2-1-2014

Vehicle Classification For Automatic Traffic Density Estimation

Aya Abdelhady

The American University in Cairo

School of Science and Engineering

Vehicle Classification For Automatic Traffic Density Estimation

A Thesis Submitted to

The Department of Computer Science and Engineering

In partial fulfillment of the requirements for
The degree of Master of Science

By

Aya Salama Abdelhady

Under The Supervision of

Dr. Mohamed Moustafa

FALL 2013

ACKNOWLEDGMENT

In the name of Allah the most gracious and most merciful

I would like to thank my supervisor, Dr. Mohamed Moustafa, I would not have reached this phase without his guidance and support. I would also like to express my thanks to my thesis committee members, Dr. Amr Goneid, Dr. Florin Balasa and Dr. Mahmoud Khalil for their support, guidance and helpful feedback.

I would also like to recognize Ericsson Turkey (Part Of Ericsson Region Middle East) for sponsoring this work under the Ericsson Networked Society Innovation Center (ENSIC) program.

I want to thank my colleagues who helped and supported me throughout this process. Special thanks to Ahmed Abdallah for his constant support and help.

At last but not least, I thank my beloved parents and friends who always supported me, welcomed listening and gave me a lot of their time without complaining, It is my honor to dedicate this thesis to them all.

ABSTRACT

Automatic traffic light control at intersection has recently become one of the most active research areas related to the development of intelligent transportation systems (ITS). Due to the massive growth in urbanization and traffic congestion, intelligent vision based traffic light controller is needed to reduce the traffic delay and travel time especially in developing countries as the current automatic time based control is not realistic while sensor-based traffic light controller is not reliable in developing countries.

Vision based traffic light controller depends mainly on traffic congestion estimation at cross roads, because the main road junctions of a city are these roads where most of the road-beds are lost. Most of the previous studies related to this topic do not take unattended vehicles into consideration when estimating the traffic density or traffic flow. In this study we would like to improve the performance of vision based traffic light control by detecting stationary and unattended vehicles to give them higher weights, using image processing and pattern recognition techniques for much effective and efficient traffic congestion estimation.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Equations

LIST OF ABBREVIATIONS

SVM	**Support Vector Machine**
ANN	**Artificial Neural Network**
SIFT	**Scale Invariant Feature Transform**
SURF	**Speeded Up Robust Features**
BGS	**Back Ground Subtraction**
MOD	**Moving Object Detector**
VI	**Vehicle Identifier**
TDC	**Traffic Density Calculator**
PBAS	**Pixel Based Adaptive Segmenter**

CHAPTER 1

INTRODUCTION

Traffic congestion is now considered to be one of the biggest problems in the urban environments. Traffic problems will be also much more widely increasing as an expected result of the growing number of transportation means and current low-quality infrastructure of the roads. In addition, many studies and statistics were generated in developing countries that proved that most of the road accidents are because of the very narrow roads and because of the destructive increase in the transportation means [1].

A brute force technique to face this phenomenon is adding more lanes and new routes. Obviously, this expansion solution is just a short term solution to traffic congestion issues due to the increasing economic growth and the increasing demand for travel. Moreover, it is very crucial to address such a problem this way in developing countries like Egypt; the economic cost attached to this solution is extremely high as it requires purchasing the required land and removing pre-existing and possibly protected structure. As a result, this solution is totally infeasible. An alternative solution lies in improving the efficiency of the pre-existing infrastructure and public transport system through the use of intelligent transportation system (ITS).

Traffic light is one of the most significant factors in the management of the traffic. Traffic light signs are that signs erected at the sides of the roads to provide information to road users. It has been proven that traffic signal retiming and coordination of existing signals reduce significantly in traffic delay, energy, travel time and this consequently results in increased safety for the public [2]. Due to poor strength of traffic police, it is impossible to control traffic manually in all area of city or town. For this reason, researchers got interested in developing efficient real-time traffic signal control [3].

This idea of controlling the traffic light efficiently in real time has attracted many researchers to work in this field with the goal of creating automatic tool that can estimate the traffic congestion and based on this variable, the traffic sign time interval is forecasted. Analysis of traffic conditions showed that there are many fluctuations in the quantity of the vehicles approaching to a cross road for the same period of time. Therefore, the current automatic traffic light control using a timer which is used in Egypt at many cross roads is not realistic and such automatic tool is required to have more realistic, effective and efficient tool than the current one. As a result, an automatic real-time traffic light control has been emerging as a new field in image processing, pattern recognition, computer systems, artificial intelligence and neural networks. In this study, we are interested in efficient traffic density estimation for realistic automatic traffic control. After studying the previous approaches, we found that they do not take stationary vehicles into consideration when calculating either the traffic density or traffic flow. For this reason, we would like to present a new metric that incorporates unattended and delayed vehicles for measuring the traffic congestion to improve the performance of vision based traffic light control. Since opened door vehicles from the drivers side is the best indicator that this vehicle is going to be an unattended vehicle .Therefore, the contribution of the proposed approach highly depends on the efficiency of the recognition of the vehicles with opened doors at the drivers side in the video.

This section is organized as follows: section 1.1 describes the problem definition; section 1.2 explains the motivation and objective for working in automatic vision based traffic light control; and finally section 1.3 lists the chapters and sections in this study and what each one will talk about.

1.1 Problem Definition

Researchers now are so much interested in automatic real-time traffic congestion estimation tool as it is the most significant factor on which intelligent transportation systems are based. Some of the researchers have focused in their work on traffic flow estimation. It is measured as the rate at which vehicles pass a fixed point (e.g vehicles per minute). They used spot sensors such as loop detectors and pneumatic sensors to quantify the traffic flow [4]. However, the sensors are very expensive and need a lot of maintenance especially in developing countries like Egypt because of the road ground deformations. In addition, metal barriers near the road might prevent effective detection using radar sensors [5]. It is also found that traffic congestion also occurred while using the electronic sensors for controlling the traffic. In contrast, video based systems are much better compared to all other techniques as they provide more traffic information and they are much more scalable with the progress in image progressing techniques [6]. This is the main reason for the motivation to develop vision based tool for traffic light control in this work.

In recent years, vision based traffic light control, which is based on video processing for traffic flow or traffic density estimation, has attracted the attention of many researchers. The value of traffic density measures only the ratio between the density of the vehicles and the total density of the road. So based on this measure, the traffic control system will compare between different roads in the intersection to take the decision for the traffic light and the time interval given. However, most of the previous vision based monitoring systems suffered from lack of robustness on dealing with continuously changing environment [7] such as lighting conditions, weather conditions and unattended vehicles. All these mentioned factors considerably affect the traffic density estimation. Changes in lightening conditions and weather conditions have been tackled in many of the previous approaches [8] and they are going to be considered also in our proposed approach, but the problem that has never been addressed before and has a significant effect on the traffic pace is the stationary vehicles, specially the unattended ones. The problem with the traffic density measurement is that the traffic density of a road with stationary or unattended vehicles is the same as the traffic density of a road with no stationary vehicles.

Traffic flow counts the number of vehicles that passes through the frame during a certain time interval. However, it may give an empty road a higher priority than a congested road, because fewer vehicles are passing though the given point in that empty road. Therefore, we will concentrate on the detection of the delayed and unattended vehicles in the proposed approach for computing more informative metric about the traffic congestion in order to have more effective way of traffic light control. This metric is very similar to the traffic density, but with taking the traffic flow into consideration. So it can be considered as a combination of both traffic density and traffic flow.

1.2 Motivation and Objective

Automatic vision based real time traffic control is broadly divided into two main consecutive steps: 1) traffic congestion estimation and 2) time intervals prediction for certain traffic lights to be based on the measured traffic congestion. In most of the current research, the value of traffic density is calculated which measures only the ratio between the vehicles density, which is the white pixels produced from the background subtraction, and the road density, which is the black pixels of the background,as shown in the following equation.

$TD = \frac{SD}{RD}$

TD : Traffic Density

SD : Sum of the densities of vehicles

RD : The road density

(1)

Then traffic denisty is multilpied by 100 to measure the percentage of coverage of the road to be considered as the traffic congetsion.

On the other hand, traffic flow rate counts the number of vehicles that passes through the frame during a certain time interval. Therefore, as obvious both metrices didn't consider stationary vehicles and their effect when measuring traffic congestion.

As a result, the main goal in this research is to calculate a new metric that measures the traffic density in a more effective way, by incorporating not only vehicles that were urged to stop because of the crowd but also the unattended vehicles which have a destructive effect on the flow of traffic. As a matter of fact, unattended vehicles in the road is as the same as dividing it into two lanes. Even if the vehicle is attended but it is stationary which we call delayed, then this means that there is a huge traffic jam which obligated the vehicle to stop moving. By achieving our goal, traffic light time interval can be predicted in a more effective way based on reliable generated measurements without the need of computing other factors such as speed of vehicles.

This will be achieved by proposing a new hybrid approach of background subtraction, mutlitracking and pattern recognition to detect delayed and unattended vehicles and to be also more sensitive to the changes in the surrounding conditions, so that traffic congestion can be accurately measured.

This research is a part of a bigger project sponsored by Ericsson Turkey (Part Of Ericsson Region Middle East) under the Ericsson Networked Society Innovation Center (ENSIC) program. Another part of the project is about having an opitmization tool that compares the traffic congestion estimated at each road of the inetersection and decides the traffic lights and time intervals. Moreover, there is a website that broadcats the traffic congestion estimated so that drivers will choose the best route to reach their destinations.

[1] [2]

[1] http://shorouknews.com/news/view.aspx?cdate=05012013

[2] http://www.youtube.com/watch?v=9RAxubst

1.3 Organization of the Thesis

The rest of this paper is organized as follows: chapter 2 reviews the different approaches used for vision based automatic traffic light control. Chapter 3 gives a quick review on the theoretical foundations of blob tracking , ANN, SVM and HAAR classifiers and describes briefly SIFT and SURF feature extraction methods. In addition to explanation of the different background subtraction techniques.Chapter 4 explains our proposed system for automatic real time vision based traffic light control and the evaluation methodology. In chapter 5, we will discuss and analyze the experiments that we have performed for building the whole architecture of the system and proving the efficiency of the proposed metric by comparing it to previous approaches. Finally, in chapter 6, we will conclude the thesis and and list some directions for future work.

CHAPTER 2

LITERATURE SURVEY

Real time automatic vision based traffic light control has been recently the interest of many researchers, due to the frequent traffic jams at major junctions and its resulting wastage of time. Instead of depending on information generated by costy sensors, economic situation calls for using available video cameras in an efficient way for effective traffic congestion estimation. Thus, given a video sequence, the task of vision based traffic light control is: 1) analyze image sequences; 2) estimate traffic congestion and 3) predict the next traffic light interval. Researchers may focus on one or more of these tasks, and they may also choose different measures for traffic structure or add measures. For more comprehensive review on vision based traffic light control, see [9] [2].

In this chapter, I review some of the research that has been done in vision based traffic light control.

Previous vision based traffic light control approaches were based on traffic density estimation or traffic flow. Therefore, they differ according to the way of calculating these metrics as follows.

2.1 Images Matching Approach

In the image matching approach, one camera is installed alongside the traffic light, which will capture image sequences. An image of the road with no traffic is captured and converted into grey level, then this image is enhanced to signify signal to appear more than the noise and to also accentuate the image features. This enhancement was implemented using Gamma- correction [10] [11]. After the enhancement is done, edges are detected in the enhanced image to remove irrelevant data with preserving the important structure of the image. This can be done using Perwitt edge detection operator [10] or using canny edge detection [11].

After edge detection procedure, both reference image and other different images are captured at different time intervals are matched. The traffic light is then to be controlled based on the percentage of matching. In image matching, all edges in one image are compared to all edges in the other image.

Accuracy of this approach is highly affected by the changes in illumination and weather conditions. Furthermore, it does not take stationary vehicles into consideration. The vehicles may be moving fast on one side having matching percentage between 50% and 70% and at the other side of intersection, the matching percentage is also between 50% and 70%, but they are completely stationary.

2.2 Background Subtraction Approach

In background subtraction approaches, segmentation of objects in the real-time images was implemented using background subtraction then the traffic density was calculated as the sum of the white pixels generated by the subtraction of images. There are two types of background subtraction; static background subtraction and dynamic background subtraction. In static background subtraction, a fixed background is obtained beforehand for the road with no vehicles and is called the reference image and then it is subtracted from images at any given time. This method has been the traditional method for many years [12]. There is only one problem that arises with this approach because as a matter of fact outdoor environments are affected by many factors such as changes in the weather and illumination. Therefore, using static background subtraction alone does not give reliable results to depend on. On the other hand, in dynamic background subtraction; background is dynamically updated with varying surrounding conditions. The background can be detected dynamically in many ways as will be discussed in chapter 3.

Therefore, it is obvious that the main drawback of both static and dynamic background subtraction approaches is that they do not take many other important conditions that may affect the density estimation into consideration such as the stationary vehicles.

2.3 Neural Networks Approach

Multilayer perceptron with 400 input elements, 100 hidden neurons and 1 output neuron was used to predict the intensity of traffic flow [13]. Sigmoid function was used as activation function. Total square error for training was 0.01. The training was run on three different samples of data; morning, afternoon and evening on three different copies of neural networks to handle the variation during the whole day. It was also trained on different directions.

This approach was not able to give accurate results due to the chaos in the large cities, so the results are not only falsely affected by the daily movement of vehicles, but also by random incidents such as accidents. Moreover, stationary vehicles and the delay they may cause is still a completely un-tackled point.

2.4 Hybrid Approach of Background Subtraction and Neural Networks

A hybrid approach was proposed which consists of three sub modules: Moving Object Detector (MOD), Vehicle Identifier (VI) and Traffic Density Calculator (TDC) [8].

In MOD, moving objects are detected using dynamic background subtraction. In VI, the vehicles are classified as small, middle and big using neural networks model which consists of 14 inputs resembling 14 different object properties and 4 output layer binary valued nodes representing big vehicle, small vehicle, medium vehicle or not a vehicle. This classification was carried out as a kind of analysis that can be used in traffic congestion estimation. Then in the last phase, TDC, all video frames together are processed at one time to calculate the number of vehicles passing through the frame for a given interval of time. Successive frames were taken in time intervals of just milliseconds. Vehicles were counted based on the location of successive frames. For each vehicle type, traffic flow was measured as follows:

$denisty_i = \frac{v_i}{t}$

$denisty_i$: Traffic density of vehicle type i

v_i : Number of vehicle Type i that passed the road un time period T

t : time period

(2)

This efficiently tackles the problem of the presence of other moving objects other than vehicles such as shadows. However, this approach just counts the vehciles ignoring their status, whether they are moving or stationary.

2.5 Discussion

Based on the conducted survey, it is obvious that all the work done in vision based automatic traffic light control did not address the problem of stationary and unattended vehicles especially in the middle of the street, that significantly affects the flow of the traffic in the crowded roads.

CHAPTER 3

THEORETICAL FOUNDATIONS

In order to detect foreground objects and compute their densities, background subtraction approach should be used. After background subtraction, multracking has to be applied on the output to detect whenever any moving object stops moving so that we can detect delayed and unattended vehicles.

For objects recognition in any given image, we should use a classifier to decide the type of the input object. Artificial Neural Networks (ANN), Support Vector Machines (SVM) and HAAR are the most popular strategies for supervised machine learning and classification.

The brute force way of classification is to input the whole image into the classifier, but in this case, training and classification will take significant time. In addition, there are many repeated data that has no importance in classification [14]. Therefore, We chose to extract the most important features in the image instead of inputting the whole image. Feature extraction plays very important role in any classification methods as the accuracy of classification is always dependent on the quality of the extracted features[15]. The most well known robust features detection methods are Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF)

In this chapter, the theoretical foundations of different background subtraction approaches, Blob Tracking, Artificial Neural Networks (ANN), Support Vector Machines (SVM) and HAAR classifiers are briefly described, in addition to Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) methods, which have the ability to reduce the image to a few descriptors which are considered to be the most appealing descriptors to be used in many of the classification problems because they are distinctive and relatively fast [16]. They are also invariant to scale, rotation and illumination.

3.1 Theoretical Foundation of ANN

Neural Networks follow the same architecture of the brain, but with electronic substitutions just as in the brain, when a neuron in a neural network receives a certain number of inputs to calculate the output.

The inputs that are fed into a neuron are weighted. The weights are multiplied by the number that gives each input its relevant importance. Weights are adjusted to reduce the error which is the difference between the target output and the output pattern computed by the network. The activation function handles the neurons decision making process. Typically sigmoid function is used. Since our problem is not a linearly separable problem then we will use a multilayer perceptron with back propagation training algorithm.

In back propagation neural network, the learning algorithm has two phases. The first phase consists of inputting the training pattern to the input layer. The network by its turn will propagate that input from that input layer to the next layer. On the other hand, for the second phase is different from the target output, the error will be calculated and then propagated backwards through the middle layers from the output layer to the input layer. By that the weights updates can be calculated. For the output layer, the weights are updated by:

$$\triangle w_j k(p) = \alpha \cdot y_j(p) \cdot \delta_k(p) \tag{3}$$

Where the error is $e_k(p) = y_{d},_k\,(p) - y_k(p)$
And correspondingly the error gradient is: $\delta_k(p) = y_k(p) \cdot [1 - y_k(p)] e_k(p)$ And for the hidden layer, the weights are updated by: $\triangle w_i j(p) = \alpha \cdot x_i(p) \cdot \delta_j(p)$
Where the error gradient is: $\delta_j(p) = y_j(p) \cdot [1 - y_j(p)] \sum_{k=1}^{l} \delta_k(p) w_j k(p)$
The training will continue on all data samples until a selected error criterion is satisfied e.g. minimum error or minimum number of iterations.

3.2 Theoretical Foundation of SVM

Support vector machine is closely related to neural networks, but instead of calculating weights by solving non-convex minimization problem, they find them by solving a quadratic programming problem. Quadratic Programming is a well-studied class of optimization algorithms to maximize a quadratic function of some real-valued variables subject to linear constraints.

As a matter of fact, if sigmoid kernel function is used in SVM model it will be equivalent to two layer perceptron neural network. The main idea of support vector machine is to construct a hyper plane so that the margin that separates two classes is maximized [17].

3.3 Theoretical Foundation of Haar Classifiers

Haar classifier is based on Haar-like features. Viola and Jones[18] adapted the idea of using Haar wavelets and developed Haar-like features.Haar wavelets are single wavelength square waves (one high interval and one low interval). For the two dimensions, a square wave is a pair of adjacent rectangles (one light and one dark). Thus, these features use the changes in contrast values between adjacent rectangular groups of pixels. The contrast variances between groups of pixels are required to define light and dark regions. Haar-like features are basically three or two adjacent groups with a relative contrast variance. These Haar-like features are used to detect an object. The main advantage of a Haar-like feature over most other features is its calculation speed. The features are calculated using integral images which are an arrays containing the sums of intensity values of pixels at the left and above any pixel in the original image. Due to the use of integral images, they can be easily scaled to be used to detect objects of various sizes.

To select the specific Haar features to use, and to set threshold levels, Viola and Jones used AdaBoost machine-learning method. AdaBoost combines weak classifiers to create a strong classifier. Weak classifier gets the right answer a little more often than random guessing would which is not very accurate. However, lots of these weak classifiers combined together force for reaching the correct solution. Therefore, AdaBoost selects a set of weak classifiers to combine and assigns a weight to each. This weighted combination is the strong classifier. AdaBoost classifiers which is called cascade is a filter chain where each filter is a separate AdaBoost classifier has a fairly small number of weak classifiers. The acceptance threshold at each level is set low enough to pass nearly all the examples in the training set. The filters at each level are trained to classifiy training images that passed all previous stages. Thus, during use, if any one of these filters fails to pass an image region, that region is immediately classified as not the object otherwise it goes to the next filter in the chain. Image regions that pass through all filters in the chain are detected to be the required object. The order of filters in the cascade is based on the importance weighting that AdaBoost assigns. The more heavily weighted filters come first, to eliminate non-required-object image regions as quickly as possible.

For objects detection, a multiscale window slides over the image to calculate the Haar-like features under this area to determine is it the object or not.

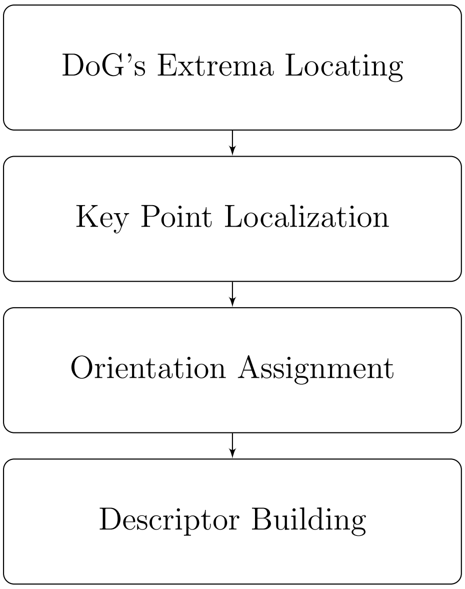

Figure 1: **The Building Blocks of SIFT**

3.4 Theoretical Foundation of SIFT

The SIFT algorithm consists of four main steps as shown in figure 1.

3.4.1 DoGs Extrema Locating

In this step all the interesting key points are detected. In order to be have scale invariance, Difference of Gaussians (DOG) is used where difference of successive Gaussian-blurred images are calculated. Interest points are only found at different scales, because the search of correspondences needs to compare them at different scales of the image. Scale spaces are always implemented as an image pyramid. Then Gaussian filter is used to repeatedly smooth them. By that the features can be found from differences of these images. Features which are repeatedly present in the Difference of Gaussians will be considered as Scale Invariant and for each of them or each pixel of them value is found to be the maximum or minimum among all its eight neighbors within the same scale and the other nine corresponding neighboring pixels in each of the neighboring scales, then it is considered to be a key point and should be kept [19].

A DoG image is represented as $D(x, y, \sigma)$
where $D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$ and $L(x, y, k\sigma)$ is the convolution of the original image I(x,y) with the Gaussian blur $G(x, y, k\sigma)$ at scale $k\sigma$
Therefore, $L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$
Hence a DoG image between scales $k_i\sigma$ and $k_j\sigma$ is just the difference of the Gaussian-blurred images at scales $k_i\sigma$ and $k_j\sigma$.

3.4.2 DoGs Extrema Locating Key Point Localization

A lot of key points will be produced from the Scale-space extrema detection, but we need only the most stable ones .Therefore, low contrast features and poorly localized features along an edge should be discarded because they are very sensitive to noise. For each key point, neighboring data is interpolated to accurately determine the interpolated location of the extremum, to give more stability in order to improve the matching. The interpolation is done by the quadratic Taylor expansion of the Difference of Gaussian scale-space function,$D(x, y, \sigma)$, with the candidate key point as the origin:
$D(x) = D + \frac{\sigma D^T}{\sigma x}x + \frac{1}{2}x^T\frac{\sigma^2 D}{\sigma x^2}x$
x̂ is The location of the extremum which is determined by the differentiation

of this function with respect to x and setting it to zero.
$\hat{\text{x}} = -\frac{\sigma^2 D^{-1}}{\sigma x^2}\frac{\sigma D}{\sigma x}$
Then we minimize to get true location of extrema.

For filtering low-contrast key points, second-order Taylor expansion D(x) is also used to discard key point whose corresponding value is less than 0.03. In addition, points with strong edge response in one direction only should be discarded using Trace and Determinant of Hessian to check if ratio of principal curvatures is below some threshold, r, we check: $\frac{TR(H)^2}{Det(H)} < \frac{(r+1)^2}{r}$
Where TR is the Trace of Hessian and Det is the Determinant of Hessian.

3.4.3 Orientation Assignment

For Rotation Invariance, all features are rotated to go the same way in a determined manner. This is done by developing the histogram of Gradient directions and the rotate to most dominant.

3.4.4 Descriptor Building

The points around the key point are sampled and the gradients and coordinates are orientated by the previously determined orientation. Then each region is divided in to sub regions. A histogram is then created for each sub region with 8 bins.

3.5 Theoretical Foundation of SURF

SIFT and SURF (Speeded-Up Robust Features) approaches employ slightly different ways of features detection [20]. SURF uses a very basic approximation of hessian metric for detection which is somehow similar to DoG [21], but it relies on integral images to reduce the computation time, so it is called the Fast-Hessian detector. With box ?lters, the second order Gaussian derivatives are approximated, and can be evaluated very fast using integral images, independently of size. Integral images allow for fast computation of box type convolution filters. The entry of an integral image $I_{\sum}(x)$ at a location X=(x,y) is the sum of all pixels in the input image I of the rectangular region that is formed by the point x and the origin, $I_{\sum}(x) = \sum_{i=0}^{i<x} \sum_{j=0}^{j<y} I(i,j)$. $I_{\sum}$ calculation takes only four additions to compute the sum of the intensities over any upright, rectangular area, independent of its size.

Due to the use of box filters and integral images, the same filter will not be applied iteratively to the output of a previously filtered layer, but instead box filters of any size are applied directly at the same speed on the original image and even in parallel. Therefore, the scale space is analyzed by upscaling the filter size rather than iteratively reducing the image size. The computational efficiency is the main motivation for this type of sampling is its.

3.6 Theoretical Foundation of Background Subtraction

Since many subsequent processes are greatly dependent on the performance of this step, it is important that the classi?ed foreground pixels accurately correspond to the moving objects of interests. In addition. it is very critical to keep history of moving vehicles so that they will not disappear after stopping.

There are two types of background subtraction; static background subtraction and dynamic background subtraction. In static background subtraction, a fixed background is obtained beforehand for the road with no vehicles and is called the reference image and then it is subtracted from images at any given time. This method has been the traditional method for many years but it is highly affected by the changes in the weather and illumination conditions[12]. On the other hand, in dynamic background subtraction; background is dynamically updated with varying surrounding conditions. The background can be detected dynamically in many ways as follows.

3.6.1 Theoretical Foundation of Frame Difference

Background model is replaced with the previous image. It is one of the most common techniques used in background segmentation. As the name itself suggests, frame differencing involves taking the difference between two frames and using this difference to detect the object. Objects that stop are no longer detected.

The two consecutive gray scaled images are differentiated and their absolute difference is used to identify the movement between frames. The noise collected due to differencing is removed by applying the threshold value to the images. The threshold value is always between [0.43 - 0.45]. Pixels below the threshold are removed from the differenced frame leaving behind objects of interest [22].

3.6.2 Theoretical Foundation of Weighted Moving Mean

The background model at each pixel location is based on the pixels recent history. This history is a weighted average where recent frames have higher weight, therefore, the background model is computed as the average from the pixels history.

It is recursive, it doesnt require to store all the previous images, it just uses weighted average technique to average sequence of frames. The weighted average is the average in which each quantity to be averaged is assigned a weight. These weightings determine the relative importance of each quantity on the average [23].

3.6.3 Theoretical Foundation of Weighted Moving Variance

The weighted moving variance acts exactly as the weighted moving mean but just instead of using the mean it uses the variance.

3.6.4 Theoretical Foundation of Gaussian Mixture Model V1

One of the most prominent and most widely used methods are those based on Gaussian Mixture Models (GMM). Pixels is modeled as a mixture of weighed Gaussian distributions.It models each background pixel by a mixture of K Gaussian distributions (K is a small number from 3 to 5). Different Gaussians are assumed to represent different colors. The weight parameters of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static [24].

3.6.5 Theoretical Foundation of Gaussian Mixture Model V2

In this version, pixels which are detected as background are used to improve the Gaussian mixtures by an iterative update rule. This allows the system learn faster and more accurately as well as adapt effectively to changing environments. It is also known to have a low memeory complexity [25].

3.6.6 (Godbehere-Matsukawa-Goldberg) GMG

It is a probabilistic foreground segmentation algorithm that identi?es possible foreground objects using Bayesian inference with an estimated time-varying background model. The background model is basically distributions on RGB colorspace for every pixel in the image. The estimates are adaptive; newer observations are given higher weights than old observations to overcome the changes in illumination [26].

3.6.7 Theoretical Foundation of Multilayer BGS

It is a robust multi-layer background subtraction technique which takes advantages of local texture features represented by local binary patterns (LBP) and invariant color measurements in RGB color space. LBP can work robustly with respective to light variation on rich texture regions but not so efficiently on uniform regions [28].

3.6.8 Theoretical Foundation of Eigen Background SL/PCA

To detect these moving objects, eigenspace is adaptively built that models the background. This eigenspace model describes the range of appearances (e.g., lighting variations over the day, weather variations, etc.) that have been observed. The eigenspace could be generated from a site model using standard computer graphics techniques [29].

3.6.9 Theoretical Foundation of Adaptive SOM

This approach built background model by using a competitive neural network similar to the Kohonen Self-Organizing Map (SOM). For each pixel, a neuronal map consisting of 3x3 weight vectors is defined. The incoming source pixels are mapped to the node whose model is most similar to the pattern according to a Euclidean distance metric, and then the weight vectors in a neighborhood of such node are updated. The set of weight vectors act as a background model that is used for background subtraction in order to identify moving foreground pixels. [30].

3.6.10 Theoretical Foundation of Fuzzy Adaptive SOM

A modified version of Adaptive SOM that uses a fuzzy rule to update the neural network background model. The fuzzy updating of the background helps to make the model more robust to illumination changes in the scene [31].

3.6.11 Theoretical Foundation of Pixel Based Adaptive Segmenter

This approach models background by a history of most recently observed pixel values. The foreground decision depends on a decision per-pixel threshold between current model and current image. The background is updated to allow for background changes based on a per-pixel learning parameter. The novel idea in this approach is that two controllers are used with feedback loops for the decision thershold and learning parameter.

3.7 Theoretical Foundation of Blob Tracking

The main goal of tracking is to detect the vehicle since the moment it enters the scene and during the whole duration it is apparent in the scene. Most importantly, by tracking objects we can detect stationary ones. Therefore this involves object detection and then tracking. We used open-source OpenCV library called cvBlob as we found it to be the most suitable algorithm for our proppsed approach after testing and comparing other alogorithms such as, Kanade-Lucas-Tomasi (KLT), Template matching, CamShift and Kalman Filter.

Blob detection should be implemented before tracking to find blobs in the given frame. The input to blob detection should be a binary image, where the foreground objects are white with black background. For this reason, the background subtraction is used to segment the foreground objects from the background to be used as an input to this phase.

Blob detection is done by finding connected components in the image. The labeling algorithm which is used to implement the blob detection is based on an algorithm by Fu Chang, Chun-Jen Chen and Chi-Jen Lu called "A linear-time component-labeling algorithm using contour tracing technique". This algorithm simply detects the connected components in the given image which are called blobs.

We used the tracking algorithm implemented by Andrew Senior, Arun Hampapur, Ying-Li Tian, Lisa Brown, Sharath Pankanti and Ruud Bolle provided by cvBlob library. This approach is dependent on the blobs detected before as shown in figure 2. In this approach, a distance metric is

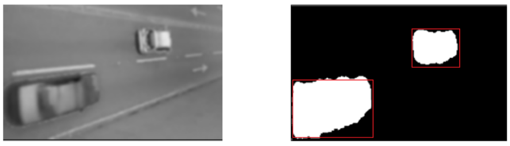

Figure 2: **Blobs Tracking**

constructed in which the distance between each of the foreground regions (new blobs) and all the currently active tracks (existing blobs) is computed. This distance metric is threshold to obtain a binary metric which is called correspondence metric that shows the possible matches between tracks (already existing blobs) with foreground regions (new detected blobs in the current frame). The correspondence matrix has rows correspond to existing tracks and columns to foreground regions in the current segmentation. The correspondence matrix is then analyzed to update the blobs (vehicles). The update of the blobs will be done by categorizing them according to the analysis into:

1. Existing object
2. New object
3. Merge detected
4. Split detected.

This is done by comparing the distances of the existing blobs (vehicles) with the new detected tracks (blobs/vehicles).

1) Existing Object: For the previously existing vehicles (blobs) in the correspondence matrix, there will be at most one non-zero element in each row or column associating each track with one foreground region and each

foreground region with one track, respectively.

2) New object: When there is a column with all zero elements, then it means that a new object has appeared in the scene which is not associated with any previous track. Therefore, a new track (for a new vehicle) will be created. On the other hand, rows with all zero elements are logically considered to be tracks that are no longer visible (because they left the scene, or were generated because of artifacts of the background subtraction).

3) Merge detected: In the corresponding matrix also it is possible to have a column with more than one non-zero entry. This case means that we are having merging objects because two or more tracks (existing blobs/ vehicles) will correspond to one foreground region (new blobs). In other words, when a single track corresponds to more than one bounding box, all those bounding boxes are merged together, and processing proceeds.

4) Split detected: Merged objects may split, for example, two vehicles very close to each other, walk away from each other, a single track will then correspond to multiple foreground regions. This will appear in the corresponding matrix by having more than one non-zero element in a row of the correspondence matrix. Any two objects tracked as one should separate, the parts continue to be tracked as one until they separate sufficiently that both bounding boxes do not correspond to the track, and a new track will be created.

CHAPTER 4

PROPOSED APPROACH FOR VISION BASED TRAFFIC LIGHT CONTROL

In this research, we will create a hybrid approach by combining background subtraction , multitracking and classification techniques which have been tested together and proved to give more accurate and efficinet measurement for traffic congestion [8]. By adding delayed and unattended vehicles detection, they will be given higher weights. As a result, roads having these vehicles will be given higher priority to take the green light and for longer times than other roads having less stationary and unattended vehicles. Thus, the proposed approach is employed to give very effective traffic light management system.

This chapter is organized as follows: On section 4.1 I explain the proposed approach and its components; section 4.2 explains the evaluation methodology we will use for comparing the performance of the different classification methodologies using different feature sets; and finally section 4.3 defines the used tools in the system.

4.1 The Proposed System Architecture

According to the conducted research on the work done before for traffic congestion estimation, it is obvious that many important factors that affect the traffic are obviously not taken into consideration. Therefore, the approach

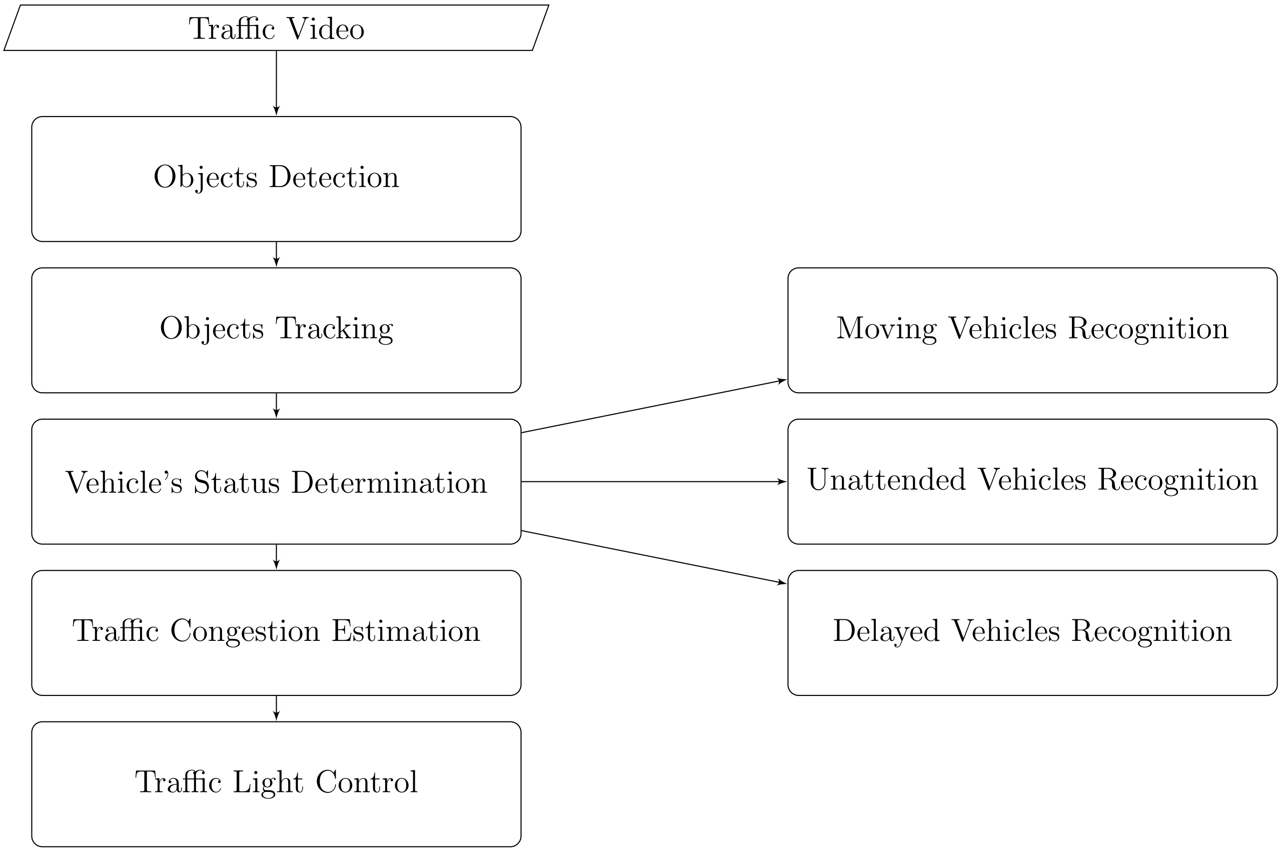
Traffic Video
Objects Detection
Objects Tracking
Vehicle's Status Determination
Traffic Congestion Estimation
Traffic Light Control
Moving Vehicles Recognition
Unattended Vehicles Recognition
Delayed Vehicles Recognition

Figure 3: **The Proposed Architecture for Automatic Traffic Light Control**

proposes a new methodology to detect unattended and delayed vehicles and give them higher weights for more accurate traffic congestion estimation. The proposed system consists of five main components as shown in the figure 3.

4.1.1 Objects Detection

Foreground objects are detected using a background subtraction technique that is variant against different external factors such as weather and illumination (day, night or time within a day). The output of background subtraction is also used to compute the densities of the vechiles.

4.1.2 Objects Tracking

Then foreground objects should be tracked to detect if the object is moving or stationary and to be able to give its dimensions to the next component which will define the status of the vehicle for more accurate traffic congestion estimation. Moreover, by tracking the vehicles we will be able to know if it has left the boundaries of the frame. For this component, blob tracking is used.

4.1.3 Vehicle's Status Determination

As shown in figure 4, the status of each vehicle is determined to give a higher weight for stationary vehicles especially the unattended ones. If the object is moving, then we will extract the subimage including this object from the video frame based on the dimensions given by the blob detector and tracker. After this, a classifier will be used to make sure that the object inlcuded in this subimage is a vehicle to remove the noise encountered due to the changes in external factors such as, shadows and other unnecessary information that may lead to incorrect traffic congestion value [32].

If it is a staionary object, then delayed and unattended vehicles recognition takes place. This component is my main contribution as it has not been applied before in any of the proposed traffic congestion estimation systems.

Given the dimensions of the stationary object calculated by the tracker,

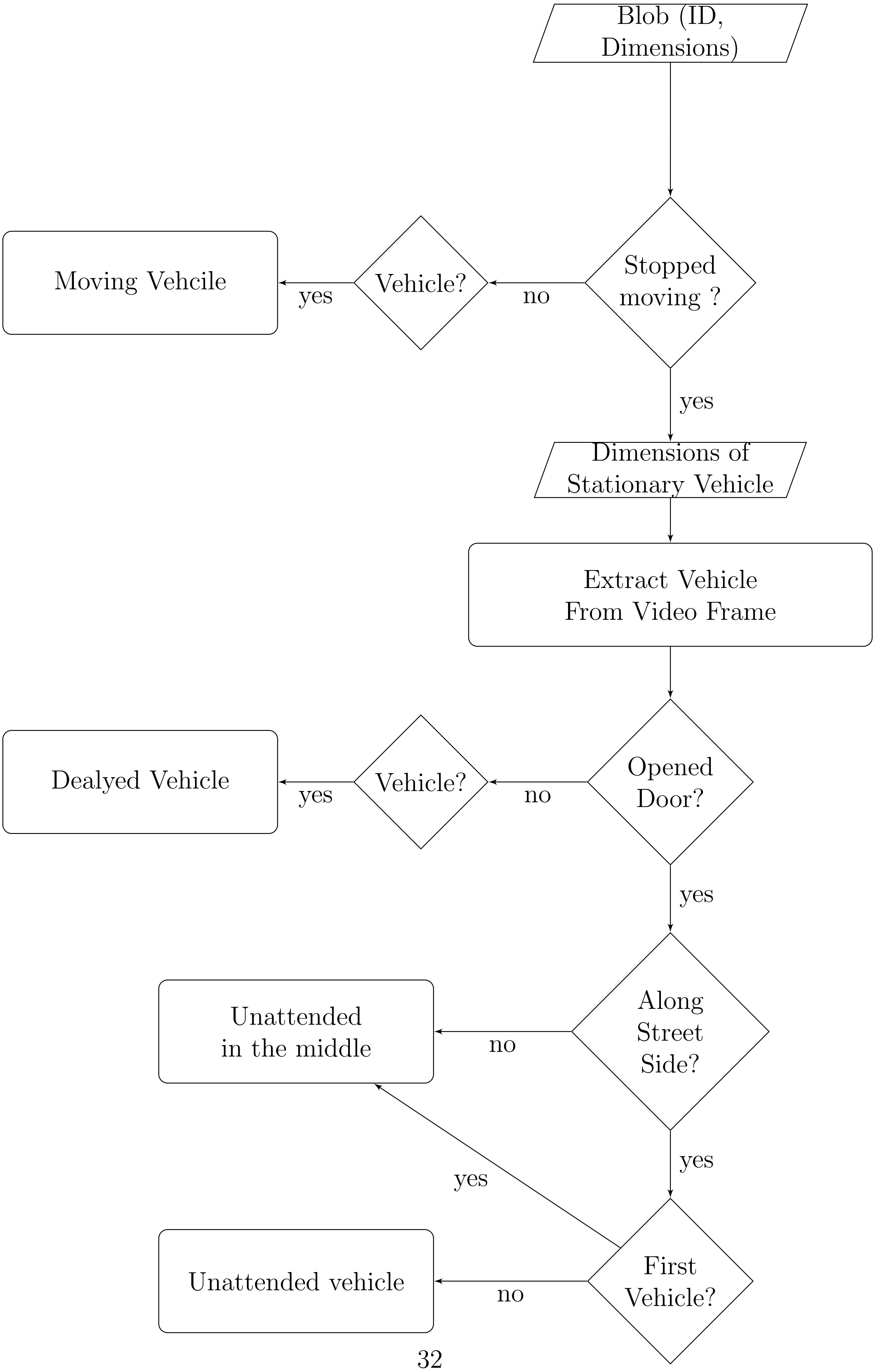

Figure 4: **Vehicle's Status Determination**

the subimage within these dimensions will be extracted from the video frame to classify the object included.

This object is classified to know whether it is a closed vehicle or a vehicle with an opened door. If the object was not detected as an opened door vehicle and it is recognized as a closed vehicle then it will be considered as a delayed vehicle until the tracker gives information that it started moving again. But if the object is a vehicle that has an opened door at the drivers side (for example, left side in Egypt and it depends on the traffic direction) as shown in figure 5, then it will be considered as an unattended vehicle because this means that the vehicle's driver is going to leave it for a while. If the dimensions of the detected unattended vehicles are in the middle of the street, then it will be given a higher weight more than the rest of the vehicles as it delays the path of the traffic. On the other hand, if the vehicle is found to be along the sides of the street, then we should check the status of the vehicles within the dimensions in front and behind that vehicle along the side in which the unattended vehicle was detected. If no unattended vehicle is detected in front or behind that detected unattended vehicle, then this unattended vehicle will also be considered as unattended in the middle and accordingly, it will also be given a higher weight. The vehicle will still be considered as an unattended vehicle until the tracker gives information that it started moving again.

Even if there is more than one opened door in the vehicle, this vehicle may be detected as an opened door beacuse of the presence of the features of the opened door. However, this vehicle is actually stationary not delayed or obligated to stop beacuse of the traffic jam, so it still hinders the traffic as

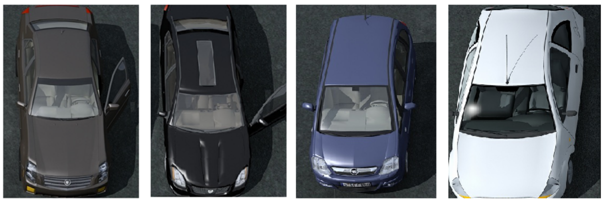

Figure 5: **Vehicles with Opened and Closed Doors at the Drivers Side**

same as unattended vehicle. Even if this would be for a shorter time. This is why it would also be given a higher weight. In all conditions, when it starts moving again, the penalty is removed.

In order to recognize unattended vehicles, we will use a classifier to decide whether the input vehicle is with an opened door at the driver's side or not. We tested three types of classifiers, Artificial Neural Networks (ANN), Support Vector Machines and HAAR. In addition, we tested Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) methodologies to find the best appropriate one to our model to extract the most important features in the image so that they are used as input to the ANN and SVM classifiers. The input of the ANN and SVM classifiers is of a fixed size, but the output of the SURF or SIFT algorithm is of a different size for each input image [33]. To solve this problem, we tested different methodologies to find the best one that suits our model. The first one is to cluster features normally with respect to their descriptors (multidimensional clustering). We tested also simpler ways other than clustering for entering a fixed size inputs to the classifier. Thus, before the features extraction, the image is divided into fixed number of sub-images and then one interesting feature is extracted from each sub-image. This simple substitution of clustering gave better results. However, we found that these approaches depend on that the objects exaclty fit in the detected blobs. Blobs are sometimes smaller or bigger due to the illumination conditions. As a result, we extracted subimages bigger than the dimensions given by the blob detector. For this reason, we also tested the Haar calssifer as it uses a multiscale sliding window when searching for its robust Haar featues and aslo in high speed, which is needed in our real time appliacation. And because Haar calssifier only recognizes if the object is the object of interest or not, we had to use another Haar classifier for closed door vehicles recognition when testing the Haar classifier for opened doors vehicles recongnition.

4.1.4 Traffic Congestion Estimation

The new proposed metric measures the traffic congestion by checking the status of the vehicles as demonstrated before, if it is not a stationary vehicle, then its density will be added without any additional value to the sum of the total density . Otherwise, if it is an unattended vehicle in the middle, then its density will be raised by a factor and if it is not unattended in the middle, then it will be raised by another factor less than that of the unattended in middle. If it is delayed, its density will also be raised by a smaller factor . Lastly, the sum of the densities will be divided by the sum of all the pixels, just as the traffic denisty metric. Perspective distortion is also taken into account so that pixels of vehicles closer to the camera are given the same pixel weight as the ones further back. The new proposed metric is summarized in the following equation.

D (vehicle) = D (vehicle) ? if it is a moving vehicle

D (vehicle) = 2 *D (vehicle) ? if it is an unattended vehicle in the middle

D (vehicle) = 1.5*D (vehicle) ? if it is an unattended vehicle but not in the middle

Where D is the regular density -number of weighted pixels- of a given vehicle.

(4)

4.1.5 Traffic Light Control

The traffic density estimator will then forward the traffic congestion measurement from every direction of the road intersection to the traffic light controller module. Hence, the traffic light switching time can be adjusted to prevent having a green light in an empty path or a red light in a crowded path simultaneously as traffic congestion is estimated to give higher weights to congested roads. There will be a kind of load balancing using any of the optimization techniques. This component should be determined to give the best performance using the available produced information from the previous components.

4.2 Evaluation Methodology

The main contribution in the proposed approach is based on unattended vehicles recognition. As been demonstrated in the proposed approach unattended vehicles recognition depends on classifying the input objects images into vehicles with closed door or vehicles with opened door from the drivers side since opened door at this side gives high probability that the vehicle will be unattended in the consequent frames. For this reason, the performance of the classifying algorithms, ANN, SVM and HAAR classifier, will be compared on different sets of inputs preprocessed in different ways. In addition, the detection of the unattended vehicles is highly dependent on the output of background subtraction for the detection and tracking the motion of the vehicles. Moreover, we use the output of background subtraction for calculating the traffic densities of the vehicles. Thus, we tested different background subtraction approaches. Furthermore, we also compared between the known metrics, traffic density and traffic flow, against our new proposed metric, to prove that this metric is much more informative and that it gives better represntation for the road status . By applying these experiments, we were able to choose the appropriate building blocks for the proposed approach for an efficient traffic light management system and prove the improvement achieved in traffic cingestion estimation by using the proposed metric.

In classification problems, the accuracy is typically evaluated by using testing data which is different from the training data set. Therefore, by knowing beforehand the classes of the testing images, efficiency of the classifiers prediction can be evaluated using the well-known measures of the supervised learning classifiers which are accuracy and F-Measure. All of these measures depend on the following four parameters:

TP: positive-labeled objects that were correctly classified.

TN: number of negative-labeled objects that were classified correctly.

FP: Number of negative-labeled objects that were incorrectly classified.

FN: Number of positive-labeled objects that were incorrectly classified.

4.2.1 Accuracy

The accuracy of a classifier calculates the percentage of the correctly classified objects:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

4.2.2 Precision

The precision calculates the probability of any object should be classified with this label then this is the correct classification:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

4.2.3 Recall

The recall calculates the probability that if any object should be classified with this label, then this is taken class:

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

4.2.4 F-measure

A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score::

$$F - measure = \tfrac{2*Precision*Recall}{Precision+Recall} \tag{8}$$

4.3 Tools

4.3.1 Matlab Toolbox for Machine Learning

The toolbox is a collection of the most fundamental algorithms for machine learning such as clustering, feature selection, feature extraction, linear and nonlinear discriminant, ensemble learning, classification and numerical optimization of arbitrary functions. The toolbox is made specifically for education and research. The Machine Learning modules in the Matlab Toolbox run on Matlab 7 or later versions. We used the following implemented algorithms in the Matlab: Multilayer perceptron classifier, SVM, SURF, SIFT and multidimensional clustering.

4.3.2 3D MAX

3D MAX enables us to construct digital three-dimensional object and, change the camera positions and lighting, In addition to using materials and other features enabled by the program, to create realistic renderings and animations of the 3D objects.

3D max simulations have effectively replaced physical real testing experiments; therefore we used these models in some of our preliminary experiments.

4.3.3 OpenCV

OpenCV (Open Source Computer Vision Library) is a library of real-time computer vision. This library is developed by Intel and is free for use under the open source BSD license. The library is cross-platform. We used a lot of functionalities of this library as it as its application areas include: Motion understanding, Object identification, Segmentation and Recognition, Structure from motion (SFM) and Motion tracking. We tested some of its functionalitites to find what is the best of them that suits our proposed approach such as the different background subtraction techniques, multitracking and classiffiers.

CHAPTER 5

EXPERIMENTAL RESULTS AND EVALUATION

As mentioned before, trafffic density and traffic flow have been used intensively as the metrics for measuring the traffic congestion. However, these metrics are not representative as our proposed metric. Therefore, in our experiments we will show the difference between the output of the three metrics.

Some components of the proposed approach affect considerably the computation of this new proposed metric. Therefore, we have conducted different experiments on background subtraction and opened door vehicles recognition to reach the best structure to achieve the main goal of the proposed approach which is effective traffic light management.

This chapter is organized as follows: section 6.1 explains the process of collecting the data needed for conducting our experiments. Section 6.2 compares the performance of 12 different background subtraction approaches, while section 6.2 compares the accuracy of SVM and ANN for unattended vehicle's detection and also compares the effect of using the features extracted by SIFT and SURF which are also reduced to a fixed size in different ways on the performance of both SVM and ANN approaches. Section 6.3 shows the effect of workig on one part of the image on the performance of classification. For the HAAR classifier, section 6.4 compares it to SVM and ANN classifiers.

For using the HAAR classifier on real vehicles images, section 6.5 displays the results of using vehicles retrieved by the blob detector. Finally, section 6.6 demostrates that the new proposed metric obtained after combining all the components of the hybrid approach is more representative to the traffic congestion of the road than the ordinary traffic density metric and traffic flow by comparing their values.

5.1 Data Collection

The main contribution of the proposed approach is based on opened door vehicles recognition.For this reason, we conducted some experiments on 3DMAX models of vehicles,which were very close to the real ones to test the feasibilty of this approach. However, they were very clear and sharp without any noise unlike the video frames. In addition to that we wanted to process real video freames because training our classifiers on these models wouldn't give accurate results in the real time application. As a result, we collected images of real vehicles.

We planted cameras in the campus of the university on the lighting poles in the parking area. We also tested different positions of the camera in order to give a view close to the roads at intersections.

We used a logitech 750e camera because it is high quality video (960 x 720 @15fps) so it makes it easier to recognize vehicles and other objects. This camera also has 130-degree, wide-angle lens which gives twice the viewing area compared to standard cameras which would be helpful when used on wide roads with more than two lanes. Although, it is a weatherproof camera that can capture in shine, summer or winter, sometimes it couldn't resist the hot weather and so it stopped recording.

We placed the cameras at 6 meters height with about 80 degrees angle from bottom of camera and 85 degrees from middle of lens.

we recorded many videos at different times of the day to collect as much as possible data to train our classifier and to test our proposed approach.

It rarely happened to find a vehicle that stops and opened its door inside the parking area because the vehicles only stops in the parking position beside or in front of the pavement. We didn't place our cameras beside these places because we want a road that imitates the intersection roads with normal traffic flow. Thus, the biggest challenge was in collecting samples for opened doors vehicles. Analyzing the videos that we recorded through the day to find opened door vehicles took a lot of time. As a result, we had to go physically to people and ask them in person to stop their vehicles and open their doors.

In figure 6, a sample of the collected data is shown.

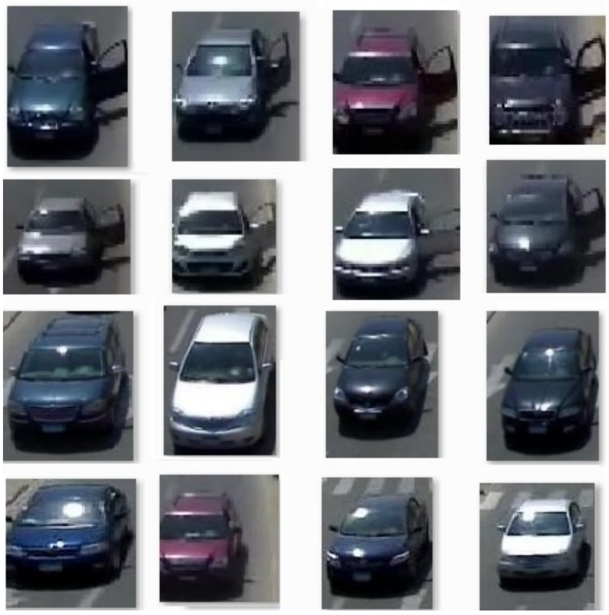

Figure 6: **Images of Real Closed and Opened Doors Vehicles**

The data collection took about three months because of the difficulty of asking vehicles drivers to stop and open their doors as this was hindering the flow of the traffic inside the campus. Therefore we were waiting for the time that not more than three vehicles are passing by. Moreover, in order to encourage drivers to stop for us and make them sure that we don't need a ride, we thought of distributing juice and other candies. As a result, we had time to ask them to open their door. We were also keen to stop different models of vehicles at different places of the road to cover as much as possible differnt cases.

We then processed the video farmes to extract subimages of the vehicles form the sequence of video frames with keeping the aspect ratio.

We were able to collect 400 images for opened door vehicles and more than 1000 images for closed door vehicles.

5.2 Experiment-A: Background Subtraction

Objective

Background subtraction segments foreground objects more accurately in most cases compared to other common moving object detection methods, and detects foreground objects even if they are motionless. However, one drawback of traditional background subtraction methods is that they are susceptible to environmental changes, for example, gradual or sudden illumination changes. The reason for this drawback is that most methods assume a static background, and hence one needs to update the background model for dynamic backgrounds. The update of the background model is one of the major challenges for background subtraction methods.

The used multitracking algorithm is highly dependent on the blobs detected as mentioned before in the proposed approach section. Therefore, the output of the background subtraction is very crucial to our system as it produced blobs that will be tracked and based on it all the other components will take its place. Most importantly, the output of background subtraction is used to calculate the densities of the vehicles. For the importance of this step, we will compare the performance of 12 various background subtraction techniques.

Method

We compared different Background Subtraction Techniques implemented in OpenCV C++ Background Subtraction Library:

1. (StaticFrameDifferenceBGS)Static Frame Difference
2. (FrameDifferenceBGS) Frame Difference

3. (WeightedMovingMeanBGS) Weighted Moving Mean
4. (WeightedMovingVarianceBGS) Weighted Moving Variance
5. (MixtureOfGaussianV1BGS) Gaussian Mixture Model V1
6. (MixtureOfGaussianV2BGS) Gaussian Mixture Model V2
7. (GMG) GMG
8. (PixelBasedAdaptiveSegmenter) Pixel-Based Adaptive Segmenter (PBAS)
9. (MultiLayerBGS) Multi-Layer BGS
10. (DPEigenbackgroundBGS) Eigenbackground / SL-PCA
11. (LBAdaptiveSOM) Adaptive SOM
12. (LBFuzzyAdaptiveSOM) Fuzzy Adaptive SOM

Results

The results are shown in the following subsections from subsection 5.1.1 to 5.1.12.

5.2.1 Static Frame Difference

The time taken by this algorithm is 10.558 seconds; which is very short time as it does not need much time

The output blobs as shown in figure 7 are full of discontinuities and even when there is no vehicles there were some noise produced because there were changes in illumination.

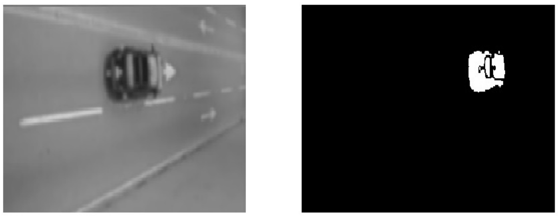

Figure 7: **Static Frame Difference Output**

5.2.2 Frame Difference

The time taken by this algorithm is 10.583 seconds; which is very short time .

The sample of output in figure 8 shows that it is of low quality, which is expected as it just subtracts each image from the image before so we will found a lot of discontinuities.

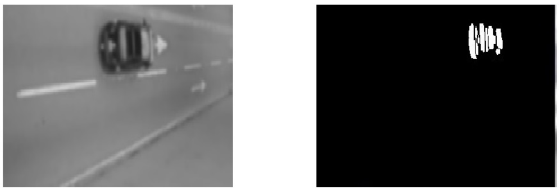

Figure 8: **Frame Difference Output**

5.2.3 Weighted Moving Mean

The time taken is 10.089 seconds and the quality of the output as shown in figure 9 is much lower than that of other approaches and therefore it would need a lot of preprocessing.

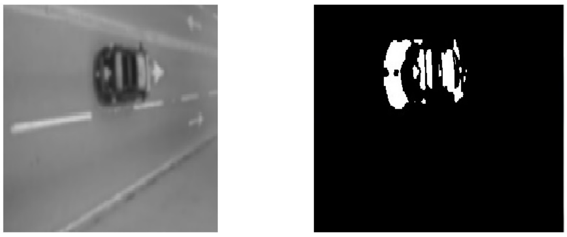

Figure 9: **Weighted Moving Mean Output**

5.2.4 Weighted Moving Variance

The time taken by this algorithm is 15.949 seconds.

The output as demonstrated in figure 10 is better than the weighted moving mean and is almost as good as static frame difference. This output also needs a lot of preprocessing to be used.

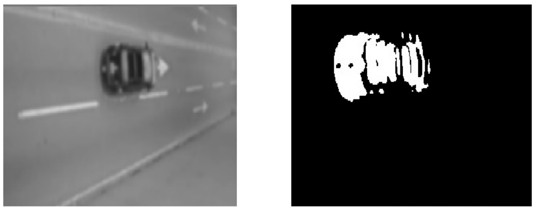

Figure 10: **Weighted Moving Variance Output**

5.2.5 Gaussian Mixture Model V1

The time taken is 10.972 seconds and as shown in figure 11,it produced the worst output.

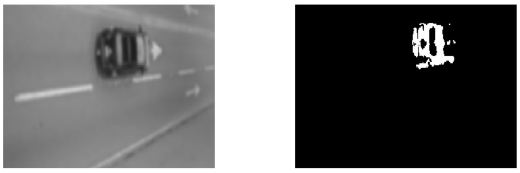

Figure 11: **Gaussian Mixture Model V1 Output**

5.2.6 Gaussian Mixture Model V2

The time taken by this algorithm is 15.949 seconds which is the longest time take.

The output in figure 12 is better than that of version 1 but still of low quality and has a lot of discontuities.

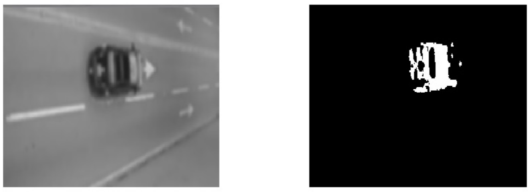

Figure 12: **Gaussian Mixture Model V2 output**

5.2.7 (Godbehere-Matsukawa-Goldberg) GMG

The time taken by this algorithm is 15.696 seconds.

As shown in figure 13, the output has less discontuities than many of the other previously discussed approaches.

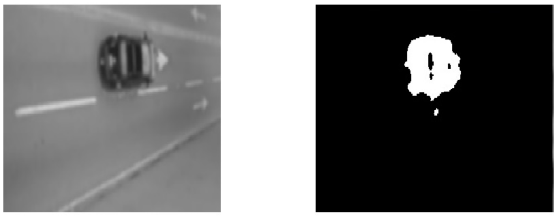

Figure 13: **GMG Output**

5.2.8 Multilayer Background Subtraction

The time taken by this algorithm is 42.56 seconds as shown in figure 14, the produces blobs still contain discontinuities.

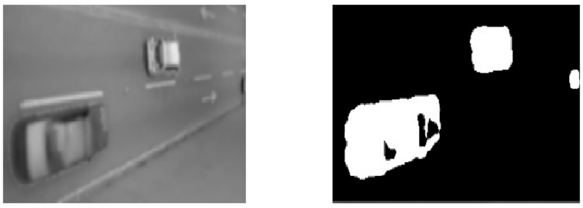

Figure 14: **Multilayer BGS Output**

5.2.9 Eigen Background SL/PCA

The time taken by this algorithm is 10.679 seconds.

As shown in the sample of output in figure 15, the output needs a lot of preprocessing to be used as it has noise and a lot of discontuities.

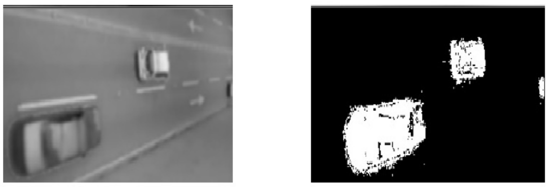

Figure 15: **Eigen Background SL/PCA Output**

5.2.10 Adaptive SOM

The time taken by this algorithm is 13.696 seconds which is short time.

As shown in figure 16, the output is very similar to the output of Eigen Background SL/PCA whish is of a very low quality.

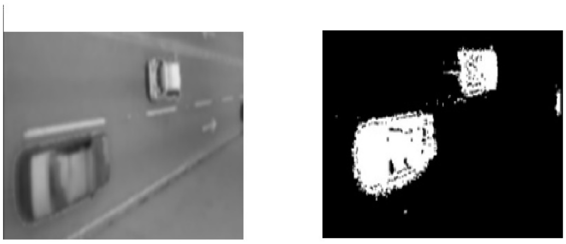

Figure 16: **Adaptive SOM Output**

5.2.11 Fuzzy Adaptive SOM

The time taken by this algorithm is 19.234 seconds

The sample of blobs shown in figure 17 are also of low quality and proves that they need preprocessing before being used by the blob tracking algorithm.

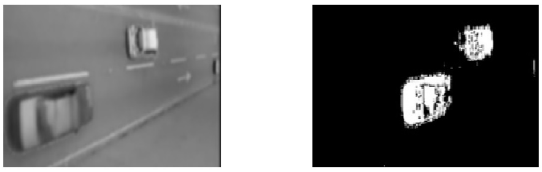

Figure 17: **Fuzzy Adaptive Sum Output**

5.2.12 Pixel Based Adapive Segmenter

The time taken by this algorithm is 56.78 seconds which is the longest time taken.

As shown in figure 18, this output is the best compared to all others, but it took the longest time. But we should consider that it does not need any preprocessing before being used by the blobs tracker. In addition, with powerful processor this time will be dramatically reduced. Most importantly, this approach keeps the history of the produced blobs so as to keep detecting the objects even after stopping their motion.

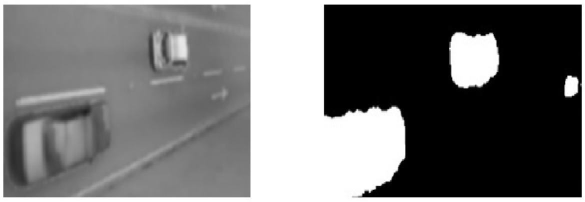

Figure 18: **Pixel based adaptive segmenter Output**

Discussion

We can conclude that pixel based adaptive segmenter gives better results than most of the background subtraction approaches. However, PBAS takes long time to detect the foreground objects but this problem time can be handled by using powerful processors. Moerevoer, we didnt need extra preprocessing such as dilation and erosion. It also has one benefit that is not in many other dynamic background subtraction approaches, which is keeping history of vehicles so as to produce the blob even after stopping which helps us in the tracking. Otherwise, we wont be able to track the blobs after stopping beacause some other dynamic background subtraction approaches consider the object as a part of the new background model if it stopped motion for many frames.

5.3 Experiment-B: Unattended Vehicle's Detection: SVM and ANN with SIFT and SURF features reduced to fixed sizes by 2 different techniques

Objective

The contribution of the proposed approach basically depends on vehicles classification into vehicles with a closed door or with an opened door on the drivers side to be able to recognize unattended vehicles so that we can compute the new proposed metric.

A lot of experiments have been conducted in this phase because this is the most important component of the architecture of the proposed approach as it represents our main contribution on which the whole system depends. In this experiment, we compared SVM and ANN.

Method

To make the comparison between SVM and ANN, 8 experiments have been conducted to compare the performance of different combinations of building blocks to choose the best methodology for classifying vehicles as it is going to be shown in this section. In all these experiments, some parameters were set to fixed values generated from many testing iterations, such as the number of clusters and the number of iterations, the minimum error and the number of hidden neurons for training the classifiers. In addition, to the percentage of the available data that will be only used for testing, which is to 40% out of 480 images (300*450)as it was found to give the best results. It is important to consider that we are using the same sets of data for training and testing ANN and SVM.

The size of the input to the ANN and SVM classifiers was 6*128 when using the SIFT features and 6*64 when using the SURF features. For the Ann we used a hidden layer of 7 neurons.

Table 1: **SVM and ANN with SIFT and SURF Reduced to Fixed Size in Two Ways**

Features	Clustering	Classification	Accuracy		F-Measure	
			Close	Open	Close	Open
SIFT	No	SVM	81.13%	79.59%	89.57%	89.26%
SURF	No	SVM	61.19%	58.73%	75.91%	73.99%
SIFT	Yes	SVM	50%	53.96%	66.66%	70.09%
SURF	Yes	SVM	39.53%	26.71%	44.94%	39.64%
SIFT	No	ANN	50.03%	45.61%	69.30%	62.64%
SURF	No	ANN	60.76%	47.91%	77.86%	64.78%
SIFT	Yes	ANN	22.85%	28.94%	37.20%	44.88%
SURF	Yes	ANN	18.61%	28.88%	27.64%	42.93%

Results

Based on the fact we could not find other projects that detected open door vehicles, we were unable to compare our results with previous work.

The results obtained from running the experiments are shown in table 1.

Discussion

By looking at the results calculated in table 1, it was obvious that, SVM has far better results than ANN in almost all the experiments. This is not a surprising result because in most of the research done before, it is proved that SVM produces more accurate results than the ANN and it overcomes many other machine learning techniques [34]

When comparing the results obtained by using the features extracted from SIFT and SURF, according to the accuracy measure, the features extracted by the SIFT are much more representative than that of SURF, this was obvious because the best accuracy (81.31%) was obtained using the SIFT features and then classified using the SVM Classifier, but when the SURF features were used instead, the result decreased by about 20% to be 61.19%.

According to literature survey, SURF is fast and has good performance as the SIFT but it is not stable to rotation and illumination changes. The 3D MAX models used were subjected to different illumination conditions [35].

Although the clustering approach was used extensively to input a set of a fixed size to the classifier throughout the literature, for all the experiments that have been carried out, it is found that the simpler approach of dividing the image into fixed number of equal sized blocks and then taking one interesting feature(with 64 descriptors if SURF and 128 descriptors if SIFT) form each image improves the best achieved accuracy by about 20% for both the closed and opened vehicles This is probably because the clustering averages the features. Thus, it does not give the best representatives for the object. However, taking the most interesting features in each sub image does not approximate. In addition, it covers all the important features in the image.

Table 2: **Results from Cropped Images Using SVM with SIFT**

Data	Accuracy		F-Measure	
	Close	Open	Close	Open
Full	81.13%	79.59%	89.57%	88.62%
Cut	87.93%	86.79%	93.57%	92.92%

5.4 Experiment-C: Unattended Vehicle's Detection : SVM on Full Images against Cropped Input Images

Objective

It was also clear that a lot of incorrect classification occurred when using the full images whether using ANN or SVM. Therefore, we needed to make it easier for the classifier to distinguish between opened and closed vehicles.

Method

We found that the main features that distinguish the different classes are mostly in the right part of the image as shown in figure 19. Therefore, we worked on an area of size 200*300 at the right corner of the image; at the driver's side. We used SVM with SIFT features as SVM and SIFT proved in the previous experiment to give better results than ANN and SURF in all the different cases.

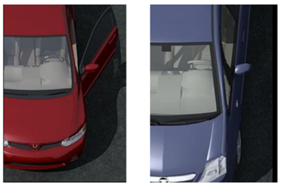

Figure 19: **Cropped Images**

Results

The results are shown in table 2.

Discussion

The best obtained accuracy of the closed vehicles was improved by approximately 6% by using the cropped images and the accuracy of the opened vehicles is improved by almost 8%. This behavior of using the full images is caused by mainly three reasons: 1) A lot of redundant features are extracted which increases the complexity of the classification 2) a lot of common features between the two classes will be extracted which reduced the accuracy of the classification and 3) the possibility of having other opened door than the drivers in the image. Therefore, the best accuracy 87.93% is obtained by working only on the front right qaurter of the image then one SIFT feature and its corresponding descriptors are extracted from each of the subimages of that quarter. After features extraction, SVM classifier runs on the descriptors.

5.5 Experiment-D: Unattended Vehicle's Detection : SVM and ANN against HAAR Calssifiers on the 3D MAX Models

Objective

After the integration of the components of the proposed approach, we found that the blobs returned by the blob detection are not always of the same size of the vehicle. Sometimes it is smaller and sometimes it is bigger due to changes in illumination and the presence of shadows. Therefore, we couldn't devide the image into subimages or work on one quarter of the image. Thus, we had to work on the SVM or ANN without sampling or cropping and as shown in table 1, the accuracy of opened door's recongition was very low. For these reasons, we needed to compare our results to the haar classifier as it uses in the sliding window to detect the objecst. So by extracting a bigger subimage from the video frame than the dimensions given by the blob detector, the vehicle can be recognized as the window is sliding over all the subimage. It also requires small processing time as it is required for our real time system.

Method

We worked on the same set of 3D MAX modles used in the previous experiments without applyig any preprocessing using the same set of training and testing which was used before by both SVM and ANN. We used the Haar classifier implemented by the OPenc CV library.

Table 3: **Comparing The Results of Haar, SVM and ANN**

Classifier	Accuracy	F-Measure
Haar	91%	97.47%
SVM	54%	70.02%
ANN	29%	45.23%

Results

When we trained the HAAR classifier on the same 3D MAX dataset used by SVM and ANN. It almost detected all the opened door vehicles in the testing images, the only problem was that there were some false positives. The HAAR classified many of the closed vehicle as opened door vehicles. Therefore, the accuracy was almost 91% for opened doors vehicles recongnition. 87.93% accuracy was achieved before by SVM working in the same set of images. However, if compared to the SVM and ANN without preprocessing and sampling beacuse they are not applicable in our proposed approach, then Haar classifier proves to give much better results as shown in table 3. These results are achieved using SIFT with both SVM and ANN as they proved to give better results than SURF in most of the cases as demonstrated in table 1.

Discussion

This problem of misclassification by Haar is caused by the presence of big side mirrors in some of the vehicles. As shown in figure 20, this vehicle is detected as an opened door vehicle because the mirror features are very close to the mirror. The HAAR classifier moves its sliding window on the blob passed to it as mentioned in the proposed approach, and when it finds the features which it was trained on, it recognizes the vehicle as an opened door vehicle.

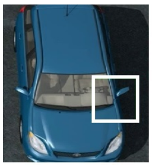

Figure 20: **Falsely Detected Opened Door Vehicle**

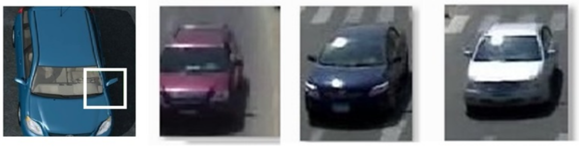

Figure 21: **Difference between 3D MAX Images and Real Images**

However, the problem of the mirrors will never be our problem in the images of real vehicles given by the blob detection in the video frame. As shown in figure 21. The mirror will never as obvious as the in clean 3D MAX Models. In addition, as obvious there is a huge difference between the closed vehicles and the opened vehicles. Furthermore, integrating the classifier with all the components of the proposed approach showed even higher accuracy since the classifier is not called except when there is a vehicle that was detected as stationary.

Table 4: **Comparing The Results of Haar on 3DMAX models and Real Vehicles**

Images	Accuracy	F-Measure
3DMAX	91%	97.47%
Real Images	65%	79.50%

5.6 Experiment-E: Unattended Vehicle's Detection :HAAR Applied on Real Images from the Video Frames

Objective

We needed a classifier that accurately recognizes opened door vehicles in any given frame from a road video sequence. Therefore, we had to test the accuracy of the HAAR classifier as it has a multiscale sliding window that can detect the objects which has the same features that it has been trained on. So by extracting larger subimages than the detected blobs, vehicles can be recognized using the Haar classifier. Forthermore, Haar features are characterized to be fastly computed and extracted.

Method

In order to train the HAAR classifier, we had to collect as many images as possible for opened door vehicles.

We had 400 images of opened door vehicles and more than 1000 images of closed door vehicles. Thus, We used 240 opened door vehicles images and 600 closed door vehicles images as negative images for training.

Results

The accuracy achieved by HAAR classifier was almost 65% for the opened door vehicles recongnition as shown in table 4.

Discussion

Working on 3d Max models gave better results as demonstrated in table 4 and the reason behind that is the presence of shadows in most of the input images as shown in figure 22. Thus the intersection of the edges of the shadows with the opened door helps the used feature extraction method to produce more clear and unique features in the case of the opened vehicles, that do not exist in the case of the closed vehicle, which consequently increases the accuracy of the performance of classification. However, this intersection was never that clear on our video frames of real vehicles. However, this accuracy is good for the current state. as we couldn't handle all the possible angles, various illumination conditions and different types of vehicles (big buses, vans, etc.) In addition, there were still few misclassification for the mirrors as doors when turning. Furthermore, not all the drivers open their doors to that extent that can be detected.

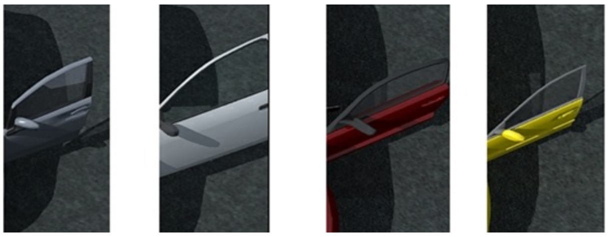

Figure 22: **Doors Intersection with Shadows**

5.7 Experiment-F: Traffic Density, Traffic Flow and the New Proposed Metric

Objective

The value of traffic density measures only the ratio between the vehicles density, which is the white pixels produced from the background subtraction, and the road density, which is the black pixels of the background and the white pixels of any moving objects detected from the background subtraction, simply we can say the whole size of the image . So based on this measure, the traffic control system will compare between different roads in the intersection to take the decision for the traffic light and the time interval given.

Assuming that we have a road with with 10 moving vehicles and for example 5 stationary vehicle or delayed vehicles. Therefore, there is traffic congestion in this road. But in the end the traffic density is the same as the traffic density of a road with no stationary vehicles. For example, another road in the intersection has 15 moving vehicles without any obstacles (stationary / delayed) will be given the same time interval as the one that have many stationary vehicles that delay the traffic flow leading to traffic congestion later on.

Another one of the most used metrics for the automatic traffic light control of the traffic flow. It is measured as the rate at which vehicles pass a fixed point (vehicles per a specified time interval). It counts the number of vehicles that passes a given point/frame during a certain time interval.

However, this metric also has problems. First of all, let us consider the case that given an intersection, one of the roads has very few vehicles moving for any reason, while another one, the opposite direction for example has much more vehicles because people most likely to go to this direction. So in this case, according to the traffic flow measurement, the first road will be considered of having much more less flow than the second one and it will be given much higher weight than the second one when taking the decision of the traffic light control. In addition, if we took only one point, we will see that in roads like in Egypt, sometimes the vehicles may not pass by the given point intersection point. Therefore, from these two sections, it is obvious that we need to combine both of the measures together in order to have the most effective and efficient way of vision based automatic traffic congestion estimation.

Method

After extraction of the foreground objects using pixel based adaptive segmenter to be variant against different external factors such as weather and illumination,

blobs were detected and tracked be able to detect if the objects left the boundaries as shown in figure 3. If the object were found that it left the boundaries of the frame, then it wasn't included in computing the proposed metric. During the presence of the blob, we used the tracker also to check its status, if it was moving or stationary. If it was moving, then given the dimensions calculated by the vehicle tracker, the subimage within these dimensions was extracted from the main video frame, HAAR classifier was used to recognize the vehicles from between the detected moving objects in order to remove the noise encountered due to the changing external factors such as, shadows and other unnecessary information. We trained another Haar classifier to detect closed door vehicles, because Haar doesnt classify the objects into closed and opened vehicles, it just gives a feedback if the object is found or not. Thus we trained another Haar classifier on 1000 closed door vehicles as positive images and 5672 random images of road, clowds, trees etc. as negative images.

If it is a stationary object, then also given the dimensions calculated by the vehicle tracker, the subimageat these dimensions was extracted from the main video frame to detect if there was a vehicle that had the door opened at the drivers side(left side in Egypt and it depends on the traffic direction) in this subimage as demonstrated in figure 4. If it was found then the object was considered as an unattended vehicle until the tracker notifies that it started moving agaib because this means that the vehicle's driver is going to leave it for a while. Otherwise, if no opened door vehicle was found then haar classifier is used again to find if there is a closed vehicle within the subimage. If the was object was detected to be a vehicle, it was considered to be a delayed vehicle until the tracker gives another information about it motion again. Lastly, the proposed metric was calculated by doubling the densities of unattended vehicle and raising the densities if delayed vehicles by a factor of 1.5. Then, the sum of densities of all vehicles was divided by the sum of all the road pixels, just as the traffic density metric.

Perspective distortion was taken into account so that pixels of vehicles closer to the camera are not given the higher pixel weight as the ones further back.

Results

Black windows are drawn around the moving vehicles, double line black windows are drawn around the delayed vehicles and the detected unattended vehicles are surrounded by white windows.

In figure 23, there are 2 moving vehicles so there are no penalties. Thus, the new propsed metric value is 0.14 or 14 % of the road is covered.

The value of traffic density is 0.14 or 41%

The value of traffic flow is 2

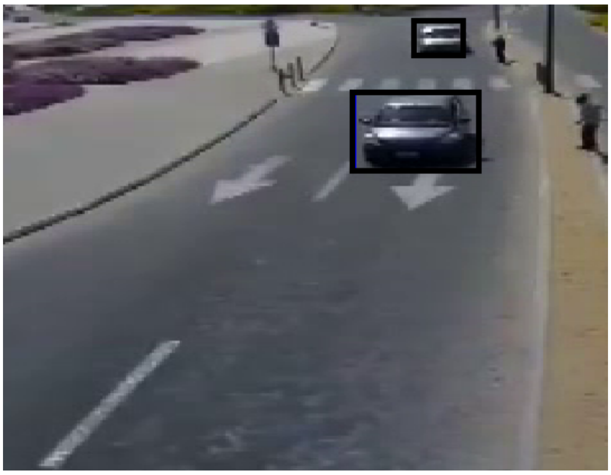

Figure 23: **Video Frame with 2 Moving vehicles and No Stationary Vehicles**

In figure 24, is one unattended vehicle because it was detected before as an opened door vehicle and one moving vehicle. Since according to the proposed approach, density of unattended vehicle is doubled as to give it a higher weight because of the delay it may cause. Thus, the value of the new proposed metric is changed to be 0.2 which means that 20% of the road is covered.

The value of traffic density is 0.14 or 14%

The value of traffic flow is 2

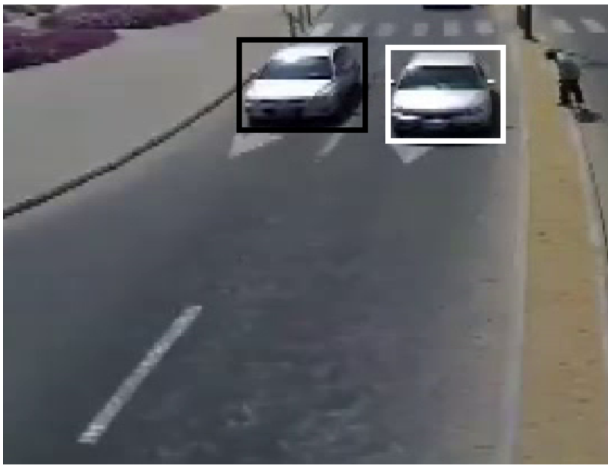

Figure 24: **Video Frame with 1 Unattended Vehicle and 1 Moving Vehicle**

In figure 25, there is one stationary vehicle that was not recognized before as unattended vehicle , so it is recognized as delayed vehicle. In this frame there is also one moving vehicle. As a result, the density of the delayed vehicle is raised by a factor of 1.5. Thus, the new propsed metric value is 0.17 or 17 % of the road is covered.

The value of traffic density is 0.14 or 14%

The value of traffic flow is 3

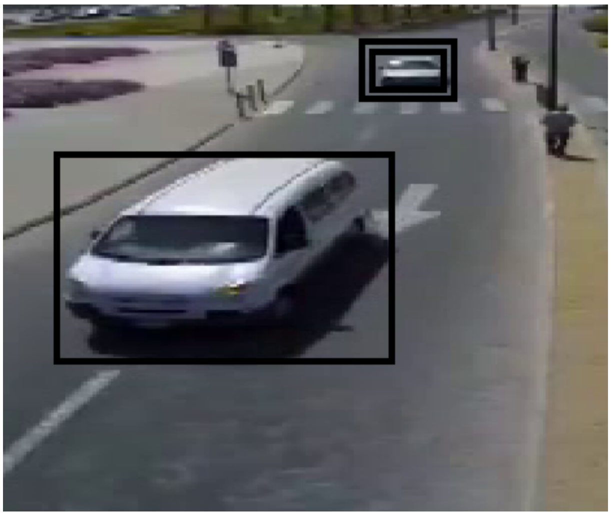

Figure 25: **Video Frame with 1 Delayed Vehicle and 1 Moving Vehicle**

In figure 26, there are two stationary vehicles. One opened door vehicle and one delayed vehicle. Thus, the new propsed metric value is 0.21 or 21 % of the road is covered.

The value of traffic density is 0.22 or 22%

The value of traffic flow is 2

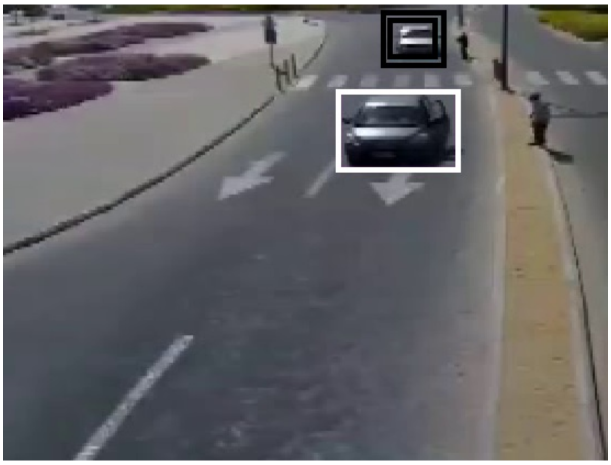

Figure 26: **Video Frame with 1 Delayed Vehicle and 1 Unattended Vehicle**

In figure 27, there are 3 moving vehicles so there are no penalties. Thus,the new propsed metric value is 0.21 or 21 % of the road is covered.

The value of traffic density is 0.21 or 21%

The value of traffic flow is 3

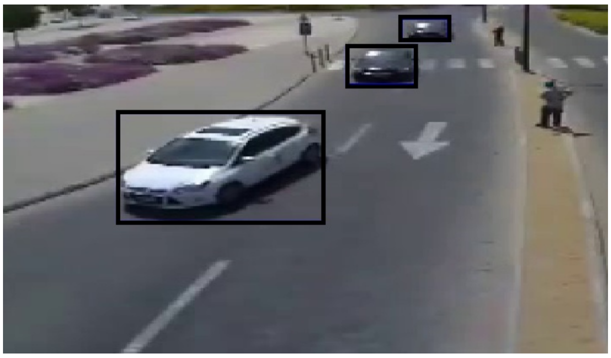

Figure 27: **Video Frame with 3 Moving Vehicles and No Stationary Vehicles**

In figure 28, there are two stationary vehicles. One was recognized as an opened door vehicle and the other one was detected before as unattended because it was recognized as an opened door vehicle in a previous frame. In the frame there is also one moving vehicle. Since according to the proposed approach, the opened door vehicle is considered as an unattended vehicle. Therefore, the densities of the two stationary vehicles are doubled as to give it a higher weight because of the time delay that they cause in the road. Thus, the value of the new proposed metric is 0.31 or 31% of the road is covered.

The value of traffic density is 0.21 or 21%

The value of traffic flow is 3

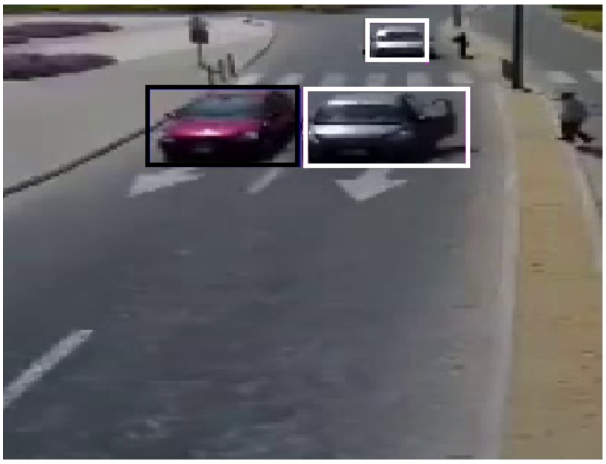

Figure 28: **Video Frame with 2 Unattended Vehicles and 1 Moving Vehicle**

In figure 29, there is one stationary vehicle that was detected as unattended vehicle because it was recognized as an opened door vehicle in a previous frame, one stationary vehicle that was not recognized before as unattended vehicle , so it is recognized as delayed vehicle. In this frame there is also one moving vehicle. As a result, the two stationary vehicles are given higher weights to compute the proposed metric. The density of the unattended vehicle is doubled, while the density of the delayed vehicle is raised by a factor 1.5. Thus, the new propsed metric value is 0.29 or 29 % of the road is covered.

The value of traffic density is 0.21 or 21%

The value of traffic flow is 3

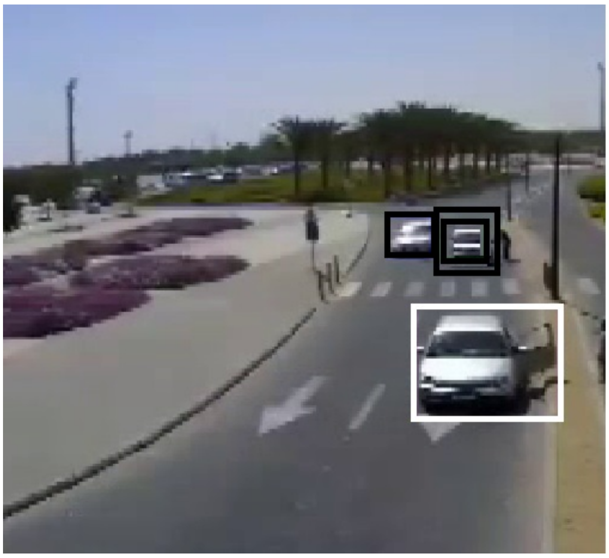

Figure 29: **Video Frame with 1 Unattended vehicle, 1 Delayed and 1 Moving vehicle**

AS shown in figure 30, this frame has a stationary delayed vehicle, and 2 closed moving vehicles. As a result, the new propsed metric value is 0.24 or 24 % of the road is covered.

The value of traffic density is 0.21 or 21%

The value of traffic flow is 3

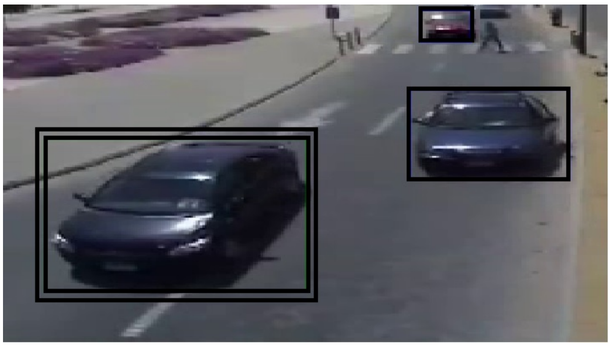

Figure 30: **Video Frame with 1 Delayed Vehicle and 2 Moving Vehicles**

Discussion

Most of these cases were repeated more than 10 times during the recorded videos of the road traffic but we have just taken some illustrative samples to compare between the different metrics.

In the first four examples that we handled here in our experiments, only 2 vehicles passed in the given frame in 20 seconds. As a result, the traffic flow was evaluated as 2 for all the frames. The same for the last four examples, only three vehicles passed in 20 seconds. Therefore, the traffic flow was 3 for all the different conditions of the vehicles.

In figure 24, there was one stationary vehicle that was detected as an opened door vehicle in the previous frames, so its density was doubled to make the traffic congestion evaluated by 20%. Therefore, the value computed by the new proposed metric showed that there is more jam in the road captured in this frame than that in figure 23 which was 14%. There was also one stationary vehicle in figure 25, but it was just a delayed vehicle. Therefore, its weight was raised by only 1.5 to make the traffic congestion be 17%. Moreover, in figure 26, there were one unattended vehicle and one delayed vehicle. Therefore, the value computed by the proposed metric is 22% which demonstrates that it had higher congestion than all the other different cases of 2 vehicles given in frames 23, 24, 25 and 26. However, in all the cases, the ordinary traffic density metric gave the same value which is 14%.

For the given case in figure 28, proposed metric (31%) also gave higher weights for the two unattended vehicles. Consequently, it was more obvious that the road is more jammed than that in figure 27 (21%) which had all its vehicles moving. While in the case given by figure 29 (29% traffic congestion), there was also two stationary vehicles, but one of them was just delayed . Therefore, it would be given a lower weight than that of unattended vehicles, which is 1.5 its weight. Thus, this road is considered to be more congested than a road with three moving vehicles, but not as much as in figure 28 and more than in figure 27. For figure 29 (24% traffic congestion), a higher weight was given for only one vehicle because it was just delayed while traffic density gave the same value, 21%, for the four cases of the three vehicles.

Obviously, in all the given video frames, the previous traffic density metric gave the same value for the different conditions of the two vehicles or three vehicles. Thus, in all the given cases, there is no difference between moving and stationary vehicles and therefore the traffic density is the same. However, the new proposed metric gave different values for the different conditions in the given examples. It has highly distinguished between all the presented cases.

To sum up, the new proposed metric is more informative and representative than the other mentioned metrics that were widely used in previous work. The metric was able to dramatically distinguish between the different conditions of the vehicles for more efficient automatic traffic light control.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Automatic vision based traffic congestion estimation has recently become one of the growing areas of research related to traffic systems and traffic light control.

Research in automatic vision based traffic congestion estimation hasn't taken the effect of the unattended and delayed vehicles on the flow of the traffic into consideration when measuring both traffic density and traffic flow

The main objective of this research work was to compute a new metric that incorporates stationary vehicles when measuring the traffic congestion and test its impact on representing the traffic congestion in any given video frame. The approach followed to achieve these objectives was: 1) suggesting a mechanism for stationary vehicles detection (background subtraction, blob detection and mutlitracking); 2) proposing a methodology for unattended vehicles detection; 3) proposing a metric that gives more importance for road with stationary vehicles specially unattended vehicles during the automatic traffic light control; 4)combining those methodologies for the aim of traffic congestion estimation; and 5) Comparing the results to the previous metrics used in previous work for traffic congestion estimation to prove the effectivess of the new proposed metric in the traffic congestion representation.

Following our proposed approach, we started by testing different approaches for background subtraction approach to be able to select the approach to help us to detect and track the vehicles in our video frames. We found that PBAS keeps the history of vehicles so as to produce the blob even after stopping to be able to track them and detect their stoppage. This approach also did not need extra preprocessing such as dilation and erosion for the produced blobs. According to our experiments, blob tracking gave very satisfying results by working on the output of PBAS.

The new proposed metric depends on the unattended vehicles detection. Therefore, we conducted various experiments to detect opened door vehicles because opened door vehicles are the best indication for getting multiple stationary vehicles in many of the successive frame. We used 3D MAX models for testing the feasibility of the proposed approach in the begining.

In our experiments we used Matlab Toolbox for machine learning to implement Artificial Neural Networks(ANN), Support Vector Machines (SVM), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). For this comparison, our data set consisted of 480 synthetically generated images by 3D MAX software for both open door and closed doors vehicles from an up-front camera point of view. The image resolution used was 300 x 450 pixels. We used 60% of the data for training and 40% for testing. In ANN we used one hidden layer consisting of 7 neurons, while in SVM we used a Radial Bias Function(RBF) kernel. And for the input, SIFT created 128 descriptors for each feature while SURF created 64. Both ANN and SVM were given the features extracted by SIFT and SURF to compare their output. These features varied according to the vehicle inserted. The input to SVM and ANN had to be fixed in size so we used K-means clustering algorithm to cluster features with respect to their descriptors (multidimensional clustering). We also examined a simpler approach, which divided the image before the extraction of the features into 6 sub-images to extract one feature from each. The latter approach gave 20% higher accuracy, this was due to the fact that clustering takes the average of the features which give lower accuracy.

We found that the main features that distinguish the different classes are mostly in the right corner of the image. Therefore, we worked on an area of size 200*300, which improved the accuracy by 8%. Moreover, applying edge detection on the images using the canny filter before feature extraction improved the accuracy significantly by 10 to 20%.

According to accuracy measurements, SVM produced better results than ANN in all the cases. Moereover, The features extracted by the SIFT were much more representative than that by the SURF features as the best accuracy (87.93%) was obtained using the SIFT. These features were extracted from the cropped door images then reducing them to fixed size by dividing the images into equivalent subimages and taking one interseting feature from each and passing them to the SVM classifier.

Therefore we found that it was best to pre-process the images before classification by cropping the image to only the right quarter of the image at the driver's side. The next step was to extract the SIFT features and their corresponding descriptors from subimages and passing them to the SVM classifier.

We couldn't depend on using the dimensions the detected blobs to extract subimages that contain the foreground objects, becaue these blobs are not alywas accurate due the illumination conditions. Therefore, these subimages do not always include the whole object which affects the results. In addition, taking only the right corner to detect the opened door vehicle was impossible as sometimes the blob. Thus, the Haar classifier was tested as it uses the sliding window that detects the desired faetures at any scale and also at high speed.

HAAR classifier achieved 91% accuracy without any preprocessing which is better than the best accuracy achieved by the SVM and ANN without cropping or sampling.

The only problem was the HAAR classifier that it classified some closed doors vehicles as opened doors ones. This problem of missclassification was because of big side mirrors in some of the vehicles. However, the mirrors are not that sharp as doors in video frames of real vehicles. Moreover, integrating the classifier with all the components of the proposed approach showed even higher accuracy since the classifier is not called except when there is a vehicle that was detected as stationary. Thus, the possibility of having false negatives will be reduced dramatically in the application. Therefore, the HAAR classifier was considered as a convenient classifier for our proposed approach to be applied for real time application.

For training the HAAR classifier to detect vehicles in the video frame sequence, we collected real data samples from inside the parking area of the university campus. The accuracy obtained after working on real data is 65 %. Which is very satisfying as we could not get images for all the different perspectives, angles, various illumination conditions and different sizes and types of the vehicles.

We trained another Haar classifier to detect closed door vehicles, because

Haar doesn't classify the objects into closed and opened vehicles, it just gives a feedback if the object is found or not. As a result, we couldn't depend on the Haar classifier that detects opened door vehicles to detect closed vehicles too. For this reason, we trained another Haar classifier on 1000 closed door vehicles as positive images and 5672 random images of road, clowds, trees etc. as negative images and it gave accuracu of about 80 %.

After the integration of the different components of the proposed approach to compute the new proposed metric, we were able to see the difference between our metric and the previously used approaches such as traffic density and traffic flow. Our metric was able to distinguish between the different conditions of the road, so as to minimize the delay of the vehicles.

There are different directions for extending this system. One direction could be further improving our classifier by training the classifier on various illumination conditions and different types of vehicles such as buses. In addition to giving different weights to vehicles according to their speed nit just when they completely stop motion. Another direction for future work could be addressing background subtraction drawbacks, e.g., shadow detection and removal. Finally, given that this research work is part of a bigger system which also includes traffic light control based on the information retrieved from the traffic congestion metric, one possible direction could be detecting the vehicle's plate and the vehicle's number to impose fines on the unattended vehicles in the prohibited places .

REFERENCES

[1] Vasconcellos, E., ”Traffic Accident Risks in Developing Countries: Superesiding Biased Approaches,” *ICTCT extra workshop*, pp. 04611-003, 2005.

[2] Gupta, P., G.N, Purohit and Dadhich, G.N, ”Approaches for Intelligent Traffic System: A Survey,” *International Journal on Computer Science and Engineering (IJCSE)*, 2012.

[3] Srinivasan, D., Choy, M.C. and Cheu, R.L., Neural Networks for Real-Time Traffic Signal Control, *ITS(7), No. 3*, pp. 261-272, 2006.

[4] ,Tomnescu, O.,Madalina, I., Elena, A. and Batros, I.” Adaptive Traffic Light Control Ssytem Using AD HOC Vehicular Communications Network,” *ISSN Vol. 74*, pp. 1454-2358, 2012.

[5] X. Ji, Z. Wei, Y. Feng,” Effective vehicle detection techniques for traffic surveillence systems,” *J. Vis. Commun. Image R. 17*,pp. 647-658, 2006

[6] Vikramditya,D., Amol, P., Kshitij, P. and Rathod S.S., Image Processing based Intelligent Traffic Controller, *Undergraduate Academic Research Journal, ISSN, V.1*, pp. 2278-1129, 2012.

[7] Liu, Y. and Payeur, P., Vision-based detection of activity for traffic control, *IEEE CCECE* 2003.

[8] Ozkurt, C. and Camci, F., Automatic Traffic Density Estimation and Vehicle Classification for Traffic Surveillance System Using Neural Networking, *Mathematical and Computational Applications, V. 14*, pp. 187-196, 2009.

[9] Bai, L., Tompkinson, W. and Wang, Y., Computer Vision Techniques for Traffic Flow Computation, *Journal of Pattern Analysis and Applications, v.7*, pp. 365-372, 2005.

[10] Choudekar, P., Banerjee, S. and Muju, M.K., Implementation of image processing in real time traffic light control, *International Conference on Electronics Computer Technology, v.2*, pp.94 -98, 2011.

[11] Sanchi, S., Real time Traffic light control and Congestion avoidance system, *International Journal of Engineering Research and Applications (IJERA) ISSN, Vol.2*, pp. 925-929, 2012.

[12] Elgammal, A., Duraiswami, R., Harwood,D., and Davis, L. S.,Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proceedings of the IEEE, vol. 90*, pp. 1151-1163, 2002.

[13] Sergey, V. and Vasilij N. , Artificial Neural Networks for Adaptive Management Traffic Light Objects At The Intersection, *Proceedings of the 10th International Conference Reliability and Statistics in Transportation and Communication (RelStat10)*, pp. 457-462, 2010.

[14] Akilandeswari, U., Nadu,T., Nithya,R. and Santhi, B., Review on Fea-

ture Extraction Methods in Pattern Classification,European Journal of Scientific Research, " *ISSN 1450-216X Vol.71*, pp. 265-272, 2012.

[15] Angelova, A., Matthiesx, A., Helmick, D., and Perona, D., Dimensionality reduction using automatic supervision for vision-based terrain learning,, *Robotics: Science and Systems Conference*, 2007.

[16] Bay, H., Ess, A., Tuytelaars, A. and Gool V., "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding , vol.110*, pp. 346359, 2008.

[17] C.J.C.Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 2, 121-167, (1998).

[18] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001.

[19] P. M. Panchal, S. R. Panchal and S. K. Shah, A Comparison of SIFT and SURF, International Journal of Innovative Research in Computer and Communication Engineering , 1, 2320 9798, 2013.

[20] Mikolajczyk, K., A performance evaluation of local descriptors, *CVPR, Vol. 2*, pp. 257 263, 2003.

[21] Mikolajczyk, K., Indexing based on scale invariant interest points, *ICCV, Vol. 1*, pp. 525 531, 2001.

[22] Deep J. Shah, "Motion Based Bird Sensing Using Frame Differencing and Gaussian Mixture", UCR Undergraduate Research Journal Vol.2, 2008.

[23] Bijan Shoushtarian, Helmut E. Bez, "A practical adaptive approach fordynamic background subtraction using an invariant colour model andobject tracking", Pattern Recognition Letters 26 ,2005, pp. 526.

[24] J. Barcel, Models, Traffic Models, Simulation, and Traffic Simulation, in Fundamentals of Traffic Simulation, vol. 145, J. Barcel, Ed. New York, NY: Springer New York, 2010, pp. 162.

[25] C. S. Group, Generic Mobility Simulation Framework (GMSF). [Online]. Available: http://gmsf.sourceforge.net/. [Accessed: 28-Apr-2013].

[26] A. Godbehere, A. Matsukawa, K. Goldberg, "Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation", American Control Conference, Montreal, 2012, pp. 4305-4312.

[27] VanetMobiSim - Newcom — Institut Eurecom — Politecnico di Torino. [Online]. Available: http://vanet.eurecom.fr/. [Accessed: 29-Apr-2013].

[28] The smarter, faster way for model-based system development — VisSim. [Online]. Available: http://www.vissim.com/. [Accessed: 29-Apr-2013].

[29] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, SUMO - Simulation of Urban MObility: An Overview, in Simul 2011, The Third International Conference on Advances in System Simulation, 2011.

[30] SUMO - Simulation of Urban MObility - TraCI. Institute of Transportation Systems at the German Aerospace Center.

[31] G. Kotusevski and K. A. Hawick, A Review of Traffic Simulation Software, Technical Report, vol. 320. Massey University, Auckland, 2009.

[32] Wang, J. M. , Chung, Y. C. , Lin, S. C. , Chang, S. L. , Cherng, S. and Chen, S. W. , Vision-Based Traffic Measurement System, *Intl Conf. on Pattern Recognition, vol. 4*, pp. 360-363, Cambridge, 2004.

[33] Schmitt, D. and McCoy, N., Object Classification and Localization Using SURF Descriptors, 2011.

[34] Huang, C., Song, K., Kim, S., John, R.G., Davis, P., Jeffey, G. and Samuel, N., " Use of Dark Object and Support Vector Machines to Automate Forest Cover Change Analysis," *Remote Sensing of Environment, vol.112*, pp. 970-985, 2008.

[35] Luo Juan, and Oubong Gwun, A Comparison of SIFT, PCA-SIFT and SURF, International Journal of Image Processing (IJIP), Vol. 3, Issue 4, pp. 143-152, 2013.