Automatic Vehicle Detection and Driver Identification Framework for Secure Vehicle Parking

*Department of Electrical & Computer Engineering, North Dakota State University, USA.
†{zahid.mahmood, samee.khan}@ndsu.edu
*Faculty of Electrical Engineering, Mathematics, and Computer Science, University of Twente, Netherlands.
†t.ali@utwente.nl
‡Department of Electrical Engineering, COMSATS Institute of Information Technology, Pakistan.
†skhattak@ciit.net.pk
Department of Computer Science St. Francis Xavier University Antigonish, NS, B2G 2W5, Canada.
†ltyang@stfx.ca

Abstract—In recent times, automatic face recognition algorithms are playing a key role in several security applications. In this paper, we develop a framework for enhancing the security of vehicle parking spaces. The proposed framework can be divided into three separate steps. In first step, a vehicle in the input image is spotted. In second step, driver face is located. In final step, a robust face recognition algorithm identifies the driver by comparing the face image with face images in a database. On successful identification of the driver face, vehicle is allowed to enter in parking area. To detect vehicle and face(s), we use Adaptive Boosting algorithm and Haar-like features, while driver face identification algorithm uses Eigenfaces for feature selection and Euclidian distance for classification. To test the face identification, we simulate a challenging situation where only a single facial image of a driver is available and four different poses (+45°, +35°, 0°, and -35°) faces are used as probe (test). These four poses are chosen because a driver’s face orientation can vary while sitting on driving seat from frontal (0°) up to +45°.

Simulation results show very high detection and identification results regardless of the facial pose variation. The results demonstrate the feasibility of developed framework to be deployed in any public vehicle parking area.

Index Terms—Eigenface, Euclidian distance, Face Recognition

I. INTRODUCTION

OBJECT detection is a crucial methodology for independent entity and driver support systems. An autonomous object detection system helps in enhancing security applications¹, avoiding collisions, and monitoring traffic [1]. Real-time objects, such as vehicle detection continuously notifies drivers with crash warning for safe driving. During past two decades, many algorithms, for example, [3]-[5] have been proposed for object/vehicle detection and recognition. Vehicle detection algorithms are computationally complex as they treat the input images continuously. In [6], researchers presented real-time vehicle location scheme utilizing mobile platform. The work has three contributions. Initially, road modeling scheme is proposed to limit detection area followed by application of efficient features and Eigencolors to track vehicle. Finally, chamfer distance classifier is applied to classify the vehicles. However, the work lacks discussion of the application of algorithm in a parking area for security purposes. In [7], authors presented an automatic vehicle plate detection system for car parking areas. However, the work lacks discussion on detection and recognition on low resolution plates. A novel algorithm to recognize license plates was discussed in [8]. The proposed method had the ability to automatically extract and perform license plate recognition based on characters from a captured image. The work also lacks discussion on feasibility of the developed work for security purposes. The performance of the developed algorithm suffers severely for low resolution license plates. In [9] the proposed work only focused on license plate recognition using super resolution method. The performance of the algorithm drops rapidly, if the license plate is broken/affected with mud. In [10], the published work focused on locating the position of the car already parked in a parking area. There was no guarantee that the parked car was authentic/not authentic. Recently, a large number of papers, for example, [11]-[13] have been published focusing only on object detection and recognition, which motivate development of security applications based on face detection and recognition. Our main contributions are given below:

- We develop a complete working system in which a vehicle and driver face is detected followed by a face recognition algorithm to recognize the driver’s face.
- We simulate a challenging task where there is only one gallery (training) face sample is available and four different poses (+45°, +35°, 0°, and -35°) faces are used as probe (test). These four poses are chosen because a driver’s face orientation can vary while sitting on driving seat from frontal (0°) up to +45°.
- We conduct detailed experiments on 1278 images in parking-lots to demonstrate the usefulness of the proposed system. Fig. 1 shows an overview of the developed framework.

¹http://www.disastercenter.com. United States disaster center reports more than 3 million vehicle robberies from the year 2011 onwards.
II. VEHICLE DETECTION USING ADABOOST

For vehicle detection, we use Adaptive Boosting (AdaBoost) algorithm. The AdaBoost algorithm generates a robust final classifier by linearly linking large number of weak classifiers. For vehicle and drivers face finding in current context, we use the approach presented in [14]. Vehicle classifier is generated by AdaBoost training. For a set of training data, Algorithm 1 picks several weak classifiers from many Haar features and combines them into a resilient classifier as described by Eq. (1).

\[
h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\
0 & \text{otherwise}
\end{cases}
\]

where \(x\) is the input image, \(h_t(x)\) are weak classifiers, \(\theta\) is the threshold, and \(h(x)\) is the final strong classifier. Algorithm 1 explains the pseudo code and technical details of AdaBoost algorithm. Moreover, Algorithm 1 reveals that a true classification of a sample results in weight drop, otherwise it remains unaffected. In general, the AdaBoost classifier learns a series of large number of weak classifiers and threshold \(\theta\) to generate the classifier explained in Eq. (1). We compute the Haar features using the Integral Image (see Eq. 2). The cascade manner rapidly filters the irrelevant image areas that do not possess vehicles.

We collect 6200 positive samples (vehicles) and 11000 negative samples (non-vehicles) to train the classifier. The data (total of 6200 + 11000 = 17,200 images) used for training is picked from our developed database of vehicle images. Few examples of training samples are shown in Fig. 2, where vehicles indicate positive training samples, while all others represent negative training samples.
Fig. 2: Training samples

An image captured in any parking area may contain other objects, such as pedestrian, cautionary, warning, or signal poles. Therefore the only purpose of vehicle detection is to restrict the input image zone by reducing the irrelevant information. Vehicle detection provides the exact position of car arriving at the parking entrance. We train the AdaBoost classifier in 32 stages having 1100 weak classifiers.

III. VEHICLE DRIVER FACE DETECTION USING ADABOOST

Face detection part of our framework determines the location of the drivers’ face while sitting in vehicle. Face detection procedure is also similar to the facts described in Algorithm 1 except the use of positive face samples used during training. To detect a drivers’ face, the AdaBoost initiates with by simple classifiers that cluster face like regions. The weak classifiers in early stage of training are simple and immature. The AdaBoost procedure as described in Algorithm 1 gradually rejects non-face driver images. Likewise the Eq. (1), final driver’s face classifier outputs with very high precision and accuracy. To accurately detect a drivers’ face, we use the Haar features (Fig. 3).

![Haar-like features](image)

Fig. 3: Haar-like features

As mentioned in previous section, we compute features using Integral Image \( I(x, y) \) as narrated in Eq. (2).

\[
I(x, y) = \sum_{x^* < x, y^* < y} i(x^*, y^*) \quad (2)
\]

where \( i(x, y) \) is untreated image. Drivers’ face detection, using Haar-like features with the AdaBoost algorithm is achieved in real-time. The detected drivers’ faces are used as input to a robust drivers’ face recognition module to examine the identity of the questioned face.

IV. VEHICLE DRIVER’S FACE RECOGNITION

A. Face Recognition Theory

The face recognition module decides the possibility of the match of each face in a database to the input face image [18]. The recognition algorithm yields one of the following three results: (a) an irrelevant object, which is surely not a human face, (b) relevant object, which is a human face, but not found in the stored drivers’ face database, and (c) actual object, which is a human face found and stored in the drivers’ face database. To recognize the drivers’ face in real-time, we employ the commonly used Eigenfaces approach [15]. Our earlier study [2, 18] proved Eigenfaces based face recognition algorithm to be more robust in pose classification. Fig. 4 depicts the 32 Eigenfaces calculated from set of 1500 drivers’ facial images (i.e., \( M = 32 \) and \( N = 1500 \)).

![A set of M = 32 Eigenfaces](image)

Fig. 4: A set of \( M = 32 \) Eigenfaces

B. Face Recognition Mechanism

Drivers’ face recognition is applied and achieved in following simple steps.

1) Decompose every facial image(s) into trivial sets, which may be referred as a featured face images.
2) Center all featured face images, which may be referred as principal components or more specifically the Eigenfaces of initial training set.
3) Project/expose the centered images obtained in above step into a large Face Space (FS). This is performed by multiplication with Eigenface basis.
4) Compute the Euclidean Distances (ED) of the test image and the projected/exposed centered images.
5) For correct face classification, test image(s) has a minimum Euclidean distance (computed in 4th step of Section IV (B)) with the stored face image from the specific drivers’ facial image.

After the drivers’ facial image(s) is/are projected on to the large FS, there will be three outcomes as described in Section IV (A). On correct recognition of drivers’ face, a driver is concluded as authentic and person with vehicle is allowed to enter/exit in/from parking-lot. We find that the Eigenfaces based face recognition algorithm learns and recognizes new drivers’ face images in an unsupervised manner. This characteristics of the Eigenfaces based face recognition algorithm helps to achieve the face recognition in real-time.
V. SIMULATION RESULTS

We perform comprehensive experiments on SYS-7047GR-TRF machine with on board 128 GB of RAM. To perform the evaluations we use the driver face dataset that consists of 1500 faces of size 90x102 pixels.

The proposed framework comprises three major modules: vehicle detection, driver face detection, and face recognition module. To understand the performance of developed framework, it is necessary to examine the results of each module independently. In Sections below, we briefly explain and evaluate each module of the developed framework.

A. Camera Installation

Our objective is to spot vehicles that appear in the transmitted vehicle images. We install a camera on the side of the parking entrance. As the vehicle arrives in the parking area, the driver has to pull down the side mirror to display the face. Particularly, we consider and treat the images captured from various distances. The camera installation provides us the flexibility that from how far we can detect and recognize the faces. Fig. 5 shows the installed camera.

![Camera installation](image)

Fig. 5: Camera installation

B. Vehicle Detection Module

Fig. 6 shows few real life captured images with detected vehicles by our employed vehicle detection algorithm and the practicability of our developed framework according to real world situations. Vehicle detection is successfully achieved on images containing numerous vehicles with variations illuminations and vehicle pose deviation. The vehicle images shown in Fig. 6 are captured from various distances from installed camera near parking entrance. Top right image in Fig. 6 represents the trivial case of vehicles detection, while the top left and bottom two images of Fig. 6 represent the challenging situation with many other objects in the image. Particularly, the top left and bottom right images in Fig. 6 are captured from a considerably large distance, where most of the vehicles are detected. As the vehicle is detected, a rectangle is drawn around the vehicle. Occluded vehicles or extremely small sized, such as 10x10 pixels are not detected by vehicle detection module and are not considered for next steps.

![Sample images illustrating vehicle detection](image)

Fig. 6: Sample images illustrating vehicle detection

C. Face Detection Module

After the vehicles are detected in the input image, in the next step we locate driver face. The face detection phase determines the possible location of the driver’s face within rectangle enclosing the detected vehicle. We test 1278 images with different number of vehicles and drivers faces in various pose and achieved an average accuracy of 97.03%. Fig. 7 shows driver’s face detection. It is evident that these images have different size of vehicles, face, and variation in face pose. However, we observe that face detection module does not process enormously blurry and occluded face images.

![Captured outdoor images illustrating drivers’ face detection (white rectangle) on images clicked from various distances](image)

Fig. 7: Captured outdoor images illustrating drivers’ face detection (white rectangle) on images clicked from various distances

D. Driver Face Recognition Module

For face recognition experiments the stored drivers’ face database and is arbitrarily subdivided into two subgroups: (a) the gallery set, denoted by \((Z)\) and (b) probe set, denoted by \((Q)\). For face recognition, we simulate a very challenging task where there is only a single frontal image in the gallery and four different probe images with variation in pose. The four different pose are: +45\(^\circ\), +35\(^\circ\), 0\(^\circ\) (frontal), and -35\(^\circ\) (see Fig. 8). The face recognition algorithm is initially trained with the gallery and then face recognition is performed on probe set to calculate the Euclidian distance.
During simulations, we also used the PIE database [16] for initial testing and benchmarking purposes for face detection and recognition module. However, the actual experiments are performed on driver face database of 1500 subjects designed and developed in our lab. Fig. 9 shows the database of various drivers’ faces of both genders. Each driver has one image stored in the database.

![Fig. 8: Sample probe driver face images illustrating four different poses (a) +45° (b) +35° (c) 0°(frontal), and (d) -35°](image)

![Fig. 9: Online database of various drivers. This figure shows 140 faces](image)

Vehicle detection, driver’s face detection, and face recognition algorithms have been implemented in C language. Table 1 depicts the execution time of complete framework along with each module. Experiments are performed on 1278 diverse images captured by installing a camera as shown in Fig. 5.

**Table 1: Complete System Execution Time**

<table>
<thead>
<tr>
<th>Vehicle Detection (s)</th>
<th>Driver Face Detection (s)</th>
<th>Driver Face Recognition (s)</th>
<th>System (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1211</td>
<td>0.1137</td>
<td>0.2845</td>
<td>0.5193</td>
</tr>
</tbody>
</table>

E. Extra Investigation of Face Recognition Algorithm

We execute numerous experimentations to confirm that the drivers’ face recognition algorithm is precise, accurate, and robust. Our earlier work [11] indicated that face recognition accuracy rapidly increases when number of samples in gallery is increased. For this specific work, the major job of the framework is to investigate the performance of face recognition module when only one sample is present in gallery. During face recognition experiments a major hurdle is to classify the facial images with pose variation. In our earlier study [2, 20], we find Eigenfaces based approach to be more accurate in pose classification. Therefore, to develop a robust framework, we analyze image resolution with various pose. Figure 10 shows the recognition experiments on face image size of 30x30, 20x20, and 10x10 pixels. A low resolution essentially implies that image has been captured from a long distance and vice versa. We extensively test the developed system at parking entrances/exits of Fargo, Moorhead, and NDSU, USA by collecting the face images of employees, workers, and students and summarize the results in Fig. 10.

Our key observations are:

- The employed face recognition algorithm has 100% classification accuracy for a face size of 30x30, 20x20, and 10x10 pixels for pose 3 (frontal).
- Face recognition algorithm has 100% classification accuracy for all the four poses for face size of 30x30 pixels and above.
- For pose 1 (+45°), the face recognition algorithm has 80% classification accuracy for face size of 20x20 and 10x10 pixels.
- For face size of 20x20 pixels, pose 2 (+35°) and pose 4 (-35°), the face recognition algorithm has same (60%) classification accuracy. This finding is particularly important, as state-of-the-art face recognition systems [17] struggle in classifying same angle and opposite face pose orientations. Moreover, we conclude that the human face is not a distinctive entity. Various factors, for example, lighting, makeup, tattoos on face, scars, face or sun creams, occlusions (eyeglasses or scarf), moles, and low-resolution can drastically change the appearance of face(s) [19].
- For face size of 20x20 and 10x10 pixels, the face recognition algorithm has 60% classification accuracy for pose 2 (+35°).
- For extremely small face size of 10x10 pixels, pose 2 (+35°) and pose 4 (-35°), face recognition accuracy is not same. In this scenario pose 2 (+35°) has higher accuracy than its counterpart.
- For face size of 30x30 pixels and below, only frontal facial image should be captured and identified.

![Fig. 10: Recognition accuracy of face image size of: 30x30, 20x20, and 10x10 pixels](image)
F. Discussion on Developed Framework

We believe that the developed automatic vehicle/driver face detection and drivers’ face recognition framework can practically be installed in any parking area and has the following advantages.

- The developed framework is scalable, meaning that state-of-the-art object (vehicle or face) detection and face recognition algorithms can be added, removed, and modified to improve or investigate the accuracy of the framework.
- The proposed framework is particularly essential for scenarios where security officers manually check the identity of the vehicles at most parking entrances.
- The developed framework can be used to prevent vehicles robberies by keeping an accurate record of the entered/exit vehicles with driving personnel.
- System can be used to keep track of the entrance and exit timings of any vehicle and employees.
- The detailed image resolution and face pose analysis of driver face recognition will essentially enhance the security of staff, building, or any other physical location.

VI. CONCLUSIONS AND FUTURE WORK

We developed a complete working system for enhancing vehicle security applications. The developed system has three separate steps: vehicle detection, driver face detection, and driver face identification. On successful identification of driver’s face, vehicle is allowed to enter/exit in parking. The application of this system will ensure only authentic vehicles are allowed and parked in any public car parking area. For face identification, we simulated a very challenging task, when we had only single gallery face image, which was used for training. Four dissimilar driver pose facial images ranging from frontal to +45° were used as a test for the drivers’ face recognition element. Simulation results show very high detection and identification results against variation in facial pose and low resolution images. The results demonstrate the feasibility of developed system to be deployed in any vehicle parking area, such as airports. However, we detect vehicles from frontal and side view and therefore this is a restriction on the input image to the system. Face detection and recognition was performed on variation in poses, such as +45°, +35°, ±0°, and ±35°.

In future, further research could be carried out for detection and identification of drivers face from tinted vehicles arriving/exiting at/from parking-lots. Moreover, research could be oriented towards enhancing degraded images captured in strong sunlight to develop a robust vehicle classifier. Furthermore, we aim to develop a parallel version of vehicle detection, face detection, and face recognition algorithms to increase the performance gain.

REFERENCES