Eurasian Journal of Analytical Chemistry
ISSN: 1306-3057 OPEN ACCESS 2018 13 (SP): 345-350

# Vehicle Traffic Analysis Using Yolo

C. Rajesh Babu, G. Anirudh*

Received 08 November 2018 ▪ Revised: 30 November 2018 ▪ Accepted: 06 December 2018

***Abstract:*** Traffic density specifically in the crowded urban areas is at an all-time high. It requires highly accurate and fast traffic analysis systems for capturing data to produce insights and for surveillance purposes. The data of vehicle traffic collected over a time period can be used to find traffic density patterns and procure insights which can be used for improving the traffic management. Existing hardware-based techniques for traffic analysis include magnet based loop detectors embedded inside the road provide useful data, but also has a significant downside: physical damage over a period of time, which reduces their functionality and accuracy. Even most of the software based techniques perform well to an extent, however they can only detect moving vehicles. To solve this issue, in this paper proposes to use a convolutional neural networks based algorithm known as You Only Look Once (YOLO). This paper proposes to create an end-to-end traffic analysis system which can take video as the input, process the video using YOLO algorithm and produce the output report using which insightful analysis can be obtained. The data is obtained from a surveillance camera to evaluate this model.

***Keywords:*** Yolo, Convolutional Neural Network, Deep Learning, Video-based System, Traffic Analysis.

## INTRODUCTION

Vehicle traffic analysis aims to count the number of vehicles on the road, presented in the video. This data is then recorded to produce the report. Such systems are widely used for surveillance and modern traffic management systems. Various image based methods have also been implemented, some of them include: Using Edge Detection, Blob Tracker Detection, Background Subtraction, Expectation Maximization Algorithm. Many of these methods have been employed in the past and have been successful to an extent in determining the vehicle traffic density[1].

Though the above mentioned techniques have achieved good progress, these works solely focuses on detection of moving vehicles only. This is not realistic as there may be occur various scenarios, such traffic jam and signals where the vehicles are in a static position over a period of time. In such cases, the above mentioned algorithms will not detect the vehicles[2]. Also, in a crowded environment, there occurs shadow and reflections which are inaccurately predicted by the existing models.

Therefore, to address the above issues, we propose a convolution neural network(CNN) based algorithm called YOLO to be used in the vehicle traffic analysis systems. This algorithm can detect static vehicles and also ignore the shadows and reflections. As a result, it produces a very accurate and fast detection which can be used in a traffic analysis system.

Our main contribution can be divided into three aspects:

- We propose an end-to-end CNN based vehicle traffic analysis system
- The proposed system uses YOLO algorithm, a fast CNN based algorithm than can overcome the shortcomings of existing techniques
- The result of the video detection is produced as a report which can be used for further analysis

The rest of the paper follows as – in section 2, the related work is reviewed. In section 3, the proposed model is detailed and explained. In section 4, the implementation of the system is explained. In section 5, the future work and scope for improvement is mentioned.

C. Rajesh Babu, Assistant Professor, Department of Computer Science, SRM Institute of Science and Technology. Email: rajeshbabu.c@ktr.srmuniv.ac.in

G. Anirudh*, Student, Department of Computer Science, SRM Institute of Science and Technology. Email: g.anirudh15@gmail.com

## RELATED WORK

In this section, we briefly review the previous work about dense crowd counting, background subtraction and blob detection

### Background Subtraction

Extraction of moving foreground objects from a static background is background subtraction. The foreground object is separated from the background based on its movements as the background remains static. The result of this technique can be used for specific target tracking and to detect the direction of movement of vehicles.

Its main use cases involve vehicle detection and monitoring, recognizing human actions, tracking human computer interaction, tracking movement of various objects and digital forensics [11].

Advantages include that is algorithm is extremely easy to implement, ease-of-use and its very fast.

Disadvantages include that the framing accuracy depends on object tracking speed and frame rate of the video. The memory requirements of the model is very high. The model does not give good results in the following conditions –

- If a bi-modal background exists
- If there are many slow moving or static objects in the frame
- If the video frame rate is low but movement of objects are fast and if there is abrupt changes lighting condition of the scene from time to time [23].

### Edge Detection

The edge of an image is the boundary where there are changes in parameters such as reflection of surface, changes in illumination and the distance between the viewer and the visible surface. Variations in physical aspects of the image can occur in various ways including the changes in parameters such as color, intensity and texture of the image.

It is used to detect an object based on the edge of the object. This is done by image analysis where the changes in the above mentioned parameters occur. The algorithm as soon as it detects a change in parameter values, analyses the texture and color of the image and finds the edge of an object[5].

The advantages are that it is one of the simplest and efficient methods available for day time vehicle density detection. Noises can be removed, so the image is directly detected and noises are filtered out without pre-processing. There are no complex mathematical operations, hence high speed performance.

The Disadvantages are that there is no mechanism for shadow removal and handling high occlusion areas. It produces comparatively poor results. There is chance for multiple vehicles to be counted as single object in high occlusion areas.

### Blob Detection

A square that is partially or fully filled is called a blob. and any square that can be reached from the original square by either horizontal or vertical movements. A blob can be detected by horizontal and vertical movements to traverse through pixels[5].

The background of an image is filled with static blobs, which is removed using blob filtering. Two types of blob detection techniques can be used –

1. Differential method which is based on the derivative function with respect to the image position
2. Local extrema methods which is based on finding the local minima and maxima of a function.

Using either or both of the methods, a list of blob is created to find the blobs containing only the vehicles. By using this list, the static blobs are removed.

However, in a cluttered environment where occlusion is high, blob detection cannot function properly as it cannot filter the overlapping vehicle images and the show of these images

## PROPOSED DETECTION MODEL

You only look once (YOLO) is a single convolutional neural network (CNN) that can predict multiple bounding boxes along with the class probabilities for those boxes simultaneously. YOLO directly optimizes performance as it trains on full images[1].

The Architecture of YOLO has 24 convolutional layers and 2 fully connected layers as shown in the figure.

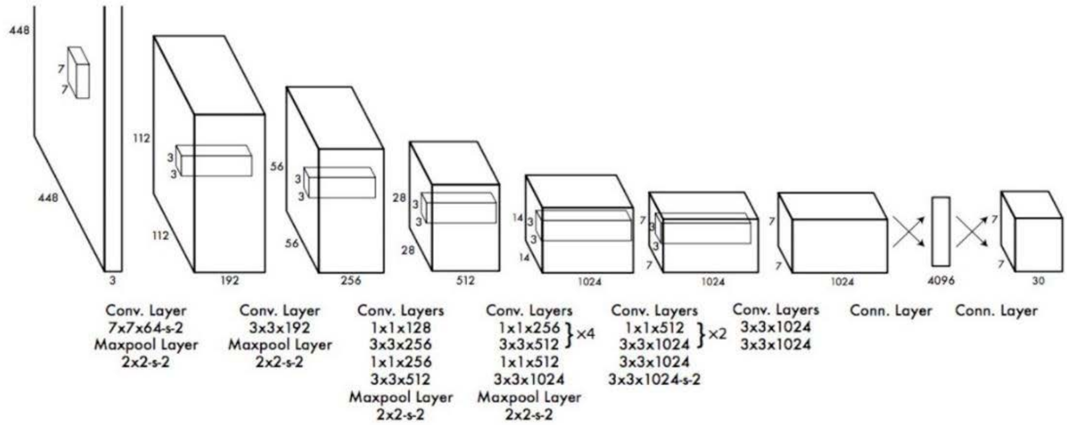


Fig.1: Architecture of YOLO

YOLO architecture is similar to that of a fully convolutional neural network and passes the image (n x n) once through the FCNN and the output obtained is (m x m) prediction. YOLO employs a single convolutional network that has the ability to predict multiple number of bounding boxes simultaneously and then calculate the confidence score for each box. Each and every grid cell produces the bounding boxes and generates the confidence scores for them. These confidence scores reflects the confidence level of the model as the box contains an object. The confident score of the cell should be zero, if no object exists in that cell.

During test time, the individual probability of the class is multiplied with box confidence score, to generate the confidence score of each box using[1]:

$$\Pr(\text{Class}i\ |object) * Pr(object) * IoU = Pr(Class\text{i}) * IoU \qquad (1) \qquad [1]$$

Where,

Pr(Class$i$/Object) is the individual probability of the class

Pr(Object) is the confidence score of the box

IoU = Area of overlap / Area of Union

Pr(Class$i$) is the confidence score of each box

Our system takes input in the form of video. YOLO algorithm takes the video and divides it frame-by-frame. Each frame is treated as an image. Background subtraction technique is applied to separate the static background. Then the difference of objects between the images is found so as to detect the moving and as well as static objects.

YOLO has several advantages. To begin with, YOLO is very fast [1]. It can produce highly accurate results without compromising on the detection speed.

Second, YOLO uses the full image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO uses the full image during training and testing and as a result can produce a quick and efficient image detection.

Third, YOLO can detect static and as well as moving objects. This is because, it divides the input video into image frames and computes the difference between the two frames to detect the object.

## EXPERIMENTAL ANALYSIS

The implementation proposed in this paper is executed in:

- OS – Windows 10 Pro 64 bit
- CPU – Intel® Core i7 – 8550U
- RAM – 16 GB
- No Graphic Card

### Training of Algorithm

YOLO requires some files to start training which includes a path to training frames in a text file, soring the action class name in a text file, saving the weight files in a path, YOLO architecture configuration file and Pre-trained convolutional weights.

The training of the algorithm is done using the common objects in context (CoCo) dataset. It contains more than 80 object categories but we use it train the YOLO algorithm to detect vehicles.

**Vehicle Analysis System**

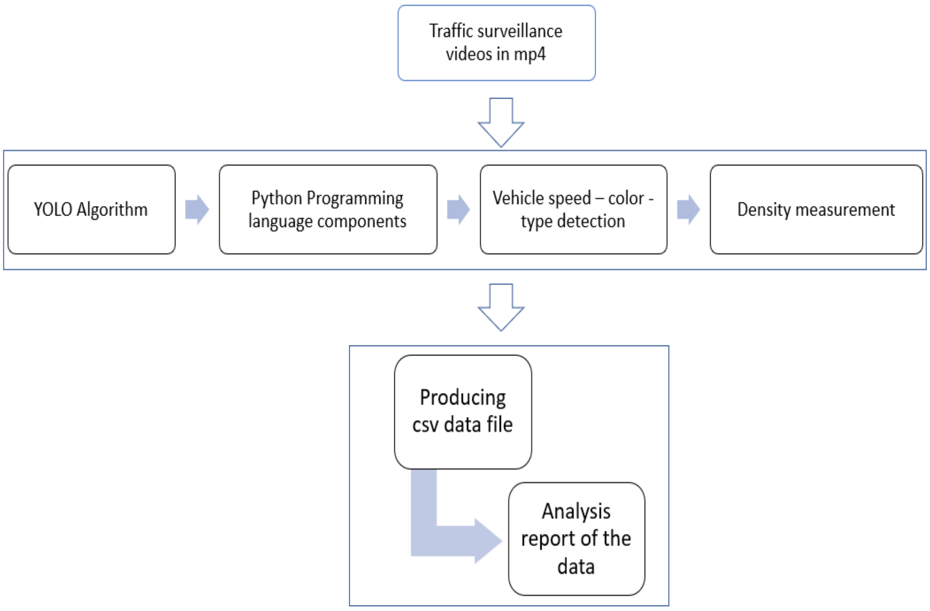


Fig.2: Vehicle traffic analysis system representation

The system architecture includes three components which is – Data Collection, Object detection and report generation.

Data collection is where the input video is collected in mp4 format. The object detection part involves the application of YOLO algorithm to the model for vehicle detection along with open CV and tensorflow api to detect the speed, color and type of the vehicle. All these details are then written in a csv file from which further first-level analysis report can be done.

The following is a figure, representative of the output produced by the vehicle traffic analysis system. Here, the vehicles are marked using a box that displays the type of the vehicle detected. This is done by using the YOLO algorithm along with tensorflow and OpenCV api as explained in the previous sections. The number of vehicles in the frame is also counted using the api. The counting accuracy can be found by:

$$Accuracy = \frac{No.Of\ Correct\ \ detections}{No.Of\ Ground\ \ truth\ detections} \qquad (2) \qquad [1]$$


Fig.3: Vehicle traffic analysis representation

The data collected from the system is then written in a csv file. Below is the representation for the same.

Table 1: Representation of the CSV file produced

| Vehicle type | Vehicle color | Vehicle speed in km/h |
|---|---|---|
| car | blue | 50 |
| truck | yellow | 70 |
| car | black | 75 |
| bike | red | 45 |
| truck | red | 47 |
| bike | pink | 40 |
| car | orange | 55 |
| car | white | 63 |
| truck | green | 50 |
| bike | blue | 43 |
| lorry | green | 52 |
| car | white | 43 |
| bike | blue | 77 |
| truck | yellow | 60 |
| bike | black | 30 |
| car | green | 32 |

By using the data in the table, a first level analysis report can be generated as shown in the below scatter plot representation.

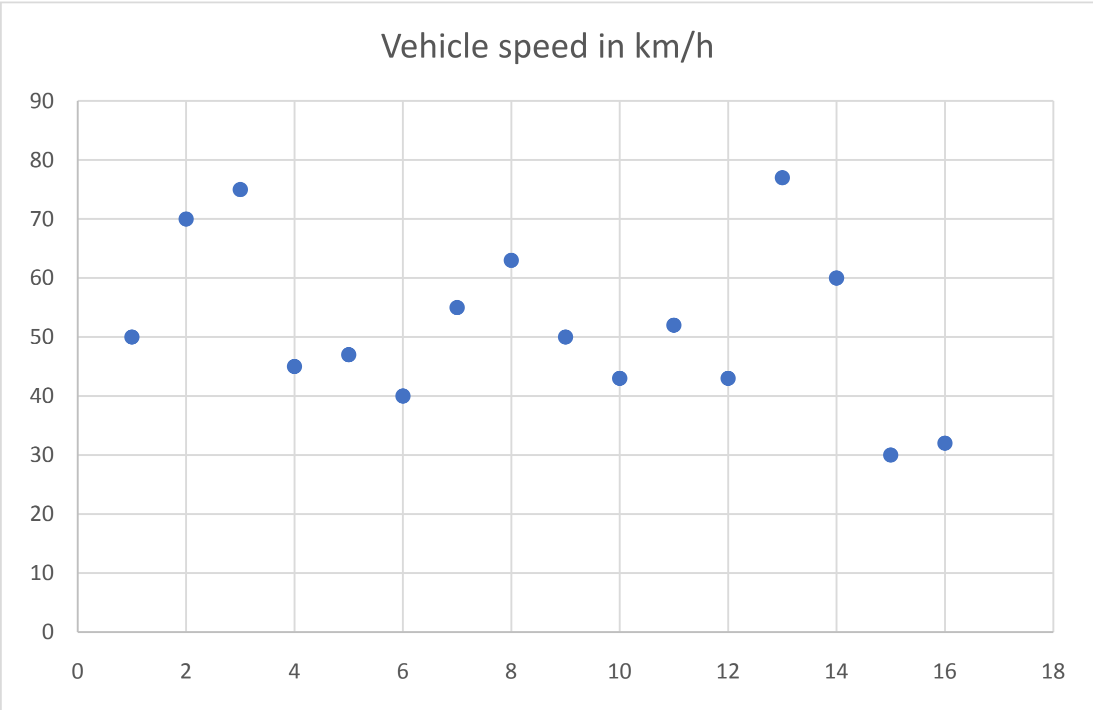


Fig. 4: Representation of the analysis report

The report is a scatter plot that visualizes the data produced and saved in the csv file. Using this, useful insights such from the produced data can be found, which can be used to improve the traffic surveillance and management.

## CONCLUSION

In this paper, an end-to-end traffic analysis system that uses the YOLO algorithm for vehicle detection is proposed. It takes a video in the format of mp4 as the input, applies YOLO algorithm for object detection, uses tensorflow api to detect the speed, color and type of the vehicle and finally produces the detected data in a csv file format, which can be used to produce first level analysis report. Traffic prediction using the generated historical data will be considered in the future works.

# REFERENCES


[1] J. Redmon, S. Divvala, R. Girshick, A. Farhadi (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

[2] S. Ren, K. He, R. Girshick, and J. Sun (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*.

[3] Ryus, M. Vandehey, L. Elefteriadou, R. G. Dowling, andB. K. Ostrom (2010).Highway Capacity Manual. Washington, DC, USA:Transportation Research Board.

[4] C. Ozkurt and F. Camci (2009). Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. Math. Comput. Appl, vol. 14, no. 3, pp. 187–196.

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov (2014). Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pages 2155–2162. IEEE.

[6] M. Sun, Y. Wang, T. Li, J. Lv, J. Wu(2017) Vehicle Counting in Crowded Scenes with Multichannel and Multi-task Convolutional Neural Networks, *J. Vis. Commun. Image R.*

[7] P. Viola and M. J. Jones (2014). Robust real-time face detection. International journal of computer vision, 57(2):137–154.

[8] C. L. Zitnick and P. Doll´ar (2014). Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014, pages 391–405. Springer.

[9] B. Hariharan, P. Arbel´aez, R. Girshick, and J. Malik (2014). Simultaneous detection and segmentation. In Computer Vision–ECCV 2014, pages 297–312. Springer.

[10] Nilakorn Seenouvong , Ukrit Watchareeruetai and Chaiwat Nuthong (2016). A Computer Vision Based Vehicle Detection and Counting System. IEEE Conference

[11] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun(2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.

[13] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W.Smeulders (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

[14] C. L. Zitnick and P. Doll´ar (2014). Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer.

[15] J. Dong, Q. Chen, S. Yan, and A. Yuille (2014). Towards unified object detection and semantic segmentation. In *Computer Vision–ECCV 2014*, pages 299–314. Springer.

[16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I.Williams, J. Winn, and A. Zisserman (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

[17] S. Gidaris and N. Komodakis (2015). Object detection via a multiregion & semantic segmentation-aware CNN model. *CoRR*, abs/1505.01749.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition(CVPR), 2014 IEEE Conference on*, pages 580–587.

[19] S. Gould, T. Gao, and D. Koller (2009). Region-based segmentation and object detection. In *Advances in neural information processing systems*, pages 655–663.

[20] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[21] V. Vinodhiniand and C. Rajesh Babu (2016). Spectral Resolution Enhancement Using Optimum Wavelet Filter for Multi Spectral Image. I J C T A, 9(15). pp. 7197-7204 © International Science Press

[22] S., Lakkimsetty, S.M., Amutha, B (2017). An effective way of accident detecting and notifying mechanism through android applications, Batchu. APRN Journal of apllied sciences.

[23] Amutha, B., Nanmaran, K (2015). An obstacle avoidance algorithm with spatial and temporal constraints for visually impaired. International journal of apllied research engineering.