A Machine Learning Approach For Bangladeshi Banknote Classification

Mahamudul Hasan	170104112
Sajid Hasan	170104123
Rakesh Roshan Paul	170104130

Project Report

Course ID: CSE 4214

Course Name: Pattern Recognition Lab

Semester: Fall 2020

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September 2021

A Machine Learning Approach For Bangladeshi Banknote Classification

Submitted by

Mahamudul Hasan	**170104112**
Sajid Hasan	**170104123**
Rakesh Roshan Paul	**170104130**

Submitted To

Faisal Muhammad Shah, Associate Professor

Farzad Ahmed, Lecturer

Md. Tanvir Rouf Shawon, Lecturer

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September 2021

ABSTRACT

Automatic currency note identification depends inexorably on the currency note features of a particular nation, and feature extraction impacts the recognition capacity immediately. A very excellent use in banking systems, and other fields of trade, can be an automated bank note recognition system. It can also help those with vision impairment. Although bill money identification devices in Bangladesh are unusual, they are vastly utilized in other nations. In this paper, we have proposed a machine learning based recognition scheme for Bangladeshi banknote recognition. The recognition system takes preprocessed images of paper currency, and extracted features using Histogram of Oriented Gradients (HOG), and then fed into linear SVM model for recognition. The experimental result shows the performance of the proposed system on a challenging dataset that can recognize available Bangladeshi paper notes with an average of 95%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image recognition has grown increasingly important in recent years, in the area of pattern recognition. In real life, an image can contain anything. They can be a text on a document, a car license plate, an iris in the eyes of a person, a face of an individual and many more. Likewise paper currency recognition is an important area of image recognition, can be used at banks or other marketplaces effectively. By far, several techniques to solve the challenge of paper money identification and verification have been proposed. In recent yerars, ANN based pattern recognition techniques are widely applied to money recognition. For The automatic fake note inspection system, these pattern recognition based system can be the best comparator for examination of human eyesight. In our research, we have implemented a Bangladeshi Banknote Currency Detection System using traditional machine learning algorithms and have shown the comparsion of their performance evaluation. After observing the comparative results, the best machine learning model is selected which has overcome many newly emerged ANN based model's performance.

Chapter 2

Literature Reviews

The authors of [1] presented an invariant approach for extracting features not susceptible to change of scale, rotation and banknote translation. This approach reduces the variety of data and improves the reliability of banknote categorization. In addition, its computational complexity is small to fulfill real-time classification needs for banknotes. The computation is easy to conclude the processing and categorization of banknotes in real time. Then, the banknote face and value are predicted by a three-layer back-propagation network. There is a strategy in [2] that only uses a portion of the original data set that is collected using random mask based on neural network. The structure reduction method of this NN is effective for time series data and its Fourier power spectra. The recognition system in [3] collects scanned pictures of banknote samples scanned with low cost optoelectronic sensors and then transmitted into a backpropagation algorithm using multilayer perceptron. The authors tried to simplify the system to make it easy to implement in hardware. The authors of [4] introduced a technique for paper currency identification system that incorporates three currency properties, such as size, color and texture. This approach restricts the size of the currencies of various countries. The authors utilizes image histogram, plenitude of different colors in a paper currency were computed and also the Markov chain idea was used to represent the paper currency's texture. In both automated dispensing devices and in many automatic banking processes, newly introduced low-cost automation equipment [5] that can recognize banknotes are widely employed. An innovative approach of recognizing paper currencies utilizing the appropriate weights and neural network, using two classifiers, namely the weighted euclidean distance is proposed in [6]. This approach is focused on identifying and extracting certain paper currency characteristics and also involves different processing stages.Many factors, such as picture size, edge detection, the Euler number and coefficient of correlation play key roles in the identification process. An effective method is introduced in [7] for recognizing Euro currency using axis symmetrical mask and two image sensors. It is also crucial for the industrial application of this suggested approach to assess the actual euro currency in the commercial market. Thai currency recognition system was built using Neural network on Digital Signal Processors units is presented in [8]. The authors of [9] developed a device that can help visually impaired individuals to detect paper currency. The algorithm employs a variety of characteristics to categorize the banknotes and subjects have quickly gained their know-how and could use the banking identification system with confidence. This paper [10] overcome many of the currency image restrictions like old, dirty and torn banknotes using Euclidean distance of features between the input image and the template image, where the input images were firstly divided into RGB channels to extract HSV, edge, and grey-level co-occurrence matrix.

Chapter 3

Data Collection & Processing

The ’Bangla-Money-Dataset’ dataset was gathered from a git repository[11], where the images in this dataset have been captured by mobile device. This dataset contains a total of 1637 train and 333 test images of nine Bangladeshi banknote images, each of which is (120, 250, 3) pixels in size. ’1 taka’, ’2 taka’, ’5 taka’, ’10 taka’, ’50 taka’, ’100 taka’, ’500 taka’, and ’1000 taka’ photos are among the nine categories.

Following the data gathering procedure, we have used several data preprocessing techniques to turn the data into a machine-readable format, allowing the machine to identify the image with ease.

For image data preprocessing, we firstly used the normalization technique to modify the range of pixel intensity values across the entire training dataset. Its primary function is to convert an input image into a set of pixel values that are more recognizable to the human eye. After normalization, we have done histogram equalization which has highlighten the edge of the image so that the machine learning model can easily detect the edges.

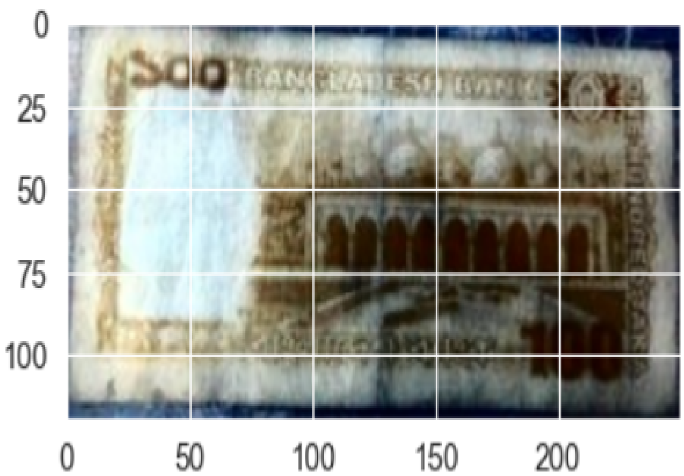

Figure 3.1: Preprocessed Image Data.

The size of dataset is very small which may cause the machine learning model not to classify well. So, to overcome this low numbered dataset problem, we have increased the size of our collected dataset using different augmentation technique which have been imported from Python's OpenCV library. We have applied the horizontal flip, vertical flip and keep them along with the original image. Hence, the dataset size become increase to 4911 images which was before around 1637. The distribution of our dataset has been stated below.

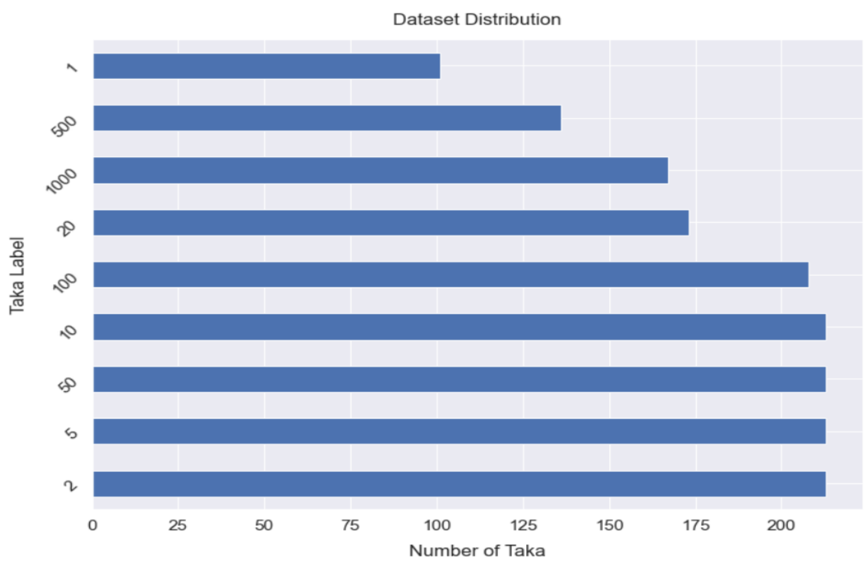

Figure 3.2: Dataset Distribution Before Augmentation

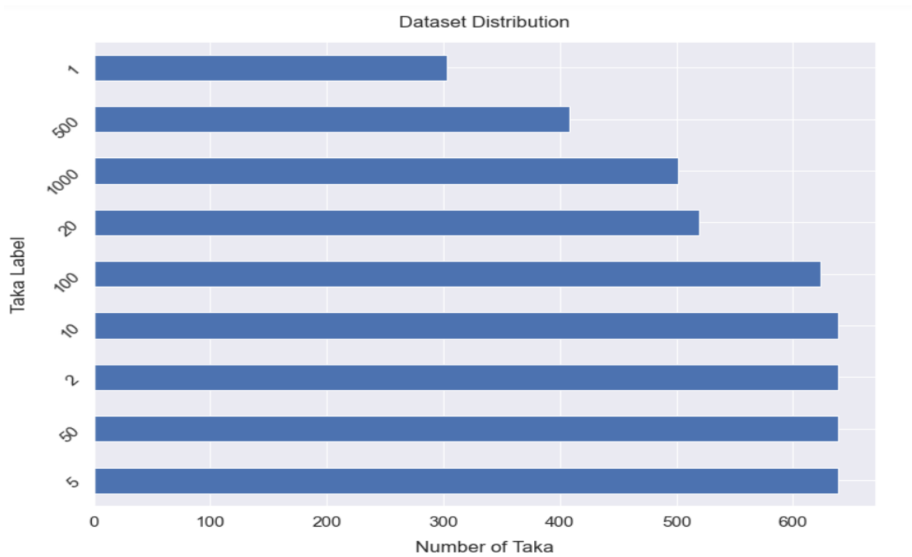

Figure 3.3: Dataset Distribution After Augmentation

Chapter 4

Methodology

In our proposed methodology, we have first collected the Bangladeshi Banknote images from 'Bangla-Money-Dataset[11]. We preprocessed the photos using the Normalization and Histogram Equalization methods after gathering the image data in order to turn the images into machine intelligible form. Then, to enhance the size of our training set, we flipped the original photographs vertically and horizontally under Data-Augmentation to increase the size of our data. We performed feature engineering on our preprocessed-augmented picture dataset after doing data augmentation. For feature extraction using handcrafted machine learning approaches, we have applied different descriptors on our images such as ORB (Oriented Rotational Brief), SIFT (Scale Invariant Feature Transform) and HOG (Histogram of Oriented Gradients).

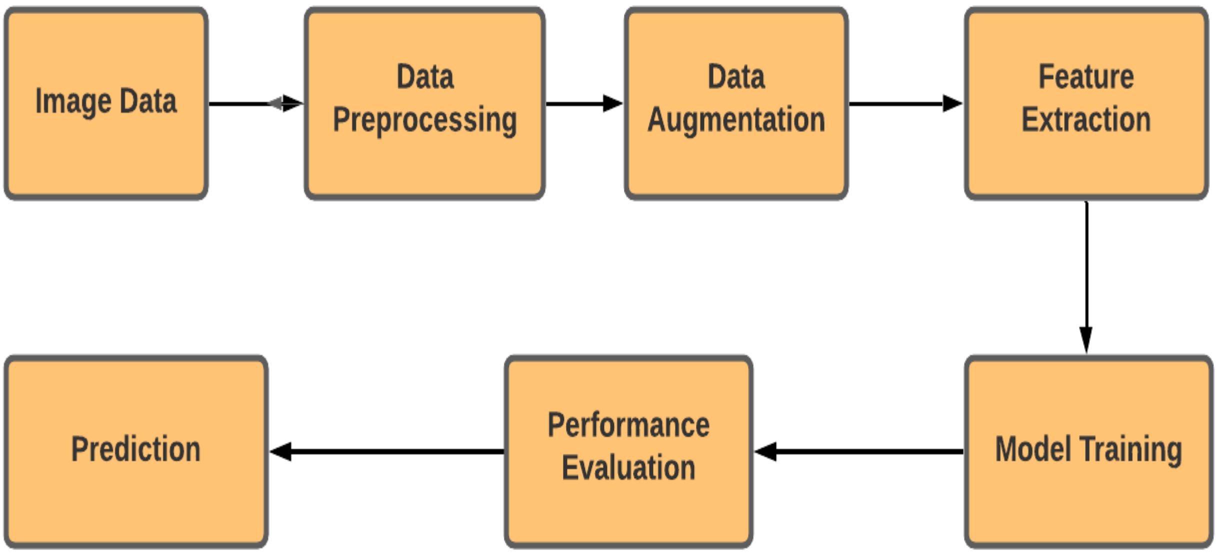

Figure 4.1: Flow diagram of the whole process.

After extracting the features using these descriptors, we have feed them to various machine learning model implemented on different classifiers to train these model. Finally, we evaluated the performance of these trained models on our test dataset.

4.1 Feature Extraction

Feature engineering is the most prominent part of our Bangla Currency Recognition System. The better our extracted features are, the better the model can classify the banknote images. The extraction of image features depend basically on the proper detection of image color, edges, background and foreground information. The better the image edges and exterior outlines are highlighted, the more features can be extracted.
Many algorithms are available in Python's OpenCV package for this feature extraction procedure in machine learning. These algorithms detect features and for each feature return a descriptor. Descriptors are used to convert picture features into computer language that is vector-based. There are many decriptors provided by OpenCV and among all ORB (Oriented Totational Brief), SIFT (Scale Invariant Feature Transform) and HOG (Histogram of Oriented Gradients) are well known in case of generating feature descriptor. Here, SIFT and ORB both descriptors are good for matching images. But on the other hand, HOG is good for object detection from an image. HOG is one of many traditional computer vision and machine learning techniques that is widely used for object detection. The 5 steps of the HOG Feature Descriptor are:

- Preprocessing (Color Normalization and Resizing).
- Computing the Gradients.
- Spatial / Orientation Binning (Dividing the image into cells).
- Block Normalization.
- Get the HOG Feature Vector.

4.1.1 Color Normalization and Resizing:

The original paper[12] mainly found that 64×128 is the ideal image size that has the ratio 1:2. Like 128×256 or 256×512. So, for better feature extraction, we have to resize our images in the dataset into 128×256 shape so that the images maintain 1:2 ratio for better features. The original paper[12] also found that both RGB and LAB color spaces perform identically. But using gray-scale images reduces performance. This means that HOG feature descriptor works best on colored images. And that's why we haven't converted our images into gray-scale images.

4.1.2 Computing the Gradients:

Computing the gradients of an image in computer vision reveals those locations where the pixel gradient intensities change. Thus, this leads to a lot of useful information.

In the research[12], the kernels used to calculate the gradients are:

Vertical gradient kernel: $[-1,0,1]$

Horizontal gradient kernel: $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

These kernels lead the main role in finding vertical edges and horizontal edges. After concatenating the vertical edges and horizontal edges, we get the full outer edge of objects in our image which is extensively used in time of image recognition.

4.1.3 Spatial / Orientation Binning and Calculating the Gradients:

The next step is dividing the image into 8×8 cells. Then we calculate the gradients for all the 8×8 cells. As this is an rgb image, hence each cell has 8x8x3 = 192 pixels. And the gradient of each cell has the magnitude and direction (2 values). So, each cell has 8x8x2 = 128 values as the gradient information. The gradients and directions are each 8×8 blocks containing numbers. They are represented using 9 orientation bins.

4.1.4 Block Normalization:

In the previous step, we divided the image into grids of 8×8 cells and calculate the gradients for each cell. According to the authors of the paper[12], gradient values can vary according to the lighting and foreground background contrast. So, to counter this issue, we can normalize the cells. In such cases, block normalization tends to perform better than single-cell normalization. We have grouped 9 cells together to make a block. We call this as 3×3 block normalization. The authors find that both, 2×2 block normalizations and 3×3 block normalization work well[12].

4.1.5 Getting the HOG Feature Vector:

The final step is obtaining the HOG feature vector. After we calculate all the block normalizations, we concatenate them into a single vector to get the final feature vector.They amount upto a total of 16740 dimension feature vector for each image.

4.2 Comparison of Classification Models

After getting the final feature vector for each image in our dataset, we have used a machine learning algorithms like Linear Support Vector Machine (SVM), Decision Tree, Logistic Regression, Naive-Bayes, kNN (k-Nearest Neighbors), RandomForest classifier and Stochastic Gradient Descent (SGD) to carry on image recognition task. To carry on the image recognition task, we first have to train these machine learning models by feeding them image descriptors extracted by HOG (Histogram of Oriented Gradients) image descriptor. After training these machine learning algorithms, we have done the performance evaluation test on our test dataset to evaluate our trained model on unseen data.

Chapter 5

Experiments and Results

Though accuracy is the most commonly used performance metrics, to get a better under-standing, we have used several other performance metrics: precision, recall, and F1 score for the evaluation of our proposed method. Table 5.1 shows the performance comparison of our used models. High precision relates to the low false-positive rate. For Linear SVM classifier the precision, recall, f1 and accuracy scores are 95%, 95%, 95% and 95% respectively. This model shows really good results on the unseen test data. For Logistic Regression classifier the precision, recall, f1 and accuracy scores are 95%, 95%, 95% and 94.71% respectively which also very good result. For SGD the precision, recall, f1 and accuracy scores are 94%, 94%, 94% and 93.5294% respectively. For Naive Bayes the precision, recall, f1 and accuracy scores are 82%, 80%, 80% and 80.2941% respectively. For kNN the precision, recall, f1 and accuracy scores are 91%, 91%, 90% and 90.5882% respectively. For Random Forest the precision, recall, f1 and accuracy scores are 82%, 77%, 77% and 78.5294% respectively. And lastly, for Decision Tree the precision, recall, f1 and accuracy scores are 44%, 44%, 43% and 44.1176% respectively where this is very poor result among all the classification model.

Table 5.1: Performance Comparison of the used models

Model	Precision	Recall	F1 Score	Accuracy
Linear SVM	95	95	95	95
Random Forest	82	77	77	78.5294
Naive-Bayes	82	80	80	80.2941
KNN	91	91	90	90.5882
SGD	94	94	94	93.5294
Logistic Regression	95	95	95	94.71
Decision Tree	44	44	43	44.1176

In our approach the Logistic Regression, kNN, Linear SVM, and SGD show very good accuracy above 90%. On the other hand, Naive Bayes and Random Forest show moderate accuracy 78% and 80% respectively. But the Decision tree model show very poor result which is 44%.

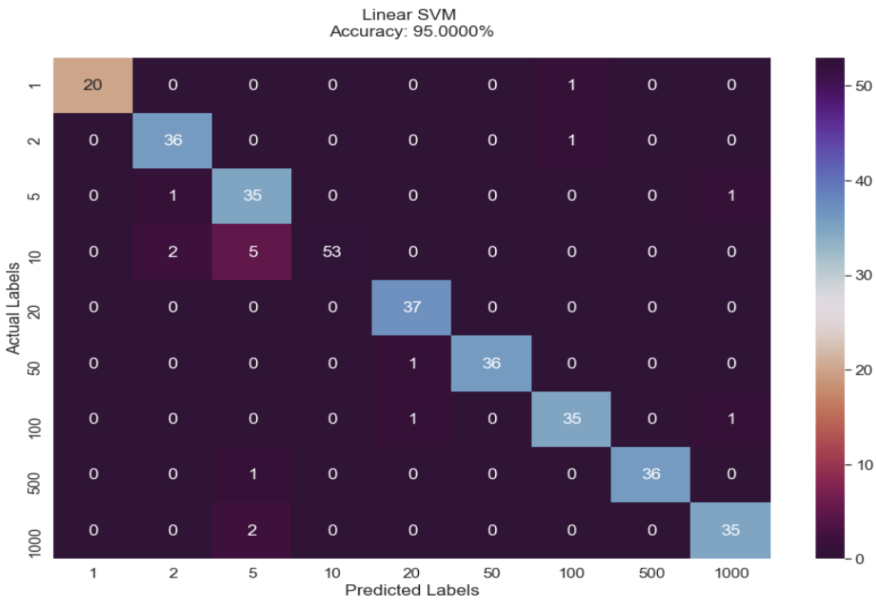

Figure 5.1: Confusion Matrix of SVM.

<-----------SVM------------->

	precision	recall	f1-score	support
1 Taka	1.00	0.95	0.98	21
2 Taka	0.92	0.97	0.95	37
5 Taka	0.81	0.95	0.88	37
10 Taka	1.00	0.88	0.94	60
20 Taka	0.95	1.00	0.97	37
50 Taka	1.00	0.97	0.99	37
100 Taka	0.95	0.95	0.95	37
500 Taka	1.00	0.97	0.99	37
1000 Taka	0.95	0.95	0.95	37
accuracy			0.95	340
macro avg	0.95	0.95	0.95	340
weighted avg	0.95	0.95	0.95	340

Figure 5.2: Precision,Recall,F1 Score of SVM.

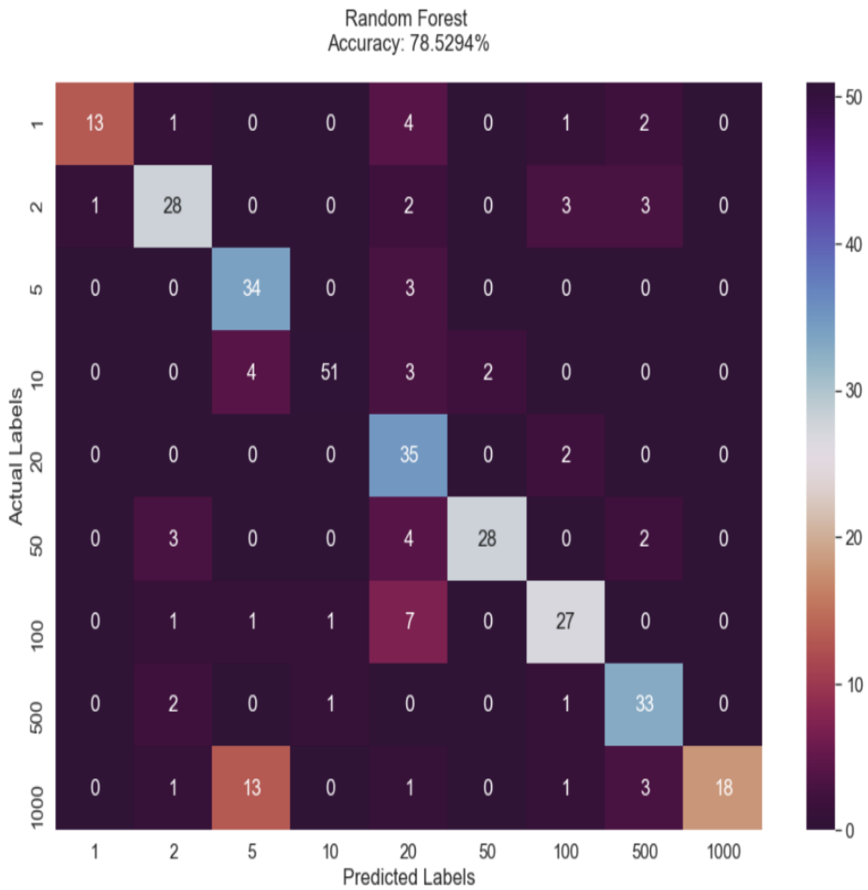

Figure 5.3: Confusion Matrix of Random Forest.

<------------RandomForest------------->

	precision	recall	f1-score	support
1 Taka	0.93	0.62	0.74	21
2 Taka	0.78	0.76	0.77	37
5 Taka	0.65	0.92	0.76	37
10 Taka	0.96	0.85	0.90	60
20 Taka	0.59	0.95	0.73	37
50 Taka	0.93	0.76	0.84	37
100 Taka	0.77	0.73	0.75	37
500 Taka	0.77	0.89	0.82	37
1000 Taka	1.00	0.49	0.65	37
accuracy			0.79	340
macro avg	0.82	0.77	0.77	340
weighted avg	0.83	0.79	0.78	340

Figure 5.4: Precision,Recall,F1 Score of Random Forest.

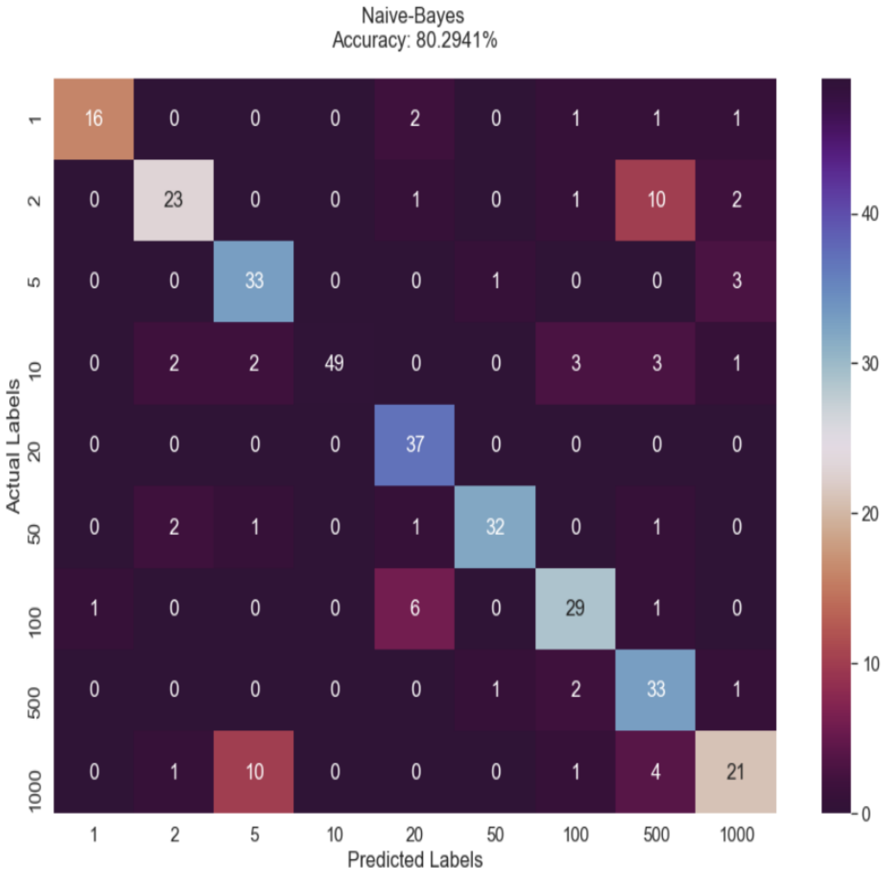

Figure 5.5: Confusion Matrix of Naive Bayes.

```
                <-----------Naive-Bayes------------>

              precision    recall  f1-score   support

      1 Taka       0.94      0.76      0.84        21
      2 Taka       0.82      0.62      0.71        37
      5 Taka       0.72      0.89      0.80        37
     10 Taka       1.00      0.82      0.90        60
     20 Taka       0.79      1.00      0.88        37
     50 Taka       0.94      0.86      0.90        37
    100 Taka       0.78      0.78      0.78        37
    500 Taka       0.62      0.89      0.73        37
   1000 Taka       0.72      0.57      0.64        37

    accuracy                           0.80       340
   macro avg       0.82      0.80      0.80       340
weighted avg       0.82      0.80      0.80       340
```

Figure 5.6: Precision,Recall,F1 Score of Naive Bayes.

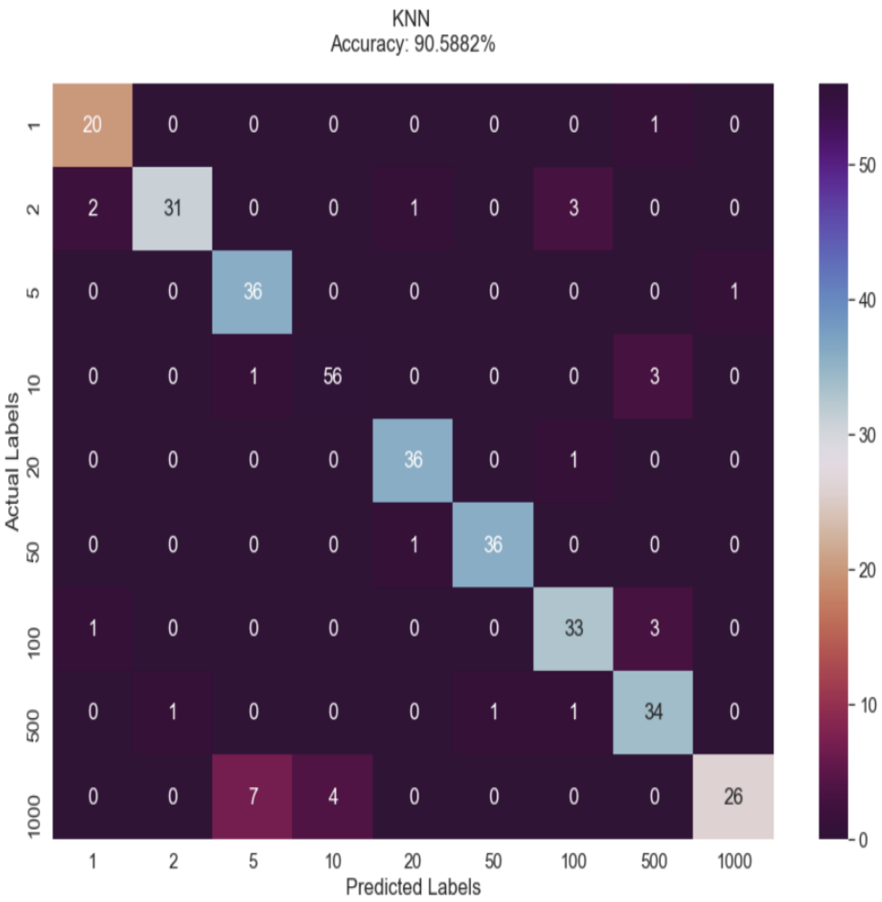

Figure 5.7: Confusion Matrix of KNN.

<------------KNN------------->

	precision	recall	f1-score	support
1 Taka	0.87	0.95	0.91	21
2 Taka	0.97	0.84	0.90	37
5 Taka	0.82	0.97	0.89	37
10 Taka	0.93	0.93	0.93	60
20 Taka	0.95	0.97	0.96	37
50 Taka	0.97	0.97	0.97	37
100 Taka	0.87	0.89	0.88	37
500 Taka	0.83	0.92	0.87	37
1000 Taka	0.96	0.70	0.81	37
accuracy			0.91	340
macro avg	0.91	0.91	0.90	340
weighted avg	0.91	0.91	0.90	340

Figure 5.8: Precision,Recall,F1 Score of KNN.

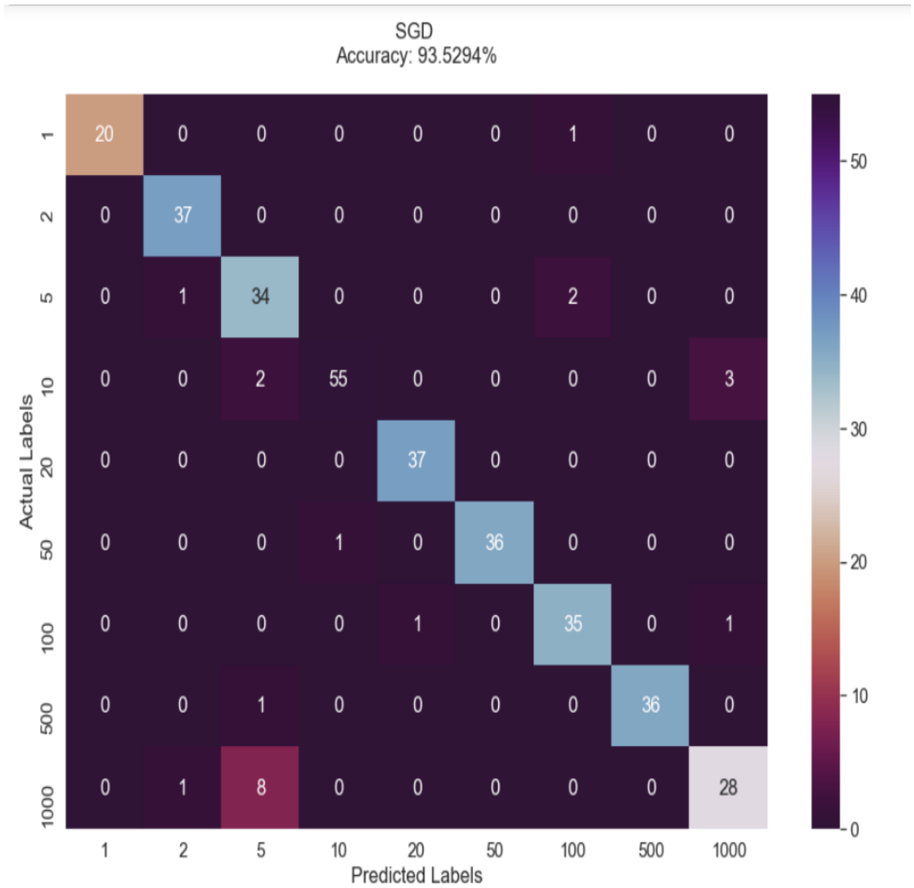

Figure 5.9: Confusion Matrix of SGD.

<-----------SGD------------>

	precision	recall	f1-score	support
1 Taka	1.00	0.95	0.98	21
2 Taka	0.95	1.00	0.97	37
5 Taka	0.76	0.92	0.83	37
10 Taka	0.98	0.92	0.95	60
20 Taka	0.97	1.00	0.99	37
50 Taka	1.00	0.97	0.99	37
100 Taka	0.92	0.95	0.93	37
500 Taka	1.00	0.97	0.99	37
1000 Taka	0.88	0.76	0.81	37
accuracy			0.94	340
macro avg	0.94	0.94	0.94	340
weighted avg	0.94	0.94	0.94	340

Figure 5.10: Precision,Recall,F1 Score of SGD.

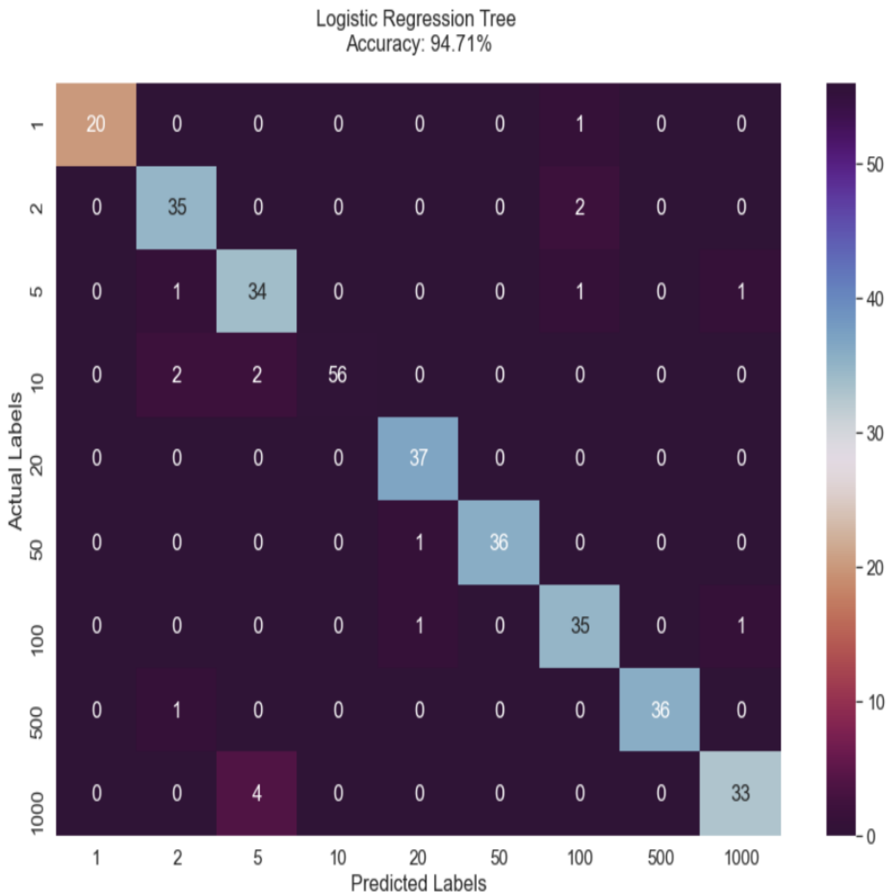

Figure 5.11: Confusion Matrix of Logistic Regression.

<----------Logistic Regression------------>

	precision	recall	f1-score	support
1 Taka	1.00	0.95	0.98	21
2 Taka	0.90	0.95	0.92	37
5 Taka	0.85	0.92	0.88	37
10 Taka	1.00	0.93	0.97	60
20 Taka	0.95	1.00	0.97	37
50 Taka	1.00	0.97	0.99	37
100 Taka	0.90	0.95	0.92	37
500 Taka	1.00	0.97	0.99	37
1000 Taka	0.94	0.89	0.92	37
accuracy			0.95	340
macro avg	0.95	0.95	0.95	340
weighted avg	0.95	0.95	0.95	340

Figure 5.12: Precision,Recall,F1 Score of Logistic Regression.

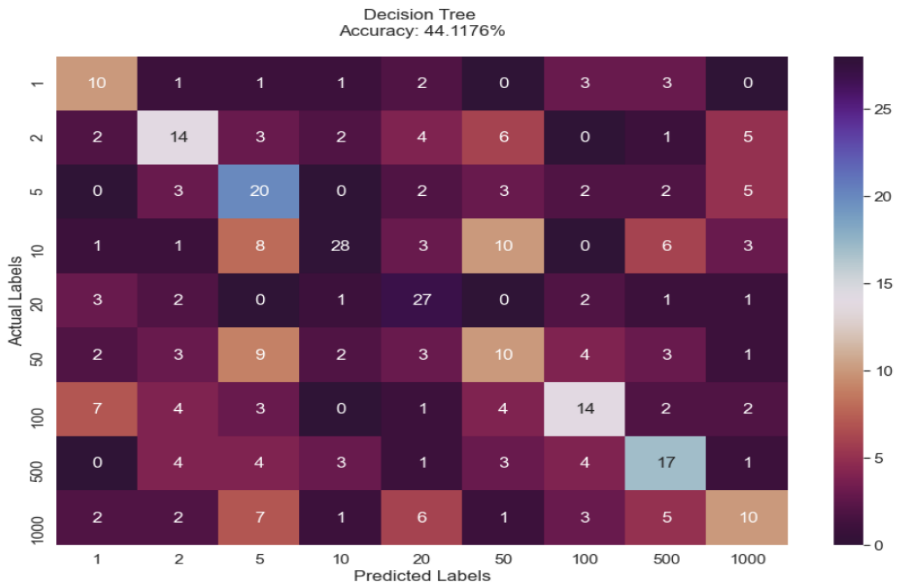

Figure 5.13: Confusion Matrix of Decision Tree.

```
                 <----------Decision Tree------------>

               precision    recall  f1-score   support

     1 Taka       0.37      0.48      0.42        21
     2 Taka       0.41      0.38      0.39        37
     5 Taka       0.36      0.54      0.43        37
    10 Taka       0.74      0.47      0.57        60
    20 Taka       0.55      0.73      0.63        37
    50 Taka       0.27      0.27      0.27        37
   100 Taka       0.44      0.38      0.41        37
   500 Taka       0.42      0.46      0.44        37
  1000 Taka       0.36      0.27      0.31        37

   accuracy                           0.44       340
  macro avg       0.44      0.44      0.43       340
weighted avg      0.46      0.44      0.44       340
```

Figure 5.14: Precision,Recall,F1 Score of Decision Tree.

Note Type	Predicted Testing Samples	Actual Testing Samples	Recognition Accuracy
1 Taka	20	21	95.24
2 Taka	37	37	100
5 Taka	35	37	94.59
10 Taka	36	37	97.3
20 Taka	36	37	97.3
50 Taka	36	37	97.3
100 Taka	35	37	94.59
500 Taka	35	37	94.59
1000 Taka	53	60	88.33

Table 5.2: Prediction Comparison of the Taka Samples over Linear SVM Model

5.1 Samples

Figure 5.15: Examples of Our Predicted Samples.

Chapter 6

Future Work and Conclusion

Automatic recognizing devices for banknotes in Bangladesh might open up a new possibility. Automation in hardware devices can help visually impaired individuals to detect paper currency. We have proposed a machine learning based model for recognizing Bangladeshi banknotes. The technique used preprocessing of the images using normalization and histogram equalization. Features were extracted using different descriptors like HOG and then fed into linear Support Vector Machine (SVM). A challenging dataset with diverse circumstances was utilized to test the performance of our proposed model. It produces effective outcomes for various types of paper currencies of Bangladesh. We have employed different efficient image processing methods and algorithms to produce precise and trustworthy results and found a very attractive accuracy of 95%. Our future goal is to develop a deep learning based model and to build a low-cost banknote reader machine to help the visually impaired people.

References

[1] D. Gunaratna, N. Kodikara, and H. Premaratne, "Ann based currency recognition system using compressed gray scale and application for sri lankan currency notes-slcrec," *Proceedings of World academy of sciecne, engineering and technology*, vol. 35, pp. 235–240, 2008.

[2] F. Takeda and S. Omatu, "High speed paper currency recognition by neural networks," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 73–77, 1995.

[3] N. Jahangir and A. R. Chowdhury, "Bangladeshi banknote recognition by neural network with axis symmetrical masks," in *2007 10th international conference on computer and information technology*, pp. 1–5, IEEE, 2007.

[4] H. Hassanpour, A. Yaseri, and G. Ardeshiri, "Feature extraction for paper currency recognition," in *2007 9th International Symposium on Signal Processing and Its Applications*, pp. 1–4, IEEE, 2007.

[5] A. Frosini, M. Gori, and P. Priami, "A neural network-based model for paper currency recognition and verification," *IEEE transactions on neural networks*, vol. 7, no. 6, pp. 1482–1490, 1996.

[6] E. Althafiri, M. Sarfraz, and M. Alfarras, "Bahraini paper currency recognition," *Journal of Advanced Computer Science and Technology Research*, vol. 2, no. 2, pp. 104–115, 2012.

[7] F. Takeda and T. Nishikage, "Multiple kinds of paper currency recognition using neural network and application for euro currency," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 2, pp. 143–147, IEEE, 2000.

[8] F. Takeda, L. Sakoobunthu, and H. Satou, "Thai banknote recognition using neural network and continues learning by dsp unit," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 1169–1177, Springer, 2003.

[9] Z. Solymár, A. Stubendek, M. Radványi, and K. Karacs, "Banknote recognition for visually impaired," in *2011 20th European Conference on Circuit Theory and Design (ECCTD)*, pp. 841–844, IEEE, 2011.

[10] J. Akter, M. K. Hossen, and M. S. A. Chowdhury, "Bangladeshi paper currency recognition system using supervised learning," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, pp. 1–4, IEEE, 2018.

[11] "Github bangla money dataset." https://github.com/nsojib/Bangla-Money-Dataset.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This project report was generated on Wednesday 29th September, 2021 at 4:20am.