RESEARCH ARTICLE OPEN ACCESS

Individual Handwritten Basic Bengali Character Recognition Using SVM Classifier

Tarek Bin Zahid, Miftahul Jannat Rasna, Mst Fouzia Hoque

(Electrical & Electronic Engineering, University of Dhaka, Bangladesh)

Abstract:

One very popular and useful application of machine learning is character recognition. Very few scientific works regarding character recognition involving Bengali characters have been done in the past years. The aim of this paper was to train a simple classifier that can detect individual handwritten basic Bengali characters. Our work involved collection of basic Bengali character samples, segmentation, noise removal and using different ML algorithms for training the model to test for optimum accuracy in recognition of the basic Bengali characters. Our results suggest that, using SVM classifier best suits our job with an overall accuracy of 86.7%.

Keywords — **support vector machine, Bengali character, handwritten, supervised learning.**

I. INTRODUCTION

Machine Learning (ML) teaches computers to do what comes naturally tohumans and animals: learn from experience. Algorithmsused in ML comprise computational methods to "learn" information directly from datawithout relying on a predetermined equation as a model or being explicitly coded to solve a problem. Character recognition is a classic supervised classification machine learning problem. Many academic and research works have been done and published on English characters but unfortunately very few on Bengali characters. Handwritten characters are harder to recognize since they vary by person, style, and do not have any specific pattern. In order to use machine learning to train a model which can detect individual handwritten characters with reasonable accuracy, different techniques have been employed in this paper to process collected sample data and different algorithms have been applied to finally select the best model by analysing the accuracy of the model.

II. SUPPORT VECTOR MACHINE

For finding the best separating line, Support Vector Machine is widely known. It looks for the closest points which are called "support vectors" and the name "Support Vector Machine" came because points are like vectors. And also, the closest points support the best line [1].

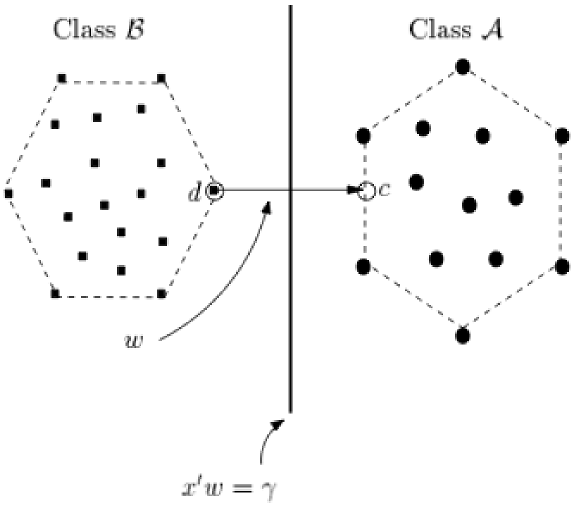


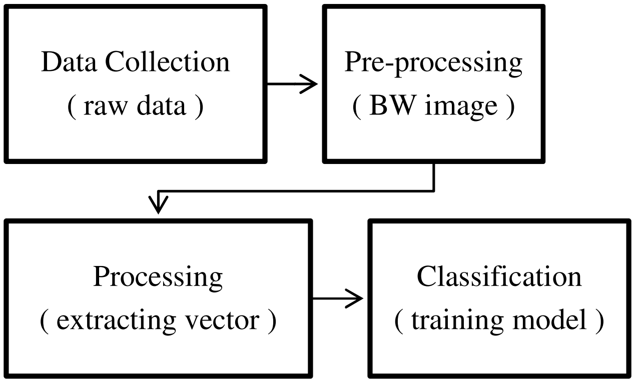
Fig. 1 The closest two points of the convex hulls determine the separating line [1]

After finding out the closest points, the SVM connects them by drawing a line (the line labelled 'w' in Fig. 1). The drawing is done by performing vector subtraction (point A- point B). Then the line that bisects and is perpendicular to the connecting line is declared to be the best separating line by the SVM.

III. APPROACH

Over the years, Bengali character recognition has gained a considerable interest among the researchers due to emergence of computationally demanding application like data mining and document classification. Contrary to characters from other languages, Bengali characters presents unique set of challenges when trying to train a model.

General steps followed in a machine learning process:



A. Data Collection

We collected data by employing students from different batches of the Department of Electrical &Electronic Engineering, University of Dhaka, Bangladesh and through kind permission from Computer Vision & Pattern Recognition Unit, Indian Statistical Institute, to use their dataset [2].

The data obtained from ISI are individual basic Bengali characters and already segmented. Some samples of their data from different classes are showed in Fig. 2.

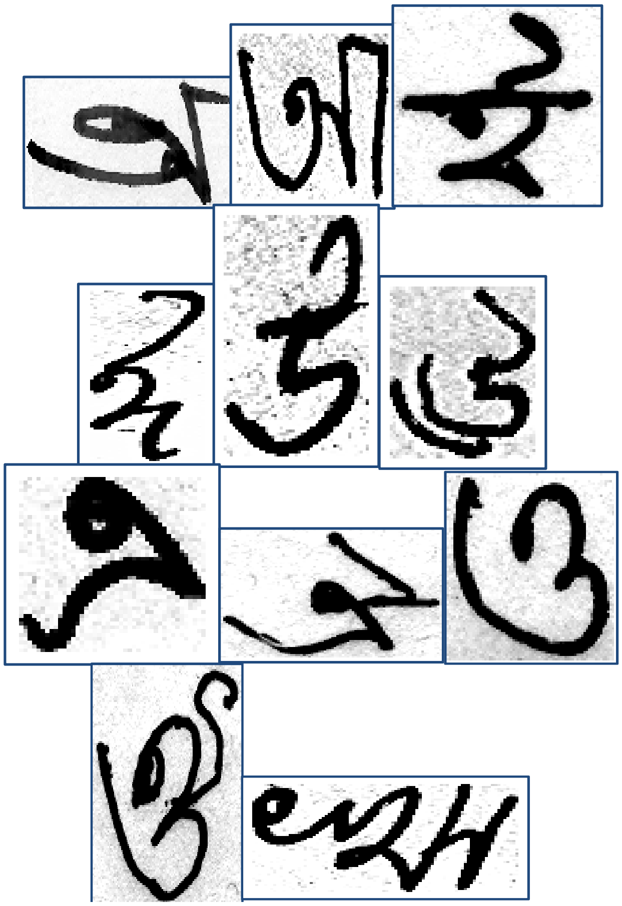


Fig. 2Data samples of all ten basic Bengali Characters from ISI haandwritten Bengali basic Character database

The data provided by ISI are very noisy and of different dimensions. Also the sample classes are highly uneven in size. The most number of samples are from অ and least number of samples are from ঋ having 740 and 423 samples respectively.

We also engaged forty-two students from the Department of Electrical and Electronic Engineering, University of Dhaka to collect samples. Each student filled an A4-sized paper with five characters from each class. This process was prompt and took few hours.After scanning the A4 pages, we performed segmentation onto the scanned data. After basic segmentation (without noise removal) we have obtained about 200 samples for each character class. Fig. 3 shows Few samples after segmentation from first four character classes.

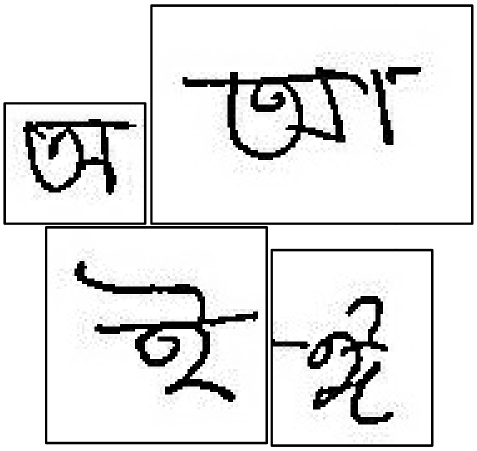
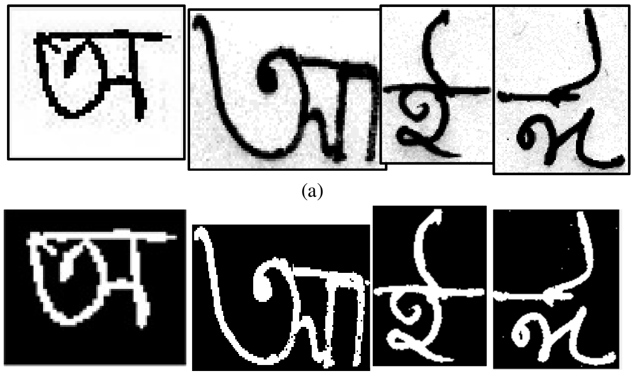


Fig. 3 Samples from first four classes of locally collected handwritten basic Bengali characters after segmentation

Though the locally collected data are very less noisy compared to that of collected from ISI, these contain morewhite spaces. These white spaces have been eliminated during data pre-processing.

B. Data Cleaning

Data cleaning has been performed on both thelocally and imported dataset. The images of all the samples for each basic Bengali character have been converted to binary values and converted to black and white by image thresholding [3] in Matlab so that all color information are discarded and astray marks are removed. Thresholding process also solves the problem of local samples that were of blue ink. Fig. 4 shows some noisy data and corresponding binary-converted data.



(a)

(b)

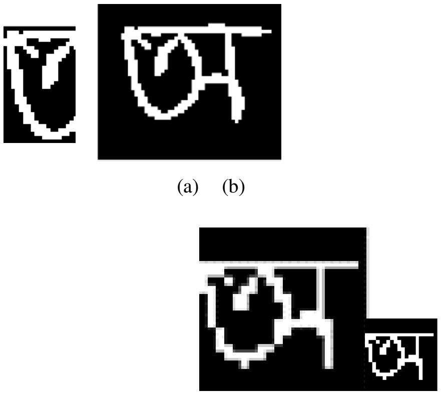
Fig. 4 (a) Samples of noisy data and (b) samples of B/W binary-converted data from first four classes of handwritten basic Bengali characters

C. Data Transformation

We have applied pixel-based detection method, our final aim was to convert all binary pixel values of the images into a row vector.

First, we have started processing the samples by removing rows and columns which doesn't include any information; rows and columns which has all 0 values (Fig. 5(a)). So, this process changes the image sample into rectangular shape (Fig. 5(b)). In order to maintain 1:1 aspect ratio by including minimum possible rows or columns evenly on both sides of the rectangular image, we performed bit-padding on all of the images (samples). This caused the rectangular image to be square in shape having 1:1 aspect ratio (Fig. 5(c)).

Finally, as the image is square in shape we can down convert to smaller resolution maintaining 1:1 aspect ratio. We converted it into 20x20 pixels (Fig. 5(d)).

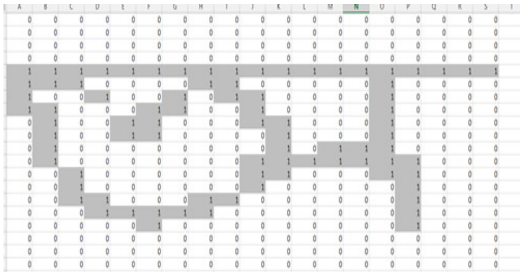


(a) (b)

(c) (d)

Fig. 5 Data transformation

The array presented in the Fig. 6 is a 20x20 pixel image, where the black portion of the image is represented as 0 and the shape of the character is represented as 1.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	
1	0	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	
1	1	0	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0	
0	1	0	0	1	1	0	0	0	1	1	0	0	0	1	0	0	0	0	
0	1	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	
0	1	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	
0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	
0	0	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	
0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	
0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	
0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig. 6 A 20x20 pixel image presented as a pixel array

The pixel array has been then converted into a 1x400 elements long or 400 features row vector [4] also known as feature vector. A feature vector is an n-dimensional vector of numerical features that represent some object.

This process is continued for each of 11 classes, each class having 600 samples. Therefore, we get 600*11=6600 feature vector, each one having 400 features.

D. Defining Class

We have assigned each of the11 character classes a unique class number from 0 to 10 (Table 4.1). Defining Class:

We have assigned each of the11 character classes a unique class number from 0 to 10 (Table 1).

TABLE I

Class name	Class
0	অ
1	আ
2	ই
3	ঈ
4	উ
5	ঊ
6	ঋ
7	এ
8	ঐ
9	ও
10	ঔ

IV. TRAINING THE MODEL

For training our model, Matlab's built-in 'Classification Learner' app has been utilised. From the 'APPS' we selected 'Classification Learner' and then started a new session. From the workspace we imported the variable which holds 401x600 vectors into the learner. We selected first 400 columns as 'Predictor' or input features and 401st column as 'Response' or output class name. Cross-validation folds have been selected as 20 folds. We started the training session by pressing 'Start Session'.

From the drop-down list containing the list of different algorithms we choose 'All' and press 'Train'. Training took some time. Then, after training all the classifiers, we noticed cubic SVM classifier has the highest rate of accuracy. We further performed several 'Advanced' techniques on the cubic SVM classifier to obtain a better accuracy.

V. ADVANCED TECHNIQUES PERFORMED ON CUBIC SVM

We have used all the features out of 400 and relied on PCA for data reduction. We have set kernel scale to 10, box constraint level 1, and choose multi class method as one-vs-all as opposed to default value one-vs-one. It took us a total of 21 one iterations to get the maximum accuracy of 86.7%. We later exported the selected classifier into the workspace for our own testing.

VI. PARTICULARS ABOUT THE CHOSEN CLASSIFIER

A. Model Number: 21

Status: Trained
Accuracy: 86.7%
Prediction Speed: ~1100 obs/sec
Training Time: 27.803 sec

B. Classifier

Preset: Cubic SVM
Kernel Function: Cubic
Kernel scale: 10
Box constraint level: 1
Multiclass method: One-vs-All
Standardize data: true

C. Feature Selection

All features used in the model, before PCA

D. PCA

After training, 70 components were kept.

Explained variance per component (in order): 6.8%, 3.5%, 2.8%, 2.2%, 2.0%, 1.8%, 1.7%, 1.6%, 1.5%, 1.4% (variance of least important components hidden).

VII. RESULT ANALYSIS

Since we have trained our model we will analyze the accuracy of our model both by using standard performance measure tools and manual data input.

Matlab provides quite a few numbers of standard tools which we will use to test the performance of our model. The tools include confusion matrix and ROC curve.We will be analyzing the standard tools for first four classes.

A. *Confusion Matrix*

We plotted our confusion matrix for our model for first four classes (Model 21):

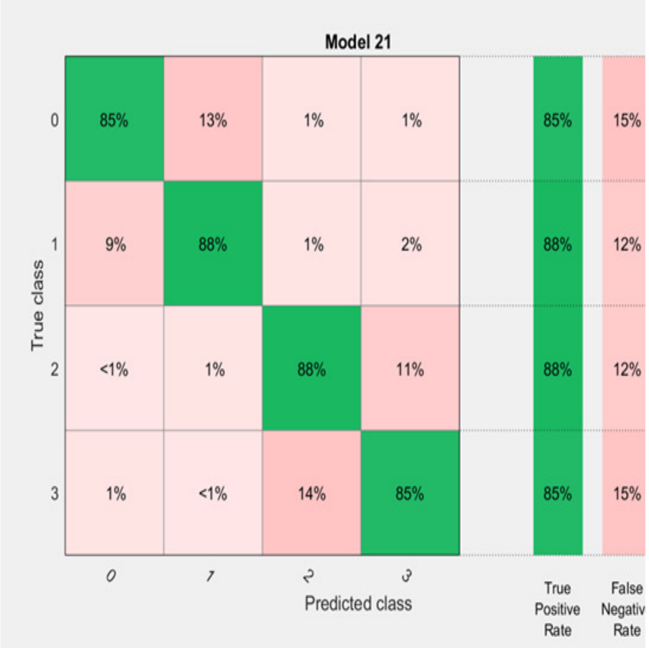


Fig. 7 Confusion matrix plotted for first four character classes of model 21

As we see from our model's confusion matrix, the green diagonal cells, which represented numbers of correct prediction of class is 85% or higher. We also see from here that highest false negative for each class is also the adjacent class. Which means the highest confusion which occurs during prediction is between similar looking characters like অ and আ or ই and ঈ. The overall accuracy of our model 86.7%. So, we can confidently say true positive rate for other class is around 85%.

B. *ROC Curve*

The Receiver Operating Characteristics (ROC) curve is a plot depicting the trade-off between the true positive rate and the false positive rate for a classifier under varying decision thresholds [5]. Accuracy is measured by the area under the ROC curve (AUC). An area of 1 represents a perfect test; an area of .5 represents a worthless test. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system [6]:

.90-1 = excellent (A)
.80-.90 = good (B)
.70-.80 = fair (C)
.60-.70 = poor (D)
.50-.60 = fail (F)

Fig.8, 9, 10 and 11 show areas under ROC curve for positive classes 0, 1, 2 and 3.

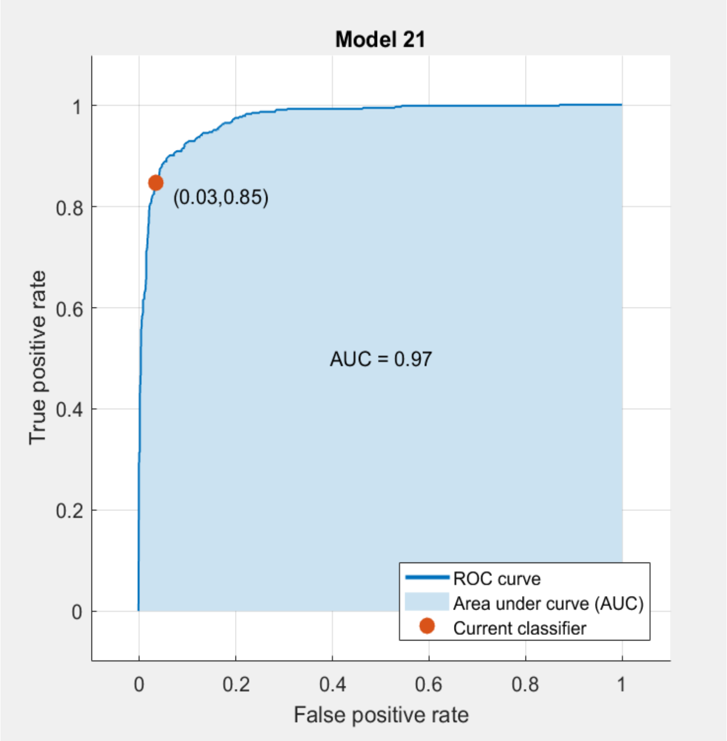


Fig. 8 Area Under ROC Curve for positive class 0

For positive class 0, AUC=0.97

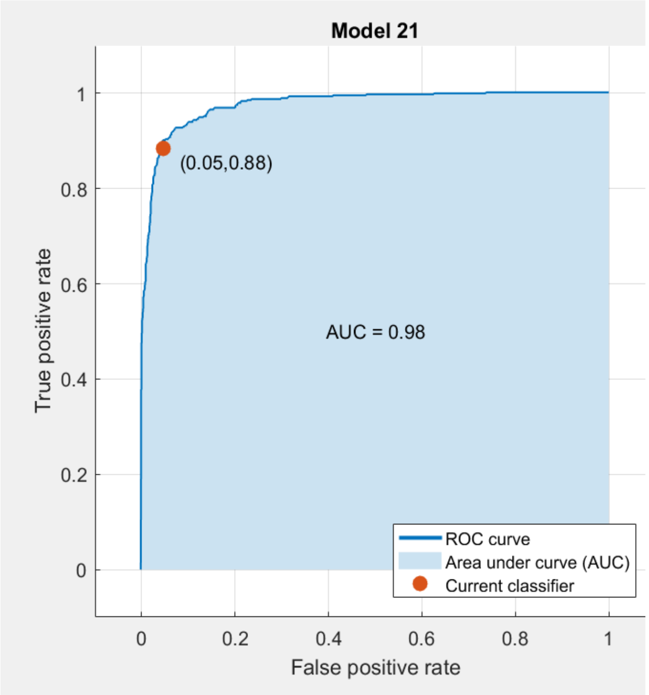


Fig. 9 Area under ROC curve for positive class 1

For positive class 1, AUC=0.98

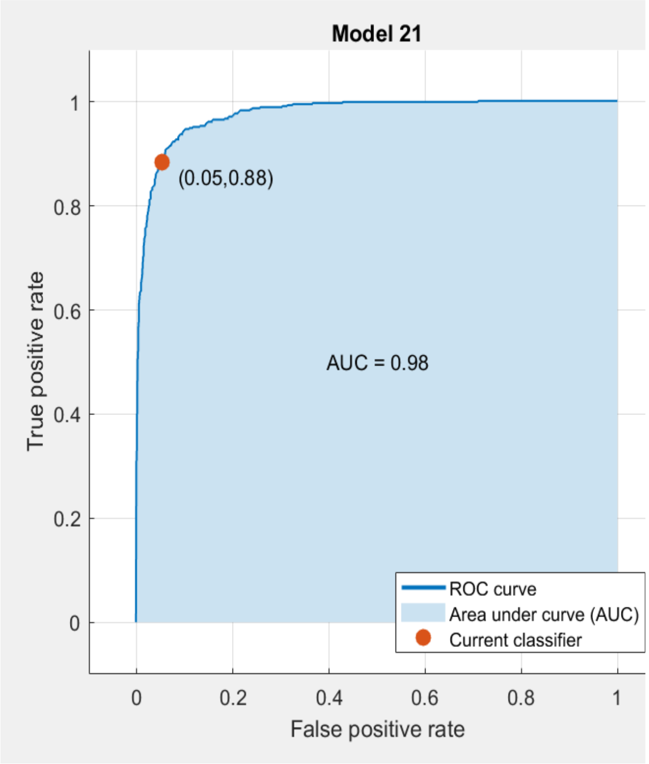


Fig. 10 Area under ROC curve for positive class 2

For positive class 2, AUC=0.98

Areas of the ROC curve for the first four classes of our model come well above .90 giving an excellent accuracy.

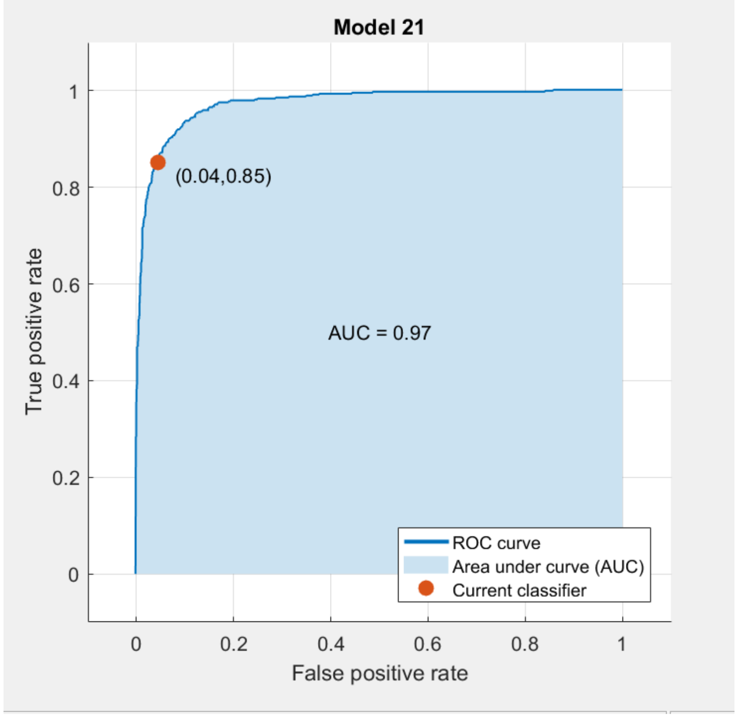


Fig. 11 Area under ROC curve for positive class 3

For positive class 3, AUC=0.97

VIII. Conclusion

SVM is a popular choice for handwritten character recognition for its high accuracy. It has been observed, almost always a SVM classifier outperform all other classifiers in handwritten character recognition problem [7]. We got a similar result in our project supporting our choice for SVM. We have used three techniques to analyze the performance of our trained model. Firstly, we have used confusion matrix to see the prediction rate for the first four class. The lowest true positive rate was 85% which is good enough since our overall accuracy was 86.7%.

Then we did a ROC analysis. Accuracy is measured by the area under the ROC curve (AUC). An area of

1 represents a perfect test; an area of .5 represents a worthless test.

All of the AUC of the first four classes came out to be .97, .98, .98, .97 respectively.

We have trained the model once at first using all the learning algorithms we obtain a preliminary idea as to what algorithm will suit best for our project and give the highest accuracy. All types of SVM (linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian, coarse Gaussian SVM) came out in the top, and among them cubic SVM had the highest accuracy. So we went ahead with cubic SVM and trained it multiple times, each time changing its PCA, kernel scale, box-constraint level and cross validation folds. Changing the technique to one vs all from one vs one, increased the accuracy of our classifier significantly.

After many iterations we choose the 21st model since it has the highest accuracy.

Limitations of this work includes absence of all the Bengali Characters and smaller dataset. But it can be improved and enhanced as well by increasing the number of training samples, adding more variables and better feature processing methods.

ACKNOWLEDGMENT

The voluntary contribution of the undergrad students of the Department of Electrical & Electronic Engineering, University of Dhaka made it possible to complete the paper.

REFERENCES

1. *stats.stackexchange.com/questions/23391/how-does-a-support-vector-machine-svm-work.*
2. www.isical.ac.in/~cvpr/
3. *https://www.mathworks.com/disccovery/image-thresholding.html*
4. *https://en.wikipedia.org/wiki/Feature_vector*
5. *www.hnk.ffzg.hr/bibl/iti2007/104%20Data%20Mining/104-06-121.pdf*
6. *Http://gim.unmc.edu/dxtests/roc3.htm*
7. *[https://link.springer.com/article/10.1007/s10032-009-0084-x].*