A simple vehicle classifier in a video sequence

Juan David Gil López
Ingeniero de Sistemas y Computación
Universidad Tecnológica de Pereira, Pereira, Colombia
Email: juangilopez@sirius.utp.edu.co

Juan Sebastián Silva López
Ingeniero de Sistemas y Computación
Universidad Tecnológica de Pereira, Pereira, Colombia
Email: sebastian@sirius.utp.edu.co

***Abstract*—This paper describes the implementation of a simple algorithm based on computer vision and machine learning for vehicle classification in a video sequence of a camera located in an specific position on the street. This algorithm can be divided in several stages as follows, Background substraction, vehicle, detection, tracking and vehicle classification. This report will be focused briefly on the first step and in a more detailed way on the latter. For background substraction an implementation of Mixture of gaussians was used, for vehicle classification a simple feature classifier was implemented, taking advantage of the camera position, a few features were extracted from each detected vehicle that allowed to try several techniques in order to achieve the goal of classifying vehicles in four categories. The tests shows that in the constrained scenario presented here, this is a reliable algorithm for vehicle classification.**

Keywords*—Computer vision, Traffic counting, Background susbtraction, Mixture of Gaussians, object tracking, Traffic Surveillance, Vehicle classification

I. INTRODUCTION

In the purpose of extracting traffic variables there has been many ways of completing this task, using manual counting, using microwave sensosrs and inductive loops. As an alternative to these methods, the use of surveillance cameras has been increasing in order to extract traffic variables, because is less expensive and less invasive than other techniques, in this implementation a camera above the street is placed in order to record a video and after that run the algorithm to perform the traffic counting and classification(see figure 2). A traffic counting system using cameras can be divided in four stages, segmentation, detection, tracking and classification.

In segmentation, the objective is to extract the foreground objects from the background, in this case we will be extracting moving objects, which makes the segmentation easier than in an static image. Several techniques have been developed for segmentation using movement, each one of those methods has advantages and disadvantages, among the most popular methods are: Background substraction, optical flow and statistical learning. The background substraction techniques are the most used techniques, beacause of the disadvantages of the other methods [1]. The background substraction was introduced in [2].

For the classification stage there are a lot of methods , that can be easier or more complex depending on the geometrical and spacial constraints of the site where the counting is going to take place. For example in [3] the authors used several kernelised support vector machines to make the vehicle classification, 3d colours histograms are used as features to represent cars and using a road-side camera they achieved good classification rates. In [4] the authors used a camera in front of the road, similar to the implementation presented here, they created a data set with serveral vehicles in which only the front of the vehicle was extracted analogue to the face extraction in an eigenface implementation, with this data set and the eigenvectors extracted, when a new vehicle was detected it's eigenvector was extracted and compared with the eigenvectors dataset using the minimum distance method, achiveving classification rates of 100%.

In this work, the specific background substraction algortithm used was described in [5] y [6], supported by the opencv library. In the classification step, a simple feature classification scheme is proposed and proved with several linear and non linear classifiers, showing the accuracy obtained by all the different technicques. This report is then organized as follows: in section IIA the segmentation algorithm used is briefly discused to show an insight of how it works, in section IIB the feature extraction part and the classification is explained, finally in section III and IV results and some conclussions are shown related with this work.

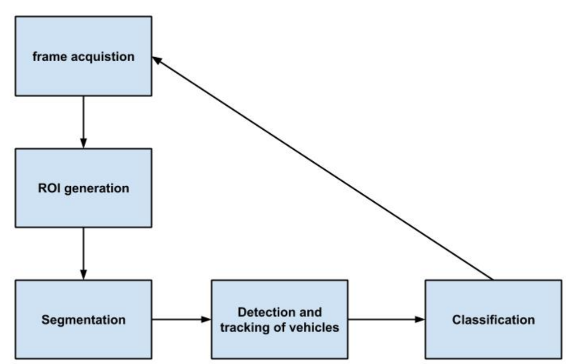

Figure 1. General Pathline of an implementation of a traffic classification and counting system

II. METHODS AND MATERIALS

A. Background substraction

In general terms, a probabilistic model for the process of background subrtraction can be formulated as follows: for each pixel value in time, call it $x(\vec{t})$, will be categorized as a foreground(FG)/background(BG) object depending on an estimation with and uncertainty [5], using a bayesian decision R:

$$R = \frac{p(BG|x(\vec{t}))}{p(FG|x(\vec{t}))} = \frac{p(x(\vec{t})|BG)p(BG)}{P(x(\vec{t})|FG)p(FG)} \quad (1)$$

Given the fact that there is no previous knowledge of the scene as well as the specific form or position of the objects in it, it's complicated to establish how will it be the relationship between the FG and BG pixels, in that way you assume that $p(FG) = p(BG)$, this means that initially the probabilty of a pixel being FG or BG is the same, also it's supposed an uniform distribution for FG value over time so $P(x(\vec{t})|FG) = c_{FG}$. In that way a pixel can be classified as BG if the next condition is reached:

$$p(x(\vec{t})|BG) > c_{thr}(= R_{c_{FG}}) \quad (2)$$

Where c_{thr} is a fixed threshold. Thus $p(x(\vec{t})|BG)$ corresponds to the model of the background that you have for an specific time. It's recmonded to the reader to read more about this model in [5]. The advantage of using the method proposed in [5] is related with the fact that it has a shadow detection implementation, and this with some other postprocessing stages gives an improved segmentation, The achieved segmentation looks like the one example showed in the figure 3, the black and white image is frequently called the foreground mask FM.

Figure 2. Image of a different recording with the same camera placement in the implementation presented here, note that the vehicles are "approaching to camera". The blue box represents a vehicle detected in the image. image taken from [7].

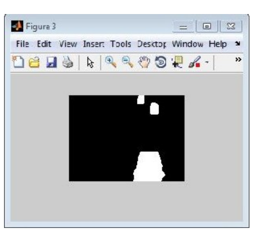

Figure 3. Example of a foreground mask. image taken from [8]

With the FM a blob(elements in white) detection and tracking over frames algortihm was implemented, when the blobs fullfilled the conditions of a car movement, a picture of the vehicle was taken, obtaining vehicle images like the ones in figure 4.

B. Classification

When a blob was detected as a vehicle, only three variables were extracted from the vehicle image obtained, the width and height of the image where mainly taken as the main features, and given the fact that the constraints of the enviroment in which the camera was placed placed,one can assume that the size of each vehicle image also gives an intuition of its type. For this implementation we assume four types of cars: private car, taxi, Large vehicle and motorcycle.

Given the extracted variables before, one can suppose that it's not enough to use that variables in order to classify between the classes to decide(see figure 5), for example between the private cars and taxis there is a dificulty to decide whether the detected was the first or the second, for that reason was neccesary to introduce a new variable, in this case, related with the color .

In the scatter plot in figure 5 it‘s intuitively to think that if a variable measuring the proportion of pixels yellow painted over the total amount of pixels in the vehicle image was given, then the cars with yellow colors such as taxi could have this measure greater than the ones that are not yellow colored (due to in the dataset, all the yellow cars are taxis, there are not yellow private cars), and then in a 3 dimensions space, a classification surface or hyperplane could work in order to separate data. In that way a measure of the yellow color level in the vehicle image was extracted, transforming the original RGB image obtained to another encoding where the information related with which color a pixel represented could be extracted. Thus, the HSV encoding was the one suitable to fulfill these task, because the H channel is related with the kind of color that pixel is representing. After transforming the vehicle image, the result obtained for extracting the yellow level was the one ilustrated in figure 6 .

Figure 4. Some of the vehicle pictures taken in some of the recordings developed

So three variables where extracted, width, height and yellow color proportion. With those three variables was neccesary to implement an strategy for classifying the vehicles in a corect manner, and given the fact of the constraints and the considerations taken before, it was possible with a simple classifier to find a linear or non linear function capable of decide which car type a vehicle image represents. For this implementation, several classifiers worked before were adapted for this problem, trained and used to make decisions. All the data set is min-max normalized; each feature in feature vector has values between 0 and 1.

For the purpose of this work four classifiers were implemented:

- Linear least squares classifier(LSQ).

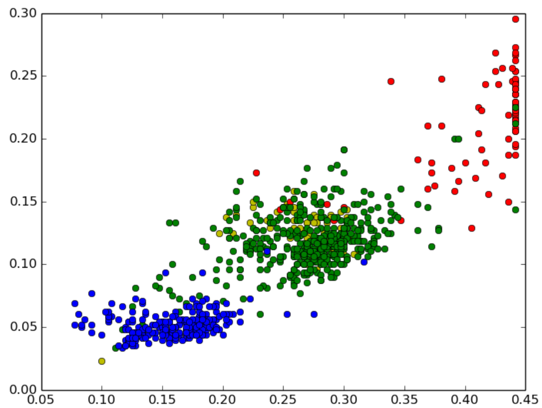

Figure 5. Each point represents the dimensions of the vehicle image (height,width). blue (large cars), red (private cars), yellow(taxis), green(motorcycles).

Table I. MEAN ACCURACY FOR LINEAR CLASSIFIERS, WITH STANDAR DEVIATION

	acc.	std. dev
LSQ	0.916	0.0002
LNB	0.747	9.21 e -5
LR	0.958	8.85 e -5

- Linear naive bayes classifer(LNB).
- Linear regression(LR).
- Neural networks as: multiple binary classifiers(one class vs the other ones), multiple class classifier and multiple regression.

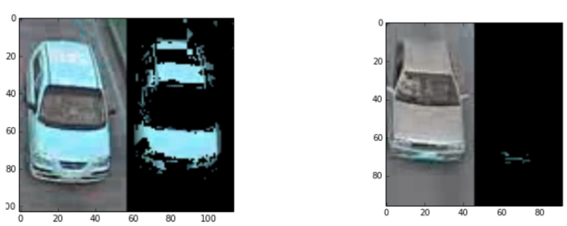

Figure 6. Vehicles with yellow color level highlighted, to the left a taxi, to the right a private car.

III. RESULTS

A dataset of 840 vehicle images was extracted, with this dataset the training and validation was made. The video from which the vehicle images were extracted was recorded previously in one of the main city avenues of Pereira. For each classification method a cross validation of five partitions was implemented, then, the mean accuracy of the prediction over a validation data set was obtained with a respective value of deviation. In table I the results of the three main linear classifiers is consolidated.

It is possible to see that the best classifier in this case is the linear regression. With neural networks several tests where developed, depending on the variable parameters, for example: the number of neurons(m), the type of basis function(bf) and also the learning rate(lr). For this implementation only M and BF was changed in order to see the differences of accuracy. Nevertheless in the case of multiple regression a variation in the learning rate was made because there was precission problems, and changing the learning rate helped to fix them.

Table II. RESULTS WITH MULTIPLE BINARY CLASSIFIERS

	param	r1	r2	r3	r4	r5
NMBC	*m*	7	7	4	2	7
	lr	0.1	0.01	0.01	0.01	0.01
	bf	sig	sig	sig	sig	tanh
	acc	0.747	0.964	0.966	0.873	0.705
	sd	0.0006	2.97e-5	8.71e-5	0.0001	0.006
	best	0.779	0.97	0.97	0.886	0.779

Table III. RESULTS WITH MULTIPLE CLASSIFICATION

	param	r1	r2	r3	r4	r5
NMC	*m*	7	7	4	2	7
	lr	0.1	0.01	0.01	0.01	0.01
	bf	sig	sig	sig	sig	tanh
	acc	0.747	0.964	0.966	0.913	0.705
	sd	0.0006	7.97e-5	3.71e-5	0.0001	0.006
	best	0.779	0.973	0.976	0.934	0.779

Table IV. RESULTS WITH MULTIPLE REGRESSION

	param	r1	r2	r3	r4	r5
NMR	*m*	7	7	7	12	7
	lr	0.1	0.01	0.0001	0.0001	0.0001
	bf	sig	sig	sig	sig	tanh
	acc	0.639	0.639	0.639	0.639	0.561
	sd	0.0004	0.0004	0.0004	0.0002	0.0001
	best	0.66	0.66	0.66	0.666	0.574

IV. CONCLUSSIONS

- For this specific dataset, the multiple regression gives poor result for the parameter configuration given in table IV, different to the iris data set, which for several parameter configurations had a good classification accuracy.
- The linear regression is as far as much good as a neural network classifier as is shown in table II,III and I, this could be related with the fact that the dataset presented here is almost linearly separable (see 5), in fact, a neural network with low hidden activation units has a good classification accuracy.
- It's recommendable to use for example scale and rotational invariant features in order to have a more general classification system, that is to have a classification system trained with several recordings in some strategic scenarios to have more general dataset to the one generated here, thus one can have a classifier capable of doing vehicle type predeictions in several scenarios.
- No other algorithms were tried in this implementation given the fact that with simple linear and non-linear classifers was possible to have a good classification accuracy.
- It is supposed that with high accuracy a problem of overfitting could be happening, nevertheless, after several tests in each partition for validation the optimal parameters obtained by the gradient descent algorithm were checked in order to see if they differed a lot,

which would have suggested an overtraining problem, but this was not the case, the parameters obtained in each partition of cross validation were very similar.

- This implementation is a good solution for vehicle classification but several scenario constraints must be fulfilled, for example: in a new avenue, the camera must be placed at a similar height to the one used here for recording the video and extracting the dataset, another constraint is to have a well behaved segmentation algorithm with a shadow scheme, like the one used in this implementation, having a poor performance in segmentation stage could lead to several errors in next stages in the vehicle classification an counting pathline.

REFERENCES

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, Jul 2002.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, pp. –252 Vol. 2.

[3] Z. Chen, N. Pears, M. Freeman, and J. Austin, "Road vehicle classification using support vector machines," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 4, Nov 2009, pp. 214–218.

[4] W. Wang, Y. Shang, J. Guo, and Z. Qian, "Real-time vehicle classification based on eigenface," in *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on*, April 2011, pp. 4292–4295.

[5] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2, Aug 2004, pp. 28–31 Vol.2.

[6] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773–780, 2006. [Online]. Available: http://doc.utwente.nl/57717/

[7] [Online]. Available: http://i.stack.imgur.com/1xic6.jpg

[8] [Online]. Available: http://www.ijser.org/paper/Vehicle-Detection-and-Tracking-from-Video-frame-Sequence.html