Exploring an efficient Handwritten Manipuri Meetei-Mayek Character Recognition using Gradient Feature Extractor and Cosine distance based Multiclass k-Nearest Neighbor Classifier

Conference Paper · December 2017

CITATIONS
5

READS
1,282

3 authors:

Kishorjit Nongmeikapam
Indian Institute of Information Technology (IIIT) Manipur
60 PUBLICATIONS **315** CITATIONS
SEE PROFILE

Wahengbam Kanan Kumar
North Eastern Regional Institute of Science and Technology
23 PUBLICATIONS **98** CITATIONS
SEE PROFILE

Mithlesh Prasad Singh
Amity University
5 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Exploring an Efficient Handwritten Manipuri Meetei-Mayek Character Recognition Using Gradient Feature Extractor and Cosine Distance Based Multiclass k-Nearest Neighbor Classifier.

Kishorjit Nongmeikapam
Dept. of CSE
IIIT Manipur, India
kishorjit@iiitmanipur.ac.in

Wahengbam Kanan Kumar
Dept. of ECE
NERIST, Nirjuli, India
wahengbam.kanankumar@gmail.com

Mithlesh Prasad Singh
Dept. of CSE
MIT, Imphal, India
mike77info@gmail.com

Abstract

In this paper, a new approach for efficiently extracting cognition out of a total of 56 different classes of handwritten Manipuri Meetei-Mayek (Indian language) is being described. Although character recognition algorithms has been researched and developed for other Indian scripts, no research work has been reported so far for recognising all the characters of the Manipuri Meetei-Mayek. The work begins with a thorough literature survey of existing works which highlighted the need of a good feature extractor as a pre-requisite for training the classifier. The limitations are experimentally removed using multiple sized cell grids using Histogram of Oriented Gradient (HOG) descriptors as feature extractor. HOG being a gradient based descriptor is very efficient in data discrimination and very stable with illumination variation. For efficient classfication of the HOG features of the Manipuri Meetei-Mayek, the robust k-Nearest Neighbor was tweaked suitably to recognize all the 56 classes of the script. The proposed approach resulted in an overall accuracy of 94.29% with a training time of about 540.81 seconds.

1 Introduction

Handwritten character recognition is increasingly gaining momentum owing to its applicable areas which can significantly reduce time But developing a more dependable approach or more technically 'a system' for recognizing handwritten characters for such regional scripts still poses a challenge to researchers. Moreover, handwritten Meetei-Mayek characters tend to be much more complex in comparison to common English characters due to the presence of modifiers, shape and structure. These factors demand a sophisticated pattern recognition algorithm that will be able to efficiently handle the challenging task of classifying these characters. In this paper, the design of an OCR system for handwritten Manipuri Meetei-Mayek is being discussed. The history and origin of Meetei-Mayek can be found in detail in the literatures by [Wanghemcha ,2007; Mangang, 2003; T.C. Hodson, 1908]. Manipuri or Meeteilon is one of the scheduled language of India and also the official language of Manipur, which is one of the state located in the North-Eastern part of India. The script contains a total of 56 characters which can classified into five different categories: Iyek Ipee/Mapung Iyek which consists of 27 alphabets, Cheitek Iyek (8 symbols), Lonsum Iyek (8 letters), Khudam Iyek (3 symbols) and Cheishing Iyek which consists of 10 numeral figures. The basic characters or the Iyek Ipee only appear as the main character of a word which may be modified by adding one of the extended symbols or Vowel modifiers to produce the required pronunciation. All the original characters of the Manipuri Meetei-Mayek alphabets are drawn, winded and wreathed based on the features of the human anatomy. Accordingly, the names of the alphabets are the names of the different parts of the human body from where they are derived [Mangang, 2003]. The Meetei-Mayek characters for which recognition are performed in the current work is shown in Fig. 1(a) along with the meaning against their names.

2 Related Works

Introduction of Manipuri Meetei Mayek OCR is in the infant stage whereas many research works have already been carried out on other Indian Scripts of different languages. Section 2.1 and 2.2 highlights the research works carried out on popular Indian languages and Manipuri Meetei-Mayek respectively.

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 328–337

2.1 Research Works on other Indian languages

Rani et al. focussed on the problem of recognition related to Gurumukhi script, they used different techniques for extracting features such as projection histogram, background directional distribution (BDD) and zone based diagonal features. These features extraction techniques were classified using SVM classifier as 5-fold cross validation with RBF (radial basis function) kernel. They achieved a very high accuracy of 99.4% using a combination of BDD and diagonal features with SVM classifier. [Rani et al., 2012]. Pal et al. proposed a system for recognizing offline Bangla handwritten compound characters using Modified Quadratic Discriminant Function (MQDF). Using a 5-fold cross validation technique they were able to obtain an accuracy of 85.90% from a dataset of Bangla compound characters containing 20,543 samples [Pal et al., 2007]. Sharma et al. proposed a scheme for unconstrained offline handwritten Devnagri numeral and character recognition using quadratic classifier based on feature obtained from chain code histogram. They were able to achieve an average accuracy of 98.86% for Devanagri numerals and 80.36% for Devanagri characters [Sharma et al., 2008]. Basu et al. presented recognition system for handwritten Bangla alphabet using a 76 element feature set which inluded 24 shadow features, 16 centroid features and 36 longest-run features. The recognition performances achieved for training and test sets were 84.46% and 75.05% respectively [Basu et al., 2005].

2.2 Research Works on Manipuri Meetei-Mayek

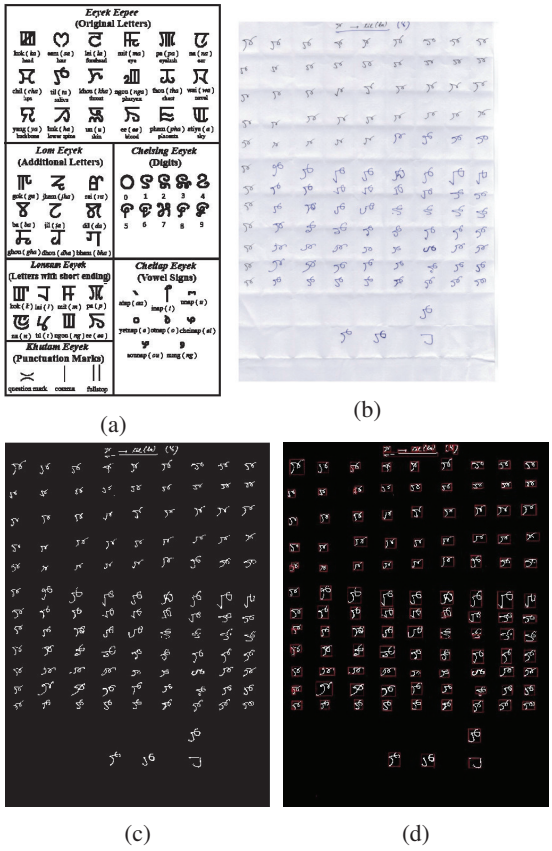
Maring and Dhir described the recognition of Meetei-Mayek numerals for both handwritten as well as printed. Gabor filter was used for feature extraction and classification was carried out using SVM. The experiment was carried out using 14x10 pixel images and overall accuracy of 89.58% and 98.45% were achieved for handwritten and printed respectively [Maring and Dhir, 2014]. Romesh et al. described the design of OCR system for handwritten text in Meitei Mayek alphabets using ANN. The database consists of 1000 samples from which 500 samples were considered as training database and the remaining samples were kept for testing and validation purpose. They observed that success of the system depended on the feature used to represent the character as well as on the segmentation stage of the test image [Romesh et al.,2014] . Chandan and Sanjib in their literature prsented a support vector machine based handwritten numeral recognition system for Manipuri script or Meetei-Mayek. They used various techniques for extracting features such as background directional distribution (BDD), zone-based diagonal, projection histograms and Histogram Oriented features which were then classified using SVM as 5-fold cross validation with RBF kernel. They were able to achieve a maximum accuracy of 95% [Chandan and Sanjib, 2013]. Romesh et al. described a way for simulating and modelling handwritten Meitei Mayek digit using backpropagation neural network approach. They were able to achieve an overall performance of 85% [Romesh et al., 2012]. Thokchom et al. proposed methods for training backpropagation network with probabilistic features, fuzzy features and combination of both features for recognising handwritten Meetei-Mayek characters. They were able to achieve an accuracy of 90.3% for the proposed 27 class classifier neural network with a combination of probabilistic and fuzzy features [Thokchom et al., 2010].

3 System Design

The motivation of this paper is to propose a robust method for classifying offline handwritten Meetei-Mayek characters. The work began with a thorough literature survey of the existing works in Manipuri Meetei-Mayek script. It was realized that so far no literature exist which can successfully or efficiently classify handwritten Meetei-Mayek alphabets and numerals, which is due to the complex nature of the script. However, previous works reported on numerals alone were quite successful as reported in section 2.2 under the heading '*Research Works on Manipuri Meetei-Mayek*'. In the present work, the HOG feature extractor is used prior to the k-NN classification process. A thorough discussion is being highlighted by considering the experimental results for selecting the suitable combination of HOG cell size and the optimal value of neighbor ('k') that can yield the maximum accuracy. By keeping the HOG feature extractor fixed, two different distance metrics that may be used with k-NN classifier is also being compared, which are Euclidean distance met-

ric and Cosine Similarity or distance metric.

To begin with, all the acquired sample images are pre-processed to remove noise as well as for extracting them individually. The pre-processing steps are discussed in section 3.1. As a first approach, based on the work by [Dalal and Triggs, 2005], section 3.2 below describes a procedure for efficiently discriminating feature sets from handwritten Manipuri Meetei-Mayek script using Histogram of Oriented Gradient (HOG) descriptors, the affects of different cell sizes on the length of the extracted features are also studied. Their feature extractor worked by dividing up an image into small spatial regions or cells, each of these regions accumulated a local 1-D edge orientations over pixels of the cell, the combined histogram entries formed the representation. In this work, multiple cell sizes for extracting HOG features have been considered in order to determine which size yielded better results for our current classification problem. The extracted feature vectors were used as training data for the k-NN classifier. Thus, we were able to obtain a significant increase in overall or average accuracy.



(a) (b) (c) (d)

Figure 1: (a) Meetei-Mayek Script (b) A Sample of the Handwritten Character 'TIL (Ta)' (c) Pre-Processed Image (d) Each elements are detected and then encapsulated prior to extraction of each one of them

3.1 Processing the Handwritten Image

In this section, the stages prior to recognition stage is being described.

3.1.1 ***Image Acquisition:*** In this stage raw data is created and collected. A total of 5600 handwritten samples were collected from people having different handwriting styles. Secondly, the image samples were scanned using a scanner and saved as jpeg file. A sample of the acquired handwritten image for the letter 'ꯇ (TIL)' is shown in fig. 1(b).

3.1.2 ***Pre-processing*** In order to make the image suitable for further processing the acquired images must be pre-processed. The term pre-processing refers to removal of any form of noise that is corrupting the useful data so that efficiency as a result of it is not decreased. For a character recognition tasks, a binary image is sufficient to work with, so the input gray image is suitably transformed using thresholding. Morphological erosion is performed so as to close the discontinuities between some letter, square shaped structural element having size equal to 2 is selected for the purpose. Morphological erosion is a simple operators in mathematical morphology which is usually performed in binary images or grayscale images. The purpose of the operation is to erode or decay the boundaries of regions of the foreground pixels (i.e. white pixels), and therefore the areas of foreground pixels shrink in size, and holes within those regions become larger. The morphologically eroded image is finally converted into a binary image[16]. Fig. 1(c) shows the final image after pre-processing.

3.1.3 ***Extracting Individual elements*** Prior to extracting each elements from the binary image so obtained in the previous step, each of them must be labelled so that automatic extraction from them is possible. For this purpose each of the elements are bounded by rectangular boxes. It can be seen from fig. 1(d) that the size of each of the boxes differ due to the fact that some character are bigger than others and vice-versa. The bounding box property for each object is an array having 4 ele-

ment which is formatted as [x, y, w, h], where (x,y) represents the row-column coordinates of the upper left corner of the box. w and h are the width and height of the box. The next step is creating a 4 column matrix that encapsulates all of these bounding box properties together, where each row denotes a single bounding box. It is necessary to define a good illustration of these bounding boxes, and thus a red box is drawn around each characters that was detected. Now, the final task is to extract all of the characters and placing them into a cell array because the character sizes are uneven, so putting this into a cell array will accommodate for the different sizes. A cell array is a type of container used for indexing data called cells, each cells may contain any type of data. Commonly they may contain combinations of text and numbers, or list of strings, or numeric arrays of varying sizes. Now simply looping over every bounding boxes that we have and then extracting the pixels within each of them will result into a character which can be placed in a cell array. Thereafter, using a loop function each of the characters in the cell array are written in to the directory for further usage.

3.2 Feature Extraction using Histogram of Oriented Gradient descriptors

Detecting features in Meetei-Mayek script is a complicated task due the similarity complex of each characters. The very first requirement is a robust feature detector which conforms to the shape or structure of the input image so that characters can be discriminated cleanly. The current study inclines on the issues of feature set extraction from Handwritten Meetei-Mayek Script using the Histogram of Oriented Gradient (HOG) descriptors. The features extracted by multiple cell-sized HOG features are used as training data for multiple classifiers, the details of which are stated in section 3.3.2. The method evaluates normalized histograms of gradient orientation of images in a dense grid. The most simple explanation being because the shapes and appearance of object can be characterized easily by using a distributing the edge detections even without exact knowledge of the corresponding edge positions. It is implemented by dividing up the image window into ”cells” which are small spatial regions. Each cell will accumulate a local 1-D histogram of gradient directions over the cell, and the combined histogram entries form the notation. It is also useful to properly equalize the contrast for improved invariance to shadowing or illumination effects before putting them to use. This feature is achieved by accumulating a measure of ”energy” of the local histogram over somewhat larger spatial ”blocks” or region and then normalizing all of the cell in the block. This is also referred to as *Histogram of Oriented Gradient (HOG)* descriptor. Then, cascading or tiling the detection window with a dense or overlapping grid of HOG descriptors, and using such combined feature vector with a kNN based window classifier will result in a chain detection [Dalal and Triggs, 2005].

Implementation: The implementation of the HOG feature descriptors for Meetei-Mayek script is based on the research work by Dalal and Triggs,2005. The detector has been tested in our Manipuri Meetei-Mayek database which roughly comprises of 56 different classes multiplied by 100 samples each. The training images comprises roughly of 56 different classes times 75 samples each. Pre-processing procedure detailed in section 3.1 is used to segment each of the character samples and finally the images were resized to 50x50 pixels. For testing, the remaining 25 samples for each of the character/class are used to validate how well the classifier performs on data that is different than the training data. Although, this is not the most representative data set, there is enough data to train and test a classifier, and show the feasibility of the approach.

The data which are used for training the classifier are the HOG feature vectors extracted from the input training images. Hence, it is important that the feature vector encodes a sufficient amount of information about the object. With the variation in cell size parameter, the amount of information encoded by each feature vectors can be observed. Each of the pixels in the image calculate a weighted vote for an edge orientation histogram channel. The weighted vote which is based on the orientation of the gradient element are accumulated into bins over local regions which is termed as cells. The orientation bins are specified as a logical scalar and they are evenly spaced from 0 degree - 180 degrees . In this case, the value of scalar less than 0 are placed into a scalar +180 degree value bin. The dark to light versus light to dark transitions contained within some areas of an

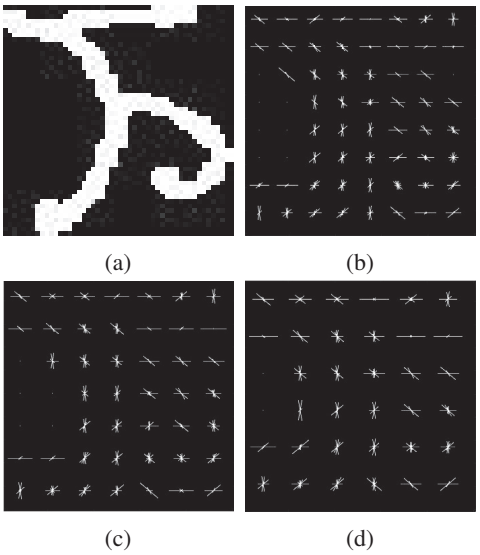


Figure 2: (a) Sample of the Pre-Processed Meetei-Mayek alphabet 'EE-LONSUM' (b) 6x6 HOG cell size (c) 7x7 HOG cell size (d) 8x8 HOG cell size.

image can be differentiated by using signed orientation. The bilinear interpolation of votes between the neighbouring bin centres can reduce aliasing for orientation as well as position. Increasing cell size can be used for capturing large-scale spatial information. It may be noted that cell size is specified as 2-element vectored form in pixels. The suppression of changes in local illumination may be reduced with increasing cell size,i.e. losing minute details as a result of averaging. Therefore, a reduction in the size of blocks will help in capturing the significance of local pixels. However, in actual practice the gradient parameters must be varied by repeatedly training and testing for identifying the optimal parameter settings. For instance, in the current work the optimal block size of HOG feature which must be maintained for efficiently recognizing Meetei-Mayek Characters is explored by considering the cell sizes viz. 6x6,7x7 and 8x8. Fig. 2 shows the features extracted using HOG descriptors for the Meetei-Mayek alphabet 'ꯎ (EE-LONSUM)'.

The extracted HOG features are returned as 1xN vector. The feature encodes local shape information from regions or from point locations within an image. Where, N is called HOG feature length and is based on the image size and the function parameter values. Let us suppose B_{image} is the number of blocks per image, C is the cell Size, N_b is the number of bins, B_o is the block overlap, B_{size} is the block size, $size_{image}$ is the size of the image. The following equations are used for appropriately deducing the the value of N.

$$N = B_{image}.B_{size}.N_b \tag{1}$$

where,

$$B_{image} = \frac{(\frac{size_{image}}{C} - B_{size})}{B_{size} - B_o} + 1 \tag{2}$$

Table 1 highlights the detected features on Manipuri Meetei-Mayek for different cell sizes. It is important to deduce the dimension of cell size that gives us the best recognition performance when combined with classifiers.

Table 1: Cell size versus HOG Feature length

Cell Size	**Length**
6x6	1764
7x7	1296
8x8	900

3.3 Classification using k-Nearest Neighbor classifier

The k-Nearest Neighbor is an example of a non-parametric type of classifier, it has been used widely as baseline classifying method in many pattern recognition applications. The input to the network consists of k nearest training samples in the feature space, while the output is a membership class. This means that, an n object is duly classified based on a vote of majority among its neighbors, the object is being classified or grouped or assigned the class which is common among its k neighbors nearest to it (it may be noted that k is a small positive integer). In case k equals to 1 then the object is assigned to the nearest neighbor. This technique is also an example of a lazy-learning or instance-based learning in which the functions are considered locally until differed during classification phase. It is also among the simplest of all machine learning tools and yet powerful. It is also quite sensitive to local distribution of data which makes it quite peculiar [Cover and Hart, 1967]

The samples used for training the network are vectors for the multi dimensional space where each of them has a class label. The training phase of the algorithm consists only of storing the HOG features which were extracted from each of the Manipuri Meetei-Mayek samples in section 3.2.

While, in the classification phase, the variable k is user defined, an enlabeled vector is also classified by specifying the class label which is the most frequent and nearest to the query point among the k training samples. The Euclidean distance is the most commonly used distance metric, the optimal value of k depends upon which types of data we are working with. Even though larger values of k has the capability to reduce noise in the classification stage, it can also make the boundaries between the different classes obscure. In multiclass classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. In the current work, for accurately classifying Handwritten Manipuri Meetei-Mayek characters, the value of k are chosen as 1,3,5,7 and 9 .

3.3.1 Distance Metric: Euclidean distance metric is the most popular and widely used similarity measure owing to its simplicity. However, the training images are not all similar necessarily in all features. Due to this limitation, in the current work, the Cosine distance metric is being investigated. A strength of it is that it can normalize all feature vectors to unit length by comparing angle between two vectors. .

Euclidean distance: Euclidean distance computes the ordinary straight line distance between any two points under consideration in the feature space or the Euclidean space. The Euclidean distance between points p and q is the length of the line segment joining them. In the cartesian coordinate system, if $p = (p_1, p_2, ...p_n$ and $q = (q_1, q_2, ...q_n$ are two points in Euclidean n-space, then the distance (d) from p to q, or from q to p is given by the Pythagorean formula:

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (3)$$

Cosine Distance or similarity It is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of $0\deg$ is 1, and it is less than 1 for any other angle. It is thus a judgement of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors at $90\deg$ have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where outcome is nearly bounded in [0,1]. The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$a.b = ||a||_2||b||_2 cos\theta \quad (4)$$

Given two vectors of attribute A and B, the cosine similarity $cos\theta$ is represented using a dot product and magnitude as

$$cos\theta = \frac{A.B}{||A||_2||B||_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (5)$$

where,A_i and B_i are components of vector A and B respectively.

The resulting similarity ranges from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality (decorrelation), and in-between values indicating intermediate similarity or dissimilarity [Manning, 2008].

Table 2: Accuracies for different pairs of HOG sizes with k (neighbors) for the Euclidean metric.

HOG Size	k	Accu. (%)	Training time (s)
6x6	1	88.2	1082.24
6x6	3	92.71	1179.24
6x6	5	89.36	1121.22
6x6	7	91.65	1042.87
6x6	9	90.21	1002.33
7x7	1	91.43	703.34
7x7	3	**94.14**	540.81
7x7	5	91.71	619.62
7x7	7	89.21	767.7
7x7	9	92.07	646.78
8x8	1	92.64	209.86
8x8	3	92.93	371.63
8x8	5	92.36	473.03
8x8	7	91.07	247.33
8x8	9	90.79	206.57

4 Experimental Results and Evaluation

The current section describes the experimental results of the handwritten Manipuri Meetei-Mayek character recognition operation using Multiple HOG feature vector with Multiclass k-NN classifier as described in section 3. The use of Cosine distance Metric based kNN classifier returned a fully trained multiclass, error-correcting output codes (ECOC) model using the training features or HOG descriptors and the class labels in the HOG

Table 3: Accuracies for different pairs of HOG sizes with k (neighbors) for the Cosine metric.

HOG Size	K	Accu. (%)	Training time (s)
6x6	1	93	1011.64
6x6	3	93.93	1141.69
6x6	5	94.07	1108.44
6x6	7	94	1261.14
6x6	9	94.07	967.43
7x7	1	93.29	468.62
7x7	3	**94.29**	502.5
7x7	5	91.14	503.56
7x7	7	93.79	543.37
7x7	9	92.43	527.4
8x8	1	93.07	205.49
8x8	3	93.71	203.03
8x8	5	93.21	231.57
8x8	7	92.71	194.03
8x8	9	92.79	340.98

feature. The *One-versus-one coding scheme* was employed. In this scheme, for each binary learner, one class is positive, another is negative and the software ignores the rest. This design exhaust all combinations of class pair assignments. The number of Binary learners is *K(K-1)/2*, where k is the number of unique class of labels [Escalera et al., 2009], [Escalera et al., 2010]. In the current study, a handwritten character Recognition for Meetei-Mayek Script based on HOG feature descriptors and trained by Cosine distance metric based kNN is successfully implemented. Three different types of HOG Cell Sizes have been considered which were examined for accuracy by training the classifier individually, i.e. 6x6,7x7 and 8x8. For each of the cell size, five different values of k are considered, which are 1,3,5,7 and 9.

In other words, the study pattern is broken up into two areas: firstly, HOG feature descriptors is used with Euclidean distance based kNN, and secondly, HOG feature descriptors is used with Cosine distance based kNN. For each of these two cases, fifteen different combinations each is being used for determining the best combination of k and HOG cell size that yields the best result. Table 2 and 3 shows the different combinations proposed herein. The time taken to train each of the different combinations of classifiers are highlighted in each area. In short, thirty different combinations or classifiers were recorded in our current work. Testing of the 30 different combinations or classifiers were performed and recorded in six different tables, i.e. two times each for 6x6,7x7 and 8x8 HOG cell sizes for Euclidean and Cosine distance metrics. However, owing to the immense size of the tables or the confusion matrices which were recorded for the current work, only the 7x7 HOG cell size with Cosine Distance based kNN success percentage for each of the 56 different classes of the script are shown in table 4 and 5 in the current paper. Some of the characters like '(PA)', '(KHOU)', '(WAI)' in table 4 have very low accuracy in comparison to other characters. For the '(PA)' character the accuracy increased significantly from 68% to 80% which is promising. However, the worst recognition rate is achieved in case of '' 'KHOU' in which the accuracy starts from just 20% and ends at a maximum of 24%. While most of the characters need to be worked on for better efficiency, some others characters like ' 'LAI' ,' 'THOU' and ' 'WAI' also needs an increase in accuracy. Despite the low accuracy readings mentioned above, there are also twenty four (24) cases out of fifty six (56) where the 100% accuracy hold all throughout the different cell sizes viz. - '(0)', '(4)', '(7)', '(8)', '(9)', '(KOK)', '(TIL)', '(NGOU)', '(YANG)', '(PHAM)', '(GOK)', '(RAAI)' , '(BHAM)', '(MIT-LONSUM)', '(PA-LONSUM)', '(NA-LONSUM)', '(TIL-LONSUM)', '(EE-LONSUM)', '(ATAP)', '(INAP), '(YET-NAP), '(OTNAP)', '(NUNG)', '(QUESTION MARK)', '(COMMA)', and '(FULL-STOP). The overall accuracy achieved by all the 30 different combinations shown in table 2 and 3 highlights a maximum accuracy of 94.29% when k=3 and HOG feature size = 7x7. The time taken to train this particular classifier was 502.5 seconds.

5 Conclusion

In this work, a novel approach for efficiently recognising Handwritten Manipuri Meetei-Mayek Characters is presented by means of comparison between the Euclidean distance Metric based kNN and Cosine distance Metric based kNN for multiple HOG feature descriptors. About 5600 handwritten samples of the 56 different classes of the Manipuri Meetei-Mayek were collected from a group of different people. The samples were then pre-processed to remove the noise in and around the letters followed by extraction of each letters

from the group. The maximum accuracy that we were able to achieve was 94.29% with a training time of just 502.5 seconds by using the Cosine similarity based kNN classification. Thus, we were able to achieve a 0.15% increase in the average recognition rate, along with 38.31 seconds decrease in training time in comparison to the commonly used Euclidean distance based kNN classifier for Manipuri Meetei-Mayek classification.

Therefore, it can be stated that the complex Meeitei-Mayek characters can be efficiently recognised by using the a combination of 7x7 cell-sized HOG descriptors with multiclass Three Nearest Neighbor (3NN) classifier. .

References

Andrew Blais and David Mertz. *An introduction to Neural Networks Pattern Learning with Backpropagation Algorithm*. Gnosis Software, Inc., July 2001.

Anita Rani, Rajneesh Rani and Renu Dhir. *Combination of Different Feature Sets and SVM Classifier for Handwritten Gurumukhi Numeral Recognition*. International Journal of Computer Applications (0975-8887) Vol. 47, No. 18, June 2012.

Cover TM and Hart PE. *Nearest neighbor pattern classification*. IEEE Trans Inf Theory 13(1):2127, 1967.

Chandan Jyoti Kumar and Sanjib Kumar Kalita. Recognition of handwritten Numerals of Manipuri Script. International Journal of Computer Applications (0975-8887), vol. 84, No. 17, Dec. 2013.

Christianini, N., and J. C. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press, 2000.

Escalera, S., O. Pujol, and P. Radeva. *Separability of ternary codes for sparse designs of error-correcting output codes*. Pattern Recog. Lett., Vol. 30, Issue 3, 2009, pp. 285297.

Escalera, S., O. Pujol, and P. Radeva. *On the decoding process in ternary error-correcting output codes.* IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 32, Issue 7, 2010, pp. 120134.

Manning CD, Raghavan P, and Schutze H. *An introduction to information retrieval*. Cambridge University Press, Cambridge, 2008.

Maring Kansham Angphun and Renu Dhir. *Recognition of Chesing Iyek/Eeyek-Manipuri Digits using Support Vector Machines*. IJCSIT, vol. 1, Issue 2, April 2014.

N. Dalal and B. Triggs. *Histograms of Oriented Gradients for Human Detection*. In Proc. IEEE Computer Vision and Pattern Recognition, pp. 1-8, 2005.

Ng. Kangjia Mangang. *Revival of a closed account, a brief history of kanglei script and the birth of phoon (zero) in the world of arithmetic and astrology*. Sanamahi Laining Amasung Punshiron Khupham (Salai Punshipham), Lamshang, Imphal, 2003.

N. Sharma, U. Pal,F. Kimura, and S. Pal. *Recognition of Offline handwritten Devnagri characters using quadratic classifier*. in Proc. Indian Conference of Computer Vision Graph. Image Processing, 2006, pp. 808-816.

Romesh Laishram, Pheiroijam Bebison Singh, Thokchom Suka Deba Singh, Sapam Anilkumar, and Angom Umakanta Singh. *A Neural Network Based Handwritten Meetei Mayek Alphabet Optical Character Recognition System*. IEEE International Conference on Computational Intelligence and Computing Research, 2014.

Renato Kresch, and David Malah. *Skeleton-Based Morphological Coding of Binary Images*. IEEE Transaction on image processing, Vol. 7, No. 10, October 1998.

Romesh Laishram, Angom Umakanta Singh, N. Chandrakumar Singh, A. Suresh Singh, H. James. *Simulation and Modelling of Handwritten Meitei Mayek digits using Neural Network Approach*. Proc. of the Intl. Conf. on Advances in Electronics, Electrical, Electrical and Computer Science Engineering - EEC 2012.

Subhadip Basu, Nibaran Das, Ram Sarkar, Mahantapas Kundu, Mita Nasipuri and Dipak Kumar Basu. *Handwritten Bangla alphabet recognition using MLP based classifier*. 2nd National Conference on Computer Processing of Bangla, pp. 285-291, Feb 2005,Dhaka.

Tangkeshwar Thokchom, P.K. Bansal, Renu Vig and Seema Bawa. *Recognition of Handwritten Character of Manipuri Script*. Journal of Computers, Vol. 5, No.10, Oct. 2010.

T.C.Hodson. *The Meitheis*. Low price publications, Delhi, 1908.

U. Pal, T. Wakabayashi and F. Kimura. *Handwritten Bangla Compound Character Recognition using Gradient Feature*. 10th International Conference on Information Technology,2007.

Wangkhemcha Chingtamlen. *A short history of Kangleipak (Manipur) part- II, Kangleipak Historical & Cultural Research Centre*. Sagolband Thangjam Leirak,Imphal, India, 2007.

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W, Hubbard, L.D. Jackel. *Handwritten digit recognition with a back-propagtion network*. in Advances in Neural Information Processing Systems 2 (NIPS*89), David Touretzky, Ed., Denver, CO, 1990, Morgan Kaufmann.

Table 4: Comparison of accuracy for each classes of the Manipuri Meetei-Mayek among different values of k (where k is the number of neighbors in kNN) for the Cosine based kNN classifier and HOG feature size of 7x7

Sl. No.	Class		Accuracy (%) 7x7 HOG Cell Size				
	Mayek	Notation	k = 1	k = 3	k = 5	k = 7	k = 9
Cheising (Digits)							
1	ꯐ	0	100	100	100	100	100
2	ꯑ	1	88	88	96	92	92
3	ꯒ	2	84	88	84	80	80
4	ꯓ	3	92	100	100	100	96
5	ꯔ	4	100	100	100	100	100
6	ꯕ	5	88	84	84	84	84
7	ꯖ	6	88	92	92	88	84
8	ꯗ	7	100	100	100	100	100
9	ꯘ	8	100	100	100	100	100
10	ꯙ	9	100	100	100	100	100
Eeyek Eepee (Main Alphabet)							
11	ꯀ	KOK	100	100	100	100	100
12	ꯁ	SAM	84	96	96	92	92
13	ꯂ	LAI	80	88	92	92	92
14	ꯃ	MIT	92	92	92	88	88
15	ꯄ	PA	68	80	80	80	80
16	ꯅ	NA	88	88	88	88	92
17	ꯆ	CHEEN	92	92	100	96	100
18	ꯇ	TIL	100	100	100	100	100
19	ꯈ	KHOU	24	24	24	20	20
20	ꯉ	NGOU	100	100	100	100	100
21	ꯊ	THOU	84	84	88	76	68
22	ꯋ	WAI	84	72	68	72	84
23	ꯌ	YANG	100	100	100	100	100
24	ꯍ	HUK	96	96	96	96	96
25	ꯎ	UN	96	96	96	96	100
26	ꯏ	EE	88	88	88	88	88
27	ꯐ	PHAM	100	100	100	100	100
28	ꯑ	ATIYA	100	96	96	96	96

Table 5: Comparison of accuracy for each classes of the Manipuri Meetei-Mayek among different values of k (where k is the number of neighbors in kNN) for the Cosine based kNN classifier and HOG feature size of 7x7 (Contd. from table 4).

Sl. No.	Class		Accuracy (%) 7x7 HOG Cell Size				
	Mayek	Notation	k = 1	k = 3	k = 5	k = 7	k = 9
Lom Eeyek (Addl. Alph.)							
29	ꯘ	GOK	100	100	100	100	100
30	ꯖ	JHAM	96	96	96	96	96
31	ꯐ	RAAI	100	100	100	100	100
32	ꯗ	BAA	100	96	96	96	96
33	ꯇ	JIL	96	96	96	96	96
34	ꯗꯤ	DIL	92	100	92	92	92
35	ꯍꯤ	GHOU	88	92	96	96	96
36	ꯗꯣ	DHOU	84	88	92	92	92
37	ꯝ	BHAM	96	100	100	100	100
Lonsum (Short Alph.)							
38	ꯀ꯭	KOK-LONSUM	96	96	96	92	92
39	ꯂ꯭	LAI-LONSUM	88	92	88	92	92
40	ꯃ꯭	MIT-LONSUM	100	100	100	100	100
41	ꯄ꯭	PA-LONSUM	100	100	100	100	100
42	ꯅ꯭	NA-LONSUM	100	100	100	100	100
43	ꯇ꯭	TIL-LONSUM	100	100	100	100	100
44	ꯉ꯭	NGOU-LONSUM	92	96	96	96	96
45	ꯏ꯭	EE-LONSUM	100	100	100	100	100
Cheising (Vowels)							
46	ꯔ	ATAP	100	100	100	100	100
47	ꯥ	INAP	100	100	100	100	100
48	ꯨ	UNAP	92	96	96	96	96
49	ꯣ	SOUNAP	96	96	96	96	96
50	ꯣ	YETNAP	100	100	100	100	100
51	ꯣ	OTNAP	100	100	100	100	100
52	ꯦ	CHEINAP	92	92	92	88	92
53	ꯡ	NUNG	100	100	100	100	100
Khutam (Punctuation)							
54	꯫	QUEST. MARK	100	100	100	100	100
56	꯫	COMMA	100	100	100	100	100
56	꯫	FULLSTOP	100	100	100	100	100