VEHICLE NUMBER PLATE DETECTION USING DEEP LEARNING

Pragati Vilas Ayare, Kishori Sudhakar Gaonkar, Rohan Ramesh Kudtarkar
[1]Student,[2]Student,[3]Student
Department of Computer Engineering,
SSPM's college of engineering, Kankavali, Dist.- Sindhudurg, State- Maharashtra

Abstract: In this work, we tackle the problem of Vehicle number plate detection in natural scene images. Inspiredby the success of deep neural networks (DNNs) in various vision applications, here we leverage DNNs to learn high- level features in a cascade framework, which lead to improved performance on detection. Firstly, we train a 37-class convolutional neural network (CNN) to detect all characters in an image, which results in a high recall, compared with conventional approaches such as training a binary text/non-text classifier. False positives are then eliminated by the second plate/non- plate CNN classifier. Bounding box refinement is then carried out based on the edge information of the number plates, in order to improve the intersection-over-union (IOU) ratio. The proposed cascade framework extracts number plates effectively with both high recall and precision.

***IndexTerms*- Convolution neural network, Bounding Box refinement, Run length smoothing algorithm.**

I. INTRODUCTION

With the recent advances in intelligent transportation systems, vehicle number plate detection has attracted considerable research interests. It has an assortment of potential applications, for example, in security and traffic control. Much work has been done on the topic of VNPD. Plate detection means to localize the number plate and generate suitable bounding boxes, while plate recognition aims to identify the characters depicted within the bounding boxes. We tackle the problems of illumination and occlusion in the image by leveraging the high capability of convolutional neural networks (CNN). CNN's have demonstrated impressive performance on various tasks including image classification, object detection, semantic segmentation, etc.

II. LITERATURE REVIEW

A. [Max Jaderberg 2014], They used CNN for detecting words regions in the image with no downsampling for per pixel sliding window [1].
B. [C-N Anagnostopoulos 2006], They proposed a method on the basis of novel adaptive image segmentation technique (Sliding Concentric Windows- SCW) and connected component analysis in conjunction with character recognition Neural Network [2].
C. [A Graves 2009], It describes a system capable of directly transcribing raw online handwriting data. The system consists of an advanced recurrent neural network with an output layer designed for sequence labeling, combined with a probabilistic language model [3].
.

III. PROBLEM STATEMENT AND PROPOSED SYSTEM

A. Problem statements:

To develop Vehicle Number Plate Detection using the deep convolutional neural network, connected component analysis.

B. Proposed system:

Number plate detection is the first stage of the vehicle number plate detection system. In order to accelerate the detection process, we train a 4-layer CNN to classify whether the image patch contains characters or not. Here we train a 37-class CNN classifier for 26 upper-case letters, 10 digits, and a non-character class, instead of a binary text/non-text classifier. CNN is trained with grayscale characters and background images of 24x24 pixels, normalized by subtracting the mean overall training data. Given an input image, we resize it into 12 different scales and calculate the character saliency map at each scale by evaluating CNN classifier in a sliding window fashion across the image. The input image is padded with 12-pixels each side so that characters near image edges would not be missed [1].

IV. DESIGN DETAILS

In this work, we take help of the convolutional neural network and perform character-based plate detection using a multiple scale sliding window manner. Number plate composed of digits as well as upper case letters. To distinguish the number plate from another text or text-like outliers present in the image, another plate/non-plate classifier is used to remove those false positive.

A. Number plate detection:

To detect whether the image patch contains characters or not, we train 4-layer CNN classifier by running CNN classifier on the whole image, compute the character saliency map.

In table 1, the 4-layer character CNN model is given. It includes 2 convolutional layers and 2 fully connected layers. Here we use 37-class CNN classifier for 26 upper-case letters,10 digits, and a non-character class, instead of a binary text/non-text classifier.
Table1 Configuration of the 4-layer Character CNN model

Layer Type	Parameters
Soft-max	37 classes
Fully connected	#neurons:37
Dropout	Prop:0.5
ReLU	
Fully connected	#neurons: 512
Maxpooling	P:2 x 2,s: 4
ReLU	
Convolution	#filters:384,k:2 x 2,s:1,p:0
Maxpooling	P:4 x4,s:4
ReLU	
Convolution	#filters:120,k:5 x 5,s:1,p:0
Input	24x24 pixels gray-scale image

By using CNN classifier, we resize the image into 12 different scales and calculate character saliency map at each scale by evaluating CNN classifier in a sliding window fashion across the image. Characters near the image edges should not be missed. For that reason, the input image is padded with 12 pixels. Then run length smoothing algorithm (RLSA) and connected component analysis is used to generate character string bounding boxes. [5]

B. False positives elimination and bounding box refinement:

After all the scaled images are processed, bounding boxes are produced. That produced bounding boxes are filtered based on some geometric constraints such as boxes length, height and aspect ratio.

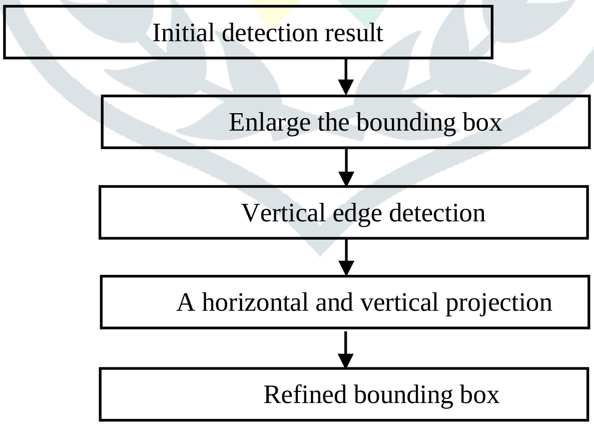

Fig: The process of bounding box refines

Then the enlarge the bounding box with 20% on each side. By using a Sobel operator, we perform vertical edge detection on a cropped number plate. Horizontal projection is applied to find the top and bottom boundaries of the number plate. After that vertical projection is carried out to get left and right bounds of the number plate.

Another plate/non-plate CNN classifier used to verify the remaining bounding boxes. In table 2, the plate/non-plate CNN model is given. This model is trained with positive samples of grayscale number plate from different countries, either cropped from real images or synthesized by ourselves and negative samples constituted by non-text image patches as well as some general text strings [5].

Layer type	Parameters
Soft-max	2 classes
Fully connected	#neurons: 2
Dropout	Prop: 0.5
ReLU	
Fully connected	#neurons: 500
Maxpooling	P: 3 x 3, s: 3
ReLU	
Convolution	#filters: 256, k:5 x 5, s:1, p:0
Maxpooling	p:2 x 2, s:2
ReLU	
Convolution	#filters: 96, k:5 x 5, s:1, p:0
Input	30 x 100 pixels gray-scale image

Table 2 Configuration of the 4-layer number plate/non-plate CNN model.

V. Implementation

A. Feature learning using Convolutional Neural Network:

In our system classifier is used to detect an image region that contains text. To classify an image patch x in one of the possible characters (or background), we extract an set of features $\Phi(x)=(\Phi1(x),\Phi2(x),\ldots..,\Phi k(x))$ and then learn a binary classifier FC for each character c of the alphabet C. Convolution Neural Network are obtained by stacking multiple layers of features. CNN consist of non-linear response function and k linear filter. The convolutional layer can be intertwined with subsampling, normalization and max-pooling layers which build translation invariance in local neighborhoods. Here, we train CNN classifier after that we get the training model which help to detect character region from given input image[1].

B. Text detection:

The process of text detection is started by computing text saliency map by evaluating character/background CNN classifier in sliding window fashion across the image.

Given these saliency maps, word bounding boxes are generated using two steps: First step is to identify lines of text. At this end, thresholding is performed to find local regions of high probability. Run length smoothing algorithm (RLSA) help to connect this region in the text line, for each row of pixels the menu ‘u’ and standard deviation ‘σ’ of the spacing between them is less than 3u-0.5σ. Finding connected components of the linked regions gives candidate text lines [1].

The second step is to split text lines into words. If horizontal spacing is less than the mean horizontal spacing for the text line then adjacent connected components are than connected using RLSA algorithm. Connected components give candidate bounding boxes for individual words. Finally, geometric constraints such as box height, aspect ratio are applied to filter bounding boxes.

C. Data set:

The Char74K and Stanford synth Both data set contain small single-character images of all 62 characters(0-9,a-z, A-Z). Char 74 contains natural images and characters and set of synthetically generated characters.

VI. Conclusion

In this paper, we have presented a vehicle number plate detection system deep learning framework. The result explores the use of CNN in feature extraction replacement. This can be extended by implementing number recognition in the bounding box.

In addition, it would be useful to use CNN for detecting bounding box.

.

REFERENCES

[1]. M. Jaderberg, A. Vedaldi, and A. Zisserman feature for text spots," in Proc. Eur. Conf. Comp. Vis.,2014, pp. 512528.

[2]. C.-N. Anagnostopoulos, I. Giannoukos, V. Loumos, and E. Kayafas, license plate recognition algorithm for intelligent transportation system applications," IEEE Trans. Intell Transp. Syst., vol. 7, no. 3, pp. 377392,2006.

[3]. A. Graves, M. Liwicki, and S. Fernandez, a novel connectionist system for unconstrained handwriting recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855868, 2009.

[4]. K. Lin, H. Tang, and T. Huang, license plate detection using image saliency," in Proc. IEEE Int. Conf.Patt. Recogn., 2010, pp.39453948.

[5]. W. Zhou, H. Li, Y. Lu, and Q. Tian, visual word discovery for automatic license plate detection," IEEE Trans. Image Process., vol. 21, no. 9, pp. 42694279,2012.

[6]. A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in Proc. Int. Conf. Mach. Learn., 2006, pp. 369–376.