

Flagged License Plate Detection And Warning

Rabia Umar^a, Hardik Paliwal^b, Avdhesh Tomar^c, Praveen Mishra^d

^{a,b,c}School of Computing Science & Engineering, Galgotias University

^dAssistant Professor, School of Computing Science & Engineering, Galgotias University

^arabiaumar153@gmail.com, ^bhardikpaliwal@gmail.com, ^ctomaravdhesh245@gmail.com,

^dpraveen.mishra@galgotiasuniversity.edu.in

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: CCTV Camera nowadays captures high-resolution video which can be used to analyse frames to fetch text by using the technique of Optical Character Recognition (OCR). FLDPW (Flagged License Plate Detection and Warning) uses Deep Neural Networks (DNN) techniques to isolate number plates from a given frame and fetch License Plate Number as plain text and compare it with predefined suspicious database and give a warning to authorities when a Suspicious License Plate (SLP) is found.

Keywords: FLDPW, SLP, DNN

1. Conversion of videos into frames

When a CCTV camera captures high-quality videos, it comes in a video format, with a certain FPS (Frames Per Second), which is a collection of frames, so a 30 FPS video will mean that there are 30 Frames in a second interval that means 30 images are there in one second.

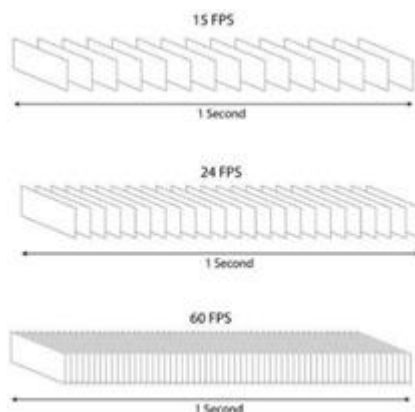


Figure 1. Decomposition of videos into frames

This paper requires analysis to be done on frames.

So, a video is decomposed into respective frames.

II. License Plate Detection:

To decrease computation and to increase accuracy, isolation of license plates is necessary. CNN (Convolutional Neural Network) is the ability of a system to automate the process to learn a large number of filters in concurrent which is specifically for a training dataset under the conditions of a specific predictive modelling problem, like image classification. The output is very specific features that can be detected anywhere on input images. The technique we are using to isolate license plates is YOLO (You Only Look Once) algorithm. Other object detection algorithms do detections on various regions which are proposed and hence end up in predicting more than one time for various regions in an image. YOLO is similar to FCNN (Fully Convolutional Neural Network) and it passes the image first through the FCNN and the yield is prediction, ROI (Region of Interest) coordinates, confidence values and class values. The ROI is a grid itself. YOLO is a reframed object detection process as a regression problem (single), direct output of ROI coordinates, confidence values and class values from image pixels. YOLO does training on full images and optimization is done directly for detection performance. The base performance of this process is ~44 FPS without batch computation on an NVidia Titan X GPU. This design offers end-to-end training and speeds are real-time along

with maintaining high average precision. The system divides the input image into a $P \times P$ grid. The specific grid cell is responsible to detect an object if that object's centre lies in that grid.

III. Comparison of YOLO with other models(I):

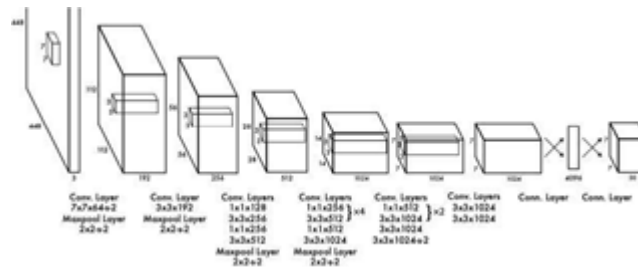


Figure 2. YOLO network architecture

The output of a grid is ROI, class and confidence values of that object. Confidence is calculated as $Pr(\text{Object}) * IOU$ (Intersection over Union). Each ROI contains 5 predictions: x, y, w, h and confidence value. (x and y) coordinates denote the centre of the box relative to the bounds of the grid cell. W (Width) and h (height) are predicted relatively to the whole image. Then the confidence prediction denotes the IOU between the box which is predicted and any ground truth box.

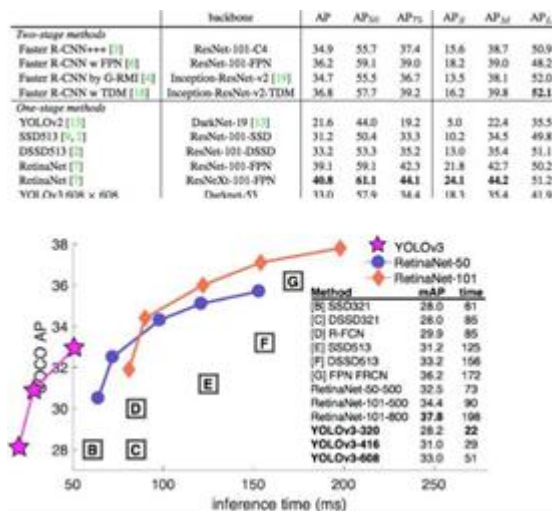


Figure 4. Performance for YOLOv3

Performance for YOLOv2 with COCO Dataset (1)

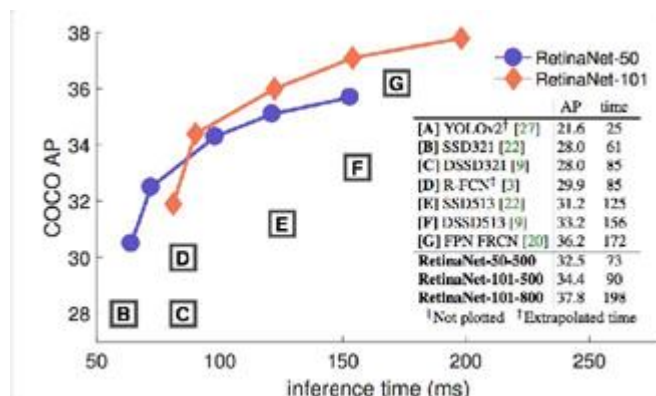




Figure 5. Speed (Ms) versus accuracy (AP) on MS COCO

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	73.9	85.5	82.9	76.6	57.8	62.7	79.4	77.2	86.6	55.0	79.1	62.2	87.0	83.4	84.7	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	89.4	49.4	75.5	57.0	87.5	80.9	81.0	74.7	41.8	71.5	68.5	82.1	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [27]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	68.8	75.9	71.4
NoC [28]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [32]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

Table 3: PASCAL VOC 2012 Leaderboard. YOLO compared with the full comp4 (outside data allowed) public leaderboard as of November 6th, 2015. Mean average precision and per-class average precision are shown for a variety of detection methods. YOLO is the only real-time detector. Fast R-CNN + YOLO is the forth highest scoring method, with a 2.3% boost over Fast R-CNN.

Figure 3 YOLO ROI on a frame (LPD)

IV. The Dataset:

The dataset we are going to use is:

Indian License Plates

(<https://www.kaggle.com/thamizhsterio>)

Which contains 10,000 images of Indian license plates along with their annotations. The reason to use this dataset is to retain accuracy. Indian license plates are a little different from license plates in other countries. Doing this will decrease the number of false positives.

V. Character recognition using OCR: (2)

OCR is short for Optical Character Recognition. OCR is “image processing technology which provides a convenient way to convert paper documents into a digital format.”. The bounding boxes from YOLO further proceed into OCR conversion, where the localized image is being converted into plain text for further computation. Here further computation references to the real-time matching and deducing flags for each match. Flag in the matching algorithm is giving a label to match found. If a match is found, the license plate number is labelled as 1, and a warning system is generated with license plate numbers with a positive match found. Along with the warning system, the database is also updated for a match with the timestamp of a match, which may later be converted to data in human-readable format, to be referenced later.

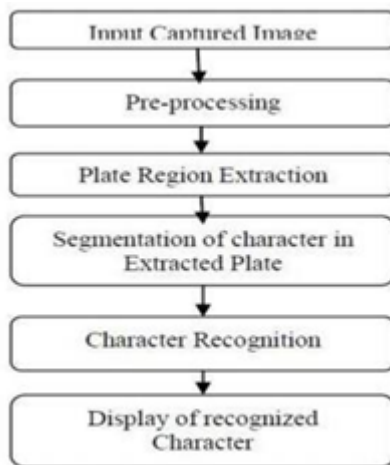


Figure 6. Working of OCR system

Optical Character Recognition works in two ways:

- a) Pattern Recognition: A numerous examples of texts in different fonts and sizes are fed into the system and compared with inputs to recognize the pattern and hence, characters.
- b) Feature Detection: OCR applications utilize the features of the scanned character to recognize a character, features like, number of curved lines, number of slopes, etc.

The OCR will give us the license plate number texts from the localized License Plate Number region of interest.

VI. The Presumed Database:

The system requires a method of targeting a license plate. The assumption made is that every targeted license plate is suspicion, the suspicion or suspect is made by traditional police work, or report. A database of license plates is maintained which may be from any criminal activity. The system currently has two working modes:

Active Mode: When the location of the suspect is narrowed down to a certain area, the license plates are fed into the system for real-time tracking. The system will actively look for top priority license plates and warn authorities, whenever a positive match is found.

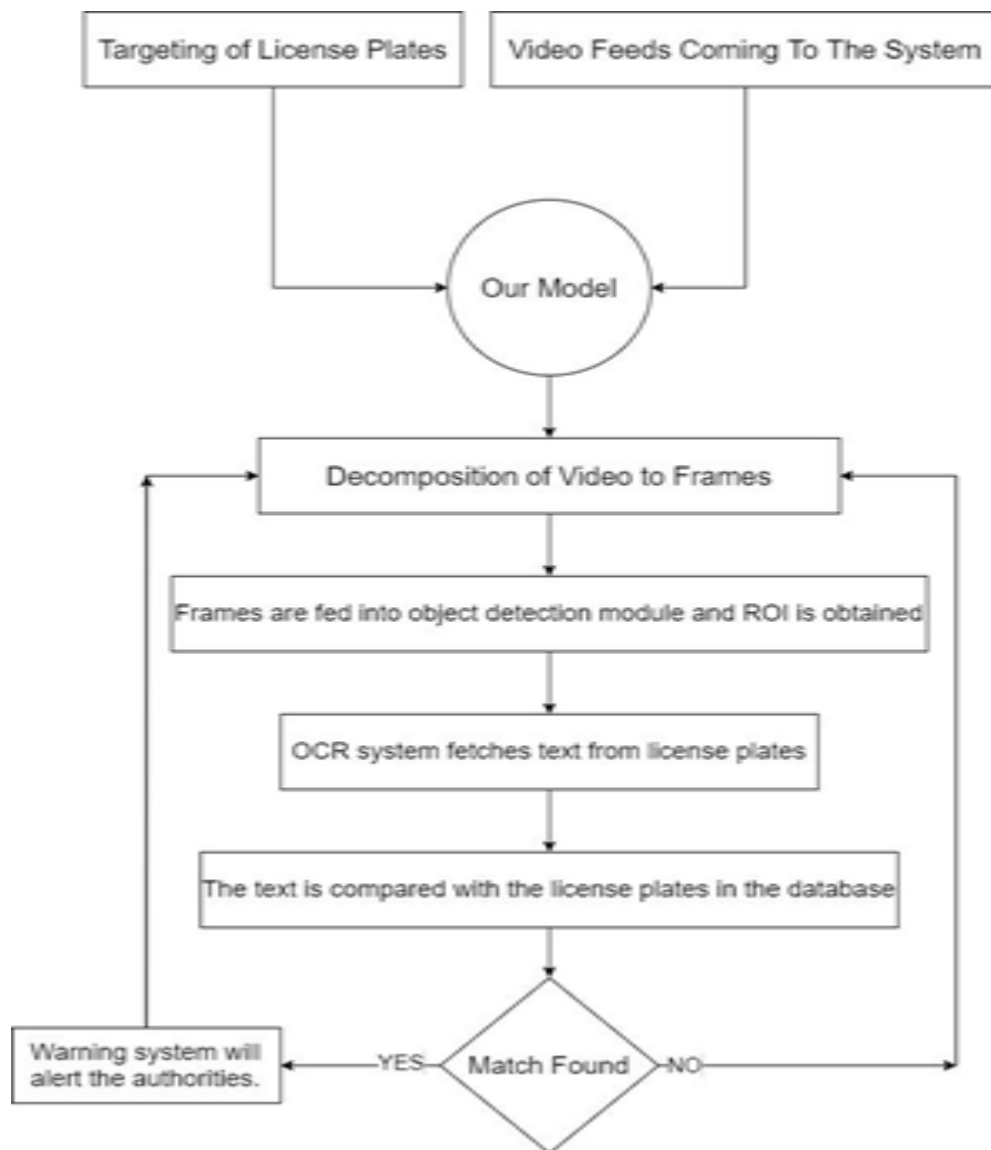
Passive Mode: When a database of flagged license plates is maintained, every license plate is targeted to gather as much information as possible. So passively working, our system will actively look for license plates in Flagged License Plate (FLP) Database and store information like, time of detection, location of detection, travel mapping, etc. Authorities will be able to fetch a detailed analysis of any specific license plates when needed.



Figure 7 Analysis of an FLP

VII. Working architecture of the system:

- a. Videos fed to the system
 - b. Video being decomposed
 - c. Frames are fed into the object detection module
 - d. Object detection module gives the output of ROIs of each detection
 - e. Using the co-ordinates of this ROI, OCR fetches the license plate number text
 - f. The text is then compared with the values in the presumed database of suspicious license plates.
 - g. If a match is found, a system of warning will be generated.
 - h. The Database is updated with each positive match
 - i. Later if an analysis on a certain license plate is done, a database is queried for positive matches and analysis can be shown.
 - j. The analysis consists of the timestamp and GPS coordinates of where the CCTV camera was on which the detection was made.
 - k. This data can be used to showcase on a map interactively.
 - l. This system gives two provisions, real-time and historical analysis.
 - m. Real-time is provisioned with real-time warning generation.
- Historical Analysis is analysing historical data, based on the database



Steps involved for Image acquisition and object ROI (Region of Interest) co-ordinates fetching.

A number plate is a pattern with very high contrast. If the number plate is too similar to the background it is difficult to see the location.

Brightness and brightness are changes as the fall of the transition to it. Morphological functions are used to extract the contrast element within the plate. The work is divided into several parts. The four basic algorithms of the ALPR system are:

1. Image Acquiring
2. Isolation of Plates
3. Breaking down
4. OCR Recognition

In the basic four-stage algorithm, first, the image containing the license plate is found, then the license plate is made locally followed by the separation of the license plate and the identity of the characters. The algorithm of the above-mentioned phase can be explained to

improve the efficiency of the system according to the need, to elaborate on this basic algorithm here we present the proposed system in the following detail:

Feeds are broken into frames and frame is inputted

Background unwanted data is removed

Plates are isolated

Characters are broken into pieces

There are two supported procedures:

Input Frame

RGB2GRAYSCALE Conversion

The image is diluted into Grayscale frames for better capturing of the characters and better reduction of noise.

Reduction of redundant data

C. Noise reduction:

We have used a central filtering technique to reduce the noise of paper and salt. We used a 3x 3 mask to get the eight neighbours of the pixel and its corresponding grey value. After using the moderate filtering technique, we will decay the image by the constructive element in any shape form. We find the morphological gradient to increase the edges, then we double the square to brighten the edges and fix in terms of double image.

Increase the contrast using the D histogram equalizer the contrast of each image is being magnified using the histogram equalizer technique.

E. Plate localization

The basic step for vehicle number plate validation is to find the size of the plate. Plates are usually rectangular. So, we find the edge of the rectangular plate. Mathematical morphology will be used to find that field. Using the Sobel adductor, we were using high litragens with high edge intensities and are identified by higher edges. Depending on the edge of the threshold value will be found from the input image.

F. Character split:

The Met Lab tool-box function provides a function called region Props (). It measures the set of properties for each labelled field of the label matrix. We use bounding boxes to measure the properties of the image field. After labelling the connecting components, this field will be corrected from the input image.

References

1. Joseph Redmon*, Santosh Divvala *†, Ross Girshick¶, Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection.
2. Redmon, Ali Farhadi. YOLOv3: An Incremental Improvement,
3. S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware CNN model. CoRR, abs/1505.01749, 2015
4. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014
5. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-CNN: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015
6. S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015.
7. R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.
8. J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In Computer Vision–ECCV 2014, pages 299–314. Springer, 2014
9. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014
10. Z. Shen and X. Xue. Do more dropouts in pool5 feature maps for better object detection. arXiv preprint arXiv:1409.6911, 2014
11. B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In Computer Vision–ECCV 2014, pages 297–312. Springer, 2014