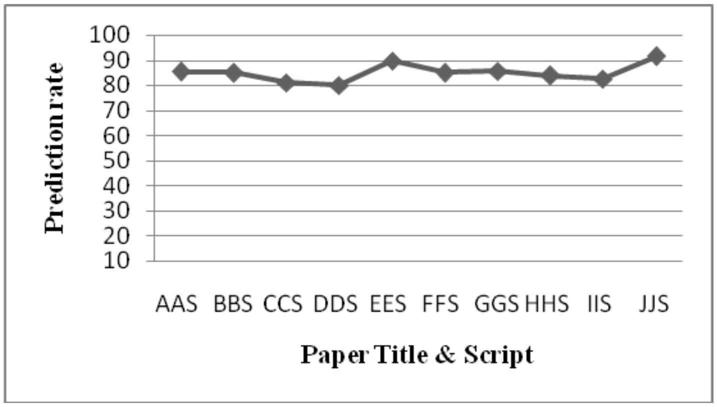


Figure 15: Perfomance measure

5. CONCLUSION

One of the widely used ancient script is Grantha which evolved during sixth to eleventh century and it was expanded to four versions named after dynasty. Grantha script slowly vanished by the emergence of Devanagari script. There are number of inscriptions accumulated by an archaeologist which has to be digitalized. This proposed methodology is to recognize the ancient script by applyingtext classification algorithm K-nearest neighbor by means of Laplacian of Gaussian (LoG) filter which can be applied by the people of Tamil development. The first phase of the research is to recognize the ancient script and the objective was achieved by the multiclass classification and neural network with 91.60% of efficiency.

REFERENCES

1. Sadanand, A. Kulkarni, L. Borde Prashant, R. Manza Ramesh, and L. Yannawar Pravin**. "Impact of zoning on Zernike moments for handwritten MODI character recognition."** In 2015 International conference on computer, communication and control (IC4), pp. 1-6. IEEE, 2015.
2. Van Phan, Truyen, Bilan Zhu, and Masaki Nakagawa. **"Collecting handwritten Nom character patterns from historical document pages."** In 2012 10th IAPR International Workshop on Document Analysis Systems, pp. 344-348. IEEE, 2012.
3. Rani, Simpel, and Gurpreet Singh Lehal. **"Recognition based classification of Gurmukhi manuscripts."**

In 2016 Symposium on Colossal Data Analysis and Networking (CDAN), pp. 1-5. IEEE, 2016.

4. Katsouros, Vassilis, Vassilis Papavassiliou, Fotini Simistira, and Basilis Gatos. **"Recognition of Greek Polytonic on Historical Degraded Texts Using HMMs."** In 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 346-351. IEEE, 2016. https://doi.org/10.1109/DAS.2016.60
5. Echi, Afef Kacem, and Abdel Belaïd**. "Impact of features and classifiers combinations on the performances of Arabic recognition systems."** In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), pp. 85-89. IEEE, 2017.
6. Kavitha, A. S., P. Shivakumara, and G. Hemantha Kumar. **"Skewness and nearest neighbour based approach for historical document classification."** In 2013 International Conference on Communication Systems and Network Technologies, pp. 602-606. IEEE, 2013.
7. Chamchong, Rapeeporn, and Chun Che Fung. **"A combined method of segmentation for connected handwritten on palm leaf manuscripts."** In 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4158-4161. IEEE, 2014.
8. Gautam, Neha, R. S. Sharma, and Garima Hazrati. **"Akkhara-Muni: An instance for classifying PALI characters."** In 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 252-253. IEEE, 2015.
9. Avadesh, Meduri, and Navneet Goyal. **"Optical character recognition for sanskrit using convolution neural networks."** In 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 447-452. IEEE, 2018.
10. Chaudhuri, B. B., and U. Pal**. "An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi)."** In Proceedings of the fourth international conference on document analysis and recognition, vol. 2, pp. 1011-1015. IEEE, 1997.
11. Vellingiriraj, E. K., M. Balamurugan, and P. Balasubramanie. **"Information extraction and text mining of Ancient Vattezhuthu characters in historical documents using image zoning."** In 2016 International Conference on Asian Language Processing (IALP), pp. 37-40. IEEE, 2016. https://doi.org/10.1109/IALP.2016.7875929
12. Liyanage, Chamila, Thilini Nadungodage, and Ruvan Weerasinghe. **"Developing a commercial grade Tamil OCR for recognizing font and size independent text."** In 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 130-134. IEEE, 2015.
13. Soumya, A., and G. Hemantha Kumar**. "Recognition of ancient Kannada Epigraphs using fuzzy-based approach.**" In 2014 International Conference on Contemporary Computing and Informatics (IC3I), pp. 657-662. IEEE, 2014.
14. Gautam, Neha, R. S. Sharma, and Hazrati Garima. **"Ariyaka: A PALI Alphabet Recognition Script."** In 2015 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 293-295. IEEE, 2015. https://doi.org/10.1109/CICN.2015.65