International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 25 Issue 5 – APRIL 2018 (SPECIAL ISSUE).

NEW GENERATION SECURED PARKING SYSTEM FOR SMART CITIES USING IMAGE PROCESSING DATA

Vignesh Srinivasan[#1],Senthil.M(Sr.G)*2, Shabari Girish N R*3, Naveen Chandra*4 and Anirudh Ganesh*5

**Department of Computer Science & Engineering, SRM INSTITUTE OF SCIENCE AND TECH*
vigneshs.301@gmail.com, senthil.m@rmp.srmuniv.ac.in

Abstract— The proposed framework is divided in to four steps: (1) vehicle detection at the parking entry junction, (2) driver's face detection (3) Detection of license plate recognition, and (4) identification of driver's face from the huge database of stored facial images and license plate. On successful identification of authorized person along with license plate, vehicle is allowed to enter in the parking-lot .The adaptive boosting algorithm is used for vehicle and face detection, while Eigen faces based approach is employed for face recognition Automatic Number Plate Recognition (ANPR) uses image processing technology to identify (license) plate of the vehicle. The objective is to design an efficient automatic authorized vehicle identification system by using the vehicle number plate. The system is implemented on the entrance for security control of a car-parking zones or area around top government offices The developed system first detects the vehicle and then captures the vehicle image. Vehicle number plate region is extracted using the image segmentation in an image. Optical character recognition (OCR) technique is used for the character recognition. The resulting data is then used to compare with the records on a database so as to come up with the specific information like the vehicle's owner, place of registration, address, etc. This is implemented and simulated in python raspberry pi hardware the system continuous to detect the empty parking areas available by the near by environment to sense the availability of the car parking areas with android app to sense the parking availability a server system will created mysql database

Keywords—ANPR-Automatic number plate recognition,Eigen faces, Face Recognition, OCR- Optical character recognition

I. INTRODUCTION

OBJECT detection is a process of locating all regions of interest in an image, such as face, player, or vehicle. Object detection is widely used in a variety of applications, for example security, access control to building, and identification systems. During the past decade, object detection and recognition algorithms have been widely researched and improved in terms of accuracy, performance, and speed. Typically, object detection and recognition algorithms require high performance computations. Increased computation performance has rendered a phenomenal growth in the computational capabilities of microprocessors. However, this increase is hardly able to catch up with the rapid advancements in image and video technology, including camera resolution, and in the computational requirements in real time applications .The license plate is detected by using object detection algorithms that has been previously developed. The Ada boost algorithm is one of the fastest ways to detect the object in the image. Our Contribution in object detection and face recognition are:

- The license plate image is captured by a camera kept at that level. The image is converted into grey-scale image
- Gaussian distortion is applied in that image to eliminate all the high details in the image since we are going to use sobel operator.
- The sobel operator gives the edge from which we can contrast the text in license plate with the image .OCR is used to detect the text in the license plate using deep learning algorithms like adaboost.
- After the face has been detected the facial features is located.
- By using 16x16 pixel search we detect the eyes and by using biometric algorithm, the other facial objects like nostrils, ears, lips etc., are detected.

The rest of the paper is organized as follow. In section II we explain the algorithms that are used in the license plate and face recognition.

II. OBJECT DETECTION AND RECOGNITION ALGORITHMS

A. *License plate and Driver face detection*

To detect the license plate and face we use the most popular and efficient Viola and Jones approach based on Adaptive Boosting (AdaBoost). This approach is mainly developed for face detection. We detect the license plate and Face recognition using Adaboost because of its ability to run in real time. The Adaboost algorithm takes the correct output from previously failed output, so that the algorithm finds the best solution in the most efficient way. This approach is innovative because of boosting and deep learning.

Fig.1: Proposed system (image captured at the parking entrance) and converted to grey scale image.

Boosting:

Boosting is a method to find accurate hypothesis by combining many weak hypothesis, each with reasonable accuracy. We develop a robust vehicle classifier by supervised AdaBoost learning. This AdaBoost learning uses the failed or weak results to bring the best and accurate results to us by eliminating all the existing weak results. The training data is selected from our own database to speed up the computation the color image converted into grey scale shown in Fig.1. An image captured at the parking lot may contain many irrelevant objects. So the main purpose of detection is to eliminate redundant areas in the image and retain only the necessary image. Detection provides the exact position of the car in the parking lot. We train the AdaBoost classifier in 28 stages having 1100 weak classifiers. The final classifier is linear combination of weak classifiers as

1. Give example images (a_1, b_1),.... (a_n, b) where $b_i = 0{,}1$ for negative & positive examples respectively.
2. for b_i=0,1, where *m* and are the number of negatives & positives respectively.
3. For t = 1; t < T; T++ do
4. Normalize the weights,
$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,i}}$$
to make w_t as a probability distribution.
5. For each feature *j*, train a classifier $h_j$which is restrictedto using a single feature. The error is evaluated with respect to
$$w_t, \epsilon_j = \sum_i w_i \,|\, h_i(a_i) - b_i|.$$
6. Choose the classifier, h_t, with the lowest error ϵ_t.
7. Update the weights:
$$w_{t+1,i} = w_{t,i}\beta_t^{1-\epsilon i}$$
Where $e_t = 0$ if example is classified Correctly, $e_t = 1$otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon t}$.
8. End
9. The final storage classifier is :

$$h(a) = \begin{cases} 1 \\ 0 \end{cases} \sum_{t=1}^{T} \alpha_t h_t(a) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t$$

Deep learning:

Deep learning algorithm is class of machine learning algorithm. We implement deep learning to detect the face with much accuracy and efficiency. This algorithm uses a cascade of multiple layers of nonlinear processing units for feature extraction. Each successive layer uses the output from the previous layers as input. To detect a face efficiently we use the following technique. From the image, the face cannot be found all of a sudden. Object detection is done so that the object (face) is found along with the nostrils, eyes, mouth, and we can conclude that object is a face. Moreover in a face we pick some specific part as the center say eyes. All the face parts are directed towards the eye so that the searching is faster. When searching the pixels, we input the data of how mouth, nostrils, looks like and this makes the searching quicker.

B. License plate Detection

Once the license plate in the car is detected, sobel operator is used to detect the edge of the numbers in the plate and OCR is used to recognize the numbers by converting it into text. Sobel operator does the operation of finding the edge by using a convolution operator between two matrices of which one is the matrix which is searched. The operator uses two 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives – one for horizontal changes, and one for vertical. If we define A as the source image, and Gx and Gy are two images which at each point contain the horizontal and vertical derivative approximations respectively, the computations are as follows:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Where * here denotes the 2-dimensional signal processing convolution operation. The x-coordinate is defined here as increasing in the "right"-direction, and the y-coordinate is defined as increasing in the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using: $\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$

Using this information, we can also calculate the gradient's direction:

$$\mathbf{\Theta} = \operatorname{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

where, for example, Θ is 0 for a vertical edge which is lighter on the right side.

C. Face Recognition

Face recognition determines the match-like hood of each face to a template element from a database. We employ the Eigen face approach that calculates an orthogonal set of M Eigen faces for a given training set of N faces, where M<<N. To recognize a face, first of all face images are centered and decomposed into small sets of featured images, that are actually the principle components of initial training

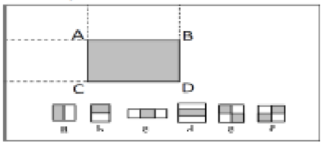


Fig.2: Computation of integral image amd Haar-like features

Later, all centered images are projected into face space by multiplying in Eigenfaces basis's. Euclidean distance between the projected test image and the projection of all cantered training images is calculated. Test image is supposed to have minimum distance with corresponding image in the training database. In Eigenfaces based algorithm as the image is projected on to the face space, there are three possible options.

1. If input image is near face space and near face class, then the individual is correctly recognized.
2. If input image is near face space, but not near known face class, then it is an unknown person.
3. If input image is distant from face space and near/distant from face class, then it is not a face image.

III. Parallel Architechture

Eigen faces based approach is one of the most successful feature based algorithm for efficient face recognition. We employ parallel processing on Eigen faces based face recognition algorithm. In the training phase, algorithm divides training images pool and assign each pool randomly to a group of processors in par pool(N), as described in Algorithm . In each par pool(N), we introduce data parallelism methodology in the Eigen face based algorithm for feature extraction of the training image. Finally, we concatenate the feature vector of each training image to form a feature matrix. In the testing phase, each test image is assigned to one parpool for feature extraction, recognition and matching.

IV. Simulation Results

We perform detailed experiments on the processor at the heart of the Raspberry Pi system is a Broadcom BCM2837 system-on-chip (SoC) multimedia processor. This means that the vast majority of the system's components, including its central and graphics processing units along with the audio and communications hardware, are built onto that single component hidden beneath the 256 MB memory chip at the centre of the board

A. License Plate Detection Module

The sole purpose of vehicle detection is to locate the position of the vehicle and confine the processing area for subsequent face detection and recognition steps. Before using our vehicle detector at the parking entrance/exit, we test the feasibility of our developed vehicle detector on real life captured images as shown in Fig. 3(a). Clearly, vehicle detection is achieved under varying illuminations. Vehicle detector successfully detects vehicles under large deviation from frontal, side, and back view.

B. Face Detection Module

Once the vehicles are detected in the input image, in next step, we proceed to locate driver face. The face detection module processes detected vehicles as its input and locates the driver face position. Our developed face detector successfully detects face pose from frontal up to ±450. Fig. 3(b) shows driver face detection result. We can see that each image having different face size, position, and pose are successfully detected. Face detection module does not process extremely blurred and occluded driver face images. Object detection time.

C. Driver Face Recognition Module

Face recognition aims at finding a person's identity from a database of known subjects. To start with driver face recognition experiments the database is partitioned into two subsets: the gallery (training) set G and probe (test) set P. For face recognition, we simulate a very challenging task such that given only one gallery image, which is frontal mug shot and four different probe images with different pose (Fig. 6). The four different poses are: +450, +350, 00 (frontal), and -350. These four poses are chosen because a driver face angle can vary under these four ranges while sitting on the driving seat. Face recognition algorithm is first trained with G images and the resultant face recognizer is applied to T images to calculate the Euclidian distances. To develop a robust and practical framework, we develop the database of 10,000 subjects in our lab


.

Fig.3: (a) License plate detection 3(b) Face recognition

Table.1 License plate and face detection time

Object detection time(secs)		**Complete Object detection time(secs)**
license plate	***Driver face***	***0.234887***
0.121104	***0.11738***	

D. Parallel Analysis of Face Recognition Module

Our primary aim of parallel implementation of the face recognition module is to achieve a significant reduction in execution time and an improved speed up over serial implementation on large datasets. As the driver displays the face, size of the detected face is found to be 80x92 pixels.

1) Scalability Analysis

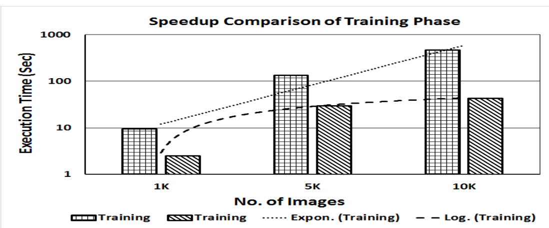
For a parallel algorithm to be scalable, with the increase in number of resources, such as the processors and the workload, the performance in terms of execution time and resources' utilization must be consistent and should not significantly degrade. We evaluate the scalability by analyzing the effects of increase in facial images and processors on the time consumption. Fig. 7(a) shows the scalability results by varying facial images and number of processors to observe the recognition time. The results show that by increasing the number of facial images five and ten times results in sudden increase in the processing time. However, substantial decrease in time consumption were observed by increasing the number of processors. On average, by increasing the number of facial images five times increases the time consumption by approximately 35.07% whereas increasing one processor results in an average decrease of 15.27% for the identification task.

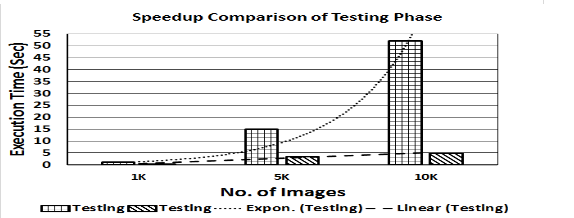
(a) $+45^0$ (b) $+35^0$ (c) 0^0(frontal) (d) -35^0

Fig. 4: Sample probe images illustrating four different poses of drive

2) Speedup Analysis

We systematically vary the training images from 1, 000 to 10,000 face images. Fig. 7(b) depicts the speed up comparison of the training phase for parallel face recognition with the serial execution. Clearly, for 1000, 5000, and 10,000 (which we refer 1K, 5K, and 10K) training facial images, the speedup is boosted to 3.55X, 4.75X, and 11.95X, respectively. The speedup comparison for testing phase is shown in Fig. 7(c). As per our framework scenario, we fed four different pose images to the face recognition module. For 1K, 5K, and 10K facial images with four different pose images as probe, the speedup is 3.85X, 5.05X, and 12.25X, respectively. Training and test phase speedups are particularly important for the cases where multiple faces with different pose are detected. Therefore, developed framework can result into significant high performance gain.





V. Conclusions and Future Work

We presented a framework of object detection and recognition for a secure vehicle parking. The proposed framework has three major steps: vehicle detection, driver face detection, and driver face recognition through parallel applied architecture. Simulation results show feasibility of developed framework to be deployed in real life. Given an appropriate training face data set, our developed framework can be a best fit for applications, such as real-time surveillance and video analytics. In future, we aim to develop the complete parallel version. Moreover, we are interested in the energy consumption of the developed system. Furthermore, research could be carried out to develop a robust object detector and recognizer from tinted vehicles.

References

1] Z. Mahmood, T. Ali, S. Khattak, L. Hassan, and S. U. Khan, "Automatic player detection and identification for sports entertainment applications," Pattern Analysis and Applications, 2015, DOI:10.1007/s10044-014-0416-4.

[2] Z. Mahmood, T. Ali, and S. Khattak, "Automatic player detection and recognition in images using AdaBoost," 9th International Bhurban Conference on Applied Sciences & Technology (IBCAST), Islamabad,2012, pp. 64-69.

[3] P. Viola and M. Jones, "Robust real-time face detection," Int. J. Comput.Vis, Vol. 57, 2004, pp. 137–154.

[4] C. A. A. Chandio, K. Bilal, N. Tziritas, Z. Yu, Q. Jiang, S. U. Khan, and C. Z. Xu, "A comparative study on resource allocation and energy efficient job scheduling strategies in large-scale parallel computing systems," Cluster Computing, Vol. 17, No. 4, 2014, pp. 1349-1367.

[5] L. Wang, D. Chen, R. Ranjan, S. U. Khan, J. Kolodziej, and J. Wang, "Parallel processing of massive EEG data with MapReduce," 18th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Singapore, Dec 2012, pp. 164-171.

[6] D. Chen, X. Li, L. Wang, S. U. Khan, J. Wang, K. Zeng, and C. Cai, "Fast and scalable multi-way analysis of neural data," IEEE Transactions on Computers, Vol. 63, No. 3, 2015, pp. 707–719.

[7] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, Vol. 3, No. 1, 1991, pp: 71-86.

[8] Z. Mahmood, T. Ali, S. Khattak, and S. U. Khan "A comparative study of baseline algorithms of face recognition," 12th International Conference on Frontier of Information Technology (FIT), Islamabad