Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 144 (2018) 259–268

Procedia Computer Science

www.elsevier.com/locate/procedia

INNS Conference on Big Data and Deep Learning 2018

Detection and classification of vehicles for traffic video analytics

Ahmad Arinaldi*, Jaka Arya Pradana, Arlan Arventa Gurusinga

Digital Services Division, PT Telekomunikasi Indonesia, Jakarta Pusat 10110, Indonesia

Abstract

We present a traffic video analysis system based on computer vision techniques. The system is designed to automatically gather important statistics for policy makers and regulators in an automated fashion. These statistics include vehicle counting, vehicle type classification, estimation of vehicle speed from video and lane usage monitoring. The core of such system is the detection and classification of vehicles in traffic videos. We implement two models for this purpose, first is a MoG + SVM system and the second is based on Faster RCNN, a recently popular deep learning architecture for detection of objects in images. We show in our experiments that Faster RCNN outperforms MoG in detection of vehicles that are static, overlapping or in night time conditions. Faster RCNN also outperforms SVM for the task of classifying vehicle types based on appearances.

Keywords: Traffic Video Analysis, Vehicle Detection, Vehicle Classification, Faster RCNN

1. Introduction

The goal of automated surveillance and monitoring systems is to remove the need of human labor for simple vision based tasks that can be performed by a computer or an automated system. The applications of computer vision systems have also been applied in various public areas such as roads, airports and retail areas. One such application of vision systems is in the task of monitoring and analyzing scenes of road traffic, with particular interest in monitoring highways and intersection. Such a system is required for effective real time traffic management systems that can detect changes in traffic characteristics in a timely manner allowing regulators and authorities the ability to quickly respond to traffic situations.

Some proposed applications for traffic video surveillance proposed are for vehicle re-identification in a multi-camera environment [18] whose potential future applications include finding travel time, traffic flow, and various traffic information relevant for policy makers and urban planners. Another approach proposed vehicle pose estimation [13], which has potential applications in vehicle tracking. A fixed camera based application of traffic video analysis is given by [10], where they used a CPU based system to count cars along with their speed on a highway. From the

* Corresponding author.
E-mail address: 925717@telkom.co.id

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.

10.1016/j.procs.2018.10.527

results of [10], they not only could count the number of vehicles but also estimate the speeds of the vehicle providing more informative traffic flow information. This information has proven valuable to urban planners and policy makers in the cities which such systems have been used.

The core of any such system that can be used effectively is the accurate detection and classification of the moving vehicles present in the video. The accurate detection of moving vehicles in a scene is a difficult problem but given a static camera position is somewhat relieved. The successful detection and classification of vehicle classes is essential to extract important traffic flow information needed by regulators such as vehicle counts, average estimated speed of the vehicles, driver behavior (for example preferred lane usage) and violations of traffic rules (such as trucks using the high speed lanes). For all of these, an accurate and reliable vehicle detector is required to extract the relevant information from the traffic scene videos.

Currently Deep Learning with a convolutional architecture (CNN) have emerged as a popular method for solving problems related to visual object recognition either in images or videos and has given state of the art performance in various visual recognition tasks, such as image classification, object detection and localization and image segmentation. For traffic vehicle analysis, deep learning has been used for large scale image based vehicle classification [19], where a deep CNN was used to classify images of road vehicles into 6 classes, including large buses, cars, motorcycle, minibus, trucks, and vans. They show that using deep CNNs for vehicle type classification achieves state of the art results on pre-cropped images containing only vehicles.

For the detection and classification of traffic scene videos, we implement and compare two systems. The first system we implement is a pipeline model comprising two different subsystems, the first is the mixture of Gaussian background subtraction which we use to detect the positions of the moving vehicles in the video. From these detection areas we form a bounding box of the image and classify the vehicle type using a support vector machine (SVM) classifier trained to classify 6 classes, including cars, delivery cars (pickups, vehicles with containers), trucks, large trucks, and buses. The second system that we implement and analyze in this paper is a system based on the deep learning architecture Region Convolutional Neural Networks (RCNN). The RCNN is trained to simultaneously detect and classify the vehicles in a video frame of the traffic scene. From these bounding box detection of the vehicles in the traffic scene, we can perform short term tracking on the position of these detected vehicles and we can extract information such as estimated speed and lane position.

In this paper, we compare the performance aspects of vehicle detection and classification using mixture of Gaussian (MoG) background subtraction + SVM vehicle classification model compared to the Faster RCNN based method that detects and classifies the vehicle classes simultaneously. From these experiments we find several weaknesses of the MoG + SVM system that make it unsuitable for traffic video analysis of dynamic scenes in a real world environment. We see that the Faster RCNN method as a appearance based method outperforms MoG in detecting vehicle that are overlapping or in low light night time conditions. We show in our experimental results from both quantitative analysis of the detection performance and the qualitative analysis based on cross validation classification accuracy that the Faster RCNN based method is more suitable for the problem of traffic video analysis. We then build a system that can estimate other important information about the vehicles such as estimated speed and lane usage from the results of the vehicle detection system. These are the main contributions of this paper.

2. Detection and Classification of Vehicles

2.1. Related Works

The problem of detecting and classifying objects in videos is an important problem to solve in building autonomous surveillance systems. Many algorithms have been proposed to help solve this problem, ranging from background subtraction based methods to classifier based approaches for detecting and classifying objects in videos. The various approaches to this problem each have their own advantages and disadvantages, and designing traffic video analytic solutions should consider the type of algorithm used to best fit the task at hand.

Background subtraction methods are methods for detecting new objects in an image that does not exist in a reference background image [2]. The basic principle is that a new image with several objects to detect is subtracted by the reference image producing a new image that encodes the difference between the two images. A threshold value is used to increase tolerance of background subtraction to noise that may be present in the video. Finally, a blob detector is used to detect the objects. Each blob is counted as one object, and detected. A classification algorithm can be passed onto the object to classify and fine tune the detected object results from background subtraction. A more complex method for background subtraction based on mixture of Gaussian (MoG) models that can not only detect pixels of foreground objects but also the shadows they cause is proposed in [20]. This model of background subtraction has also been implemented for detecting the movement of people in videos [1].

Detection of specific objects in images is a difficult, due to the nature of objects in images is that are often of different sizes, different orientations, and overlapping objects that causes occlusion of the object of interest to be detected. These problems require a detection algorithm that has several properties, such as translation invariance (invariant to different locations of object of interest in the image), rotation invariance (invariant to the rotation of the object in the image), and scale invariance (invariant to the size of the objects in the image). A common approach is to use machine learning methods that learn a representation directly from the available data to train a model. Popular methods use low level features such as SIFT [12], HOG [3], and Haar [17] combining them with a machine learning method to classify the objects. This approach is known as the "Feature + Classifier" approach.

2.2. *Faster Region Based CNN (RCNN) for Object Detection and Classification*

A new and popular approach is to use deep convolutional neural networks that can learn discriminative features directly from the input images for a specified task in a supervised manner. The deep convolutional neural network uses many layers of convolution filter sets that learn a hierarchical representation of the input image data, where lower level convolutional layers will learn to detect simple features such as lines and textures, while higher level convolutional layers will learn features that are combinations of the lower level features. Hence in this paper we use features that are learned from the available data using a convolutional neural network, where the feature kernels are learned from the available training data. This allows the model to build features that can generalize to the whole dataset. Currently deep convolutional neural networks have been proven to give state of the art results in visual recognition tasks, as in [11]. The power of deep neural networks come from the hierarchical representation of the features that such a deep neural network is able to build during training of the model. This allows deep CNNs to learn rich and meaningful features that prove powerful in classification problems. These rich features can be utilized for a variety of visual recognition tasks, such as classification, image segmentation, or even object detection as proposed in the RCNN architecture [5]. It is stated that the performance of the RCNN object detector can be improved significantly by using a CNN architecture for the feature extraction stage, such as using larger and deeper models such as VGG network [16] and the 50-layer residual network [7].

To build our system, we first require a model that is able to detect and find the locations of objects of interest to our task, for example humans and cars. For this purpose we choose the RCNN model which is a deep learning model designed specifically for detecting and localizing objects in an image. The RCNN model was first proposed by Girshick in [5]. The RCNN proposed in [5, 6] uses region based proposals based on a selective search algorithm that generates up to 2000 region proposals for one image during test time. These region proposals are then warped using an affine transform into 227x227 size images, that are then fed into a pre-trained neural network, for example AlexNet [11] and ResNet [7]. The resulting output of the deep convolutional network is a vector of feature size 4096, which is used as an input to an SVM that then is trained to classify these features into an object class. The SVMs perform a 1 against else classification hence for N classes objects to classify a number of N SVMs is required to be trained. This first model of the R-CNN has some disadvantages caused by the inefficiency of training 3 different modules, a region proposal module based on selective search algorithm followed by feature extraction using a deep convolutional neural network, and the SVMs used to perform classification.

These weaknesses are addressed by Fast RCNN proposed by the same authors in [4]. Fast RCNN modifies certain inefficient operations in traditional RCNN to improve training speed and accuracy of the model. First, Fast - RCNN gets rid of the SVMs used in classification of bounding box images and a bounding box regressor for fine tuning bounding boxes, and instead opts for using softmax classification and as a result they have combined formerly 2 models (a CNN feature extractor and SVM classifier) into one single model that needs to be trained only once. Second, is that Fast RCNN uses a bounding box regressor to fine tune the region proposals, this results in a higher accuracy and faster training due to this module being trained in conjunction with the classification network. It is claimed that these improvements allow the Fast RCNN to train up to 9 times faster compared to regular RCNN whilst at the same time achieving a higher performance. RCNN is slow since for every area of a proposed region bounding box, a forward pass of the CNN feature extractor and the SVM classifier is used. Fast RCNN proposes to rectify this by simplifying this process to only a single pass forward using the CNN of an image, hence for every image, the CNN is used only once. The features for each single bounding box then need to be extracted from the output of the CNN. This is achieved through a process called Region of Interest (RoI) Pooling, which extracts the appropriate features form the produced feature map. This is done due to the observation that multiple objects in an image share many features with other objects in said image, thus there is no need to recalculate these feature each time for every object in the image, hence the features need to be shared. These features that are in the form of a fixed length vector are then fed into a softmax classifier and a regressor to classify the object within the bounding box and fine tune the location that are optimized jointly.

Fast RCNN however still has certain drawbacks, caused by the selective search process that proposes that proposes regions from the image to the CNN. It is shown in [14, 15], that this region proposal stage can be replaced by a special neural network layer called the Region Proposal Network (RPN). This RPN takes as an input a whole image and then proceeds to generate several bounding boxes as region proposals for the next classification step. The RPN shares full image features with the convolutional neural network that is used to produce the image features, hence region proposal can be done at the same time as feature extraction and classification of the image. This RPN is trained to generate high quality region proposals, as a single network along with the Fast RCNN network. This RPN can be seen as a attention model that focuses the rest of the network to the region of interest. These RPNs are trained to predict a wide range of bounding boxes using a predefined set of anchor boxes, that are set at a range of scale and aspect ratio values. Anchor boxes are parameters used to determine the region proposal of the RPN, and for each box RPN not only outputs a four number bounding box area, but also a two element score that is the softmax probability of the box containing an object. In Faster RCNN, a pyramid scheme of several anchor boxes are used as the regression function. These anchor boxes have several properties that make them suitable for object detection and classification, such as translation invariance and multiple scale (and aspect ratios) representations. Training is then done in an alternating manner between the RPN and Fast RCNN network. Hence, Faster RCNN removes all of the unnecessary processing required by the selective search method to generate region proposals and the whole system can be trained as a single entity end-to-end. The scheme of Faster RCNN and the anchor boxes are given in Fig. 1.

3. Implementation Details

3.1. The Dataset

In our experiments, we use two different datasets of traffic videos. The Indonesian Toll Road dataset is our own dataset which was taken at two prominent Indonesian toll roads, which are the Jagorawi Toll and the Kapuk Toll. For the Jagorawi toll data, we record our videos at Ramp 2 Taman Mini Indonesia Indah (TMII) Jasa Marga, Indonesia (Coordinates: longitude = -6.287124, latitude = 106.877644). The video dataset is in 4096x2160 resolution with 22.0 frames per second. The dataset was taken from a pedestrian bridge manually using a camera. For the Kapuk Toll Road data, we record our videos at the Kapuk Toll Gate (Coordinates: longitude = -6.121997, latitude = 106.768772). In

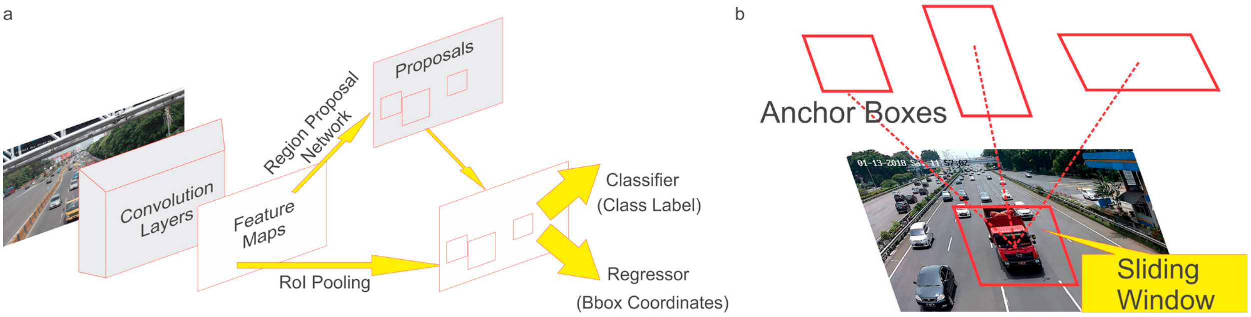


Fig. 1. (a) Faster RCNN architecture; (b) Faster RCNN anchor boxes.

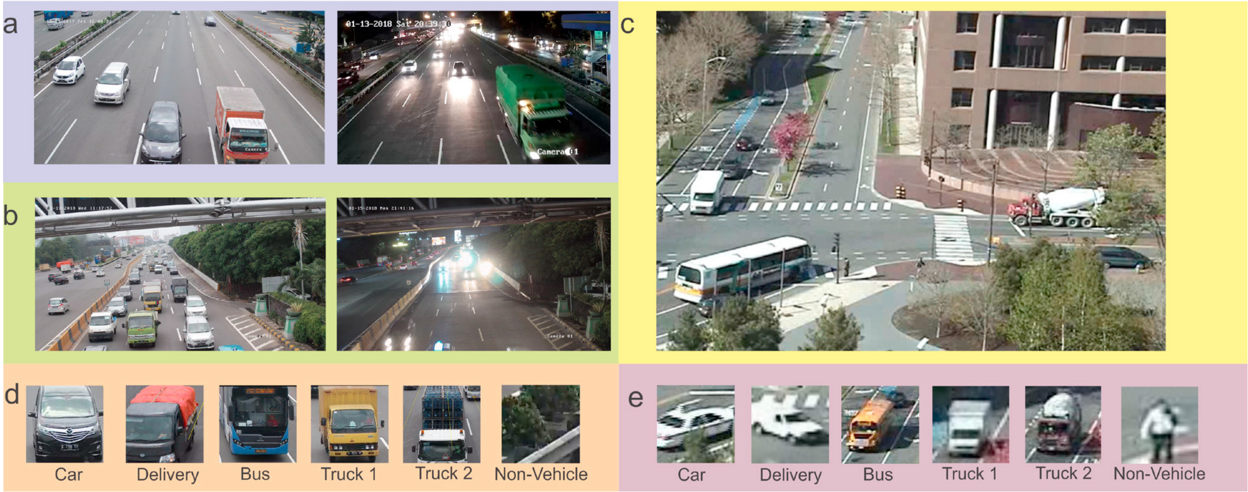


Fig. 2. (a) TMII toll scene; (b) Kapuk toll scene; (c) MIT traffic scene; (d) Indonesian toll vehicle classes; (e) MIT traffic vehicle. classes.

total, we collect almost a week worth of video data at these two locations, with night videos also included in our dataset.

The second dataset we use is a public dataset, MIT traffic, which is designed for research on traffic scene analysis and crowded scenes. It consists of a traffic video sequence 90 minutes long. It is recorded by a stationary camera. The size of the scene is 720 by 480. It is divided into 20 clips. Visualizations of the scenes in the datasets we use are given in Fig. 2.

For the MoG detection method, we use MoG to detect changes in the pixel values and subtract the background images leaving only foreground objects in the video. These foreground objects are then extracted using a bounding box segmentation to extract appropriate images to train an SVM model to classify the images based on the image class, which includes the five vehicle classes mentioned before along with an additional non-vehicles class to also learn the various false positives produced by the MoG background subtraction method. The images are then resized to 64x64 patches for the Indonesian Toll Road dataset and 32x32 patches for the MIT Traffic dataset. This is due to the distances of vehicles in the MIT traffic dataset being farther and thus the image patches are smaller.

For evaluating the SVM performance, we use 5 folds cross validation, i.e. a 80% training and 20% testing split on the image patches extracted using the MoG background subtraction method making sure that images of the same vehicle from the same video does not appear both in the training and testing sets. The classes are cars, delivery cars (pickups, vehicles with containers), trucks (trucks 1), large trucks (trucks 2), and buses. We try to keep a balanced distribution of the training/evaluation data of vehicle image patches for each of the classes to ensure the generalization capability of the trained model.

We train and evaluate the Faster RCNN model on images extracted from the traffic video scenes. From the images we extracted, we sampled 1058 images from various locations and conditions of the traffic scenes for annotation. We do the same for the MIT Traffic dataset, but we only sample 353 images. We use Faster R-CNN to train based on these images to detect the vehicle classes provided in Fig. 2. For evaluation purposes, we use a 5 fold cross validation scheme, i.e. a 80% training and 20% testing random split of the data.

3.2. Implementation Details of the MoG + SVM Vehicle Detector and Classifier

For the purpose of detecting and classifying vehicles in a traffic scene video, we propose 2 methods. The first method uses a combination adaptive background subtraction technique based on mixtures of Gaussian for the vehicle detection phase and an SVM classification stage to detect the type of vehicle. The diagram of this system is given in Fig. 3.

We use MoG as a background subtraction method to detect the moving vehicles in the image. The MoG method returns a mask of pixel values which it believes are foreground objects to be detected. This mask however is still very

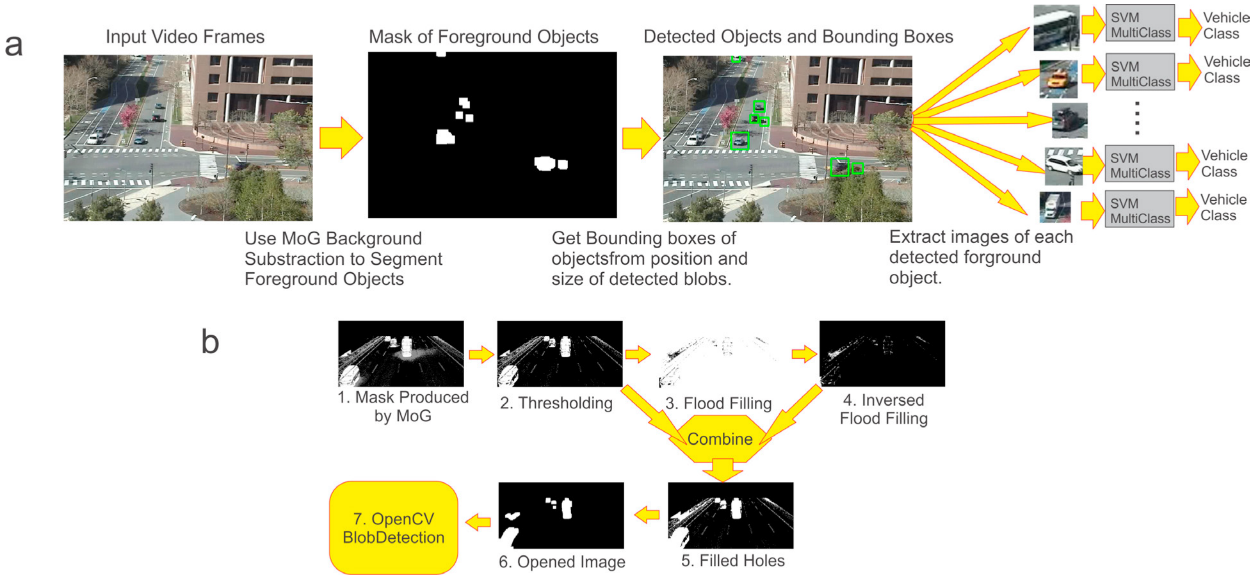

a
Input Video Frames
Mask of Foreground Objects
Detected Objects and Bounding Boxes
SVM MultiClass
Vehicle Class
SVM MultiClass
Vehicle Class
SVM MultiClass
Vehicle Class
SVM MultiClass
Vehicle Class
Use MoG Background Substraction to Segment Foreground Objects
Get Bounding boxes of objectsfrom position and size of detected blobs.
Extract images of each detected forground object.
b
1. Mask Produced by MoG
2. Thresholding
3. Flood Filling
4. Inversed Flood Filling
Combine
7. OpenCV BlobDetection
6. Opened Image
5. Filled Holes

Fig. 3. (a) Scheme of MoG+SVM method for detection and classification of vehicles in traffic scenes; (b) Detailed graph of image processing steps for detection of vehicle blobs from background subtraction.

rough and not suitable yet for blob detection to detect the vehicles. Hence, we require several image processing steps to detect blobs in this mask image, as shown in Fig. 3. The resulting detections are passed on to an SVM classifier to determine the class of the vehicle.

We use two types of SVMs in our experiments, the first is a Linear SVM model and the second is a radial basis function SVM (RBF-SVM). For the Indonesian Toll Road dataset, we set the input to the SVM as 64 × 64 color images that are flattened into a 1-Dimension vector. For the MIT Traffic dataset we use only 32 × 32 images considering the vehicles in the MIT Traffic dataset are considerably smaller than the vehicles in the Indonesian Toll Road dataset.

3.3. Implementation Details of the Faster RCNN Vehicle Detection and Classification

The second model we implement for the purpose vehicle detection and classification for traffic video analysis is a model based on the Faster RCNN [14, 15]. The Faster RCNN model works on the images, which are frames of the videos, and detects the bounding boxes and classes of any vehicles that may be present within said image frame, as we have shown in Fig. 1. We train the Faster RCNN on images where we have annotated the vehicles based on the six classes mentioned before and also the locations of the bounding boxes of these vehicles.

The Faster RCNN model thus can do both vehicle detection and classification of the vehicle types in one single model. To evaluate the Faster RCNN model we trained, the accuracy metric of bounding box classification of the vehicle types is used. To measure whether a bounding box has successfully enclosed a vehicle, we compare the bounding box area with the ground truth bounding box and measure the intersection over union. We declare the bounding box as correct if the intersection over union is more than 0.5.

The anchor boxes of the Faster RCNN determine the initial regions to be detected by the model as shown in Fig. 1. The sizes of the bounding boxes must be tuned to match the range of sizes of the objects that the model must detect. While the aspect ratio determines the ratio of the height and width of the objects to be detected. Due to the different scales of the vehicles in the Indonesian Toll Road data and the MIT Traffic data, we use different sets of aspect ratios and anchor box sizes. In the Indonesian Toll Road data, we use anchor box sizes of [64, 128, 256, 512] and aspect ratios of [1:1, 1:1.5, 1.5:1]. While for the MIT dataset, we use anchor box sizes of [32, 64, 128] and aspect ratios of [1:1, 1:2, 2:1].

3.4. Tracking Moving Vehicles and Vehicle Behavior

Our traffic video analysis system requires the estimation of driver behavior such as driving speed and lane usage. To achieve this, our system needs to perform more than simply detecting and classifying images of vehicles. To this end, we also need a tracking mechanism to track the movements of the vehicles that have been successfully detected by the image based methods. We make use of two tracking algorithms available in OpenCV, the median flow tracker based on [9] and the KCF tracker based on [8]. For the Indonesian Toll Road data we use the median flow tracker. This is due to its ability to detect regular motion of the vehicles moving in straight lines due to the restricting toll lanes and the limited region of interest making sure that scale changes of the tracked objects do not change significantly. While for the MIT Traffic dataset, we use the KCF based Tracker. We choose this tracker since the vehicles in the MIT Traffic dataset exhibit more diverse movement being in a traffic intersection, along with greater variety of scale changes in the tracked objects. Tracking is then done for 30 consecutive frames. During this time, we calculate various behavioral statistics of the vehicles, such as average speed and lane usage.

Once we have tracked the changes of the vehicle positions in 30 frame span where we perform short term tracking, we then estimate the vehicle speed. We use the principle of triangle similarity for this task. For this task, we need a marker object that is unique to that camera scene. Given that we know the real width of this marker object, W in meters, and it's position at a known distance from the camera, D in meters. We then measure the width of that marker object in pixels as it appears within the video, P in pixels. Given this marker object, we can then calculate the focal point of the camera, F as:

$$F = \frac{PD}{W} \tag{1}$$

Once we know the value of F, we can then estimate the distance of other objects in the camera, D' in meters. To do this, we need an estimate of the real world object width, which we can assume is a priori knowledge, since the width of vehicles in the real world is known, W' in meters. For this paper, we assume that vehicles and delivery cars have a width of 1.8 meters, while buses and trucks have a width of 2.5 meters. We also need to know the vehicle width in pixels, P' in pixels, for this we use the width of the bounding boxes detected by the vehicle detection algorithm. Using the previously calculated value of F, we can calculate the estimated distance of vehicles, D' to the camera using:

$$D' = \frac{W'F}{P'} \tag{2}$$

To estimate the speed, we need to calculate the difference in distance of the vehicle in question during a fixed time period. We use a difference of five frames to estimate the vehicle speed. This is done to take an average estimation of speed from those five frames, and to minimize the effects of an unstable tracking bounding box on the speed estimates. We know the video rate in frames per second, fps, and thus we can estimate the speed of the vehicle as:

$$Est.Speed = \frac{3.6(D'_t - D'_{t-5})}{\frac{5}{fps}} \tag{3}$$

We use the factor of 3.6 to convert the speed from m/s into km/hour. Using this method we avoid having to make complicated 3D calibrations of the video scene. Due to needing a marker object for the estimation of the vehicle speeds, we only perform speed estimation on the Indonesian Toll Road dataset and not on the MIT Traffic dataset. Our speed estimation method is quite different from the speed estimation method proposed in [10]. We rely on a fixed time equation (that is we calculate speed after tracking for 5 frames), rather than relying on a fixed distance.

The next thing we would like to analyze from the traffic videos is the lane usage of the various vehicle types in the traffic videos. This is required to make a profile of the lane usage and to detect violations of traffic rules by the drivers. The lanes in each scene are defined manually by an annotator for each different scene. These lanes are defined by two points, a start point and a end point of the lanes, which we denote as p_1 and p_2. Given the center point of the bounding box enclosing the vehicle in question in the 2D image as p_3, and all these points are represented as 2D vectors. e determine the distance of this vehicle to each of the defined lanes as follows:

$$Distance = \frac{|(p_2 - p_1) \times (p_1 - p_3)|}{|(p_2 - p_1)|} \tag{4}$$

4. Experimental Results and Analysis

4.1. SVM Vehicle Classification Results

For the first model we build, we use an SVM based classifier to classify the vehicle images detected using MoG background subtraction. The classifier is trained to predict the label of the given image patch into six classes, which include five vehicle classes and a Non-Vehicle class encompassing all background objects detected by the MoG background subtraction. The importance of this Non-Vehicle (background) class is for backwards error correction of the bounding boxes detected by MoG background subtraction and blob detection. These patches are then classified by the SVM classifier. We present the five fold cross validation accuracy results as explained in section 3, in Table 1 for the linear SVM and Table 2 for the RBF-SVM.

Table 1. Linear SVM five-fold cross validation results.

Data	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Indo. Toll Road	.419	.581	.581	.378	.432	.478 ± .096
MIT Traffic	.529	.599	.618	.606	.605	.591 ± .035

Table 2. RBF-SVM five-fold cross validation results.

Data	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Indo. Toll Road	.623	.613	.609	.402	.480	.545 ± .099
MIT Traffic	.394	.488	.413	.529	.471	.459 ± .055

4.2. Faster RCNN Vehicle Classification Results

The second model we implement for traffic video analysis is the Faster RCNN based model for vehicle detection and classification. This model performs both the detection and classification of vehicles in the image simultaneously. The results of the five fold cross validation accuracy is presented in Table 3.

Table 3. Faster RCNN five-fold cross validation bounding box classification accuracy results.

Data	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Indonesian Toll Road	0.509	0.779	0.788	0.777	0.507	0.672 ± 0.150
MIT Traffic	0.704	0.687	0.679	0.688	0.710	0.694 ± 0.013

We obtain these accuracy results based on the classification accuracy of the predicted classes, which are the five vehicles classes as stated before and a catchall background class. Hence, he Faster RCNN performs, as with the SVM based models perform classification on six output classes. To determine whether a box encompasses an object when compared to the ground truth labels, we use the intersection over union metric. Where a value of intersection over union higher than 0.5 indicates the object has been successfully detected. We give the results of the Faster RCNN vehicle detection and classification in Fig. 4. Detection and classification of vehicles for night time data poses a challenge for the MoG background subtraction vehicle detection detector due to the bright lights and various reflections in the traffic scene. We see that Faster RCNN does not suffer from this issue, since it is trained to detect vehicles based on appearances and not based on the changes in pixel value which is sensitive to lighting and shadows.

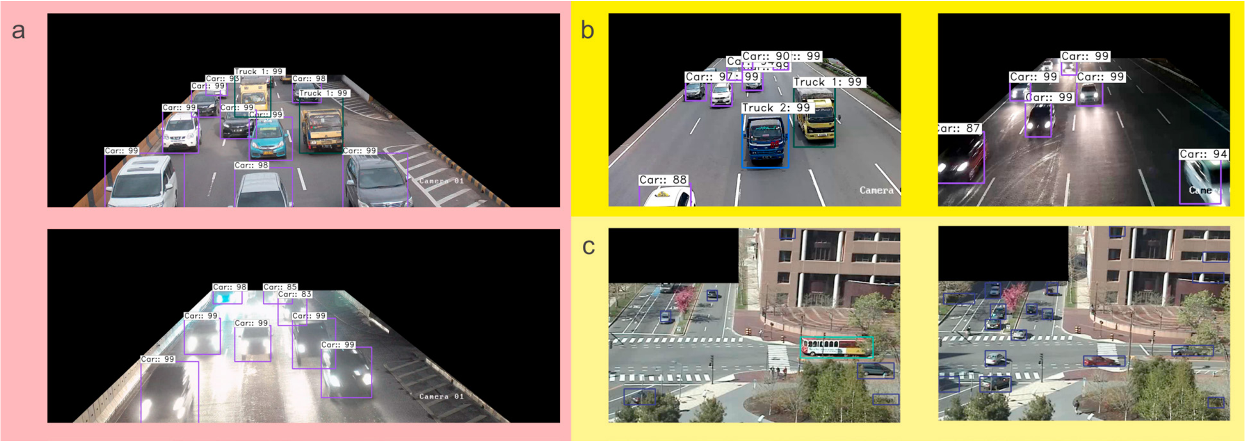


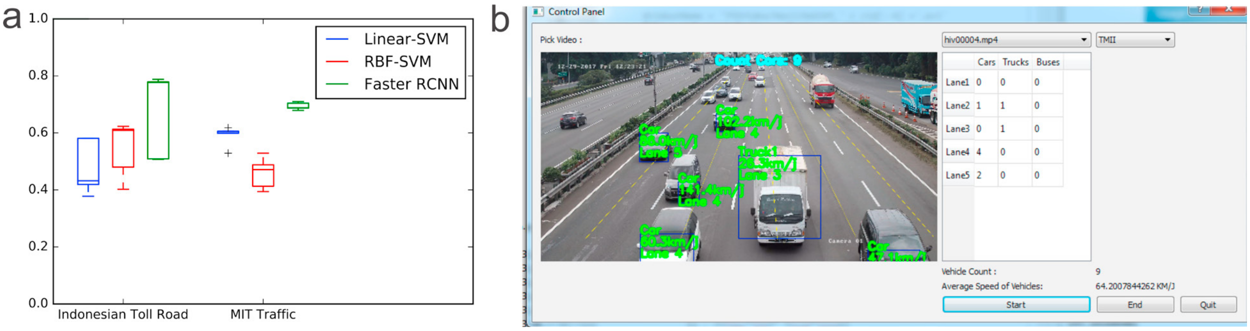
Fig. 4. (a) Results on the Kapuk Toll scene; (b) Results on the TMII Toll Scene; (c) Results on the MIT Traffic scene.

4.3. Comparison of Classification Accuracy Results

We compare the classification accuracy results of five fold cross validation between the SVM based models and the Faster RCNN model for the classification of vehicle types. We present these results in Figure 14, along with an example interface of our system. We can see that for both the Indonesian Toll Road dataset and the MIT Traffic dataset, Faster RCNN outperforms both SVM based classification models in terms of cross validation accuracy. Another interesting observation is that the ranges of accuracy results vary greatly in the Indonesian Toll Road dataset when compared with the results of the MIT Traffic dataset. We believe this is due to the greater variability present in the Indonesian Toll Road dataset, which not only consists of two different locations, but also varying times of lighting conditions (day and night). While the MIT Traffic dataset has only one scene with a uniform lighting condition (recording was done for only 90 minutes).

5. Conclusions

In this paper we have presented a system for automatic traffic video analysis. This system can automatically count the number of vehicles, classify the vehicles by type, estimate the speed of the moving vehicles and determine lane usage. To achieve this purpose, we have implemented and compared two systems for vehicle detection and classification, a system based on MoG background subtraction + SVM classifier and a system based on the Faster RCNN. From our experiments we see that the Faster RCNN is more suitable for the problem of detection and classification of vehicles in a dynamic traffic scene with moving vehicles. On the task of detecting moving vehicles in traffic videos, Faster RCNN outperforms MoG background subtraction. This is due to the weaknesses of Mixture of Gaussian background



Pick Video : hiv00004.mp4 TMII

	Cars	Trucks	Buses
Lane1	0	0	0
Lane2	1	1	0
Lane3	0	1	0
Lane4	4	0	0
Lane5	2	0	0

Vehicle Count : 9

Average Speed of Vehicles: 64.2007844262 KM/J

Fig. 5. (a) Comparisons of cross validation accuracy results; (b) Example of the system interface for traffic surveillance.

subtraction when dealing with vehicles that are overlapping, too close to one another, or even during night time conditions where pixel values are dominated by bright headlights. We also find that on the task of classifying the vehicle types based on appearances, Faster RCNN also outperforms the SVM classifier. We have also built an application that will allow users to easily gather traffic statistics from the analyzed videos using a graphical user interface (GUI).

References

[1] Arinaldi, A., Fanany, M.I., 2017. Cheating video description based on sequences of gestures, in: 5th International Conference on Information and Communication Technology (ICoIC7), pp. 1–6.

[2] Benezeth, Y., Jodoin, P.M., Emile, B., Laurent, H., Rosenberger, C., 2010. Comparative study of background subtraction algorithms. Journal of Electronic Imaging 19, 033003.

[3] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893.

[4] Girshick, R., 2015. Fast r-cnn. arXiv preprint arXiv:1504.08083 .

[5] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587.

[6] Girshick, R., Donahue, J., Darrell, T., Malik, J., 2016. Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence 38, 142–158.

[7] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[8] Henriques, J.F., Caseiro, R., Martins, P., Batista, J., 2015. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 37, 583–596.

[9] Kalal, Z., Mikolajczyk, K., Matas, J., 2010. Forward-backward error: Automatic detection of tracking failures, in: 20th international conference on Pattern recognition (ICPR), pp. 2756–2759.

[10] Kim, S.H., Shi, J., Alfarrarjeh, A., Xu, D., Tan, Y., Shahabi, C., 2013. Real-time traffic video analysis using intel viewmont coprocessor, in: International Workshop on Databases in Networked Information Systems, pp. 150–160.

[11] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, pp. 1097–1105.

[12] Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 91–110.

[13] Ozuysal, M., Lepetit, V., Fua, P., 2009. Pose estimation for category specific multiview object localization, in: IEEE Conference on Computer Vision and Pattern Recognition, pp. 778–785.

[14] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, pp. 91–99.

[15] Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster r-cnn: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence 39, 1137–1149.

[16] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .

[17] Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. I–I.

[18] Zapletal, D., Herout, A., 2016. Vehicle re-identification for automatic video traffic surveillance, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1568–1574.

[19] Zhuo, L., Jiang, L., Zhu, Z., Li, J., Zhang, J., Long, H., 2017. Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks. Machine Vision and Applications 28, 793–802.

[20] Zivkovic, Z., 2004. Improved adaptive gaussian mixture model for background subtraction, in: Proceedings of the 17th International Conference on Pattern Recognition, pp. 28–31.