Chapter 2

Dataset creation and performance analysis of state-of-the-art classifiers with existing features

This chapter describes the creation of a Meitei Mayek off-line handwritten character dataset. The complete character set of the script is taken into consideration for the creation of the dataset. To the best of our knowledge, there are no publicly available datasets for the concerned script. Although a number of datasets have been developed for other scripts worldwide, new datasets can always be created to validate the performance of classification methods. This dataset sets an example of a new dataset of a completely different script. The existing computer vision methods can be validated against the developed dataset.

A performance analysis of two handcrafted feature descriptors viz. Histogram of Gradients (HOG) and Discrete Wavelet Transform (DWT) with four popular classifiers viz. Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Random Forest (RF) and Multi-Level Perceptron (MLP) is carried out later in the chapter. The performance of raw image pixel intensity (IPI) values is also tested with the four mentioned classifiers against the developed dataset.

2.1 Existing Datasets

There are some publicly available benchmark handwritten character datasets available for scripts like Latin [70, 111, 129], Chinese [120, 205, 240], Arabic [7, 8, 109, 136, 155] and Korean [95]. While discussing publicly available datasets on handwritten text recognition, it is important to mention the European tranScriptorium[181] and The Recognition and Enrichment of Archival Documents (READ))[1] projects. Several datasets on historical documents of four different languages viz. Spanish [172], English [2][3], German [4][5] and Dutch [6] are developed as part of these projects. Competitions around the tranScriptorium datasets were carried out in the International Conference on Frontiers in Handwriting Recognition (ICFHR) 2014 [182] and International Conference on Document Analysis and Recognition (ICDAR) 2015 [183] and those based on the READ datasets in ICFHR 2016 [184] and ICDAR 2017 [185].

Among the Indic scripts, benchmark datasets have been developed for Bangla, Devanagari, Gujarati, Telugu, Tamil, Gurmukhi, Kannada and Malayalam. Indian Statistical Institute, Kolkata has done one of the pioneering works of developing datasets of Indic scripts like Bangla (numerals, basic characters, vowel modifiers and compound characters), Devanagari (numerals and basic characters) and Oriya (numerals)[7]. Center for Microprocessor Application for Training Education and Research (CMATER), Computer Science and Engineering Department, Jadavpur University, Kolkata has developed datasets for numerals and characters of Devanagari, Telugu, Bangla and Urdu[8]. Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Canada also has created a Urdu handwritten character dataset consisting of numerals, special symbols, characters and words [180]. Datasets for isolated characters of Tamil, Telugu and Devanagari are developed by HP Lab India[9]. Recently, a character image dataset consisting of 85 frequently used Malayalam character classes has been created [125]. Works are also being done for development of Kannada handwritten character dataset for numerals [159] and a document dataset consisting of handwritten document pages, lines and words [11]. Recent works on benchmark dataset creation of Gurmukhi [102] and Gujarati[10] are also found in literature.

As far as Meitei Mayek is concerned, we could only find one publicly

[1]https://www.research.ed.ac.uk/en/projects/read-recognition-and-enrichment-of-archival-docume
[2]https://zenodo.org/record/44519#.Y7LBXXZBy3A
[3]https://zenodo.org/record/248733#.Y7LinXZBy3A
[4]https://stadtarchiv-archiviostorico.gemeinde.bozen.it/bohisto/de
[5]https://www.briefedition.alfred-escher.ch/home.html
[6]Utrechtuniversitylibrary,MV:C5,http://objects.library.uu.nl/reader/index.php?obj=1874-44915&lan=en
[7]https://www.isical.ac.in/~ujjwal/download/database.html
[8]https://code.google.com/archive/p/cmaterdb/
[9]http://lipitk.sourceforge.net/hpl-datasets.htm
[10]https://tdil-dc.in/index.php?option=com_download&task=showresourceDetails&toolid=971&lang=en

available dataset but is not completely free of cost. It consists of more than 5000 samples collected from individuals of age group 4-60 years by asking them to write down the characters on A4 size sheets[11]. The problem with the above mentioned dataset is that it does not capture the natural handwriting of the individuals because a) most individuals of the considered age group do not know how to write the script and b) since the samples were collected by asking them to write the characters, it does not take into account the unconstrained handwriting of individuals at different times and mental states. The dataset introduced in this chapter overcomes these shortcomings by firstly considering all the characters present in the character set and secondly considering only the masses who know how to write the script naturally and also their unconstrained handwriting. The developed dataset is made publicly available at http://agnigarh.tezu.ernet.in/~sarat/resources.html for the research community to use.

2.2 Creation of TUMMHCD dataset

The major steps followed for the creation of TUMMHCD are shown in Figure 2-1. Each of the steps is described in the following subsections:

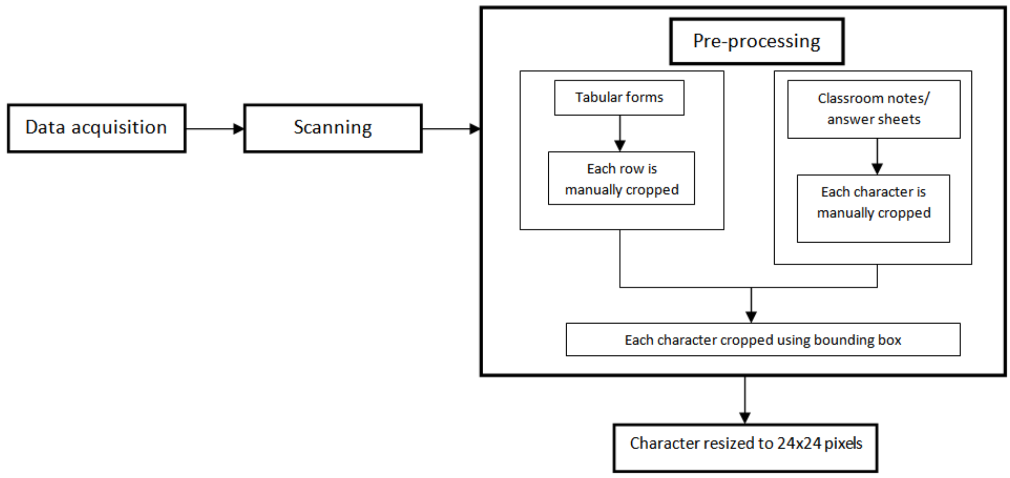


Figure 2-1: Steps followed for dataset creation

[11] https://ieee-dataport.org/documents/benchmark-dataset-manipuri-meetei-mayek-handwritten-char

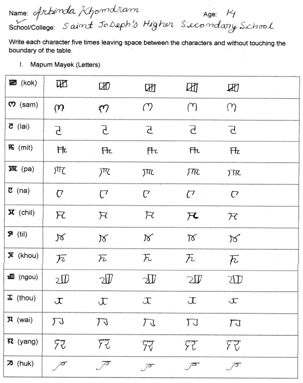
2.2.1 Data acquisition

Since Meitei Mayek was adopted recently for writing, we could not find any kind of filled out forms in any government organization or elsewhere which could be used for the dataset creation. The process of data collection is challenging also because only a small section of population can write the script in their natural handwriting, the oldest individuals being 23 years of age. Individuals older than this age group have not studied the script during their school and college days and as such they do not know how to read and write the script. Hence, the only source of getting sample data is colleges and schools. It is attempted to capture natural handwriting of the individuals as much as possible for the development of the dataset. For this purpose, data samples were collected in two phases. The first phase was handing out a tabular form to each individual where they were asked to write each character five times. Around 200 individuals were considered for this phase of data sample collection. As part of second phase of data sample collection, answer sheets were collected from 121 individuals and one page of each individual was considered for further steps of dataset creation. Classroom notes of 158 students were also collected which gave 158 pages of naturally handwritten text.

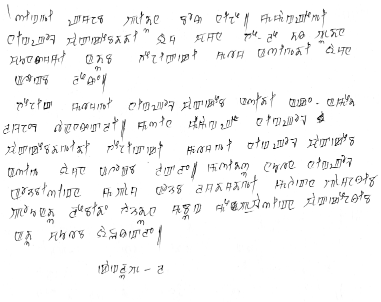
It was observed that the ten numerals and a few (four) characters viz. ꯓ, ꯚ, ꯙ and ꯘ occur for a smaller number of times in the data samples collected in the above mentioned two phases. These characters were again collected in a third phase (Refer Figure 2-2c) in the form of tabular forms from a different set of 60 individuals so that the distribution of samples over the classes is fairly equal. Therefore, a total of 539 individuals have contributed in the development of present dataset. The age group of individuals fall in the range of 12-23 years at the time of data collection (2017). One college and six different schools across varying locations of two prime districts of Manipur viz. Imphal East and Imphal West are considered for the data sample collection. Example samples of the data collected is provided in Figure 2-2. The number of writers and their age-range are presented in the pie chart shown in Figure 2-3.

2.2.2 Scanning

A canon flatbed scanner is used to scan the collected data samples at 300 dpi in grayscale format. They are saved in TIFF format.



(a) Tabular form (First phase)



(b) Answer sheet (Second phase)

(c) 10 numerals and 4 consonants (Third phase)

Figure 2-2: Samples of collected data

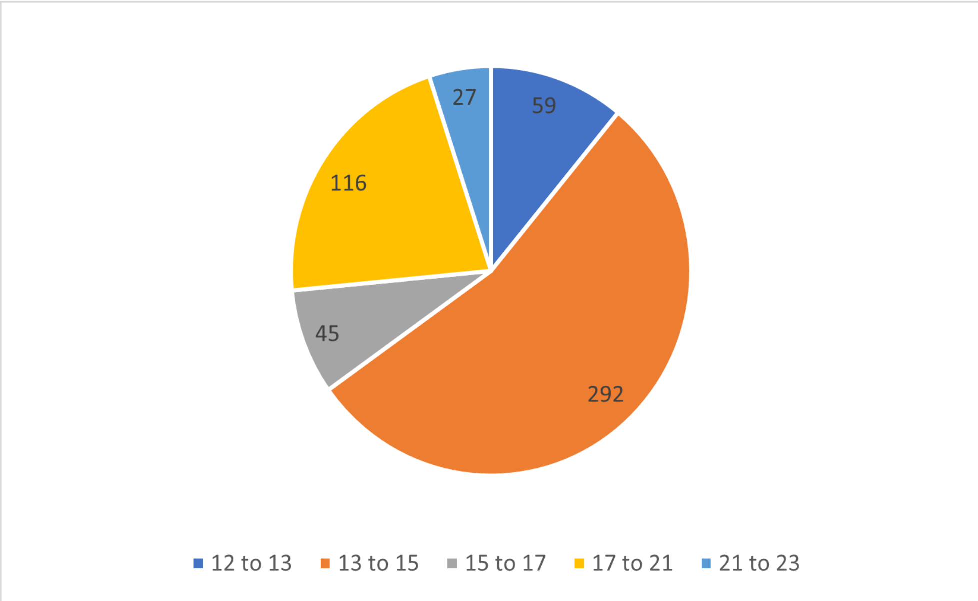


Figure 2-3: The number of writers who contributed in the data creation and their age group

2.2.3 Pre-processing

1. First, manual cropping of rows from the tabular forms is carried out. The rows are then binarized to find out the bounding box of each character. Co-ordinates of the bounding box are then used to crop the characters from the original grayscale, manually cropped rows. The steps are depicted in Figure 2-4. Undesired images whose size is smaller than height or width of 10 pixels are then discarded programmatically.

2. Further, manual scanning is done to get rid of other unwanted images which get cropped in the previous step

3. Cropping of characters in notes and answer sheets is done by manually cropping each character. Each character's bounding box is then found out using the same procedure of step 1 and cropped accordingly.

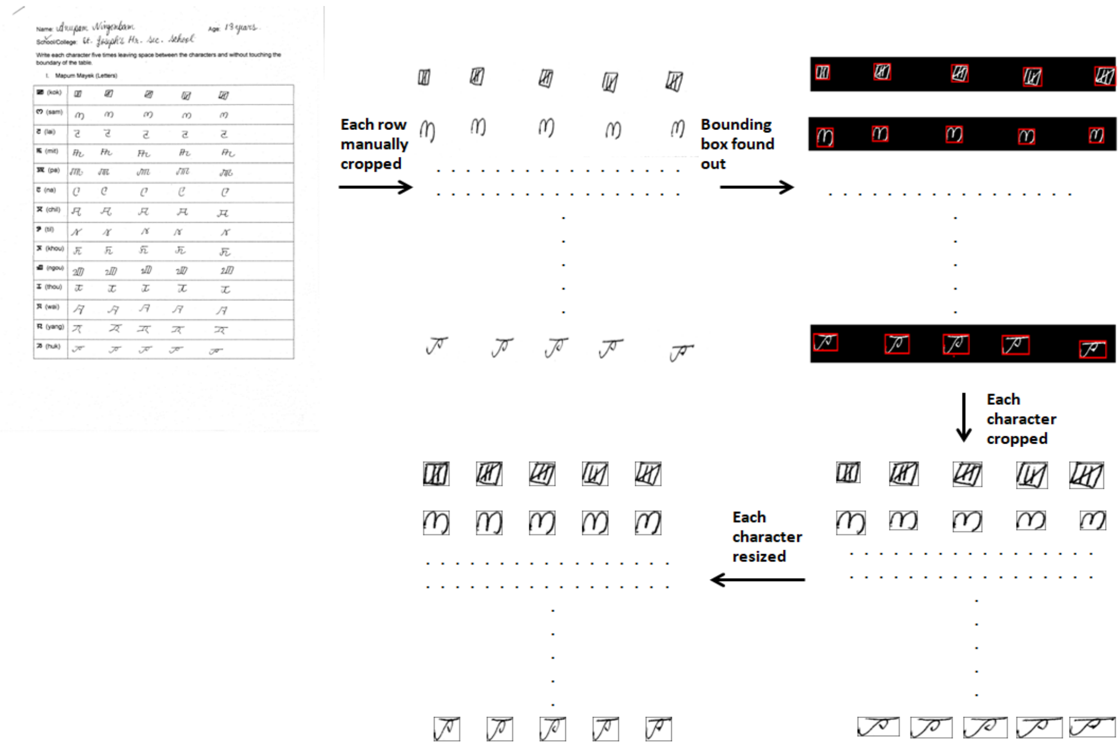


Figure 2-4: Steps for cropping out characters from tabular forms.

2.2.4 Resizing

Cropped characters from tabular forms, classroom notes and answer sheets are size normalized to fit in a box of 24x24 pixels. Resizing is done to bring the character images to a standard size. Resizing images into 28x28 and 32x32 pixels are found in HCR literature. Resizing of 24x24 pixels is considered empirically since we are taking the minimum bounding box of the characters while cropping the characters. If one wishes to use padding, then the image size can become 28x28 or 32x32 pixels .

The dataset consists of a total of 85,124 character images. In each class, images are randomly divided into training set (85%) and test set (15%). The training set and test set consist of a total of 72,330 and 12,794 images respectively. The partition of the TUMMHCD dataset into training set (85%) and test set (15%) was done in the similar manner as the MNIST dataset where the training set is 60000 (85.7%) and test set is 10000 (14.3%). Also, since the idea was to adopt CNN which requires a significantly large dataset to train, it was decided to divide the dataset in the mentioned ratio. A division of 80:20 or 90:10 can also be considered. Some random samples of the first five consonants and first five numerals are shown in Figure 2-5. The distribution of character images over 55 classes is shown in Table 2.1. The developed dataset will be referred to as TUMMHCD (Tezpur University Meitei Mayek Handwritten Character Dataset) in the later sections of the thesis.

Figure 2-5: Samples from some of the classes

2.3 Performance analysis of existing techniques on TUMMHCD

Various classification techniques have been introduced for recognition of handwritten characters over the last few decades. Classification can be carried out using IPI values or handcrafted feature descriptors and one or more classifiers. We seek to provide a performance analysis of four popular classifiers using handcrafted feature descriptors viz. HOG and DWT descriptors. The performance using IPI values is also tested with the four mentioned classifiers against the developed dataset. The four classifiers considered for our work are the top-4 performing classifiers among seven popular classifiers viz. KNN, Linear Support Vector Classifier, MLP, RF, SVM and Gaussian Naive Bayes [77]. These seven classifiers are again chosen based on the work reported in [233] where they have also selected similar set of models to develop a benchmark on their dataset.

HOG descriptor is one of the most recent and widely used feature descriptors. Other equally popular feature descriptors are SIFT [121], SURF [25] and

Table 2.1: Training, testing and total number of samples in each class

Class id	Character Name	Character Symbol	Training	Test	Total
0	Ama	꯱	1711	303	2014
1	Ani	꯲	1641	290	1931
2	Ahum	꯳	1564	277	1841
3	Mari	꯴	1592	282	1874
4	Manga	꯵	1581	280	1861
5	Taruk	꯶	1555	275	1830
6	Taret	꯷	1539	272	1811
7	Nipal	꯸	1569	277	1846
8	Mapal	꯹	1513	268	1781
9	Phun	꯰	1664	294	1958
10	Kok	ꯀ	1288	228	1516
11	Sam	ꯁ	1297	229	1526
12	Lai	ꯂ	1305	231	1536
13	Mit	ꯃ	1301	230	1531
14	Paa	ꯄ	1279	226	1505
15	Naa	ꯅ	1323	234	1557
16	Chil	ꯆ	1315	233	1548
17	Til	ꯇ	1320	234	1554
18	Khou	ꯈ	1287	228	1515
19	Ngou	ꯉ	1292	228	1515
20	Thou	ꯊ	1314	232	1546
21	Wai	ꯋ	1164	206	1370
22	Yang	ꯌ	1326	234	1560
23	Huk	ꯍ	1274	225	1499
24	Un	ꯎ	1305	231	1536
25	Ee	ꯏ	1280	226	1506
26	Pham	ꯐ	1222	216	1438
27	Atiya	ꯑ	1287	228	1515
28	Gok	ꯒ	1265	224	1489
29	Jham	ꯓ	1472	260	1732
30	Rai	ꯔ	1294	229	1523
31	Baa	ꯕ	1287	228	1515
32	Jil	ꯖ	1305	231	1536
33	Dil	ꯗ	1303	230	1533
34	Ghou	ꯘ	1438	254	1692
35	Dhou	ꯙ	1374	243	1617
36	Bham	ꯚ	1379	244	1623
37	Kok Lonsum	ꯛ	1269	225	1494
38	Lai Lonsum	ꯜ	1257	222	1479
39	Mit Lonsum	ꯝ	1284	227	1511
40	Pa Lonsum	ꯞ	1260	223	1483
41	Na Lonsum	ꯟ	1269	224	1493
42	Til Lonsum	ꯠ	1268	224	1492
43	Ngou Lonsum	ꯡ	1265	224	1489
44	Ee Lonsum	ꯢ	630	112	742
45	Aatap	◌ꯥ	1274	225	1499
46	Yetnap	◌ꯦ	1314	233	1547
47	Unap	◌ꯨ	1266	224	1489
48	Enap	◌ꯤ	1270	225	1495
49	Cheinap	◌ꯩ	1280	226	1506
50	Otnap	◌ꯣ	1268	224	1492
51	Sounap	◌ꯧ	1243	220	1463
52	Nung	◌ꯪ	892	158	1050
53	Cheikhei	꯫	595	105	700
54	Apun	꯭	1201	213	1414

ORB [176]. HOG features are easier and quicker to compute and are more widely used for image recognition tasks as compared to the other feature descriptors. Additionally, they are more accurate in capturing the object shape in an image which is desirable from image recognition point of view. For this reason, HOG descriptors have been adopted for our work. DWT, on the other hand, is mainly used to preprocess an input image. Not many works are reported in HCR using DWT descriptors. However, representation of images in the form of DWT can be a preferred choice in many image processing tasks as it provides information in both time and frequency domains of an image. Furthermore, it reduces the vector size representing the images. This introduces less training costs in the HCR systems, especially for large datasets.

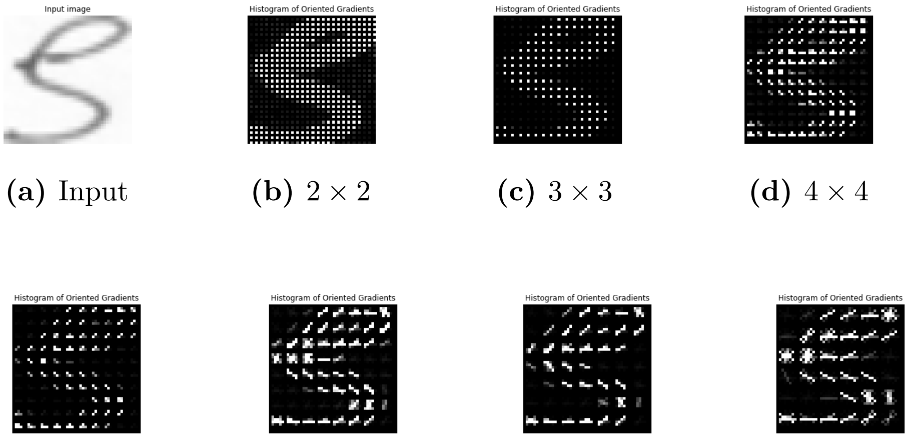
2.3.1 Recognition using handcrafted feature descriptors

Two types of handcrafted features are taken for the recognition of TUMMHCD using the above-mentioned four classifiers. They are described as follows: Firstly, HOG descriptors [42] have been employed in a number of HCR tasks [23, 31, 83, 214]. This descriptor provides histograms of directions of oriented gradients which provide useful information regarding the edges and corners present in an image. Large values of magnitude of gradients around certain portions in an image tells a lot about the object shape and texture present in the image. Secondly, feature descriptor based on DWT [47] is used. DWT is a wavelet transform in which the wavelets are sampled using a series of discrete values. Wavelet transform is the transformation of signals or functions into what is known as "wavelets". These are functions that satisfy certain mathematical requirements and are used in representing data or other functions at different scales or resolutions [68].

2.3.1.1 HOG feature descriptor extraction

For extracting HOG features, an image is resized to 48×48 pixels. Then the image is divided into a number of cells and a histogram of gradient is calculated for each cell over all the pixels in the cell forming the basic *orientation histogram* representation. Each orientation histogram is then represented using a 9-valued vector since we are using 9-bin histogram. The next step is block normalization over blocks of two cells (since we are using two cells per block) which is performed to normalize the histogram so that they are invariant to lighting variations. L2 normalization is employed for this step. The final step is to collect the HOG descriptors from all the blocks and flatten them into a single vector which is the final HOG feature descriptor for the image. For a cell size used of 6×6 pixels and block size of 2×2, this gives a final vector descriptor of length 1764.

Different cell sizes (Figure 2-6) have been considered in our work to test the accuracy on the developed dataset.



(a) Input **(b)** 2×2 **(c)** 3×3 **(d)** 4×4

(e) 5×5 **(f)** 6×6 **(g)** 7×7 **(h)** 8×8

Figure 2-6: Histogram of Oriented Gradients with different cell sizes

2.3.1.2 Two-dimensional DWT feature descriptor extraction

A wavelet is a normalized and finite waveform with zero mean and of limited duration. It can be represented as:

$$\int_{-\infty}^{\infty} \psi(t)dt \tag{2.1}$$

where $\psi(t)$ is known as mother wavelet. The mother wavelet is dilated and translated to yield child wavelets or wavelet functions. The function to be studied is processed with the child wavelets to produce wavelet coefficients [82].

In DWT, the wavelet transform is carried out by a set of discrete set of the wavelet scales and translations by following certain defined rules such that after transform, the function should be decomposed into a set of wavelets which are mutually orthogonal [170] called the basis functions.

In image processing terms, DWT is a method to transform image pixels into wavelets. And 2D-DWT can be computed by repeated application of 1D-DWT [220]. The levels of decomposition along with the coefficients obtained are shown in Figure 2-7. The coefficients at top left box at each level of decomposition which are known as approximation coefficients are used as the feature descriptors [113]. These are the lowpass filtered representation of the image found to contribute the most in classifying images and is most similar to the original image [230].

The feature descriptors are tested using SVM, KNN, Random Forest and MLP classifiers (Table 2.2, Table 2.3). The classifiers provided in the scikit-learn

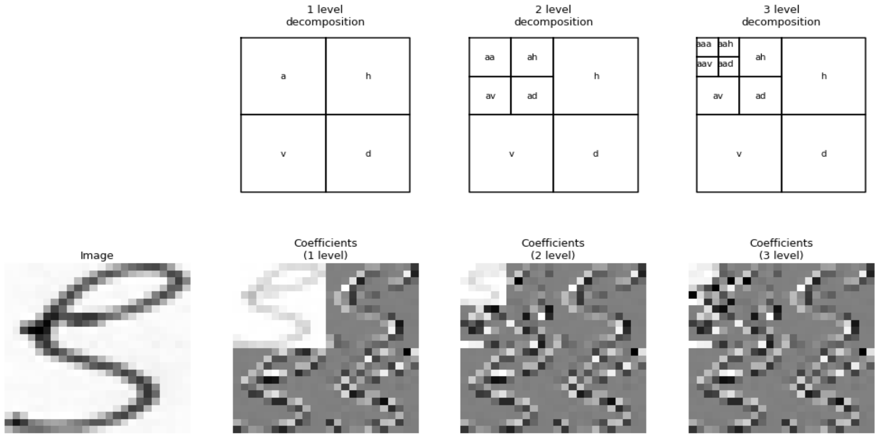


Figure 2-7: Approximation and detail coefficients of *db1* wavelet at three different decomposition levels.

Table 2.2: Recognition accuracies achieved using HOG descriptors with different cell sizes. Figures in bold signify highest test accuracy for concerned classifier

Cell-size	Feature Vector Length	SVM	KNN	RF	MLP
2x2	20736	92.42%	57.64%	80.23%	90.21%
3x3	8100	92.18%	57.32%	79.19%	90.02%
4x4	4356	93.52%	83.34%	87.77%	91.73%
5x5	2916	93.47%	84.48%	87.81%	89.98%
6x6	1764	**93.98%**	89.76%	90.32%	**92.49%**
7x7	1296	93.26%	89.36%	90.34%	89.01%
8x8	900	93.32%	**90.84%**	**91.00%**	92.33%

library [156] are used for the work. With HOG descriptor, SVM and MLP perform consistently better than other classifiers with a maximum of 93.98% (SVM), 92.49% (MLP) and minimum of 92.18% (SVM), 89.01% (MLP) recognition accuracy respectively when varying number of pixels per cell is taken. Cell-size also does not affect the performance of SVM and MLP much while that of KNN is highly affected by the number of pixels considered in a cell with the lowest test accuracy of 57.32% with cell-size 3×3 and the highest test accuracy of 90.89% with cell-size 8×8. Random Forest fluctuates in terms of recognition accuracy when the cell-size is varied giving 79.19% and 91.00% as lowest and highest recognition accuracies respectively. Best recognition accuracy is achieved by SVM with a cell-size of 6×6. It can be seen that very small cell sizes do not perform well and are computationally very expensive as the feature vector size is too large. With DWT feature descriptor, SVM, Random Forest and KNN perform fairly well while the performance of MLP is relatively bad. Wavelet db1 gives overall better accuracy than those given by db2 and db3. KNN performs the best with an accuracy of 88.24% and MLP performs the poorest giving an highest recognition accuracy of 82.90%.

Table 2.3: Recognition accuracies achieved using different wavelets and decomposition levels. Figures in bold signify highest test accuracy for concerned classifier

Wavelet	Level of decomposition	Feature Vector Length	SVM	KNN	RF	MLP
db1	1	144	**86.69%**	87.90%	86.89%	81.65%
	2	36	83.13%	**88.24%**	**87.31%**	80.91%
	3	9	55.89%	61.53%	63.36 %	59.32%
db2	1	169	85.51%	84.43%	86.17%	81.11%
	2	64	83.41%	81.98%	85.31%	82.31%
	3	25	70.29 %	61.83%	71.76 %	72.78%
db3	1	196	84.59%	83.64%	85.45%	**82.96%**
	2	81	82.24%	77.47%	80.92%	81.41%
	3	49	74.17%	65.27%	72.96%	77.22%

2.3.2 Recognition using image pixel intensity (IPI) values

Vectorized representations of IPI values of the images are fed to the classifiers. The same set of classifiers viz. SVM, KNN, RF and MLP are employed to test the accuracy using IPI values. For each classifier and for each set of parameters specified in the second column, three runs are performed and the average of recognition accuracies are provided in the third column (Table 2.4) The same procedure is followed in achieving the results shown in Table 2.2 and Table 2.3.

When IPI values are fed to the classifiers with different sets of parameters (Refer Table 2.4), recognition accuracy achieved is not significant compared to the ones shown by the same classifiers using handcrafted feature descriptors. Of the four classification algorithms considered for the present study, Random Forest achieves the highest test accuracy of 86.20% closely followed by SVM with test accuracy of 86.17%. SVM, RF and MLP show close results with highest recognition accuracies of 86.17%, 86.20% and 86.03% respectively while KNN gives comparatively lower accuracy of 81.47%.

2.4 Discussion and Conclusion

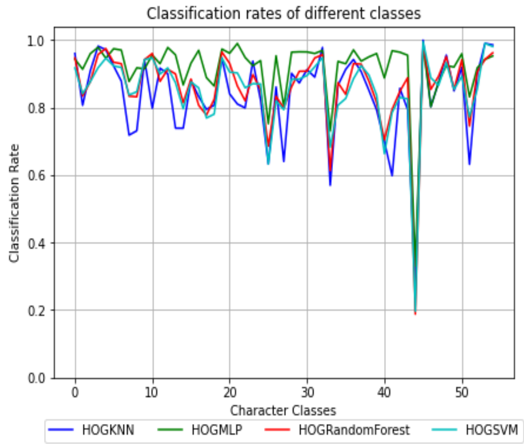
This chapter presents a handwritten dataset of complete character set of Meitei Mayek. It has a total of 85,124 images with 72,330 in training set and 12,794 in test set. A performance analysis of four state-of-the-art classifiers with three different types of features have been carried out. From the experimental results it is observed that SVM with HOG features gives the best accuracy out of all other classifier-feature combination against the developed dataset. In order to assess

Table 2.4: Recognition accuracies achieved using IPI values with different classifiers. Figures in bold signify highest test accuracy for concerned classifier

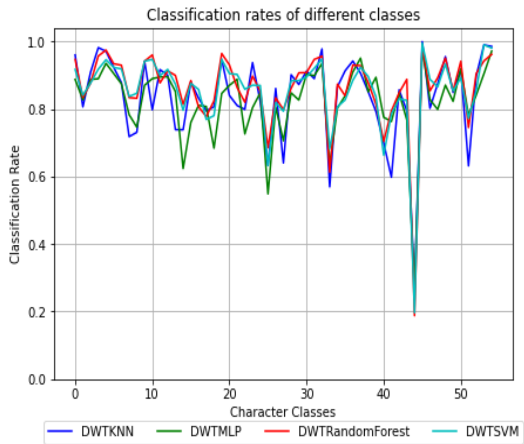
Classifier	Parameters	Recognition Accuracy (%)
Support Vector Classifier	c=1 kernel=linear	80.94
	c=10 kernel=linear	79.80
	c=10 kernel=RBF	84.65
	c=100 kernel=RBF	85.88
	c=10 kernel=sigmoid	80.16
	c=10 kernel=poly	84.67
	c=100 kernel=poly	**86.17**
KNeighbor Classifier	weights=uniform n_neighbors=1 p=1 algorithm=auto	80.98
	weights=uniform n_neighbors=3 p=1 algorithm=auto	80.18
	weights=uniform n_neighbors=5 p=1 algorithm=auto	80.05
	weights=distance n_neighbors=5 p=2 algorithm=auto	80.53
	weights=distance n_neighbors=7 p=2 algorithm=auto	79.80
	weights=distance n_neighbors=5 p=1 algorithm=auto	**81.47**
	weights=distance n_neighbors=1 p=1 algorithm=auto	80.98
Random Forest Classifier	max_depth=50 criterion=gini n_estimates=100	85.70
	max_depth=100 criterion= gini n_estimates=100	85.70
	max_depth=200 criterion=gini n_estimates=100	85.70
	max_depth=50 criterion=entropy n_estimates=100	**86.20**
	max_depth=200 criterion= entropy n_estimates=100	**86.20**
	max_depth=50 criterion= entropy n_estimates=50	84.72
	max_depth=50 criterion= entropy n_estimates=20	80.73
MLP Classifier	activation='relu',batch_size='auto',hidden_layer_sizes(100,), solver='sgd'	81.41
	activation='relu',batch_size='auto',hidden_layer_sizes(100,), solver='adam'	82.64
	activation='relu',batch_size='auto',hidden_layer_sizes(50,50), solver='adam'	84.10
	activation='relu',batch_size='auto',hidden_layer_sizes(50,50), solver='sgd'	82.61
	activation='relu',batch_size='auto',hidden_layer_sizes(50,50,50), solver='adam'	85.83
	activation='relu',batch_size='auto',hidden_layer_sizes(100,50,50), solver='adam'	85.98
	activation='relu',batch_size='auto',hidden_layer_sizes(100,100,100), solver='adam'	**86.03**

the performance of each classifier-feature combination in terms of each of the 55 classes of TUMMHCD, the best performing model for each combination is taken and a graph is plotted which is shown in Figure 2-8.

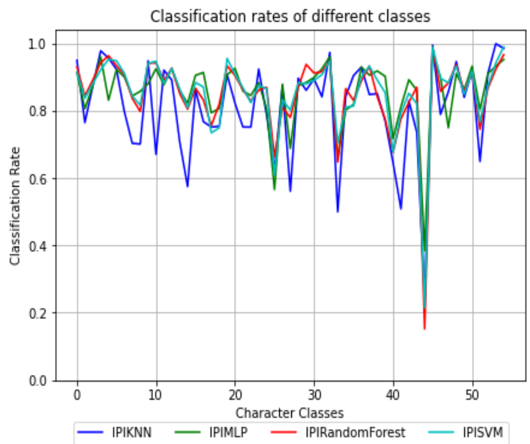
It can be observed from the figure that the relative classification accuracies of the classifier-feature combinations shown in the graph for different classes show similar pattern, i.e. if for a certain classifier-feature combination the classification rate for a class is very less compared to other classes, then similar results are also observed for other classifier-feature combinations. For example, it is clearly visible that there is a dip for characters with class ids like 25, 33, 44, etc. This means that there are some characters which have been misclassified for a greater number of times by all the classifier-feature combination. The presence of such characters leads to lower recognition accuracy of the system. It is necessary to explore more features and techniques in order to achieve better accuracy. Since deep learning,



(a) Using HOG feature descriptors



(b) Using DWT feature descriptors



(c) Using IPI values

Figure 2-8: Classwise classification rates for different feature-classifier combinations

especially CNNs learn the features on its own and due to its success in many pattern recognition and computer vision tasks including character recognition, it is employed for our dataset as well. The details are provided in the next chapter.