Bangla License Plate Recognition Using Convolutional Neural Networks (CNN)

Prepared for

Undergraduate Thesis

by

M M Shaifur Rahman

ID: 13301052

Supervisor

Moin Mostakim

Lecturer,

Department of Computer Science and Engineering

BRAC University

December 24, 2017

Declaration

We declare that, this thesis report is our own work and has not been submitted for any other degree or professional qualifications. All sections of the paper that use quotes or describe an argument or concept developed by another author have been referenced in the reference section.

____

Moin Mostakim

Supervisor

____

M M Shaifur Rahman

Student

Acknowledgement

I am grateful to my supervisor, Moin Mostakim, for his encouragement, direction, immense support and valuable ideas throughout the work.

I am also hugely indebted to Zahangir Alom, lecturer (on leave) of BRAC University (currently PhD Candidate in University of Dayton, Ohio) for sharing his valuable ideas and experiences, which helps me to make right choices in implementation phase.

I am thankful to all whom in one way or another contributed in the completion of this thesis.

Abstract

In the last few years, the deep learning technique in particular Convolutional Neural Networks (CNNs) are using massively in the field of computer vision and machine learning. This deep learning technique provides state-of-the-art accuracy in different classification task on different benchmarks such as MNIST, CIFAR-10, CIFAR-100, ImageNet [1, 2]. However, there are a lot of research has been conducted for Bangla License plate recognition in last decade. None of them are used to deploy a physical system for Bangla license plate recognition system because of their poor recognition accuracy. In this thesis, we would like implement CNNs based Bangla license plate recognition system with better accuracy that can be used for different purposes like: roadside assistance, automatic parking lot management system and so on.

Keywords: Convolutional Neural Networks (CNN), Bangla License Plate, License Plate Recognition, Neural Network

Table of Contents

I. Introduction

1.1 Problem Definition

License plate recognition systems are tremendous asset for traffic, parking, toll management and cruise control applications [3]. When it comes to security monitoring and management of any place or region, LPR systems can be used as a tracking aid to serve as eyes for the security team. In the context of safety and law enforcement, LPR systems play a vital role in border surveillance, physical intrusion and safeguarding [4-7]. Many kinds of LPR systems are developed using various intelligent computational techniques to obtain accuracy and efficiency [7]. License plate is a principle identification means for any vehicle. Nevertheless, considering fraud circumstances like alteration and replacement, LPR systems are correlated with intelligence mechanism for robustness [8].

1.2 Motivation

The task of recognition has always been easy for human beings where for the machine it is one of the sophisticated things to do. To us, vision appears to be simple, yet actually, we are processing around 60 images in every second with millions of pixels in each image. The truth is, a big part of our brain is busy doing this processing which makes it clear that this processing is a very hard thing to do for our brain. Moreover, teaching a machine to see like we do is an extremely difficult errand, not just because it is difficult to make a machine understand all the technical stuff, but since we do not know actually how it happens as the entire process is done by the central nervous system. When we see an object reflection of light from that object enters our retina and after doing some elementary analysis it is passed to the brain where the visual cortex analyses the image in a more detailed manner. This process happens in a fraction of a tiny second almost subconsciously. Though all the difficulties we human being have managed to developed a lot in the field of computer vision and we have been able to teach machines the way to seeing a thing. Nowadays machines can classify objects near to human level which can solve many problems we face in our day to day life.

Our thesis is dedicated to increasing the performance of Bangla license plate recognition.

1.3 Objective

In computer vision, image segmentation is the way of segmenting a digital image into multiple sets of pixels called super pixel. The main objective of image segmentation is to simplify the image into something which is meaningful and easier to analyze. Objects and boundaries such as line, curve etc. are located by image segmentation. A label is assigned to each pixel so that the pixels having the same label share certain characteristics. The result of semantic segmentation is set of segments which cover the image entirely. With respect to same characteristics or computed property such as color, intensity or texture each of the pixel in a region is similar. The region which is adjacent is different with respect to same characteristics. Our research is based on pixel level image segmentation. The convolutional neural network has become popular for recognition tasks. We took a six layer convolutional neural network [9] which is a latest state of art in Bangla handwritten digit recognition. Our training was performed with 1750 (One thousand seven hundred and fifty) datasets and tested with 350 (three hundred fifty) sample sets.

II. Literature Review

Yu. A. Bolotova, A. A. Druki and V. G. Spitsyn, proposed a system based on calibrated dual-camera device which is used to detect license plate of moving vehicles one of them are fixed camera and another one is pan-tilt-zoom camera. This device not only tracks multiple targets but also gets the license plate images with high quality. A convolutional neural network (CNN) is designed to be a detector and a character classifier for efficiently locating the regions of license plates and recognizing the alphabets on them [10]. G. R. Goncalves, D. Menotti and W. R. Schwartz worked on OCR approach which is based on convolutional networks (CNN) is used in License plate recognition. In this approach, the architecture of the CNN is chosen from thousands of random possibilities and its filter weights are set at random and normalized to zero mean and unit norm. Linear support vector machine (SVM) with features extraction is used in this CNN [11].R. A. Baten, Z. Omair and U. Sikder implement an unique system using "Matra" concept. “Matra” is a feature of Bangla script is used to develop a Bangla License Plate reader. In this proposed model they segmented each word as single connected components and later recognized through template matching of words [12]. J. Pyo, J. Bang and Y. Jeong develop a CNN which is used to detect Vehicle Recognition. In their proposed architecture first moving car is localized by frame difference, the resultant binary image is used to detect the frontal view of a car by symmetry filter, the detected frontal view is used to train and test the CNN[13]. P Wongta, T. Kobchaisawat and T. H Chalidabhongse in their conference paper described a CNN which is used in multi-oriented Thai-text localization in natural scene images. In this research, text confidence maps are constructed by using specially trained CNN text detector on multi-scaled images. The multi-scaled text confidence maps are merged to produce original input size text confidence map. By using Thai text characteristics, text line hypothesis is generated from a merged text confidence map. Finally, the post-processing techniques and Thai text characteristic analysis are performed to acquire text [14].O. Bulan, V. Kozitsky, P. Ramesh and M. Shreve worked on Automated license plate recognition (ALPR) is using deep localization which is actually a strong convolutional neural network (CNN). Segmentation and Optical Character recognition (OCR) is jointly using a probabilistic inference method based on (Hidden Markov models) HMMs [15]. J. Lwowski and his research team's approach was secure cloud-based deep learning license plate recognition system (LPRS). Their proposed LPRS was for smart cities which were developed based on deep convolutional neural network. In this proposed model authors used NVIDIA GPUs high performance cloud servers which was pretty much costly [16-18]. A built-in system was implemented with a GPU in order to recognize the license plate number without detection line. The deep-learning network to recognize the license plate number of the vehicle uses relatively simple AlexNet. Jetson TX1 board was used as it is low cost standalone embedded system specialized device [19].P. Zemcik and his team did a work focused on recognition of license plates in low resolution and low quality images. The authors present a methodology for collection of real world (non-synthetic) Data set of low quality license plate images with ground truth transcriptions. Their approach to the license plate recognition is based on a Convolutional Neural Network which holistically processes the whole image, avoiding segmentation of the license plate characters. Evaluation results on multiple datasets show that their method significantly outperforms other free and commercial solutions to license plate recognition on the low quality data. To enable further research of low quality license plate recognition, they also make the datasets publicly available [20].

2.1 Methodology

2.1.1 Dataset Construction

Data Acquisition: We captured nearly three thousand images of license plate. All the images were captured by 12 megapixel camera to ensure better image quality. Images were captured from different locations in Dhaka city which includes indoor parking, outdoor parking, running vehicle from road, vehicle. Images were captured in different condition to make the samples standard. Among the captured images only clearly visible license plate images were taken as primary data to ensure our train dataset more accurate. Images were captured from different angle, time and throughout the year. Figure 1(a), 1(b), and 1(c) show original images.

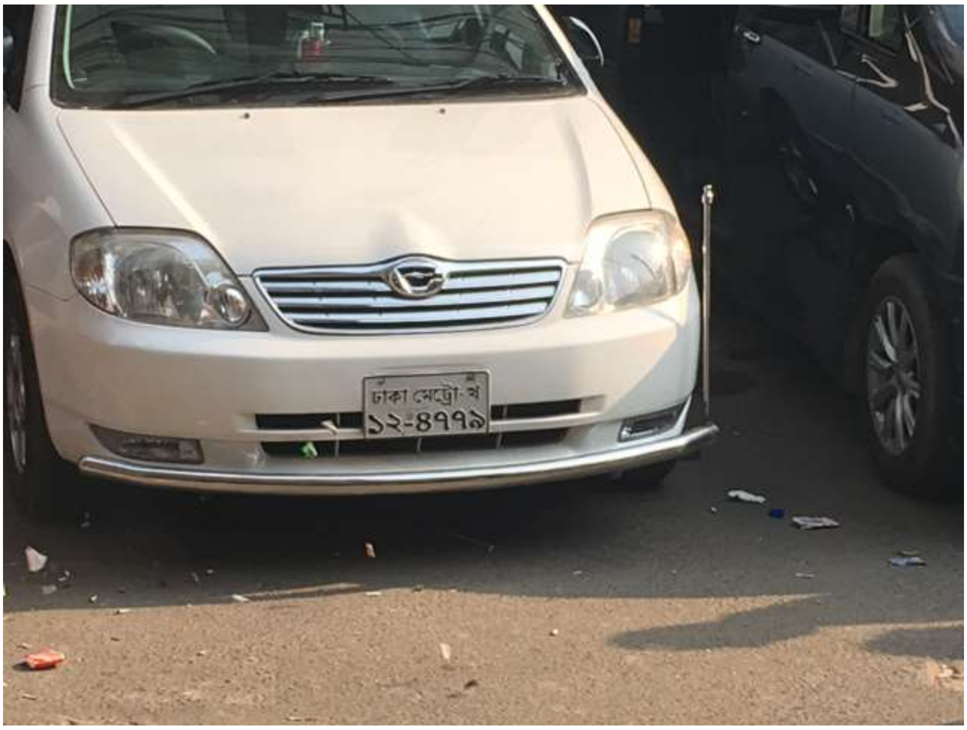

Figure 1(a): Image from Roadside Parking

Figure 1(b): Running vehicle Image from Road

Figure 1(c): Image from Roadside Parking

Data Post Processing: License plate was cropped from the main image first; lately digits and characters were sliced in 32pixels by 32 pixels each. That’s how all the train set data was prepared. Figure-1(d) shows some of our cropped images of license plate from captured images.

Figure 1(d): Cropper License Plate (LP)

In figure-1(e), figure-1(f), figure-1(g) shows sample dataset which we used to train our convolutional neural network (CNN).

Figure-1(e): Sample Dataset

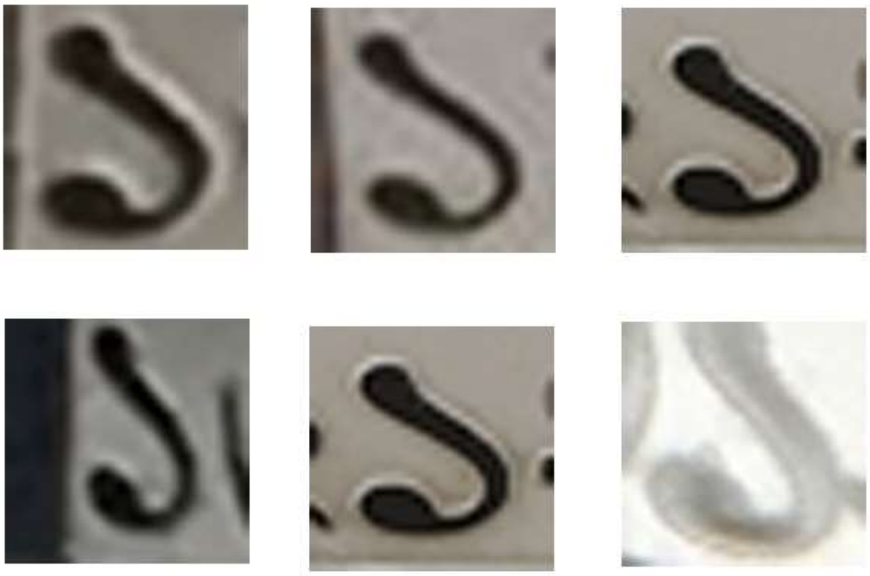

Figure-1(f): Sample Dataset

Figure 1(g): Sample Dataset

The whole process of data acquisition and dataset preparation is shown in figure-1(h)

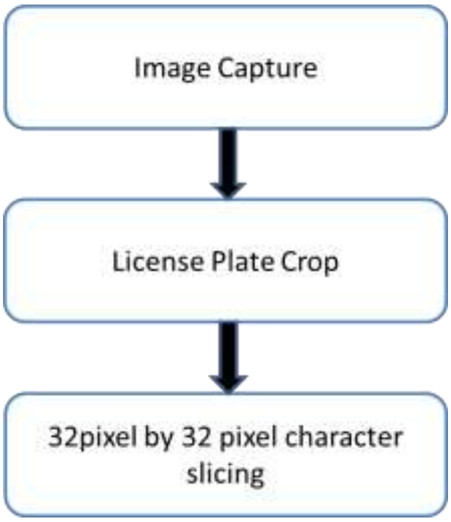

Figure-1(h): Data acquisition and dataset preparation process

2.2 License Plate Recognition

Workflow of our system is illustrated in figure Figure-1(i). At first The CNN was

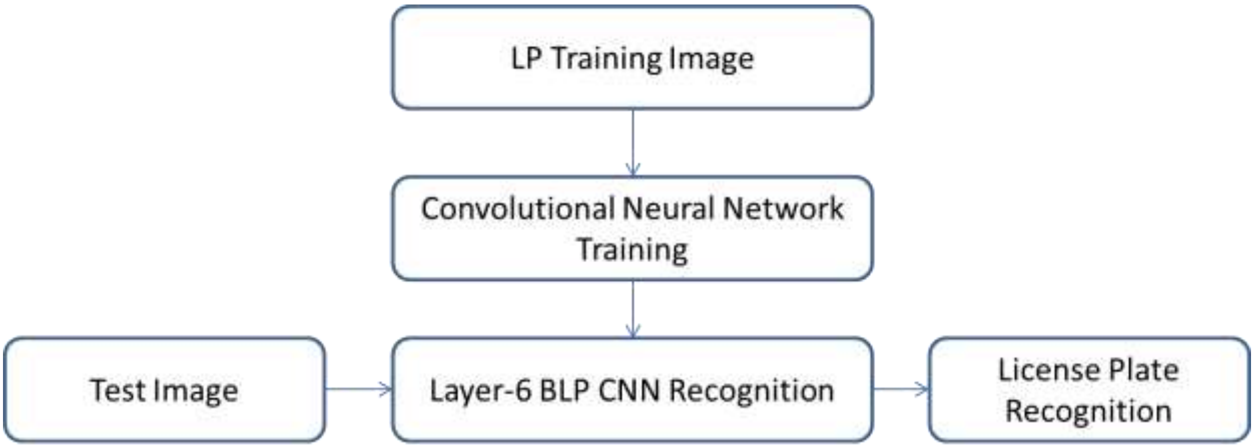

Figure-1(i): System Workflow

trained with prepared dataset of sixteen different classes which actually consists of zero to nine and some Bangla letter we found in license plate. We prepare total one thousand and nine hundred dataset where each class has at least more than one hundred numbers of samples. Training of Bangla License Plate Recognition System (BLPRS) Convolutional Network was performed with different number of datasets. We divide the dataset into train set and test set in different fold like 70% data is used to train 30% to test then again 80% to train and 20% to test 90% to train and 10% of data to test for comparing the performance of the system. Test images were taken randomly from test set to perform testing from 30%, 20% 10% respectively.

III. Convolutional Neural Network (CNN)

The CNN structure was first time proposed by Fukushima in 1980 (Fukushima, 1980)[21]. However, it has not been widely used because the training algorithm was not easy to use. In 1990s, LeCun et al. applied a gradient-based learning algorithm to CNN and obtained successful results (LeCun et al., 1998a)[22]. After that, researchers further improved CNN and reported good results in pattern recognition. Recently, Cirean et al. applied multi-column CNNs to recognize digits, alphanumerics, traffic signs, and the other object class (Ciresan & Meier, 2015; Ciresan et al.,2012)[23-24]. They reported excellent results and surpassed conventional best records on many benchmark databases, including MNIST (LeCun et al., 1998b)[25] handwritten digits database and CIFAR-10 (Krizhevsky & Hinton, 2009)[26]. In addition to the common advantages of DNNs, CNN has some extra properties: it is designed to imitate human visual processing, and it has highly optimized structures to learn the extraction and abstraction of two dimensional (2D) features. In particular, the max-pooling layer of CNN is very effective in absorbing shape variations. Moreover, composed of sparse connection with tied weights, CNN requires significantly fewer parameters than a fully connected network of similar size. Most of all, CNN is trainable with the gradient-based learning algorithm, and suffers less from the diminishing gradient problem. Given that the gradient-based algorithm trains the whole network to minimize an error criterion directly, CNN can produce highly optimized weights. Recently, deep CNN was applied for Hangul handwritten character recognition and achieved the best recognition accuracy (Kim & Xie, 2014)[27]. Figure 2 shows an overall architecture of CNN that consists with two main parts: feature extraction and classification. In the feature extraction layers, each layer of the network receives the output from its immediate previous layer as its input, and passes the current output as input to the next layer. The CNN architecture is composed with the combination of three types of layers: convolution, max-pooling, and classification. Convolutional layer and max-pooling layer are two types of layers in the low and middle-level of the network. The even numbered layers work for convolution and odd numbered layers work for max-pooling operation. The output nodes of the convolution and max pooling layers are grouped into a 2D plane which is called feature mapping. Each plane of the layer usually derived with the combination of one or more planes of the previous layers. The node of the plane is connected to a small region of each connected planes of the previous layer. Each node of the convolution layer extracts features from the input images by convolution operation on the input nodes. The max-pooling layer abstracts features through average or propagating operation on the input nodes. The higher level features is derived from the propagated feature of the lower level layers. As the features propagate to the highest layer or level, the dimension of the features is reduced depending on the size of the convolutional and max-pooling masks. However, the number of feature mapping usually increased for mapping the extreme suitable features of the input images to achieve better classification accuracy. The outputs of the last feature maps of CNN are used as input to the fully connected network which is called classification layer. In this work, we use the feed-forward neural networks as a classifier in the classification layer, because it has proved better performance compared to some recent works (Mohamed et al., 2012[28]; Nair & Hinton, 2010)[29]. In the classification layer, the desired number of features can be obtained using feature selection techniques depending on the dimension of the weight matrix of the final neural network, and then the selected features are set to the classifier to compute confidence of the input images. Based on the highest confidence, the classifier gives outputs for the corresponding classes that the input images belong to. Mathematical details of different layers of CNN are discussed in the following section.

3.1 Convolutional layer

In this layer, the feature maps of the previous layer are convolved with learnable kernels such as (Gaussian or Gabor). The outputs of the kernel go through linear or non-linear activation functions such as (sigmoid, hyperbolic tangent, softmax, rectified linear, and identity functions) to form the output feature maps. In general, it can be mathematically modeled as

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} k_{ij}^l + b_j^l)$$

Where x_j^l the output of the current layer is, x_i^{l-1} is previous layer outputs, x_{ij}^l is kernel for present layer, and $b_j^l$is the bias for current layer. M_j represents a selection of input maps. For each output map is given an additive bias *b*. However, the input maps will be convolved with distinct kernels to generate the corresponding output maps. For instant, the output maps of *j* and *k* both are summation over the input *i* which is in particular applied the *j* th kernel over the input i and takes the summation of its and same operation are being considered for *k* th kernel as well.

3.2 Subsampling layer

The subsampling layer performs down sampling operation on the input maps. In this layer, the input and output maps do not change. For example, if there are N input maps, then there will be exactly N output maps. Due to the down sampling operation, the size of the output maps will be reduced depending on the size of the down sampling mask. In this experiment, 2 × 2 down sampling mask is used. This operation can be formulated as

$$x_j^l = f(\beta_j^l down(x_j^{l-1}) + b_j^l)$$

Where *down (·)* represents a subsampling function. This function usually sums up over $n \times n$ block of the maps from the previous layers and selects the average value or the highest values among the $n \times n$ block maps. Accordingly, the output map dimension is reduced to n times with respect to both dimensions of the feature maps. The output maps finally go through linear or non-linear activation functions.

3.3 Classification layer

This is a fully connected layer which computes the score for each class of the objects using the extracted features from convolutional layer. In this work, the size of the feature map is considered to be 5×5 and a feed-forward neural net is used for classification. As for the activation function, sigmoid function is employed as suggested in most literatures

3.4 CNN with Back-propagation

In the Back-propagation steps in CNNs, the filters are updated during the convolutional operation between the convolutional layer and immediate previous layer on the feature maps and the weight matrix of each layer is calculated accordingly.

3.5 CNN with dropout

The combination of the prediction of different models is a very effective way to reduce test errors (Bell & Koren, 2007; Breiman, 2001)[30-31], but it is computationally expensive for large neural networks that can take several days for training. However, there is a very efficient technique for the combination models named “dropout” (Hinton et al.,2012)[32]. In this model, the outputs of hidden layer neurons are set to be zero if the probability is less than or equal to a certain value, for example 0.5. The neurons that are “dropped out” in the way to forward pass that do not have any impact on BP. Dropout reduces complexity of the network because of co-adaptation of neurons, since one set of neurons are not rely on the presence of another set of neurons. Therefore, it is forced to learn more robust features that are useful in aggregation with many different random subsets of the other neurons. However, one of the drawbacks of the dropout operation is that it may take more iteration to reach the required convergence level. In this work, dropout is applied in the first two fully-connected layers in Fig. 2.

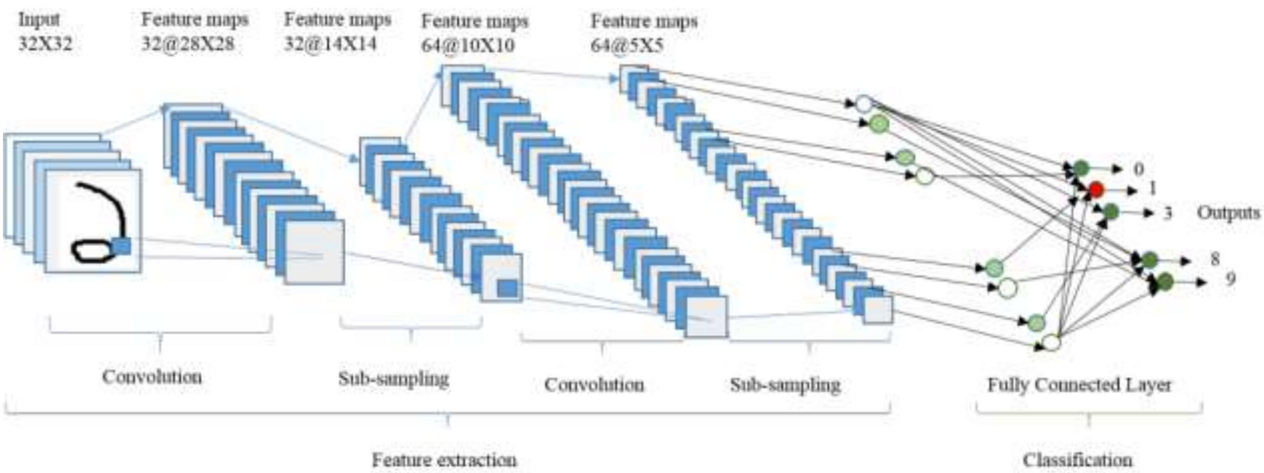

Figure 2. The overall architecture of the CNN used in this work, which includes an input layer, multiple alternating convolution and max-pooling layers, and one fully connected classification layer.

3.6 Number of parameters of CNN

In this experiment, we used six layers of convolutional neural networks. One input layer, two layers for convolution, two layers for subsampling or pooling and final one layer for classification. The first convolution layer has 6 output mapping and the second one has 12 output mapping. The parameter of convolutional network is calculated according to the following manner: 32 × 32 image is taken as input.

Table-1: Number of parameter for different layers of CNN

Layer	Operation of Layer	Number of feature maps	Size of feature maps	Size of window	Number of parameters
C_1	Convolution	6	28X28	5X5	156
S_1	Max-Pooling	6	14X14	2X2	0
C_2	Convolution	12	10X10	5X5	1,872
S_2	Max-Pooling	12	5X5	2X2	0
F_1	Fully Connected	300	1X1	N/A	93,600
F_2	Fully connected	16	1X1	N/A	4,816

The output of the convolutional layer is 28 × 28 with 6 feature maps. The size of the filter mask is 5 × 5 for the both convolution layers. The number of parameters are used to learn is (5 × 5 + 1) × 6 = 156 and the total number of the first subsampling layer, the number of trainable parameters is 0 and the size of the outputs of subsampling layer is 14 × 14 with 6 feature maps. For second convolutional layer (14-5)+1=10, 10 X 10 with 12 feature map is used and the second sub sampling filter mask size is 2 X 2. The learning parameters for second convolution layer are ((5×5+1) ×6) ×12 = 1,872 and 0 for convolutional and sub-sampling layers, respectively. In the fully connected layer, number of feature maps is an empirically chosen number which is 5 X 5 X 12 = 300 from the previous max-pooling layer provides outputs with 12 maps and 5 × 5 size of output for each input. The number of parameters for the first fully connected layer is: 300 × 12 × (5 × 5 + 1) = 93,600, whereas the amount of the final layer parameter is: 16 × (300 + 1) = 4,816. Total number of parameters is 1,00,444. All the parameters with respect to the corresponding layers is stated in Table 1, and Fig. 3 illustrates a corresponding feature extraction process in CNN.

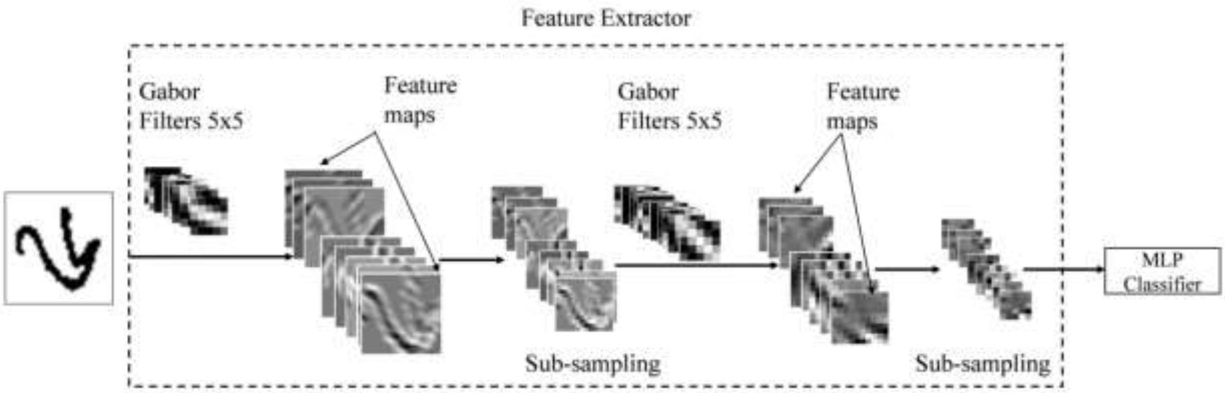

Figure 3: Visualization of feature extraction for hand written digit six with CNN.

The number of feature maps does not change in the sub-sampling layer. In addition, these layers do not require any computational parameters. We can observe that the fully connected layers are expensive in term of computational parameters of deep learning approaches compare to convolutional layer.

IV. Experiment and Result

4.1 Environment setup

We have conducted experiment with our own dataset which contains 1750 number of training samples and 350 number of testing samples. We did not apply any data augmentation or preprocessing steps for preparing this dataset. The entire experiment has been implemented with Matlab version R2015a on window environment with 4 GB RAM.

4.2 Results

During the experiment, the entire experiment has been run for several times for different combination of training and testing samples. The experimental results are shown in Figure-4(a), 4(b), 4(c), 4(d) respectively. The experimental results are shows the error rate with respect to the number of iteration are considered. Figure 4(a) shows the errors versus iteration curve for 10 epoch. Where one epoch equal to one forward pass and one backward pass of all the training examples. Here the number of iteration has been calculated based on the number samples in training set times with the number of epochs.

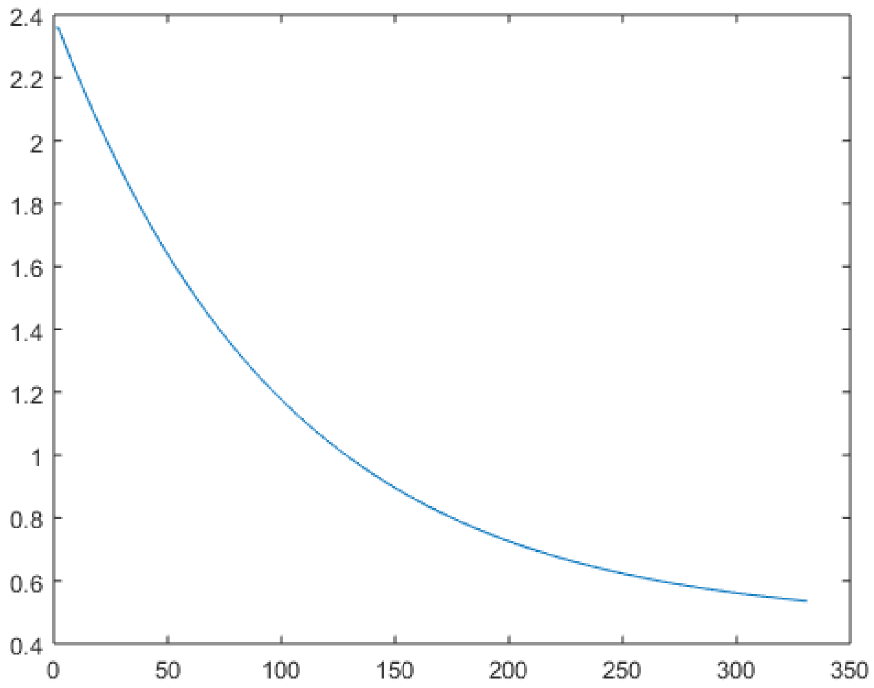

Figure 4(a): Iteration vs Error for 10 Epoch

The following Figure 4(b), 4(c), and 4(c) show the convergence curve for 100, 500, and 1000 epochs respectively.

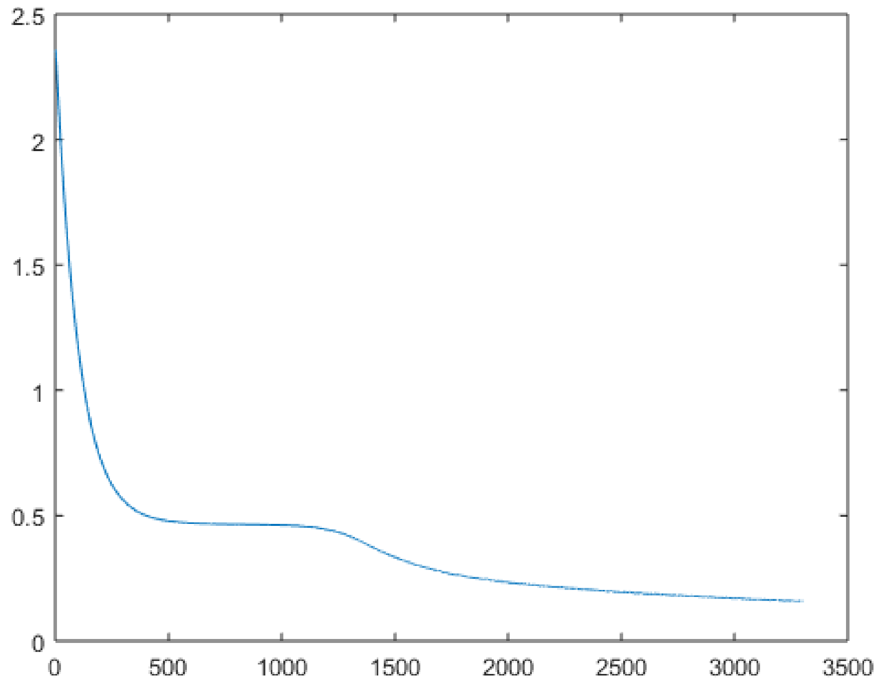

Figure 4(b): Iteration vs Error for 100 Epoch

From these figures, it can be clearly observed that the performance of the proposed system increases with respect to the number of epochs. As deep learning is a data driven techniques, here we have only uses 1750 number of samples for training the whole system. Therefore, by increasing the number of epoch or iterations we can ensure better learning during training.

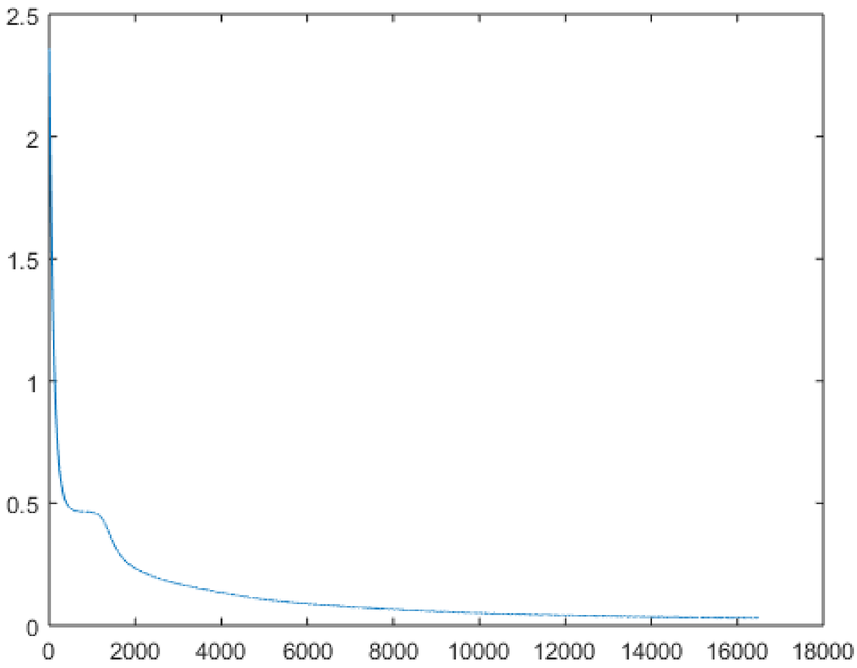

Figure 4(c): Iteration vs Error for 500 Epoch

The best convergence behavior can be observed for 1000 number of epochs which is shown in Figure 4(d).

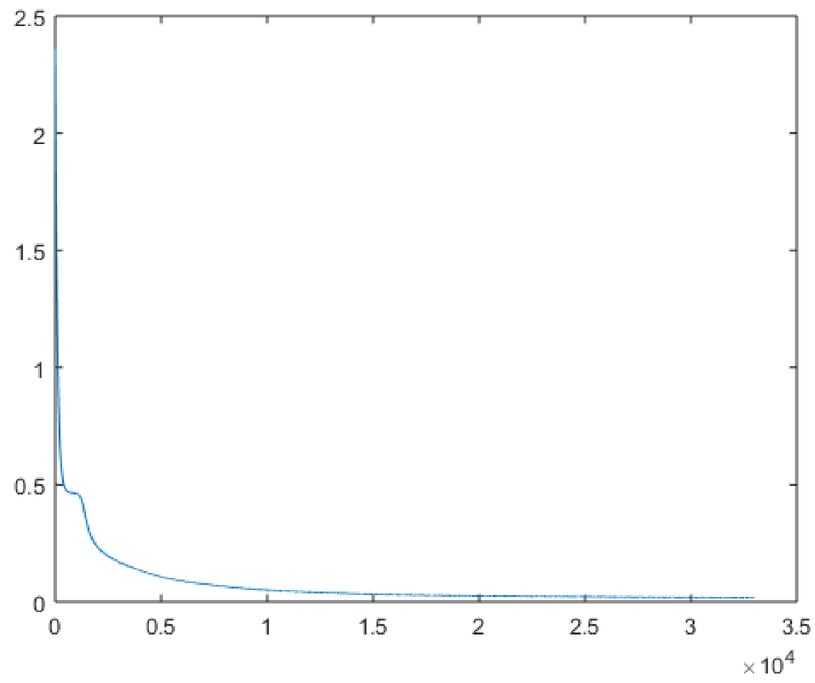

Figure 4(d): Iteration vs Error for 1000 Epoch

Finally, we have tested our proposed techniques with all of the different trained model and observed different performance based on number of inputs sample in the testing phase shown in Figure 4(e). Figure shows Accuracy for different number of test set, column 1, 2, 3, 4 represent system trained with 1300, 1500, 1650, and 1750 number of training sample respectively. We have achieved 88.67% testing accuracy for highest number of training sample which is the highest testing accuracy in this implementation other training samples also shows accuracy of 81.09%, 82.28% respectively. So, it's clearly visible that the number of dataset increases the performance.

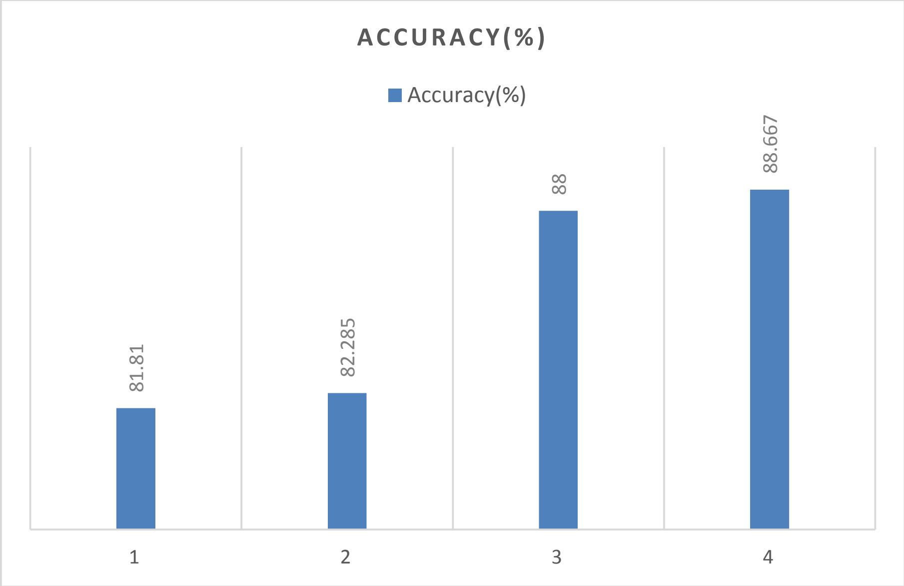

Figure-4(e): Testing accuracy for different number of input samples

4.3 Computational time

Train Sample	Average time/Epoch	Epoch	Computational Time(Sec)	Accuracy
1300	2.1762	100	217.62	70
1300	2.37	500	1185	81.8181
1300	2.208	1000	2208	81.0909
1500	2.6422	100	264.22	73.428
1500	2.52456	500	1262.28	82.285
1500	2.64115	1000	2641.15	82.285
1650	2.835	100	283.5	79.33333
1650	2.9672	500	1483.6	88.66667
1650	2.7625	1000	2762.5	88
1750	2.874	100	287.4	82
1750	3.567	500	1783.5	86
1750	2.89	1000	2890	88

Table-3: Epoch, Computational time, Accuracy chart

Observing the table-3 we can conclude that large number of dataset and epoch provides better accuracy. However, larger number of epoch takes longer time to produce output.

4.3 Limitation

The limitation that was faced for training was due to having smaller memory and computational power. A general purpose computer with 4GB of RAM and core i3 3.2GHz CPU was used to train and test the system. As for our training and simultaneous testing, the memory and computational power requirement was high but due to lack of resources the testing process consume more time.

For using any other proponent or using deeper net the memory needed is much more. If training/testing was done with GPU the time required would be 1/10 fold. For testing with higher number of epoch, we needed 6 hours of continuous testing but if conducted with GPU it would decrease drastically. A large GPU memory surely improves overall system performance.

V. Conclusion and Discussion

5.1 Conclusion

A convolutional neural network is a cornerstone for any recognition task nowadays ranging from classification, object-detection to segmentation. In addition, fully convolutional neural networks are better feature-extractors than their fully connected layer consisting counterparts. Furthermore, the deeper the architecture the better the feature extraction process although not always. Our novel idea was to implement the system for Bangla License Plate Recognition (BLPRS) and show that with CNN is a better approach to obtain better performance. In this implementation, we have achieved around 89% testing accuracy for BLPR. Another contribution of this work is to create a benchmark dataset for BLPRS which will be helpful for implement and evaluated BLPRS in the future for others.

5.2 Future Work

We are using dataset with 1750 number of samples for training our system, we have planned to increase the number of train samples to improve system performance. Moreover, our Convolutional Neural Network (CNN) which consists of 6 layers including fully connected layers and with only 6 and 12 feature maps in two convolution layers respectively. The performance of this proposed license place recognition will be increase with higher number of features maps and more layers which one of the future direction of this work.

References

[1] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012

[3] Du, Shan, et al. "Automatic license plate recognition (ALPR): A state-of-the-art review." IEEE Transactions on circuits and systems for video technology 23.2 (2013): 311-325.

[4] Abdullah, Siti Norul Huda Sheikh, et al. "Comparison of feature extractors in license plate recognition." Modelling & Simulation, 2007. AMS'07. First Asia International Conference on. IEEE, 2007.

[5] Hsieh, Ching Tang, and Yu-Shan Juan. "Wavelet transform based license plate detection for cluttered scene." WSEAS Transactions on Computers 4.1 (2005): 40-44.

[6] Sulehria, Humayun Karim, Danish Irfan Ye Zhang, and Atif Karim Sulehria. "Vehicle number plate recognition using mathematical morphology and neural networks." WSEAS Transactions on Computers 7 (2008).

[7] Sulehria, Humayun K., Ye Zhang, and Danish Irfan. "Mathematical Morphology Methodology for Extraction of Vehicle Number Plates." International journal of computers 1.3 (2007): 69-73.

[8] Villegas, Osslan Osiris Vergara, et al. "License plate recognition using a novel fuzzy multilayer neural network." International journal of computers 3.1 (2009): 31-40.

[9] M. Alom, P. Sidike, T. Taha and V. Asari, "Handwritten Bangla Digit Recognition Using Deep Learning", *https://www.researchgate.net*, 2017.

[10] "License Plate Detection and Recognition Using a Dual-Camera Module in a Large Space - IEEE Conference Publication", Ieeexplore.ieee.org, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/4373505/.

[11] Gabriel Resende Gonçalves, David Menotti, William Robson Schwartz, "License plate recognition based on temporal redundancy", *Intelligent Transportation Systems (ITSC) 2016 IEEE 19th International Conference on*, pp. 2577-2582, 2016, ISSN 2153-0017.

[12] R. A. Baten, Z. Omair and U. Sikder, "Bangla license plate reader for metropolitan cities of Bangladesh using template matching," *8th International Conference on Electrical and Computer Engineering*, Dhaka, 2014, pp. 776-779.

[13] Junghwan Pyo, Jiwon Bang, Yongjin Jeong, "Front collision warning based on vehicle detection using CNN", *SoC Design Conference (ISOCC) 2016 International*, pp. 163-164, 2016.

[14] Pitchakorn Wongta, Thananop Kobchaisawat, Thanarat H. Chalidabhongse, "An automatic bus route number recognition", *Computer Science and Software*

Engineering (JCSSE) 2016 13th International Joint Conference on, pp. 1-6, 2016.

[15] O. Bulan, V. Kozitsky, P. Ramesh and M. Shreve, "Segmentation- and Annotation-Free License Plate Recognition With Deep Localization and Failure Identification," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351-2363, Sept. 2017.

[16] Jonathan Lwowski, Prasanna Kolar, Patrick Benavidez, Paul Rad, John J. Prevost, Mo Jamshidi, "Pedestrian detection system for smart communities using deep Convolutional Neural Networks", *System of Systems Engineering Conference (SoSE) 2017 12th*, pp. 1-6, 2017.

[17] Andrew Boles, Paul Rad, "Voice biometrics: Deep learning-based voiceprint authentication system", *System of Systems Engineering Conference (SoSE) 2017 12th*, pp. 1-6, 2017.

[18] Nicolas Gallardo, Nicholas Gamez, Paul Rad, Mo Jamshidi, "Autonomous decision making for a driver-less car", *System of Systems Engineering Conference (SoSE) 2017 12th*, pp. 1-6, 2017.

[19] S. Lee, K. Son, H. Kim and J. Park, "Car plate recognition based on CNN using embedded system with GPU," *2017 10th International Conference on Human System Interactions (HSI)*, Ulsan, 2017, pp. 239-241.

[20] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík and P. Zemčík, "Holistic recognition of low quality license plates by CNN using track

annotated data," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, 2017, pp. 1-6.

[21] Fukushima, Kunihiko. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4):193–202, 1980.

[22] LeCun, Yann, Bottou, Leon, Bengio, Yoshua, and Haffner, ´ Patrick. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278– 2324, 1998a

[23] Ciresan, D. and Meier, U. Multi-column deep neural networks for offline handwritten chinese character classi- fication. In International Joint Conference on Neural Networks (IJCNN), pp. 1–6, July 2015.

[24] Ciresan, Dan, Meier, Ueli, and Schmidhuber, Jurgen. ¨ Multi-column deep neural networks for image classification. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE, 2012.

[25] LeCun, Yann, Bottou, Leon, Bengio, Yoshua, and Haffner, ´ Patrick. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278– 2324, 1998b

[26] Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images, 2009.

[27] Kim, In-Jung and Xie, Xiaohui. Handwritten hangul recognition using deep convolutional neural networks. International Journal on Document Analysis and Recognition (IJDAR), 18(1):1–13, 2014.

[28] Mohamed, Abdel-rahman, Dahl, George E, and Hinton, Geoffrey. Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, and Language Processing, 20(1):14–22, 2012.

[29] Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814, 2010.

[30] Bell, Robert M and Koren, Yehuda. Lessons from the netflix prize challenge. ACM SIGKDD Explorations Newsletter, 9(2):75–79, 2007.

[31] Breiman, Leo. Random forests. Machine learning, 45(1): 5–32, 2001.

[32] Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.