2024 IEEE 9th International Conference for Convergence in Technology (I2CT)
Pune, India. Apr 5-7, 2024

# ALPR: ResNet50 powered Bangla License Plate Detectionand OCR by Root Mean Square Propagation Optimizer and Linear SVM Classifier

Abdulla Nasir Chowdhury
*Department of Electrical and Electronics EngineeringLeading University*
Sylhet, Bangladesh
abdullahnasirchowdhury1@gmail.com

Samya Pal Summit
*Department of Information Technology Jahangirnagar University*
Dhaka, Bangladesh
summitpal3@gmail.com

Md. Fuad Ahmed Laskar
*Department of Electrical and Electronics Engineering London South Bank University*
London, Englandfuadlaskar46@gmail.com

Gulam Mahfuz Chowdhury
*Department of Electrical and Electronics Engineering Leading University*Sylhet, Bangladeshgulammahfuz@iut-dhaka.edu

Ishmam Ahmed Chowdhury
*Department of Electrical and Electronics Engineering Leading University*
Sylhet, Bangladesh
ishmamahmed_eee@lus.ac.bd

Mehedi Hasan
*Department of Computer Science and Engineering Daffodil International University*
Dhaka, Bangladesh
mehedi15-3972@diu.edu.bd

2024 IEEE 9th International Conference for Convergence in Technology (I2CT) | 979-8-3503-9447-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/I2CT61223.2024.10543675

***Abstract*—This paper implements the MATLAB Image Processing Toolbox in detecting the license plate region using several user-defined functions in order to pre-process and process the image up until the point of extraction of characters. The extracted characters were then classified by utilizingResNet50 from the Deep Learning Toolbox of MATLAB, custom training it on above a thousand images of Bangla and English characters and numbers alongside possible categories of noise extracted from the ROI after processing the image which resulted in a datastore of 103 total categories. The output is converted to a string and saved in an excel sheet to be accessed later on. In this ALPR model, the model will scan through the images of vehiclesfrom a folder in a destination specified by the code to identify the license plates and characters and perform necessary actions on them. The aim of this paper is to properly implement the Image Processing Toolbox by MATLAB in order to identify the Region of Interest and study the performance of the Linear SVM(Support Vector Machine) classifier with ResNet50 when it comes to Bangla OCR. The training and validation accuracy achieved by using the Root Mean Square Optimizer was 97.57%. The final accuracies and precision achieved while testing the model on 50% of the image dataset was 99.2%. Moreover, theER (Error Rate) and FPR (False Positive Rate)were limited within 0.02%. The model scored 100% on F1 scores and Matthews Correlation Coefficient for every category of image classified.**

***Keywords—ALPR,ResNet50,MATLABImage Processing Toolbox,Deep Learning Toolbox, ROI, convolutional neural network, Linear SVM Classifier, ER, FPR,F1 scores, Matthews Correlation Coefficient***

## I. INTRODUCTION

During this AI revolution, despite several research models being proposed, none of the research papers on ALPR has been properly implementable for the number plates or the license plates in Bangladesh. There are a couple of reasons behind the lack of a robust and implementable model for ALPR. The first point to note down is the inexistence of a large enough dataset for training a model on. Today Machine Learning refers mostly to Transfer Learning as building a data frame from scratch would consume a large amount of time, require tremendous effort and be quite heavy on the wallet. This is tedious and leads to lower accuracy. However, the scarcity of a Bangla characters from license plates dataset on Kaggle strikes out the possibility of implementing transfer learning in training a model and this unavailability of a dataset contributes heavily to the lack of an implementable model. The output of a machine learning model is heavily dependent on the dataset. Secondly, the lack of a robust structure to implement the model can be contributed to a deficiency of knowledge in video processing and image processing sections along with the lack of understanding on implementable machine learning algorithms. Some papers even deal with implementing a machine learning algorithm from scratch which is counter-productive in this age of revolutionary AI. Another common scenario is the use of template matching using a predefined function for identifying the correlation coefficient between the previously saved characters and the system outputted characters. This does not always lead to anaccurateconclusion as the Bangla number for 4 is similar to the English number for 8. That makes it worth mentioning that the license plates in Bangladesh are not always in Bangla. There may even be several font styles as well. Moreover, the characters are often scratched out so identifying them using the correlation coefficient method for template matching can be over-ruled. The more superior approach would be to make use of a traditional convolutional neural network trained on a custom dataset implementing an algorithm which uses supervised learning. This paper approaches this problem using the Resnet50 architecture which is a residual network trained on a thousand categories of images from the ImageNet dataset.

## II. LITERATURE REVIEW

In [1] a comparative study on the architecture of frameworks VGG16, VGG19 and ResNet50 for Image Classification provides an insight on the competitive performance of ResNet50. The conclusion drawn here is that ResNet50 provides the highest accuracy of the three, with 97.33% at 20 epochs whereas the other two provided 96.67% and 97.07% respectively. In [2], a performance study is conducted on VGG16, ResNet-50 and GoogleNet Deep Learning Architectures to classify breathing sounds. Here, however, GoogleNet outperformed ResNet-50 and VGG16 but only by a miniscule margin. The results of classification were 62.50%, 62.29% and 63.39%. In [3], another comparative study on the effectiveness of ResNet50, EfficeientNetB7, InceptionV3 and VGG16 models in the classification of crops and weeds was done which outlines ResNet50 as the runner-up with EfficientNetB7 as the best performer. Based on the evidence provided, ResNet50 is a fine choice for Image Classification. In [4], the efficiency of ResNet50 is again proven by the architecture's application in MPR or (Modulation Pattern Recognition) when competing against VGG16, VGG19, Inception and Xception with the accuracy of recognition improved to 85 percent.

In [5], the ResNet50 Architecture is implemented on Pest Detection System, which requires high scrutiny to detail. In [6], this architecture has been implemented for the detection of brain tumor. The results according to [6] were promising as can be seen from the F1-score, accuracy, precision and overall efficiency. In [7], it has been implemented for the detection of CMF or Copy-Move Forgery. This is another application which begs for a high level of accuracy. Here, in order to assess the performance and efficiency of the algorithm, accuracy and logarithmic loss (LogLoss) were employed and the results evaluated on the CoMoFoD (Copy Move Forgery Dataset) states that the algorithm achieves better accuracy with a reduction of the LogLoss function at 100 epochs. In [8], the implementation of ResNet50 is observed in the recognition of food and the results claimed that it showed a 99.87% training accuracy and 96.67% overall testing accuracy which is commendable.

The results drawn from [1] till [8] inspired the use of ResNet50 in this field of application as this architecture has proven itself more than worthy for its shot at Bangla OCR.

In [9], a survey on methods and techniques conducted by J. Shashirangana for automated license plate recognition is done where the team proposes that the techniques are object detection, image processing and pattern recognition. This system however implements template matching which is also seen in [12]. No matter how high the accuracy might be on custom datasets of template, it will never surpass neural networks as template matching is obsolete and impractical since it requires templates on every possible font of alpha-numerical characters which is impossible. In [10], Y.Y. Lee proposes the detection of License Plate using a CNN architecture named YOLO v3, which stands for You Only Look Once. The architecture performs very well according to the results depicted in the paper. The paper provides valuable insight in the use of DOE for detecting the license plate region and improving the model performance. In [11], Filipino License Plate Characters were recognized using Faster R-CNN with Inception V2. However, the rate of detection was only 90.011% with a recognition rate of 93.21% and an overall accuracy of 83.895%. In the comparisons made in [3] and [4], it was seen that ResNet50 beats Inception V3 easily, so it's implementation in this field would be more valid.

In [12], the application of template matching for the recognition of Bangla characters provides high accuracy with the implementation of Morphological Image Processing. However, the paper explains that the implementation of image processing is limited to the use of the sobel operator for edge detection which does not perform well in several cases. Moreover, template matching relies highly on the correlation co-efficient so the Bangla number for 4 which has a very high co-relation to the English number for 8 would provide error to the system. This method therefore, is not commendable in the field of Bangla OCR. In [13] an automatic bangla license plate recognition system for low resolution images is proposed. This paper addresses a very important detail that has to do with the work done on this field. According to this paper, previous works in this field has more to do with images that are nearer to the sensor and this paper has provided great insight in reducing the peak signal to noise ratio (PSNR) for the reconstruction of 200 test samples. The accuracy achieved was 78% and 91% for OCR on the reconstructed and ground truth images. In [14], S. Abdullah and his team had implemented YOLO v3 architecture alongside ResNet20. Their model achieves more than 85% IoU (Intersection over Union) in digit recognition. The ResNet-20-based CNN achieves an accuracy of 92.7 in recognizing Bangla characters that were present in the license plate. In [15], M.M. Sarif implemented YOLO v3 and achieved an accuracy of 97.5%. In [16], a team led by Mahmudol H. Tusar provided extra-ordinary results in Real Time Bangla License Plate Recognition implementing YOLO-v5 algorithm along with Easy OCR. Their model achieved a praiseworthy final recognition rate of 98%. In [17], the cutting edge YOLOv7 architecture, released this year, was implemented by the team led by S.U. Ahmed with an improved OCR Engine. Their model claimed a lower rate of accuracy however, which was 96% in detecting the license plate regions and 97% in the accuracy of recognition of Bangla characters unlike the paper in [16]. In [18], Bangla OCR utilizing transfer learning although not explicitly stated has been implemented and tested on three models, Inception V3, VGG16 and Vision Transformer with prediction accuracies of 98.65%, 97.82% and 96.88% respectively but not implemented for ResNet50. So far, ResNet50 has not been implemented yet in the field of Bangla OCR. In [19], K. Roy and his team decided to take an analytical approach with the goal for enhancing the automatic detection and recognition of skewed Bangla License Plates and they introduced a novel pipeline that combines deep learning and their own algorithm that transforms images of both skewed and normal plates into formats best suited for ALPR systems. In [20], an automated approach for the recognition of Bangla License Plates is proposed. This was done by implementing a YOLO model but achieved only 81% accuracy. This is impractical as much higher accuracies with YOLO models v3, v5 and v7 have been achieved in papers from [14] till [17].

## III. METHODOLOGY

The foundational step that contributes to the determination of license plate characters successfully has to do with proper implementation of the image processing functions in order to identify the license plate region.A model should be able to precisely read an image of a vehicle that is over speeding or overtaking which could lead to a blurred image, an instance of a vehicle at a poll that might lead to a crisp image or a dark and over-shadowed one or from amonochrome closed-circuit television operating under dim light conditions that would result in a grayscale image. This means that the model should be able to successfully identify and crop out the region of interest, the license plate region. In order to tackle these circumstances, several user defined functions utilizing the MATLAB Image Processing Toolbox assisted in the successful extraction of the license plate region with very high accuracy. The next step would be to identify the extracted characters of the plate. The characters are alphanumerical either be in Bangla or in English. This requires usage of either a pre-trained OCR such as the open-source library pytesseract module for easy Optical Character Recognition or the use of a pre-trained Convolutional Neural Network with a custom dataset such as the YOLO (You Only Look Once) model and others. For this paper, the residual network ResNet50 has been chosen. This is available within the Deep Learning Toolbox of MATLAB. The final step would be to evaluate whether or not the output is comparable with the characters of a standard license plate without exception. This is done by a simple comparison between the list of characters outputted with a few user defined templates.

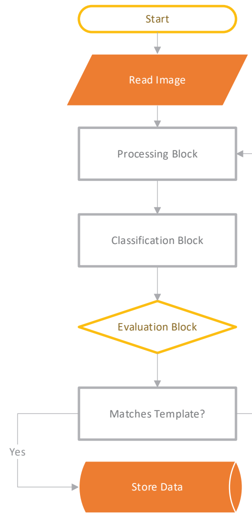



Fig. 1. Flowchart of proposed model

An additional step has to deal with writing the datafrom the extracted text and converting the extracted characters according to their respective labels into an excel file. The excel file will contain the number plate data which will be available to be accessed later on by either a program or a user.

### A. *Image Processing*

There are 3major steps in this section namely: Pre-processing, Processing and Character and Noise Extraction. These are briefly described:

#### 1) *Pre-Processing:*

In this section there are two fundamental tasks i.e., Routine Tasks and Performance Enhancement Tasks. They are explained below:

Routine Tasks:

These steps will always be performed on the image.

- Automated Image Reading: The images are read into the code from a source folder, using a user defined function that iterates once over all the files until all have been processed.
- Scaling: The image is scaled to a default size which is implemented in order to prevent diversity while performing processing operations.
- Conversion: The image is converted from an RGB to a two-dimensional binary imagein order to detect the edges using one of the four operators: Sobel, Canny, Roberts or Prewitt.
- Saving: The converted image is saved in a folder specified for Binary images.

Performance Enhancement Tasks:

These steps will be performed if the output through the model fails the Evaluation Block.

- Image Sharpening Function: This is a user defined function which is employed when the images read into the code fail to pass the evaluation block after passing through the classification block. This is a back-up function which takes into account the routine image preprocessing tasks after sharpening the image.
- Image Denoising Function: In case, the string output fails to pass the evaluation phase a second time, this block is activated. The noise in an image might cause the output to be faulty. The noise is reduced using median filter and average filter interchangeably and then sent to the Image Deblurring function or Image Sharpening function after which the routine preprocessing and processing tasks are carried on.
- Image DeblurringFunction: This is another backup function which is activated if the output fails the evaluation block a fourth time. There are threeuser-defined image deblurring functions that have been employed within this code block, one of which uses Wiener Deconvolution Technique that takes the read image, filter on the image and variance as input parameters. The other makes use of the Blind Deconvolution Technique that takes the same input parameters. The third and final function makes use of the Fourier Transform of the blurred image and that of the noise extracted from the blurred image. In this function, Fourier Transform is applied on the blurred image, then zero padding is done on a filter to adjust its size and make it equal to that of the blurred image. The Inverse Fourier Transform of the ratio between the Fourier Transform of the blurred image to the image noise is taken. The code adjusts the filter using a conditional loop to get the desired output.

The processes here will be used interchangeably to get the desired outcome and pass the evaluation block.

#### 2) *Processing:*

This section deals with image manipulation and can be divided into 2 steps.

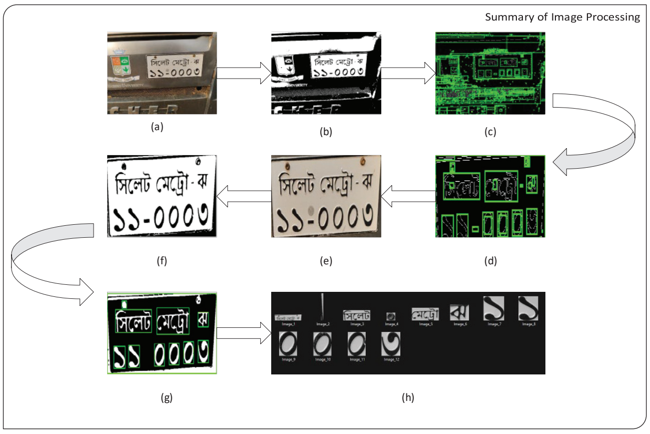



Fig. 2. Flowchart of Image Processing Block: a. Read Image; b. Apply Routine Tasks and Performance Enhancing Tasks; c. Implement Bounding Boxes on Connected Components; d. Detect ROI; e. Crop-out Original Image of ROI; f. Binarize; g. Invert; h. Crop-out CC regions from original image and save them in a folder

- Edge Detection: This is done using the 4 operators namely Sobel, Prewitt, Canny and Roberts for edge detection available within the MATLAB Image Processing Toolbox. But as all 4 can not be implemented at a single instance of the processing section, they are divided into blocks for performing the task separately if the model fails to identify the number plate region or extract the characters.

- Region of Interest: The region of interest in this case isdetermined by three algorithms, the first of which is the determination of bounding box with characters in it, that means, the one with the highest number of connected components which could either be the number plate or something unexpected. To make sure that the image is deficit of noise the bounding box with at most 18 connected components is selected, separated by the default pixel length of the characters in a license plate region which was determined through several trials and errors.

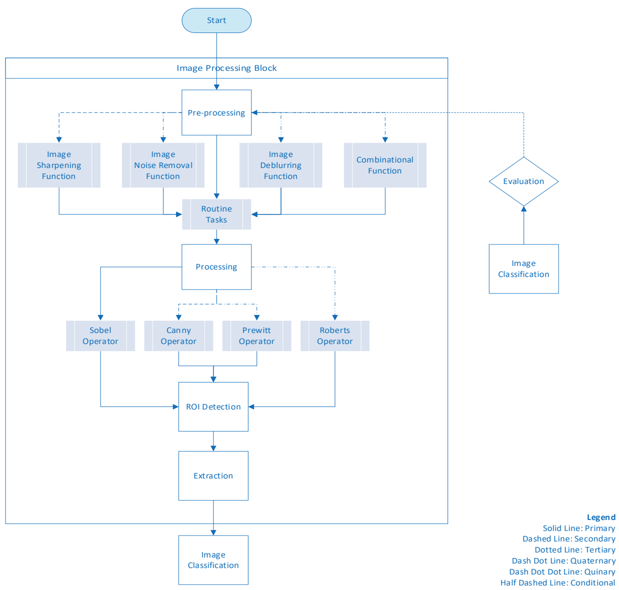



Fig. 3. Flow Chart for Image Processing Block

3. Character and Noise Extraction: Once the region of interest is determined the characters are extracted utilizing a conditional loop for extracting each of the connected components which might include noise and later saved in a folder specified for the noise and characters which will be explored by the code block for character classification.

## *B. Character Classification:*

The character classification takes place inthree phases: The Training and Testing phase, The Image Classification phase and the Evaluation phase.

Training and Testing Phase: An image data store is created for the network with 1438 categories of images within 103 folders. The net is first trained on this dataset splitting them into testing and training data. The ResNet50 CNN takes 224x224 3-channel images as input, so image preprocessing is done to convert the images into RGB. In many cases, the source image is overwritten while processing accidentally. To avoid such problems, the augmented Image Datastore function is used to convert the image size into the size suitable for the input layer. The dataset is divided into augmented image training set and augmented image testing set accordingly. The training set is fixed to 70% of the features and the testing set is the remaining 30%. The activations function is used to extract the training features of the first convolutional layer with input parameters set to the augmented training set and a minimum batch size set to 32 in order to ensure that the CNN and image data fit into memory. This function defaults to using the GPU. The output from this is saved as columns. This sort of output speeds up the procedure for multi class classification problems.

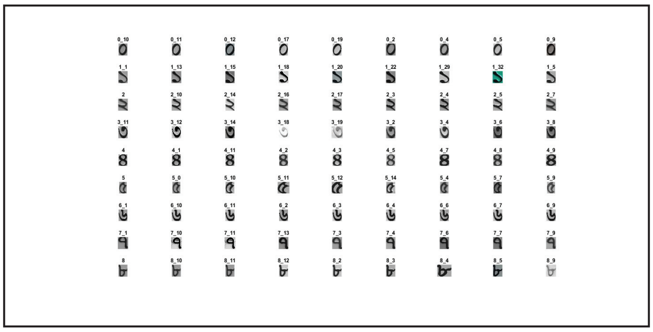


Fig. 4. Sample Training Dataset

In this case, 4 neurons have been attributed to the layer width. The layers initialized include the Image Input Layer, which reads images of size 224 x 224 x 3, the two dimensional convolution layer that takes input according to the layer width and provides padding where necessary, the batch normalization layer to reduce the training time of the CNN significantly and make it more stable through normalization of the inputs byre-centering and re-scaling them accordingly, the rectified linear unit layer which effectively speeds up the training process by mapping values that are negative to zero and accurately maintaining the positive values, the

Addition Layer to perform element-wise addition on the network layers, the two dimensional Average Pooling Layer used in order to simplify the output through nonlinear down-sampling thereby reducing the number of parameters that need to be learnt by the network, the Fully Connected Layer or the linear layer that connects the input neurons to the output neurons, the Softmax Layer to output the distribution of probability over the provided classes and the Classification Layer to classify the images accordingly. In this case, a linear SVM (Support Vector Machines) classifier was initialized for classification of the images. The layer graph function of MATLAB Deep Learning Toolbox was used to extract the layer graph of this DAGNetwork or the Directed Acyclic Graph network. For training this network, the parameters were set to using the RMSPropor Root Mean Square Propagationoptimizer over the SGDM or Stochastic Gradient Descent Mean and ADAM orAdaptive Mean optimizers as this optimizer provided higher validation accuracy as can be observed in the results section. The root mean square propagation optimizer allowed for faster convergence with better accuracy and even reduced training time. The training time was optimized by over 10% in this case. The minimum batch size was set to 32, and the network was trained for 50 epochs. The learning rate was set to the standard learning rate of 0.0001. The validation frequency was set to 5. The training images were not shuffled every epoch as this would be much heavier on the GPU and require a much larger number of epochs to be trained on. Despite not shuffling every epoch, the network yielded acceptable rates of accuracy, precision and specificity which can be observed in the results section of this paper. The network was then trained on exactly 821 image categories augmented into the specified image input layer size and applying a color preprocessing to convert the images to RGB with the layer graph and training options initialized as described above.

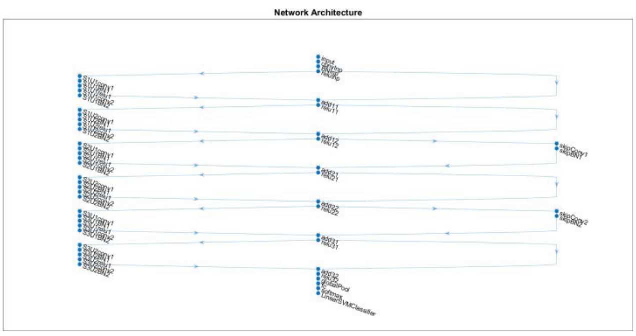



Fig. 5. Layer Graph of implemented ResNet50 Architecture

1. Image Classification Phase: After loading the custom trained net, the images saved in the folder after image processing are fed automatically to the network in the image classification phase. A new augmented image data store is initialized in this phase, where the input parameters are the size of the image, the individual images in the folder and the color preprocessing to make sure the images are always in RGB. The image features are extracted using the activations function with the newly initialized data store. Thelabels are predicted using the classifier Fit Class Error Correcting Output Codec and the test features with observations made in columns. The percentage is measured as done previously. The predicted label is extracted as a string. This extracted string output is saved in an array. The entire process is looped over until every image within the folder is classified. The array is now an array of strings. This outputis now ready for evaluation.
2. Evaluation Phase: In this phase, the output string is evaluated based on a simple conditional algorithm. The algorithm can be simply explained as a conditional statement which checks whether or not the output list is of a format that would match the strings on a number plate. In short, the output must contain 3 words and 6 numbers where the first 4 numbers are separated from the last 2 by a hyphen. The algorithm is set to ignore the hyphen's presence. However, the 3 words must match the names of the cities, the metro and the category of vehicles available within Bangladesh in a sequential manner. If the evaluation returns a negative output, the entire loop will be re-run implementing the other pre-processing functions in a sequential manner. If the list passes the evaluation, it will be converted into a string and will then be ready to enter the data frame.

### C. *Creating a Data-frame*

The output from the Image classification is taken as the input in this code block. The string is simply written into an excel sheet with a built in MATLAB function and saved in a folder for sheets. This code block also has the additional functionality of checking whether or not a car is registered. It will compare the tin plate number's expiry date with the current date and let the user know if it's legal for the road or not.

## IV. Results

### A. *Hardware Setup:*

The processingand classification tasks wereaccomplished on an Intel Core i5 10$^{th}$ Gen CPU with an MX150 GPU. The images were taken by a 50MP smart phone camera and later transferred to the PC.

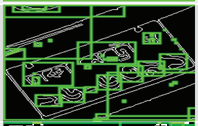
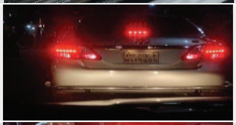
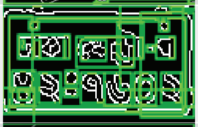
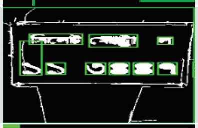
### B. *Software Setup:*

MATLAB Image Processing Toolbox, MATLAB Deep Learning Toolbox, MATLAB ResNet50 Add-on.

### C. *Image Processing Outcomes:*

The following categories for Cars, Two wheelers, Three wheelers, Buses and Trucks have been provided of image have passed the image processing phase successfully.

TABLE I. ROI DETECTION

| ROI Detection from Input Images | | | |
|---|---|---|---|
| Sl. no | Category | Read Images | ROI Detection |
| 1 | Rear View |  |  |
| 2 | Tilted 60° |  |  |
| 3 | Night Distant |  |  |
| 4 | Night, Close |  |  |
| 5 | Tilted Right |  |  |
| 6 | Clear Top |  |  |

*D. Model Accuracy:*

*a) Training and ValidationAccuracy:* The features images with labels named according to their respective characters were 821. The labels included almost all characters of the Bangla alphabets for '*Benjonborno*' and 'Shoroborno' except those for Onusshar, Bishorgo and Chondrobindu. The training set also included 26 alphabets of the English language and the words for various districts along with the word 'Metro'.

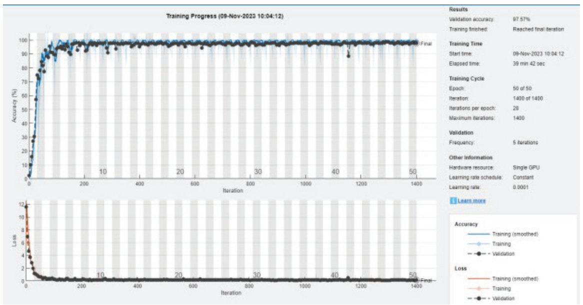


Fig. 6. Training Outcome by RMS Propagation

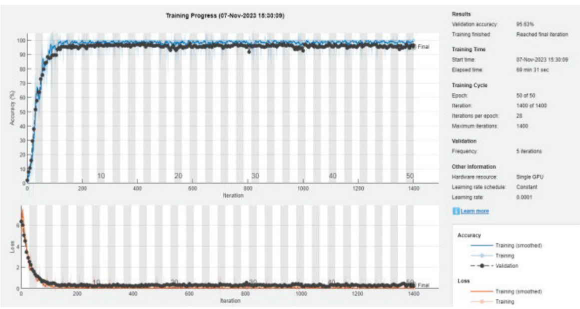


Fig. 7. Training Outcome by Adaptive Mean

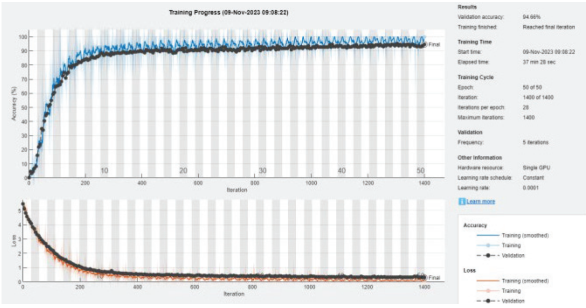


Fig. 8. Training Outcome by Stochastic Gradient Descent Mean

There are 20 categories for numbers, 10 from Bangla and 10 from English. Alongside this, 80 categories of noise in the root folder to allow the model to select one of these categories in case no output matches the characters.The training outcome by using the RMS Propagation, Adaptive Mean and Stochastic Gradient Descent Mean solvers provided 97.57%, 95.63% and 94.66% validation accuracies. So Root Mean Square Propagation was chosen as the optimizer.

*b) Performance Evaluation:* The final prediction accuracy was always above 95% with a maximum prediction of 98%. The classifier ECOC used is a compact model for Support Vector Machines which is good for prediction tasks. The model was fed 1x103 categorical class names and had 103x1 binary learners. The ultimate coding matrix was 103x103 with elements of type double.

In order to evaluate the model performance, the Accuracy, Sensitivity and Precisions were calculated, along with the F1 Scores. The Matthews Correlation Coefficient, Error Rate and False Positive Rates were also calculated.The following formulas were used to perform the calculations assuming TP as True Positives, TN as True Negatives, FP as False Positives:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\ ; \qquad Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{FP+TN}\ ; \qquad Precision = \frac{TP}{TP+FP}$$

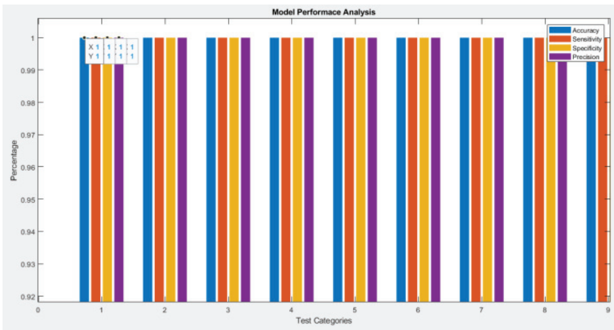



Fig. 9. Model Performance Evaluation

$$F1Scores = \frac{2 \times TP}{2 \times TP+FP+FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+TN)(TP+FN)(TN+FP)(TN+FN)}}$$

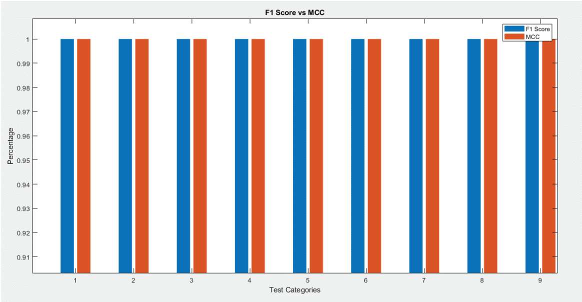



Fig. 10. F1 Scores and Mathews Correlation Coefficient

The high F1 score is an indication of the strong overall performance of the binary classification model and the high MCC depicts that all the classes were predicted well even under disproportionate conditions.

$$Error\ Rate\ = 1 - Accuracy;$$

$$FPR(False\ Positive\ Rate) = 1 - Specificity;$$

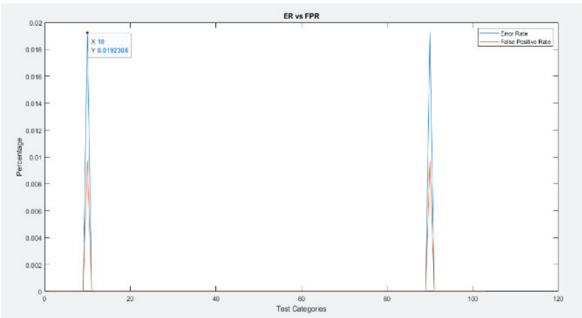



Fig. 11. Error Rate vs. False Positive Rate

The low error rate and false positive rate showcase the success of the model as can be seen from the above line graph.

TABLE II. PERFORMANCE TABLE

| Sl. No. | Predictions | | |
|---|---|---|---|
| | ***Category*** | ***Number of Vehicles*** | ***Percentage*** |
| 1 | Front View | 250 | 98% |
| 2 | Rear View | 150 | 98.4% |
| 3 | Front Tilted View | 125 | 96% |
| 4 | Read Tilted View | 125 | 96.5% |
| 5 | Front Top View | 110 | 97% |
| 6 | Rear Top View | 110 | 97% |
| 7 | Front Top Annotated View | 115 | 95.6% |
| 8 | Rear Top Annotated View | 115 | 95% |
| 9 | Reflecting Plates | 105 | 95.5% |
| 10 | Night View | 140 | 96% |
| 11 | Blurred Images | 60 | 95% |
| 12 | Dirty Images | 14 | 95.4% |
| 12 | Uneven Light Distribution | 15 | 97% |
| 13 | Scratched Out Characters | 4 | 25% |
| 14 | Color Variance | 20 | 98% |
| 15 | Several plates in single image | 16 | 25% |
| 16 | Rickshaw Plates | 5 | 0% |

As can be observed from the performance table, the highest image count is observed within the first image category. Whereas almost no emphasis has been provided in collecting images of rickshaws as the main purpose of this study is to enable the implementation of this model in the detection of vehicles that cross the speed limit, which is not deemed possible when it comes to rickshaws. Also, rickshaws do not have visible license plates most of the time and the rickshaw drivers do not require any form of license to drive on the roads.

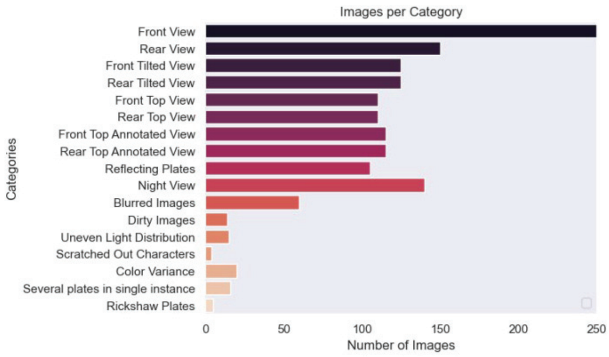



Fig. 12. Images per Category

Clearly, a drawback of the dataset is that the distribution of the images per category is not a normal distribution.

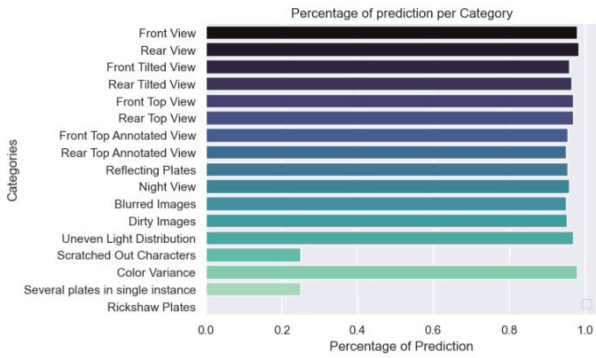



Fig. 13. Percentage of Prediction per category

The percentage of prediction however is up to the mark and almost always close to a 100%.

## V. DISCUSSION

It needs to be stated that the approach of implementing a linear SVM classifier after the Softmax layer is uncommon and unnecessary in most cases. In place of the linear SVM classifier a simple classifier would satisfy the purpose without effecting the results significantly. The sole purpose behind adopting this approach was experimentation and it yielded good results. The model was not trained for detecting Rickshaw plates or images consisting of several instances of plates. This is because the goal of this model is to be able to perfectly recognize the characters on a number plate. The image processing does the work for extracting the region of the plate. This approach is rather primitive and a neural network could be trained to identify the region of interest much more easily. The figures depicting the model output and accuracy were generated in MATLAB while the figures related to the performance table were generated using the Seaborn library of Python. There are models like ImageNet, Google Net, Mobile Net SSD, R-CNN, Faster R-CNN and many others that could be used for detecting the characters. Therefore, in a future work, a comparative study will be done to check which has the best use case in the implementation of Bangla Optical Character Recognition.

## ACKNOWLEDGMENT

Firstly, I would like to acknowledge that none of this would have been possible without the help of our parents. I am very grateful to them for their continued support. Secondly, we are grateful to our supervisor, Gulam Mahfuz Chowdhury for guiding us into the right direction and providing us succinct and precise guidelines on computer vision, machine learning and artificial intelligence. His continuous support even after being overseas has hugely helped us in accomplishing this task. Finally, I would like to express my most heartfelt gratitude to all those who have worked hard for the collection of the data contributing to the authenticity of our research work.

## REFERENCES

[1] S. Mascarenhas and M. Agarwal, "A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification," 2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), Bengaluru, India, 2021, pp. 96-99, doi: 10.1109/CENTCON52345.2021.9687944.

[2] N. Zakaria, F. Mohamed, R. Abdelghani and K. Sundaraj, "VGG16, ResNet-50, and GoogLeNet Deep Learning Architecture for Breathing Sound Classification: A Comparative Study," 2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP), El Oued, Algeria, 2021, pp. 1-6, doi: 10.1109/AI-CSP52968.2021.9671124.

[3] N. Mishra, I. Jahan, M. R. Nadeem and V. Sharma, "A Comparative Study of ResNet50, EfficientNetB7, InceptionV3, VGG16 models in Crop and Weed classification," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-5, doi: 10.1109/ICIEM59379.2023.10166032.

[4] X. Tian and C. Chen, "Modulation Pattern Recognition Based on Resnet50 Neural Network," 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP), Weihai, China, 2019, pp. 34-38, doi: 10.1109/ICICSP48821.2019.8958555.

[5] U. S, V. N. R, P. N. R and L. S, "Performance Analysis of ResNet50 Architecture based Pest Detection System," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 578-583, doi: 10.1109/ICACCS57279.2023.10112802.

[6] S. Rathore, T. Rana, U. Mittal, T. Gupta, N. Malik and M. S. A S, "Brain Tumor Detection by using ResNet-50 and Image Processing Tools," 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2023, pp. 480-483, doi: 10.1109/ICACITE57410.2023.10183081.

[7] V. Sharma and N. Singh, "Deep Convolutional Neural Network with ResNet-50 Learning algorithm for Copy-Move Forgery Detection," 2021 7th International Conference on Signal Processing and Communication (ICSC), Noida, India, 2021, pp. 146-150, doi: 10.1109/ICSC53193.2021.9673422.

[8] A. N. M. Zulfikri, F. Y. A. Rahman, S. Shabuddin and R. Mohamad, "Food Recognition based on Deep Learning Algorithms," 2022 IEEE Symposium on Industrial Electronics & Applications (ISIEA), Langkawi Island, Malaysia, 2022, pp. 1-4, doi: 10.1109/ISIEA54517.2022.9873669.

[9] J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," in IEEE Access, vol. 9, pp. 11203-11225, 2021, doi: 10.1109/ACCESS.2020.3047929.

[10] Y. Y. Lee, Z. Abdul Halim and M. N. Ab Wahab, "License Plate Detection Using Convolutional Neural Network–Back to the Basic With Design of Experiments," in IEEE Access, vol. 10, pp. 22577-22585, 2022, doi: 10.1109/ACCESS.2022.3153340.

[11] M. C. E. Amon et al., "Philippine License Plate Character Recognition using Faster R-CNN with InceptionV2," 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management ( HNICEM ), Laoag, Philippines, 2019, pp. 1-4, doi: 10.1109/HNICEM48295.2019.9072753.

[12] Islam, Md. Ashraful & Chowdhury, Gulam Mahfuz & Haque, Md.Niaz. (2021). Bangla License Plate Detection, Recognition and Authentication with Morphological Process and Template Matching. 1-6. 10.1109/INCET51464.2021.9456145.

[13] N. Haque, S. Islam, R. A. Tithy and M. S. Uddin, "Automatic Bangla License Plate Recognition System for Low-Resolution Images," 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2022, pp. 1-6, doi: 10.1109/STI56238.2022.10103289.

[14] S. Abdullah, M. Mahedi Hasan and S. Muhammad Saiful Islam, "YOLO-Based Three-Stage Network for Bangla License Plate Recognition in Dhaka Metropolitan City," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 2018, pp. 1-6, doi: 10.1109/ICBSLP.2018.8554668.

[15] M. M. Sarif, T. S. Pias, T. Helaly, M. S. R. Tutul and M. N. Rahman, "Deep Learning-Based Bangladeshi License Plate Recognition System," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ISMSIT50672.2020.9254748.

[16] Tusar, Mahmudol & Bhuiya, Md & Hossain, Md & Tabassum, Anika & Khan, Riasat. (2022). Real Time Bangla License Plate Recognition with Deep Learning Techniques. 1-6. 10.1109/IICAIET55139.2022.9936764.

[17] S. U. Ahmed, F. B. F. Maisha and M. Hossam-E-Haider, "Bangla License Plate Detection and Recognition System with YOLOv7 and Improved Custom OCR Engine," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-7, doi: 10.1109/ICERECT56837.2022.10060446.

[18] N. M. Dipu, S. A. Shohan and K. M. A. Salam, "Bangla Optical Character Recognition (OCR) Using Deep Learning Based Image Classification Algorithms," 2021 24th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2021, pp. 1-5, doi: 10.1109/ICCIT54785.2021.9689864.

[19] K. Roy et al., "An Analytical Approach for Enhancing the Automatic Detection and Recognition of Skewed Bangla License Plates," 2019 International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 2019, pp. 1-4, doi: 10.1109/ICBSLP47725.2019.201528.

[20] M. A. Al Nasim, A. I. Chowdhury, J. N. Muna and F. M. Shah, "An Automated Approach for the Recognition of Bengali License Plates," 2021 International Conference on Electronics, Communications and Information Technology (ICECIT), Khulna, Bangladesh, 2021, pp. 1-4, doi: 10.1109/ICECIT54077.2021.9641214.