Chapter 4

A multilevel recognition of TUMMHCD using fusion of features strategy

4.1 Introduction

Both traditional hand-crafted features and deep learned features have their own advantages when it comes to their performance both in terms of computation cost and feasibility. Handcrafted features can be manually designed and can be used for specific classification task such as finding a specific feature descriptor which discriminates one character from another highly shape-similar character. They are computationally less costly compared to deep features. They can also be used to train classifiers when the amount of training data available is not large enough to train a deep network. Deep features however take more time to train a classifier and require quite a large amount of training data for training to carry out. Recent works on various tasks of computer vision however show the superior performance of deep features over the traditional handcrafted features [37, 194].

The main idea behind combining the two types of features is to have both local and global features for the classification task. Global features describe the entire image, whereas local features describe the image patches (small group of pixels). CNNs, however, focus mainly on local features such as points and angles in the images while giving less attention to the global features such as contour and structural features [241]. Deep CNNs are sensitive to local contour features of an image pattern but provide no information regarding the global shape information,

which otherwise is how human visual system works [21, 22]. It is however imperative to have both the types of features in order to have a more discriminative set of features for better recognition of images. In our approach, both local and global features are employed. The contours of objects detected by HOG method hold important information regarding the local and global shapes and structures of the objects. These global structural features when fused with the deep features extracted from CNN enhances the discriminative power of the recognition system. The importance of another type of feature descriptor is also taken into account in the present work. The traditional CNNs process the image data in spatial domain. Discrete Wavelet Transform (DWT), on the other hand processes data in the spectral domain. The features in spatial domain and spectral domain tend to have different characteristics. Therefore, it is expected that both the features will have different information of an image pattern [60]. Integration of DWT in the working of conventional CNNs has shown improvement in tasks such as signal processing [105], image classification by adopting subbands of approximation and detail coefficients to feed multiple CNNs using an ensemble approach [231]. DWT is also used for achieving noise-robust image classification by replacing convolutional and pooling layers in traditional CNNs by DWT [116].

Deep networks usually learn local features in its initial layers. As the layers go deeper, the features become more abstract and generalized in nature and are called global features. Since the filter size and the receptive field are usually smaller than the size of the image, the feature maps contain local information of the image pattern. Because these features only depend on the pixels in its receptive field, they are called local features. Global features are those which depend on the pixels of the entire image. The features learned by CNNs at lower layers called low-level features are more specific to the training dataset, so one has to be careful while selecting the training dataset [140]. In works where pre-trained CNN models are employed for computer vision tasks [39, 66, 74, 196], top layers are usually used to extract high-level features. Since our dataset is big enough to train a CNN model which is sufficiently deep for our task, both low-level and high-level features might prove to be a richer feature representation of the image patterns. It is also reported in the literature that using lower-layer features can yield better results for classification using SVM and RF [20, 236].

4.2 Related work

Fusion of feature descriptors viz. handcrafted features and deep features have been explored for HCR in some scripts. Recent works have shown that performance of deep networks can be enhanced by the adoption of hand-crafted feature descriptors in addition to deep features. The local and global features used in fusion approaches can be extracted from different methods. For example, HOG is a global hand-crafted feature descriptor whereas Scale-Invariant Feature Transform (SIFT) is a local feature descriptor. Moreover, both local and global features can be extracted from a CNN depending on the layer used for the feature extraction.

Fusion of handcrafted and deep features is reported in the work of Sharif et al. [195] where HOG features were used with deep features for recognition of Bangla numerals. The HOG features were fed to an ANN and features were extracted from the last fully connected layer. These features were fused with the deep features from the last layer of a CNN. The fused feature vector was then fed to another two-layered network for the final recognition of numerals. They achieved an accuracy of 99.17% on CMATERDB Bangla numerals. In the scene text detection work performed in Tang and Wu [209], handcrafted features viz. colour, texture and geometric features are fused with deep features from CNN and fed to a Random Forest regressor based classifier. F-measure values of 0.876, 0.885, and 0.631 are obtained on three benchmark datasets. In the method devised by Sulaiman et al. [206] for writer identification, the input image is divided into several overlapping patches. From each patch, local binary pattern (LBP) based handcrafted feature descriptors and deep feature descriptors based on Alexnet model are extracted separately. Visual words are then found out for both the types of feature descriptors using k-means algorithm. Vector of locally aggregated descriptors (VLAD) encoding is then employed to find a fix-sized vector representation. VLAD encodings of both the feature descriptors are then concatenated to find the final 1-D feature vector. Extreme learning machine (ELM) based classifier is used for classification on three publicly available datasets and reported state-of-the-art results. Fusion of feature descriptors is explored for online signature verification in the work carried out by Vorugunti et al. [223]. Here, global features are calculated from local features such as stroke order, x, y-coordinates, pressure, azimuthal angle etc at each sampling point of the signature image. Deep features are extracted from a convolutional autoencoder and are fused with handcrafted features. The method outperformed state-of-the-art methods on three benchmark datasets. Decision fusion and feature fusion strategies have been combined in the work by Mangai et al. [122]. The approach is tested on three UCI datasets and superior

performance has been reported.

In image classification research, fusion of handcrafted and deep features has found an important spot in dealing with medical images. Features such as morphological, texture, colour and density features are combined with deep features [106, 194, 228]. Classifiers used are MLP, ELM and SVM. Handcrafted features along with deep features from pretrained or modified deep neural network models such as Alexnet, VGG, ResNet, GoogleNet, InceptionV3, etc. have shown state-of-the-art performances [43, 58, 72, 117, 165, 179, 217]. In the study reported by Golrizkhatami and Acan [65], a score level fusion from different levels of a CNN is fed to a nearest neighbour classifier, a feature-level fusion of statistical and temporal features is fed to an SVM and morphological features after dimensionality reduction fed to an SVM. The outputs of these three classifiers are then fed to a decision level fusion block for the final classification. It was reported that combination of the three types of feature descriptor performed better than the feature descriptors when considered separately and they could achieve state-of-the-art results.

As far as Meitei Mayek is concerned, to the best of our knowledge there are no works reported in literature on fusion of feature descriptors techniques.

4.3 Proposed methodology

The proposed multilevel methodology has two levels of recognition. The first level uses a CNN for the recognition task. The second level uses a fusion of handcrafted and deep features for recognition. The second level recognition is only carried out for those test images which are filtered by a *filtering module*. The conceptual framework of the proposed methodology is given in Figure 4-1. The dotted lines represent the flow of training images and the solid lines represent the flow of test images. The proposed methodology incorporates the following modules:

- *First level recognition module:* This module is the first level recognition using CNN
- *Softmax module:* This module deals with finding the set of classes with highest three softmax output values
- *Filtering module:* This module filters or decides whether a test image will be forwarded to second level recognition or not

- *Second level recognition module:* The second level recognition of the test images filtered by the filtering module using fused features is taken care by this module

Each of the modules is described in the following sections.

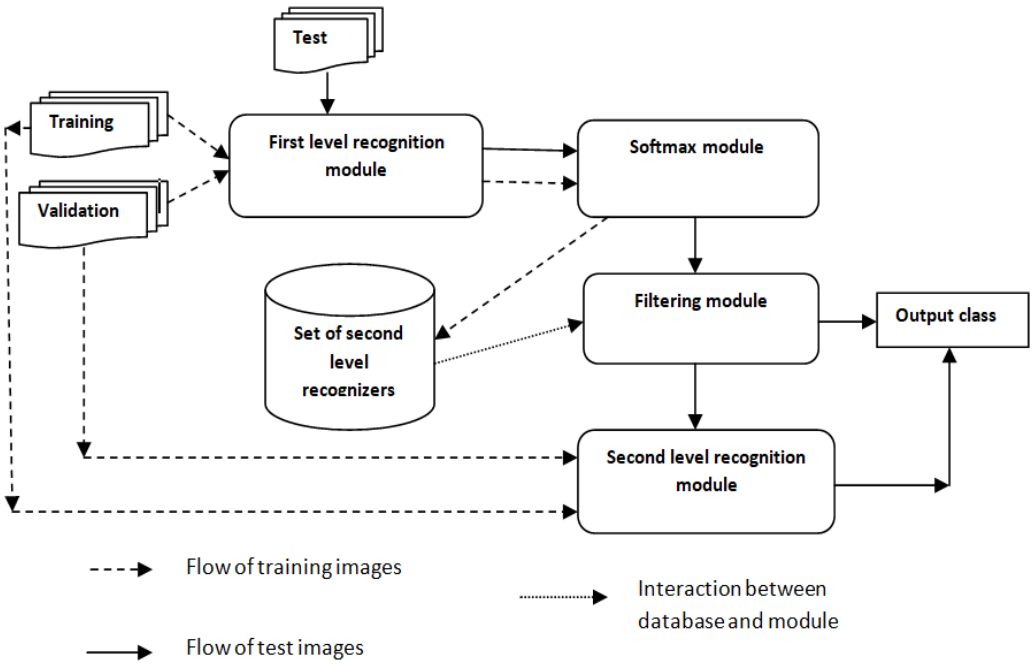

Figure 4-1: Conceptual framework flow of the proposed methodology

4.3.1 First level recognition module

The first module is the first level recognition using CNN. For training and validation purpose, our proposed CNN model is employed. The training is carried out using the same training and validation sets used earlier in the previous chapters. Maximum number of epochs used is 100. Early stopping with patience 20 is adopted. The test set is then fed to this model for the recognition to perform. The output of this first level CNN is utilized for two purposes: a) to find the first level output class of the test images and b) to find the set of classes with highest three softmax values based on the misrecognized images of the validation set. This set of classes is further used in the second level recognition and are called second level recognizers.

4.3.2 Softmax module

This module finds out the set of classes which would be considered for the second level recognition. As the name of the module suggests, the task is achieved by taking the output values of softmax function from the last layer of the CNN model. These output values will be referred to as the "softmax values" in the present thesis. The softmax values of the images in the validation set which are misrecognized by the first level CNN are the participating values for finding the set of classes for second level recognition.

4.3.2.1 Softmax function:

The softmax function is used as the activation function in the last layer of our proposed CNN model. Its function is to normalize the output of a network to a probability distribution over predicted output classes. It is represented by the following formula:

$$S(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)} \tag{4.1}$$

for $i = 1, 2, ...k$ and $z = (z_1, ..., z_k) \in \mathbb{R}^k$

For an input vector of k real numbers, the softmax function produces k output values, each output value corresponds to the probability of membership for each class.

The vector of probabilities that the softmax function outputs for a given test image thus gives an idea of the likelihood of that image belonging to the output character classes. The fact that it is a probability of belonging to a character class, it plays an important role in finding out the best potential output character classes for a test image. The set of recognizers to be considered for the second level recognition can then be decided once the potential candidates are found. The method proposed to do so is given in the next section.

4.3.2.2 Finding set of second level recognizers

A method based on softmax values of the misrecognized characters of the first level CNN is proposed. As mentioned earlier, the CNN is trained using TUMMHCD training and validation sets. To calculate the set of recognizers for the second level recognition, the images of the validation set which have been misrecognized by the

first level CNN have been taken. The notion behind this step is that the character classes of the images in the validation set which are misrecognized will likely be the same set of classes of the images in the test set which will get misrecognized. It is described as follows:
let ϑ = validation set
ϑ_i = set of images in the validation set belonging to class i
T = test set
T_i = set of images in the test set belonging to class c_i
if,

$x \in \vartheta_i$ gets misrecognized as $y \in \vartheta_j$ in the validation set ϑ,
then it is most likely that,

$p \in T_i$ will get misrecognized as $q \in T_j$ in the test set T

The algorithm to find the set of second level recognizers is given in Algorithm 1. For each misrecognized character image im in the validation set ϑ, softmax values are used. For each character class c_i, the average of softmax values of the misrecognized images in the class is calculated using equation 4.2. The softmax values of all misrecognized character images belonging to the particular character class are added and is divided by the number of misrecognized images in the character class. The character classes corresponding to three highest softmax values are taken for further steps to carry out. If the true class is one of these three character classes, then a recognizer corresponding to these three classes is taken for second level recognition. For example, for class c_1, if the character classes with three highest values are c_i, c_j and c_k, the algorithm checks if c_1 is either one of these three classes. If so then the classes c_i, c_j and c_k forms one recognizer in the second level recognition. The reason why highest three values are considered is because there are instances when the true class of the image is not one of the classes with highest two softmax values.

$$\varsigma_i = \frac{\sum_{i=1}^{n} s_i}{n} \tag{4.2}$$

where ς_i is the average softmax value of class c_i, s_i is the softmax value of a misrecognized image belonging to ϑ_i and n is the number of misrecognized images in ϑ_i.

Algorithm 1: Finding set of second level recognizers

Input: Softmax values of misrecognized images in validation set

Output: Set of second level recognizers, $\varkappa$

$\varkappa = \phi$

for $i \leftarrow 1$ **to** 55 **do**

$\varsigma_i = \frac{\sum_{i=1}^{n} s_i}{n}$

where,

ς_i is the average softmax value

s_i is the softmax value of a misrecognized image in ϑ_i

n is the number of misrecognized images in ϑ_i

Let c_x, c_y, c_z be the classes with three highest average softmax values

if c_i *is* $c_x||c_y||c_z$ **then**

$\kappa = \{c_x, c_y, c_z\}$ ▷ κ is the recognizer corresponding to classes c_x, c_y and c_z

$\varkappa \leftarrow \kappa \cup \varkappa$

end

end

return $\varkappa$

Following the above method, 28 recognizers are identified for the second-level recognition. As mentioned earlier, each of the 28 recognizers corresponds to three character classes. The 28 recognizers with the corresponding character classes are listed in Table 4.1. The class ids in the second column of the table are the class ids with descending order of softmax values. That is, for the first recognizer R_1, class with id 1 has the maximum softmax value followed by class id 8 and class id 2 has the lowest softmax value among the three of them.

4.3.3 Filtering module

The filtering module is a module through which every test image in the test set has to pass. This module decides whether a test image needs to be forwarded to the second level recognition module or not. The steps followed by the module to perform the task is given in Algorithm 2. The algorithm returns a 1 if the test image is to be forwarded to second level recognizer, else it returns a 0. The decision is made based on the softmax values of the test image obtained as the output of the first level CNN. The filtering module checks if the difference between the two highest softmax values of a particular test image is less than a threshold value of 0.70. It also checks if those two classes with highest softmax values are

Table 4.1: Recognizers for second level recognition

Recognizer id	Class ids	Character symbols
R_1	1, 8, 2	[illegible]
R_2	32, 3, 12	[illegible]
R_3	50, 33, 6	[illegible]
R_4	8, 1, 24	[illegible]
R_5	46, 9, 15	[illegible]
R_6	47, 11, 43	[illegible]
R_7	38, 15, 12	[illegible]
R_8	40, 47, 14	[illegible]
R_9	42, 18, 33	[illegible]
R_{10}	35, 32, 20	[illegible]
R_{11}	44, 25, 18	[illegible]
R_{12}	43, 15, 27	[illegible]
R_{13}	40, 28, 14	[illegible]
R_{14}	3, 18, 29	[illegible]
R_{15}	9, 30, 11	[illegible]
R_{16}	34, 18, 33	[illegible]
R_{17}	35, 20, 4	[illegible]
R_{18}	43, 10, 37	[illegible]
R_{19}	35, 12, 38	[illegible]
R_{20}	17, 28, 40	[illegible]
R_{21}	27, 15, 41	[illegible]
R_{22}	17, 16, 42	[illegible]
R_{23}	27, 37, 43	[illegible]
R_{24}	25, 44, 38	[illegible]
R_{25}	11, 47, 6	[illegible]
R_{26}	49, 4, 48	[illegible]
R_{27}	48, 45, 52	[illegible]
R_{28}	52, 45, 54	[illegible]

correspondingly same as the two classes with highest softmax values in one of the 28 recognizers identified earlier. If both these conditions are true, the test image is given to the particular recognizer in the second level for second level recognition. The value of the threshold is decided upon empirically by taking different values of 0.65, 0.70, 0.75 and 0.80. The performance results of using different threshold values for the filtering module is given in Figure 4-4.

Algorithm 2: Filtering test image

Input: Softmax values of the test image, set of second level recognizers $\varkappa$ and filtering threshold t

Output: 1 or 0

$s_t = 55$ softmax values of the input test image. $t = 0, 1, 2, ..., 54$

`/* Find the classes with two highest softmax values */`

Let s_1 and s_2 be the two highest softmax values corresponding to the classes c_i and c_j respectively, where i and j are the class indices

Let R_1 represent a recognizer with a, b, c as the class indices having three highest softmax values

if $(s_1 - s_2 < t) \wedge (i == a \wedge j == b)$ **then**
  return 1
end
else
  return 0
end

The reason why only the first two classes are considered while filtering the test images is because a number of test images failed to be forwarded to the second level recognition when the third class is also considered in making the decision. This is because the true class of most of the test images that enter the second level recognition is either the first class or the second class. However, the recognizers are trained with three classes each. Therefore, the test image still has a chance to be classified as belonging to the third class if it at all belongs to the third class. This makes the system more robust to instances where the true class of a test image is the third class, which happens in some of the cases.

4.3.4 Second level recognition module

The test images which are filtered out by the filtering module for second level recognition are fed to the second level recognition module. This module has 28

recognizers which are found by the softmax module as described earlier. Each of these 28 recognizers $(R_1, R_2, R_3, ...R_{28})$ is trained with the respective three character classes given in Table 4.1. Recognizer R_1 is trained with fused feature set of classes with ids 1, 8 and 2; recognizer R_2 with classes with ids 32, 3 and 12; and so on. The feature set used to train the 28 recognizers is a fusion of three types of features: traditional handcrafted HOG feature descriptors and deep learned features as shown in the Figure 4-2. The deep learned features are of two types: a) deep features extracted by feeding CNN with the approximation coefficients of DWT b) deep features extracted by feeding CNN with raw grayscale images. These three types of features are fused to produce the final feature set. The final fused feature set is given to an SVM for the final classification. The details are provided in the following sections.

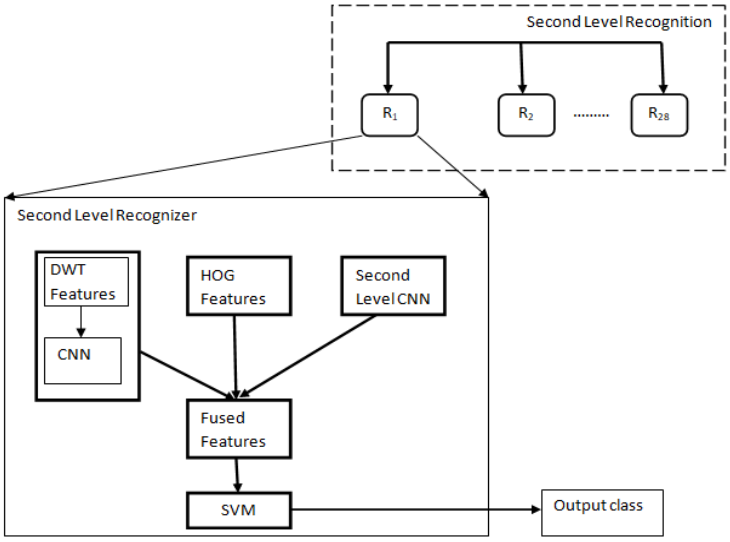

Figure 4-2: Second level recognition module

4.3.4.1 Handcrafted feature descriptor extraction

The HOG descriptor [42] tries to find the global contour like shape and structure of an image object. One important property of HOG is that it captures both magnitude and orientation of the edges in an image. In handwritten characters, it focuses on finding the edges for which change in gradient is high and thus giving significant information regarding the shape of the characters. It also discards information which is not relevant and focuses on the one which will be discriminative enough to distinguish between the character classes. The feature extraction process is carried out in the same manner as described in Section 2.3.1.1.

4.3.4.2 Deep feature descriptor extraction

The CNN model proposed for recognition of TUMMHCD is used to extract the deep features. The main layers of the CNN is shown in Figure 4-3. The input to the CNN are of two types: raw grayscale image pixel intensity (IPI) values and approximation coefficients produced by 2D-DWT. The approximation coefficients of the first level decomposition of the wavelet *db1* is used for feature extraction. The details of 2D-DWT are given in Section 2.3.1.2. The images are first size-normalized to 48×48 for the extraction of DWT-based deep features. The approximation coefficients array obtained after first level decomposition with *db1* is of size 24×24 which is fed to CNN. In essence, there are two CNNs with the same model but with two different types of inputs. Both the CNNs are trained and validated in the same fashion as the first level CNN using the training and validation sets of TUMMHCD. The only difference is the feature set used. The CNNs in the second level are trained and validated using the fused feature sets of their respective three classes.

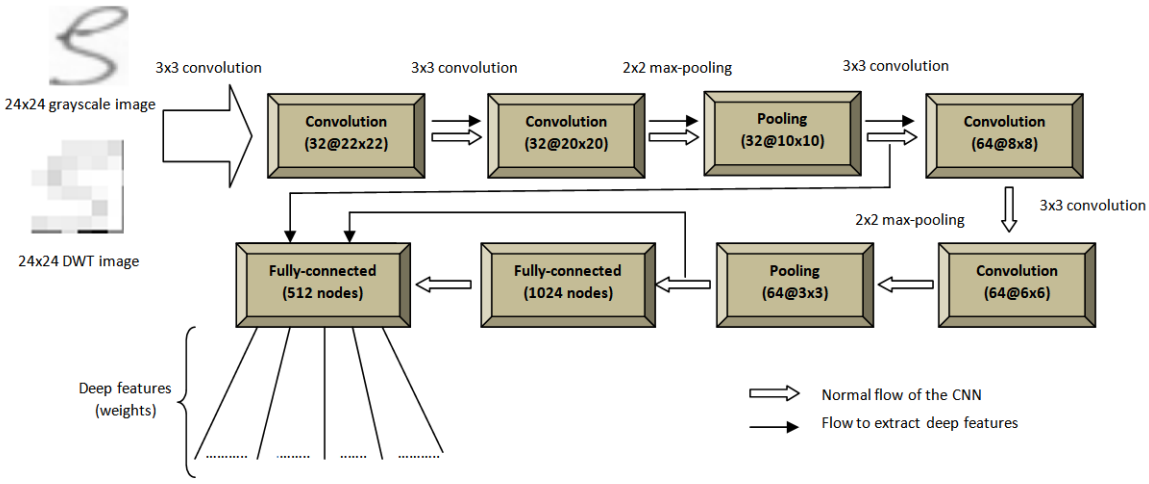

Figure 4-3: CNN architecture used for extracting deep features. $i@j \times j$ indicates i output feature maps of size $j \times j$

As shown in Figure 4-3, the deep features are extracted from two different layers of the CNN. They are the first and second pooling layers. The output feature maps of these two layers are forwarded to the last fully-connected layer having 512 nodes. The connection weights of this last fully-connected layer constitute the deep features used in the work. The reason the fully-connected layer is used as the layer to extract deep features is because the chances of network learning redundant and undesirable features are high if feature maps from the convolutional or pooling layers are directly used [199]. Therefore, it is important that the features are passed through fully-connected layer to get rid of those unnecessary features.

Each recognizer in the second level is trained and validated with their respective three classes. For the two types of input data viz. IPI values and approximation coefficients, two best models are achieved using early stopping method for each recognizer. The saved models are used to extract the connection weights between the layers which are the desired deep features. Therefore, the feature vector length of the deep features is 1024 (512 + 512) for image with IPI values and another 1024 (512+512) for image with approximation coefficients. This gives the final deep feature vector length of 2048. The length of the final fused feature set is 3812 (1764 + 2048).

4.4 Experimental results and discussion

The performance evaluation of the proposed methodology is carried out using the test set of TUMMHCD which consists of 12,974 images. Each test image first goes through the first level recognition module where the first level CNN performs the recognition task. The image either completes its recognition or is sent for second level recognition depending on the decision of the filtering module. The images which get forwarded to the second level recognition module undergo second level recognition to get classified to one of the character classes.

4.4.1 Overall system performance

Table 4.2 shows the number of test images obtained against the corresponding specifications given in the first column. For 916 test images out of the total 12,974 test images, it is found that the difference between the classes with two highest softmax values is smaller than the threshold value of 0.7. And out of these 916 images, 556 are forwarded to the second level recognition by the filtering module based on the two criteria mentioned in Section 4.3.3. It is a common intuition that not all of the 559 test images filtered by the filtering module are wrongly recognized by the first level CNN. There are 176 test images out of the 559 images that are correctly classified by the first level as well as the second level recognition modules. The correct recognition of these test images by the adoption of the second level recognition module, thus, does not contribute anything to the overall accuracy of the system.

There are also 74 test images out of the filtered 559 images which are wrongly recognized by both the recognition modules. 254 test images which are

Table 4.2: Statistics of results obtained in multilevel recognition with fusion of features strategy

Specification	Value
Number of test images passing the filtering module	916
Number of test images entering second level recogntion	559
Number of test images classified correctly by both the levels	176
Number of test images missclassified by both the levels	74
Number of test images missclassified by first level and classified correctly by second level	254
Number of test images classified correctly by first level and missclassified by second level	55

misrecognized by the first level recognition module are however recognized correctly by the second level recognition module. The second level recognizers could not recognize 55 test images out of the filtered images. These 55 test images are however classified correctly by the first level recognition module. The recognition accuracy of the overall system is 97.09%. Therefore, there is an improvement of 1.53% in the recognition accuracy from 95.56%.

4.4.2 Performance with different training scenarios

The performance analysis of the system is done for different training scenarios. The four different scenarios under consideration and the recognition accuracies achieved by adopting each training strategy are given in Table 4.3. When only the first level recognition module, i.e. the first level CNN is trained with deep features alone, recognition accuracy achieved is 95.56%. When the first level CNN is trained with the fused feature set of three different types of features without the adoption of the second level recognition module, a recognition accuracy of 96.52% is obtained. An increase of 0.96% in the recognition accuracy is observed when the recognition is carried out by the second level recognizers in addition to the first level CNN. This training scenario adopts SVM as the classifier for the classification task.

The performance of the system is also tested for the scenario where the second level recognizers are trained and evaluated with the fused feature set of all the classes in the training set. In that case, the recognition accuracy of the system is 96.32%. Finally, the training strategy adopted in the proposed methodology is considered. It has the first level recognition carried out with CNN and second level recognizers are trained and validated with the fused feature set of only the three

Table 4.3: Performance results of different training scenarios

First level	Second level	Recognition accuracy	Error rate
Trained with deep features	No second level	$95.56\% \pm 0.357\%$	4.44%
Trained with fused features (with SVM)	No second level	$96.52\% \pm 0.318\%$	4.48%
Trained with deep features	Trained with fused feature set of entire training set (with SVM)	$96.32\% \pm 0.326\%$	3.68%
Trained with deep features	Trained with fused feature set of identified three character classes (with SVM)	$97.09\% \pm 0.291\%$	2.91%

character classes identified for the respective recognizers. This strategy gives the highest recognition of 97.09% among all the four training strategies considered in the work.

4.4.3 Performance with different filtering threshold values

The threshold value used by the filtering module is decided upon empirically by taking different values as mentioned earlier. The experimental results obtained by taking different values of the filtering threshold values are shown in Figure 4-4. The performances of the top-3 performing classifiers for TUMMHCD namely SVM, KNN and RF are evaluated. The results show that the filtering threshold value of 0.7 gives the highest recognition accuracy with SVM classifier while that of 0.8 gives the lowest. With KNN and RF, the filtering threshold value of 0.75 shows slightly better results than those of the other threshold values but less than what SVM obtains with threshold value of 0.7. SVM gives an overall higher recognition accuracy than KNN and RF.

4.4.4 Performance on MNIST, DIDA and CArDIS

Performance evaluation of the proposed strategy is also carried out using the famous MNIST[111] and two newly developed datasets namely DIDA[104] and CArDIS[237]. The DIDA "single digit dataset 10k" and CArDIS datasets are divided randomly in the ratio 9:1 to obtain training and test sets respectively. The SVMs of second level are trained using 1000 images from each of the three datasets. Table 4.4 lists the details of the datasets and the experimental results obtained using the proposed methodology.

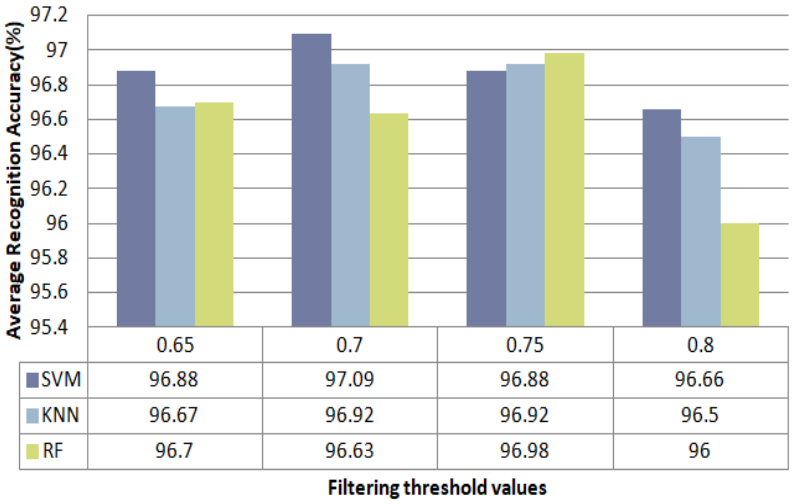

Figure 4-4: Recognition accuracies achieved using different values of the filtering threshold parameter with SVM, KNN and RF

Table 4.4: Performance of the proposed strategy on MNIST, DIDA and CArDIS datasets

Dataset	Number of classes	Training set	Test set	First-level recognition accuracy	Number of second-level recognizers	Multilevel recognition accuracy
MNIST	10	60000	10000	99.49%	4	99.56%
DIDA	10	8500	1500	98.26%	1	98.33%
CArDIS	29	55696	9839	97.73%	10	98.10%

4.4.5 Discussion

Experimental results show that some characters which are misrecognized by first level recognition module are recognized correctly by the second level recognition module. The filtering module plays a significant role in achieving the overall system performance. The filtering threshold value serves as the first layer of the filtering module. From Figure 4-4, it is evident that the filtering threshold values greater or less than 0.7 do not perform well. When the threshold values are higher, even though more number of test images might be forwarded to the second level recognition, it is not necessarily true that all those test images need to be forwarded. This implies that for a test image i_t, if the difference $(s_1 - s_2)$ of its two highest softmax values is less than 0.8 but not necessarily less than 0.7, it is unlikely that i_t is misrecognized by first level recognition module. This indicates that for those test images whose $(s_1 - s_2)$ value is less than 0.8 but greater than

0.7, the first level recognition performs well in giving the correct output classes they belong to. Having said that, when a filtering threshold value of 0.65, which is smaller than 0.7, is considered, the accuracy of the system again decreases. In this case, the number of test images forwarded for second level recognition is less and it misses out on some test images which are misrecognized by the first level recognition module that need a second level recognition.

The results obtained for different training scenarios (Table 4.3) show that when the second level recognizers are trained with the fused feature set of the entire training set, the recognition accuracy decreases from the one where only the first level CNN is trained with the fused feature set. This shows that the fused feature set does not give a very discriminative representation of the image patterns for those characters which go to the second level recognition. Even though there is an improvement in recognition accuracy from 95.56% to 96.32% with the incorporation of the second level recognition, the improvement is not as good as the one obtained with the last training scenario. The last training scenario trains the second level recognizers with the three character classes identified for each recognizer and provides a recognition accuracy of 97.09%. This however, comes with the overhead of identifying the set of character classes for the second level recognizers. For the other three datasets, maximum accuracies of 99.56%, 98.33% and 98.10% are achieved for MNIST, DIDA and CArDIS datasets using SVMs in the second level. It is observed that there is an increase in the recognition accuracies with the employment of second-level recognition.

4.5 Conclusion

A methodology for recognition of Meitei Mayek handwritten characters is proposed in this chapter. The approach adopts a multilevel recognition with two levels of recognition. A filtering module acts as the decision making module between the first and second level recognition modules. A method based on the softmax values of the first level CNN is used by the filtering module for identifying the test images which have to undergo a second level recognition. The second level recognition works with a fusion of handcrafted and deep feature descriptors with SVM classifier. The proposed methodology could achieve an improvement of 1.53% in recognition accuracy over the single-level recognition.