Automatic Vehicle Identification and Classification

Darren Karl A. Sapalo
College of Computer Studies
De La Salle University, Manila
2401 Taft Ave, Malate, Manila, 1004 Metro Manila
Email: darren_karl_sapalo@dlsu.edu.ph

***Abstract*—This paper presents an automatic vehicle identification and classification system to be able to estimate the volume of vehicular traffic from video sequences. The two primary goals of this research is (1) to be able to classify vehicles commonly found in the Philippines into six classifications: sedan, bus, jeep, truck, SUV, and others; and (2) to be able to estimate the number of vehicles that pass the road. The video processing is done offline (i.e. not in real-time) and relies on manual calibration of lane dividing lines. The primary reference (technical paper) of this automatic traffic surveillance system used two features: size and linearity. This extension research reports design challenges of automatic traffic surveillance systems in the Philippine context and offers its recommendations and opportunities for future work.**

Keywords—Intelligent traffic systems, image processing, vehicle classification, vehicle counting

I. Introduction

A goal of intelligent transportation systems is to extract traffic information such as traffic density from video streams of traffic [1]. Traffic surveillance systems aim to detect, track, and classify vehicles. Being able to detect and classify vehicles can aid in estimating traffic density and pollution levels in cities and can help traffic enforcers coordinate traffic flow.

II. Methodology

The vehicle identification and classification approach is based on the research of [1].

A. Data Collection

In this research, there were three (3) RGB video dataset collected, lasting an average of around three (3) minutes using a hand-held Samsung Galaxy S4 smart phone. The initial resolution of the videos were 1920x1080 and were reduced to 480x270. The video datasets were recorded from three places: the first video dataset was collected from the fourth floor of the Gokongwei Building of De La Salle University, Manila (DLSU) facing Taft Avenue. The first dataset covers only one side of the road, with only one general direction of vehicle movement.

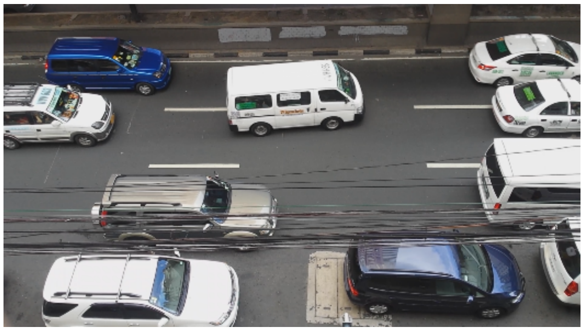

Fig. 1. Problem of occlusion: Electrical cables and parked vehicles in the Gokongwei dataset

Gokongwei Dataset. The first dataset inherently had problems of occlusion and presented common vehicle behavior based on the location: there were some electrical lines that occluded the road and vehicles and there were also some vehicles that would double-park (parking beside another parked vehicle). Figure 1 shows how the region of interest has been reduced, and how parked vehicles can affect vehicle flow on the road.

Buendia Dataset. The second venue for data collection was on the Light Rail Transit 1, Buendia Station, facing the Makati Central Business District. This dataset was able to capture both sides of the road, capturing two directions of vehicle movement left road (moving downwards) and the right road (moving upwards). This dataset proved to be better than the first one because it did not have the electrical cables that added occlusion. However, because there was an intersection underneath the LRT station, cars on one side of the two roads frequently had stalled vehicles because of the traffic stoplight as seen in figure 2. This will greatly affect segmentation based on background differencing. Common vehicles in the second dataset are sedans, jeeps, motorcycles, buses, SUVs, trucks, and others, ordered by most frequent to least frequent.

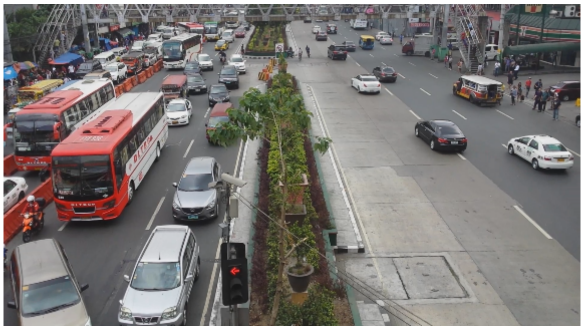

Fig. 2. Problem of stalled vehicles: Traffic has accumulated on the southbound road because of the stoplight in the Buendia dataset

Another issue with this dataset is that jeeps park on the side of the road as seen in figure 3. These parked vehicles waiting

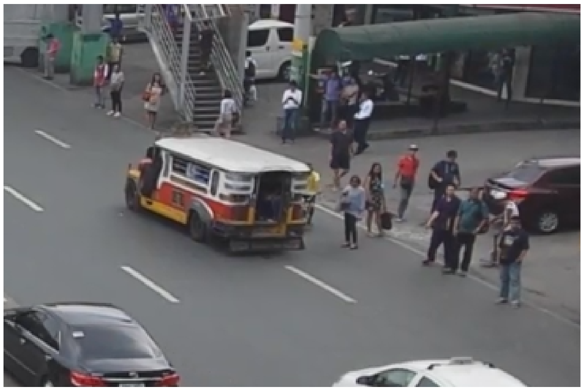

Fig. 3. Problem of parked vehicles: Vehicles such as jeeps and buses park on the side of road and in the middle of lanes to load and alight passengers

for passengers to board and alight will adversely affect the segmentation process. The background modeling module will have erroneous results as vehicles that do not move for a long time are gradually learned as part of the background.

EDSA Dataset. The third venue for data collection was on top of a walkway across EDSA Avenue, near Evangelista Street, Makati City. The video sequence faces northbound with vehicles moving southbound and lasts approximately three minutes for a total of 3567 frames. Common vehicles in the third dataset are sedans, motorcycles, SUVs, jeeps, buses, and trucks, ordered by most frequent to least frequent.

The common vehicles in the Philippines can be categorized as *sedans*, *buses*, *jeeps*, *SUVs*, *trucks*, and *others*. These are the classes used by this system for vehicle classification. An example of these classes are shown in figures 11 to 17. The *others* class includes vehicles such as motorcycles and vans.

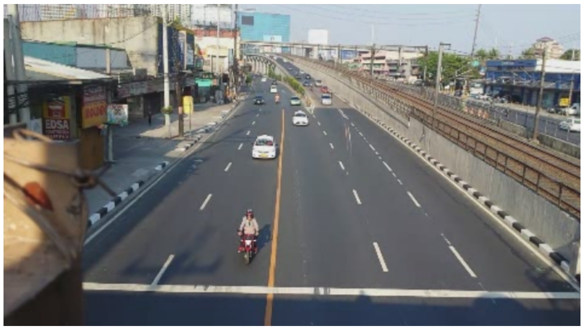

Fig. 4. Vehicles in EDSA Avenue moving southbound. There were no parked or stalled vehicles in this dataset.

Most of the noise in the traffic dataset was caused either camera movements caused by the hand, or by erroneous refocusing of the imaging device (Samsung Galaxy S4). Whenever the camera re-focused, there were significant illumination changes in the data. This occurred in tests when using other imaging devices as well (iPhone 5s).

In the study of [1], shadows were very prominent in their traffic video dataset. They propose a shadow elimination algorithm that replaces the pixel value on the region of interest with the pixel value on the background model if it has a high probability that it is a shadow pixel. The probability of being a shadow pixel was computed based on the mean and variance of shadow pixels acquired from numerous training data on shadows.

B. Stabilization

First, the video dataset was stabilized so that small camera movements caused by the minimal shaking of the camera can be reduced, if not removed. In the data collection of the EDSA dataset, a makeshift tripod was used to remove small camera movements caused by hand movements.

C. Background Modeling

Once the video has been stabilized and the movement between frames has been reduced, background modeling can be performed. The video is first converted to grayscale before the background model is acquired. There are various ways to achieve a background model from a video sequence. Initial experiments performed a simple **weighted background learning approach** to acquire the background model B_i at time i:

$$B_i(x,y) = B_{i-1}(x,y) \cdot (1-\alpha) \ + \ I_i(x,y) \cdot \alpha$$

Majority of the known background model is acquired from the previous background model B_{i-1}. The background model learns at a rate of α, acquiring only that much information from the current image I_i. Various values for α was tested: 0.05, 0.01, 0.005. This produced multiple background models which were manually inspected to evaluate its performance. A higher value for α meant that the background model learned the background faster: non-moving and slow moving objects in the scene will be accepted as the background faster. This proved to be unsuitable for the dataset, as there were slow moving vehicles. However, having too small a value for α meant that it will take longer before the background model is completed. Also, stalled vehicles, which is incorrectly learned as background, will take longer to be unlearned because of the stoplight. Empirical studies showed that the optimal value for α was 0.005.

D. Segmentation By Image Differencing

After acquiring a sufficient background model, the next step is to perform image differencing between the captured image I_i and the background model B_i to produce the preliminary result of segmentation D_i.

$$D_i(x,y) = | \ I_i(x,y) \ - \ B_i(x,y) \ |$$

The formula above acquires the distance of the current image I_i to the background model B_i by getting the absolute difference of their values. Afterwards, the binary image which represents the region of interest R_i is acquired by performing thresholding.

$$R_i(x,y) = \left\{ \begin{array}{ll} 1 & D_i(x,y) \geq D \\ 0 & otherwise \end{array} \right\}$$

If the value found at D_i is above the average error D then it means that it is part of the foreground. This produces a segmentation mask.

E. Segmentation

Although initial experiments made use of the simple weighted background learning approach in conjunction with the image differencing, an alternative tool was used to acquire the segmentation mask.

Authors of [2] developed an open source background subtraction library that provides basic to more advanced background subtraction methods. The system is available at https://github.com/andrewssobral/bgslibrary. The alternative algorithm used for segmentation was the Sigma-Delta background subtraction algorithm by [4]. The algorithm proved be effective in isolating the vehicles from the background. Figure 5 shows the segmentation result of a few cars approaching from the distance. A road mask was also applied so that noise outside of the road can be removed.

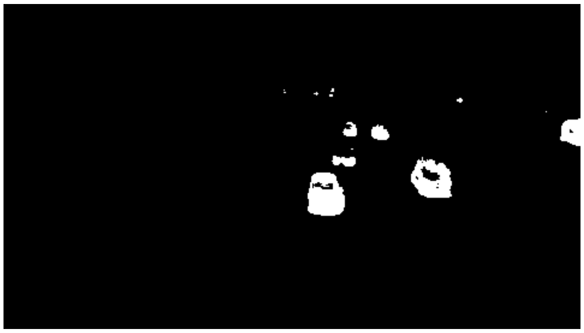

Fig. 5. The result of segmentation using the $\Sigma - \Delta$ background subtraction algorithm by [4].

F. Vehicle identification

The vehicle identification and classification algorithm used in this research is based on [1], which makes use of size and linearity as the features for vehicle classification.

The first step in vehicle identification is to detect *Blobs* (the white regions in the mask), acquire their area, and filter the blobs based on their area within a certain threshold. Blobs with area between 500 → 15000 were kept as possible vehicles. These values were acquired by analyzing the sizes of vehicles from a far distance and near the video camera. By performing this filtering, we remove the small occurrences of noise seen in figure 5.

The second step is to track the movement of the blobs with respect to the previous frame. This allows us to understand what blob in the previous frame is the same blob in the current frame. Doing this allows the system to detect the movement of a vehicle.

The third step is to acquire and normalize the features of the vehicle: size and linearity [2]. The first feature is the **size feature**, normalized by dividing the area of the blob by the width of the lane dividing lines W_{Lane_i} at the centroid of the blob p at lane i:

$$W_{Lane_i}(p) = | \quad X_{DL_i}(y_p) - X_{DL_{i+1}}(y_p) \quad |$$

The width of the lane dividing line is the horizontal distance between two adjacent lane dividing lines X_{DL_i} and $X_{DL_{i+1}}$. [1] provides a way to automatically detect lane dividing lines by analyzing a 2D histogram populated by the the positions of the centroids in the blobs in a video sequence. However in this research, the lane dividing lines are manually calibrated.

The second feature is the **linearity feature**, which analyzes the up-slanted edges of a vehicle. [1] provides figure 6, illustrating the concept of linearity. The truck on the left shows low linearity because its up-slanted edges has missing parts in the blob. On the other hand, the up-slanted edges of the bus fits perfectly on a straight line with minimal error, however it has some useless boundary points on the right side of the blob.

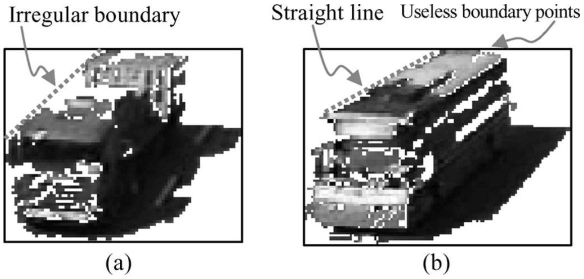

Fig. 6. Illustration by [1], showing the difference an irregular boundary of a truck (low linearity) and the straight line (high linearity) of a bus.

To be able to acquire the linearity of a blob, first the set of up-slanted edges U_{H_i} is acquired from the blob. However, because there are useless boundary points that will affect the linearity of the set of points, such as those found in figure 6b. The set must be filtered.

Given a set U_{H_i}, let the minimum bounding box be B_{H_i}. Let the distance of a point q in the set U_{H_i} to the bounding box be $d_{B_H}(q)$, and the maximum distance to the bounding box $d_{B_H}^{max} = max_{q \in U_{H_i}}\{d_{B_{H_i}}(q)\}$. The filtered set $\overline{U}_{H_i}$ is defined as:

$$\overline{U}_{H_i} = \{p \,|\, p \in U_{H_i}\,,\ d_{B_H}(p) > 0.25 d_{B_H}^{max}\ \}$$

The filtered set $\overline{U}_{H_i}$ is the set of points in U_{H_i} that is at least 25% of the maximum distance $d_{B_H}^{max}$. The above equation is an reduction of [1]'s definition of the the filtered set, which initially had another condition to filter pixels that have a high probability of being a shadow.

Using the points in the set $\overline{U}_{H_i}$, the error function below is minimized [3]:

$$E(b, m) = \sum_{i=1}^{N} (y_i - b_k - m_k x_i)^2$$

The values of m and b can be acquired by:

$$m_k = \frac{1}{K}\left(N\,\bar{X}\,\bar{Y} - \sum_{i=1}^{N} x_i\,y_i \right)$$

$$b_k = \frac{1}{K}\left(\bar{X}\,\sum_{i=1}^{N} x_i\,y_i - \bar{Y}\sum_{i=1}^{N} x_i^2 \right)$$

TABLE I. TEMPLATE INSTANCES PER CLASS

	Sedans	Bus	Jeeps	Trucks	SUVs	Others
Count	250	52	53	14	238	153

where $\bar{X}$, $\bar{Y}$, and K are the following:

$$\bar{X} = \frac{1}{N}\left(\sum_{i=1}^{N} x_i\right)$$

$$\bar{Y} = \frac{1}{N}\left(\sum_{i=1}^{N} y_i\right)$$

$$K = N\,\bar{X}\,\bar{X} - \sum_{i=1}^{N} x_i^2$$

A vehicle with its up-slanted edges such as that of figure 6b will have very minimal error when it is plotted on the straight line model. This means that $E(b, m)$ will have a value closer to 0. However, a vehicle with an irregular boundary such as figure 6a will have points that will not fit in a line perfectly. This means that there is significant error; $E(b, m)$ will have a value farther from 0.

[1] defines the linearity of a vehicle H as:

$$Linearity(H) = exp\left(-\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - mx_i - b)^2}\right)$$

By using the exp function, the linearity values of a vehicle is bounded between 0 (high error) and 1 (minimal to no error).

G. Vehicle classification

This research classifies vehicles into 6 classes: sedan, bus, jeep, truck, SUV, and others. [1] classifies vehicles based on a vehicle template library. A template has the values of the features (size and linearity) and the corresponding vehicle classification.

Table I shows the number of template instances per class in the vehicle template library. A limitation in the dataset is the minimal templates available for instances of trucks, jeeps, and buses..

To classify a vehicle, the mean and the variance of the templates within a vehicle class VC_k is acquired. Given the jth template V_j^k of vehicle class k, the mean of the rth feature is defined as:

$$m_k^r = \frac{1}{n_k}\sum_{i=1}^{n_k} f_r(V_i^k)$$

and the variance of the rth feature is defined as:

$$\sigma_{r,k} = \sqrt{\frac{1}{n_k}\sum_{j=1}^{n_k}(\,f_r(V_j^k) - m_r^k\,)^2}$$

Given a vehicle H_i and a template V_j^k in the vehicle classification VC_k, the similarity between the vehicle and the template is:

$$S_k(H_i, V_j^k) = exp\left(-\sum_{r=1}^{2}\frac{(f_r(H_i) - f_r(V_j^k))^2}{\sigma_{r,k}^2}\right)$$

The similarity of a vehicle H_i to the whole vehicle class VC_k is defined as:

$$S_k(H_i|VC_k) = \frac{1}{n_k}\sum_{V_j^k \in VC_k} S_k(H_i, V_j^k)$$

The probability of the vehicle H_i to be classified as vehicle class VC_k is the similarity to the class k over the total sum of similarities to all of the classes:

$$P(VC_k|H_i) = \frac{S(H_i|VC_k)}{S_{sum}(H_i)}$$

The classification given to a vehicle H_i is the class with the highest probability. This system assigns the classification to a tracked vehicle after the vehicle has been tracked for more than 20 frames and has passed a certain horizontal line on the road, denoting that the blob's Y position has changed from state A (far in the distance) to state B (closer to the camera).

III. RESULTS

The performance evaluation of the vehicle classification and counting is recorded using the EDSA dataset which contains 3567 frames. The total number of vehicles counted by the system was 111 vehicles, while the actual number of vehicles is 105.

Table II to Table VII shows the confusion matrices of each of the classifiers. This is presented in the Figure 7 as an overview of classifier performances.

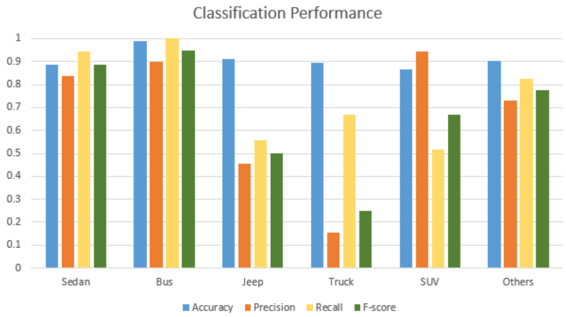

Fig. 7. Accuracy, precision, recall performance of classifiers

The results showed good performance for tracking sedans and other vehicles but did not perform as well on the other classes such as buses, jeeps, trucks, and SUVs. This may be attributed to the low number of template instances for jeeps, trucks, and SUVs encoded into the vehicle template library as seen in table I. The low number of encoded template instances

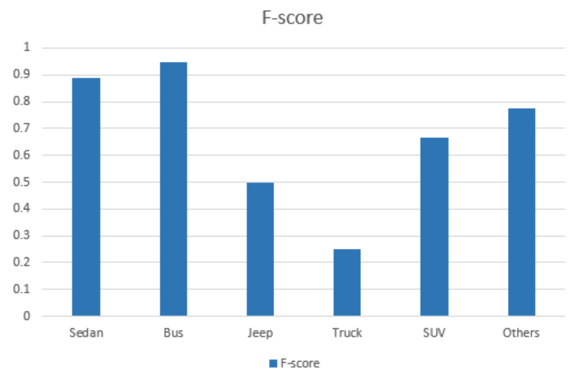

Fig. 8. F score performance of classifiers

TABLE II. SEDAN CLASSIFICATION CONFUSION MATRIX

Sedan		Prediction	
		False	True
Actual	False	50	10
	True	3	51

was directly affected by the low number of vehicles that were actually in the dataset, seen in Table IV to Table VI.

Figure 8 shows the F-score performance of the classifiers. The results show that only Sedan, Bus, and Others had an F-score that is above 0.70. Trucks had a lower F-score because it had a low recall and precision of 66.67% and 15.38% respectively.

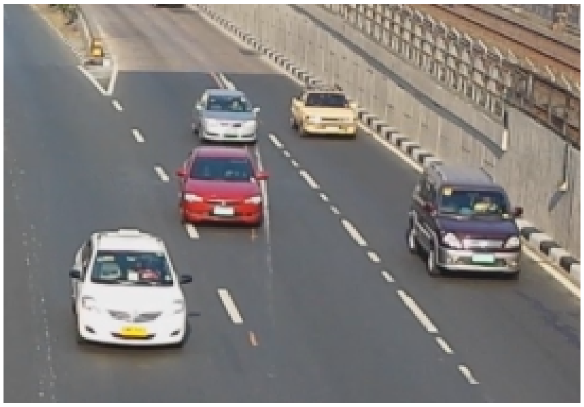

Fig. 9. Vehicle occlusion greatly affects bus classification

One of the causes of incorrectly classified buses is the inherent problem of occlusion in computer vision which is not part of the scope of this research. As seen in figure 9, vehicles near each other may cause the segmentation process to incorrectly detect separate vehicles as connected blobs. This greatly affects bus classification which highly relies on its size feature.

The Sedan classifier had an accuracy rate of 88.60%, recall of 94.44%, and precision of 83.61%. Its error rate was 11.40% with specificity of 83.33%.

The Bus classifier had an accuracy rate of 99.10%, recall of 100%, and precision of 90%. Its error rate was 0.90% with specificity of 99.02%.

TABLE III. BUS CLASSIFICATION CONFUSION MATRIX

Bus		Prediction	
		False	True
Actual	False	101	1
	True	0	9

TABLE IV. JEEP CLASSIFICATION CONFUSION MATRIX

Jeep		Prediction	
		False	True
Actual	False	98	11
	True	1	2

The Jeep classifier had an accuracy rate of 91.30%, recall of 55.56%, and precision of 45.45%. Its error rate was 8.70% with specificity of 94.34%.

The Truck classifier had an accuracy rate of 89.29%, recall of 66.67%, and precision of 15.38%. Its error rate was 10.71% with specificity of 89.91%.

The Sedan classifier had an accuracy rate of 86.61%, recall of 51.52%, and precision of 94.44%. Its error rate was 13.39% with specificity of 98.94%.

The Others classifier had an accuracy rate of 90.43%, recall of 82.61%, and precision of 73.08%. Its error rate was 9.57% with specificity of 92.39%.

Figure 10 shows a visualization of the instances in the vehicle template library, showing the size (y axis) and the linearity (x axis) of the instances and colored by class.

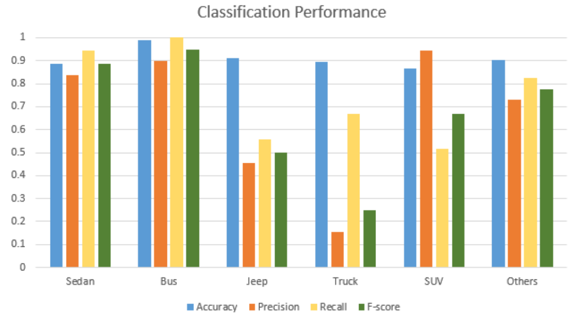

Fig. 10. An excerpt of the visualization plotting the clustering of instances in the vehicle template library.

The visualization shows the high variance in the features of sedan instances (green) as seen in their varying sizes. Bus instances (blue) were distant from the main cluster and are easily discriminated by their large size. Two clusters of the other instances (red) can easily be seen: one that clusters around $size = 1$ which is interpreted to be van instances, and

TABLE V. TRUCK CLASSIFICATION CONFUSION MATRIX

Truck		Prediction	
		False	True
Actual	False	98	11
	True	1	2

TABLE VI. SUV CLASSIFICATION CONFUSION MATRIX

SUV		Prediction	
		False	True
Actual	False	93	1
	True	16	17

TABLE VII. OTHERS CLASSIFICATION CONFUSION MATRIX

Others		Prediction	
		False	True
Actual	False	85	7
	True	4	19

one that clusters around $size = 0.25$ which is interpreted to be motorcycle instances. With the number of vehicle instances in the dataset for SUVs (orange), trucks (gray), and jeeps(yellow), the results show that their size and linearity features are clustered together and are not yet sufficient for classification.

IV. CONCLUSION

This research was able to collect and analyze vehicle datasets, to identify common problems in vehicle traffic datasets in the context of the Philippines, and to implement a variation of [1]'s vehicle identification and classification system. Results of the classification and the visualization of the vehicle template library confirm that sedan and motorcycle classification shows promising clustering results. The main challenge of bus classification is the problem of occlusion, while other classes (jeep, truck, SUVs) require more data instances to improve the classification or more blob/vehicle features to be introduced.

Noise in vehicle traffic dataset include occlusion problems caused by electrical wires. A challenge in background modeling is the common occurrence for vehicles such as jeeps, buses, and shuttles to park either because they are boarding or alighting passengers or because of traffic stop-lights. Also, the dataset shows that there are vehicles that park in the middle of two lanes, for as long as twenty seconds just to board, alight, and wait for passengers.

V. RECOMMENDATIONS

Various improvements can be made with this research. A larger dataset can be collected and more instances encoded in the vehicle template library to improve the performance of the vehicle classification approach. Future works can study other properties of blobs (such as convexity) to see if there are other features that can be used aside from size and linearity. Vehicle classification under different settings such as varying weather conditions (sunny, rainy, cloudy) or varying traffic conditions (traffic stand-still) can be explored. The main algorithm for vehicle detection and classification will need to adapt when performed on a dataset that has heavy stand-still traffic, because the background modeling approach used in this research will not be as effective on slow or non-moving vehicles.

APPENDIX A
CLASS INSTANCES OF THE VEHICLES

Figures 11 to 17 show instances of the classes found in the EDSA dataset.

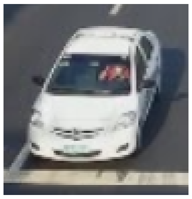

Fig. 11. A sedan is the car type that is most common in the dataset, with low height and is often used by taxis. Example of these are Toyota Altis, Vios, etc.

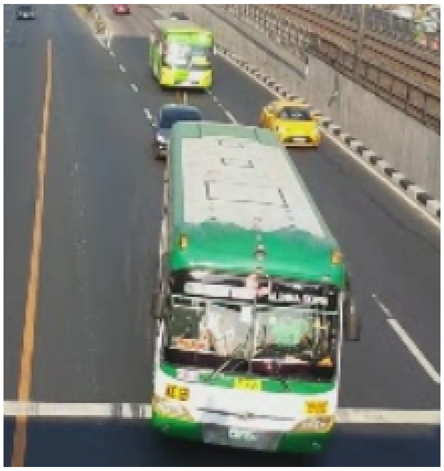

Fig. 12. Buses in the Philippines are long, tall, and wide. They occupy almost the whole space on a lane.

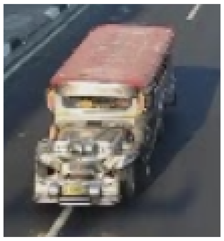

Fig. 13. Jeeps or Jeepneys are vehicles that are long with average height. These vehicles stop on the side of the road to alight and pick up passengers. The reflection of the sunlight on the road reflected from the metal casing of the jeep is sometimes recognized by the segmentation module.

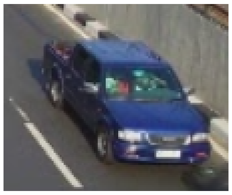

Fig. 14. The distinguising feature of trucks is their rear luggage space that can carry equipment or large objects. Trucks aren't as common in the third dataset. Industry or delivery trucks of companies were not found in the dataset.

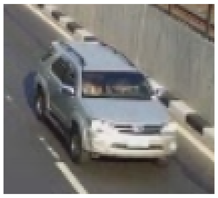

Fig. 15. Sports utility vehicles (SUVs) are larger than sedans and usually have mechanisms on the top of the vehicle for carrying equipment. Examples of these are Fortuner, Montero Sport, CRV, etc.

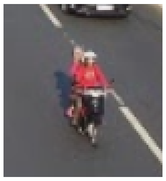

Fig. 16. Motorcycles were found in the dataset, although bicycles and scooters were not found. These vehicles fall under the *others* category.

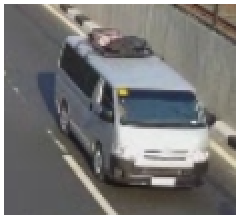

Fig. 17. Vans are long vehicles, that can carry more passengers than sedans. These vehicles fall under the *others* category.

ACKNOWLEDGMENT

The author would like to thank Dr. Joel Ilao of the College of Computer Studies, De La Salle University, Manila for his contribution as the adviser on this research project.

REFERENCES

[1] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen and Wen-Fong Hu, "Automatic traffic surveillance system for vehicle tracking and classification," in IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 2, pp. 175-187, June 2006.

[2] Sobral, Andrews. BGSLibrary: An OpenCV C++ Background Subtraction Library. IX Workshop de Viso Computacional (WVC'2013), Rio de Janeiro, Brazil, Jun. 2013.

[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, Numerical Recipes in CThe Art of Scientific Computing. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[4] Antoine Manzanera and Julien C. Richefeu. A new motion detection algorithm based on - background estimation. Pattern Recognition Lett. 28, 3 (February 2007), p. 320-328.