Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 38(1), 2011, Pages 62–71
ISSN: 1223-6934

Automatic Number Plate Recognition System

Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, and Ahmed M. Abdelsadek

Abstract. Automatic recognition of car license plate number became a very important in our daily life because of the unlimited increase of cars and transportation systems which make it impossible to be fully managed and monitored by humans, examples are so many like traffic monitoring, tracking stolen cars, managing parking toll, red-light violation enforcement, border and customs checkpoints. Yet it's a very challenging problem, due to the diversity of plate formats, different scales, rotations and non-uniform illumination conditions during image acquisition. This paper mainly introduces an Automatic Number Plate Recognition System (ANPR) using Morphological operations, Histogram manipulation and Edge detection Techniques for plate localization and characters segmentation. Artificial Neural Networks are used for character classification and recognition.

2010 Mathematics Subject Classification. Primary 68T10; Secondary 68T45.
Key words and phrases. license plate recognition, plate region extraction, segmentation, neural networks, optical character recognition, Hough transform, ANPR.

1. Introduction

The Automatic number plate recognition (ANPR) is a mass surveillance method that uses optical character recognition on images to read the license plates on vehicles. They can use existing closed-circuit television or road-rule enforcement cameras, or ones specifically designed for the task. They are used by various police forces and as a method of electronic toll collection on pay-per-use roads and monitoring traffic activity, such as red light adherence in an intersection.

ANPR can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver. Systems commonly use infrared lighting to allow the camera to take the picture at any time of the day. A powerful flash is included in at least one version of the intersection-monitoring cameras, serving both to illuminate the picture and to make the offender aware of his or her mistake. ANPR technology tends to be region-specific, owing to plate variation from place to place.

The objective of the paper is to successfully locate standard Egyptian number plate, segment characters and recognize them given a car image. The system must deal with different angles, distances, scales, resolutions and illumination conditions. The rest of the paper proceeds as follows. Section 2 presents the problem statement. Section 3 presents the proposed solution. Finally, Section 4 concludes the paper.

Received October 29, 2010. Revision received January 11, 2011.

2. Problem Statement

The main focus in this research project is to experiment deeply with, and find alternative solutions to the image segmentation and character recognition problems within the License Plate Recognition framework. Three main stages are identified in such applications.

First, it is necessary to locate and extract the license plate region from a larger scene image. Second, having a license plate region to work with, the alphanumeric characters in the plate need to be extracted from the background. Third, deliver them to an OCR system for recognition. In order to identify a vehicle by reading its license plate successfully, it is obviously necessary to locate the plate in the scene image provided by some acquisition system (e.g. video or still camera).

Locating the region of interest helps in dramatically reducing both the computational expense and algorithm complexity. For example, a currently common 1024x768 resolution image contains a total of 786,432 pixels, while the region of interest (in this case a license plate) may account for only 10% of the image area. Also, the input to the following segmentation and recognition stages is simplified, resulting in easier algorithm design and shorter computation times.

The paper mainly work with the standard Egyptian license plates but the techniques, algorithms and parameters that is be used can be adjusted easily for any similar number plates even with other alpha-numeric set.

3. Proposed Solution

In this section The process of automatic number plate recognition consists of four main stages:
(1) Preprocessing
(2) License plate localization
(3) Character segmentation
(4) Character recognition

3.1. Preprocessing. As mentioned before, the system of automatic number plate recognition faces many challenges. So, this step is essential to enhance the input image and making it more suitable for the next processing steps. The first step done in the preprocessing is to apply minimum filter to the image in order to enhance the dark values in the image by increasing their area. This is mainly done to make the characters and the plate edges bold, and to remove the effect of the light diagonal strips that appear in the characters and edges of the Egyptian license plates (see Fig 1). This process is followed by increasing saturation of the image to increase the separation between colors. Then the image is converted to grayscale (taking the luminance component of NTSC) [8]. Then increasing the image contrast to separate the background from highlights (see Fig 1) [8].

3.2. License Plate Localization. In this stage, the location of the license plate is identified and the output of this stage will be a sub-image that contains only the license plate. This is done in two main steps.

3.2.1. *Locating a large bounding rectangle over the license plate.* In this step a rectangle that contains the license plate is located (this rectangle may also has some extra parts from the four sides), and this rectangle is the input to the next step for further processing (removing the extra parts, character segmentation then recognition) [8].

FIGURE 1. (a) original image (b) after applying minimum filter (Note that the diagonal strips passing through characters and edges of the plate disappeared, and the characters became bold and their area are increased, This is important in order to have many strong edges around the characters) (c) after increasing saturation (d) after converting to grayscale

First, Sobel vertical edge detection is applied to the image. Then a threshold of 36 (This value is determined using trial and error) is applied, Such that every edge with magnitude less than 36 is considered false edge and is set to 0. Then a vertical projection (projecting on the Y-axis) of the edge detected image is taken and smoothed using an average filter with width equals 9. It's obvious that the characters of the plate along with the plate's vertical edges will have very strong vertical edges (see Fig 2). Moreover, these edges will sum up horizontally in the vertical projection and a strong peak will appear in the rows of the plate (These row will be called band). So, the approach is to take some number of peaks in the vertical projection and processing each of them individually in the next steps and when a successful band is found, the processing of the following bands is canceled. The reason behind taking more than one peak is that the image may contain objects (logos, road advertisement, etc..) that produce many vertical edges also these "false" edges may be centered in the same area so they will form a peak that may be stronger than the peak of the plate itself [8].

For each band, we take a sub-image referenced by this band and all subsequent processing will be applied on this sub-image. Now the problem is to cut the band image from the left and right to get a bounding rectangle over the license plate (Again, this rectangle doesn't have to be tight on the plate). For this sake, a vertical Sobel edge detection is applied again, but the height is larger than the width of the filter, this is to decrease the effect of false edges and noise, experimentally, the best size is 6x3 filter (see Fig 3) [8]. Again a threshold of 30 is applied for the same reason as before. Now, a horizontal projection of the edge detected band image is taken (projection on the X-axis) and smoothed using an average filter of large size this time, since

FIGURE 2. (a) Original Image, (b) vertical projection of image (c), (c) Sobel vertical edge detection after applying threshold, (d) the band that corresponds to the strongest peak (which is also in this case the correct plate band)

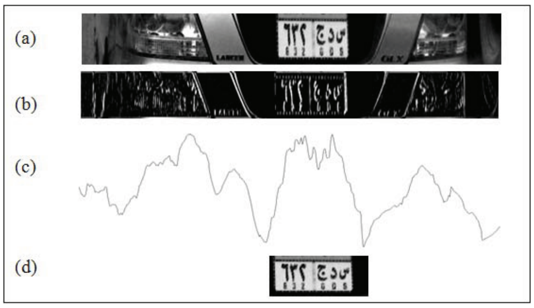

FIGURE 3. (a) band image; (b) after Sobel vertical edge detection; (c) horizontal projection for image (b) (Note that there are three peaks that corresponds to the left and right lamp and the plate); (d) the plate that corresponds to the middle peak

there are gaps between the letters and the projection will have many peaks at the x coordinates where letters exist but it will drop down in the x coordinates of the gaps. So, smoothing it with average filter of large width will resolve this problem and many number of peaks will be converted to one wide peak that represents the range of the X-axis where the plate is located in that specific band we are working with. The width of the average filter is taken to be the height of the band. Relating the height of the band with the width of the average filter is very important since over-smoothing of the projection will merge the plate peak with the other main peaks in the band like the peak got from vehicle lamps for example (and it already explained why the width shouldn't be very small). Now, a predefined number of peaks (It's already explained why we take more than one candidate peak not just the strongest one) will be selected from the smoothed projection (see Fig 3) [8].

For each peak, a sub-image is taken according to the range of current peak. So, the bounding rectangle of the license plate is located. This is will be the input to the next step.

3.2.2. *Determining the exact location of the license plate.* Using the sub-image from the last step which contains the license plate with some extra parts (if any), the following processing is applied to this sub-image. The license plate may be skewed because of the angle of the camera while image acquisition process. And it is very important to de-skew the plate to its original orientation, thus making the plate

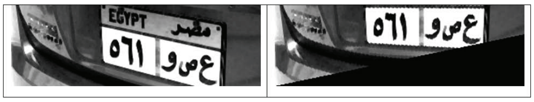

Figure 4. (a) the original image after de-skew; (b) after applying the de-skew (shear) operation using Hough transform, It is clear that the plate became axes aligned

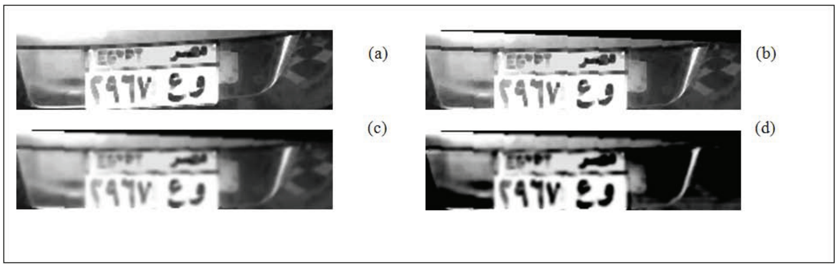

Figure 5. (a) is the an output image from step 3.2.1 where we got a bounding rectangle over the plate but has some extra parts from all directions; (b) After applying the de-skew operation; (c) after applying Gaussian smoothing; (d) after applying the morphological operation Image = Image - BottomHat(Image) (Note that the characters became more bold and they are enhanced and their area is increased)

aligned with the X and Y axes (The reason behind its importance will be clear below). So a Hough transform is applied to the horizontally edge detected image in order to find the shear parameters by which the image can be de-skewed to retrieve the standard orientation (see Fig 4) [8], [1][12]. After this operation we have a plate with its axes aligned with the X and Y axes. Then a Gaussian smoothing filter is applied to smooth the image and remove noise. Then a morphological operation consists of subtracting the bottom-hat of the image from the image itself is done using a structuring element of a horizontal line of length 150 [7]. This operation makes the characters of the plate bold and increases the characters area along with the effect of increasing contrast, and subsequently this will ease the process of segmentation and recognition afterward (see Fig 5).

All the above is considered a preprocessing for this step. Next, we aim at finding the exact band of the plate. In other words, the goal of this step is to cut the top and bottom extra parts of the previously cut rectangle (but this time the cut will be accurate because we have limited the area we are working with and moreover we de-skewed the plate). This is done using the same idea we used previously to get the plate band. It consists of applying Sobel vertical edge detection, then applying a threshold, then doing a vertical projection (projecting on the Y-axis), Then getting the strongest peak in this projection and cut the image accordingly using the range of this peak, thus cutting the exact plate band from the image and leaving the top and bottom extra parts (see Fig 6). This time just the strongest peak is taken since we already limited the possibility that false edges appear when we cut a rectangle around the plate and we are sure that the vertical edges produced by the plate's characters are summed up correctly in a limited number of rows due to the de-skew operation.

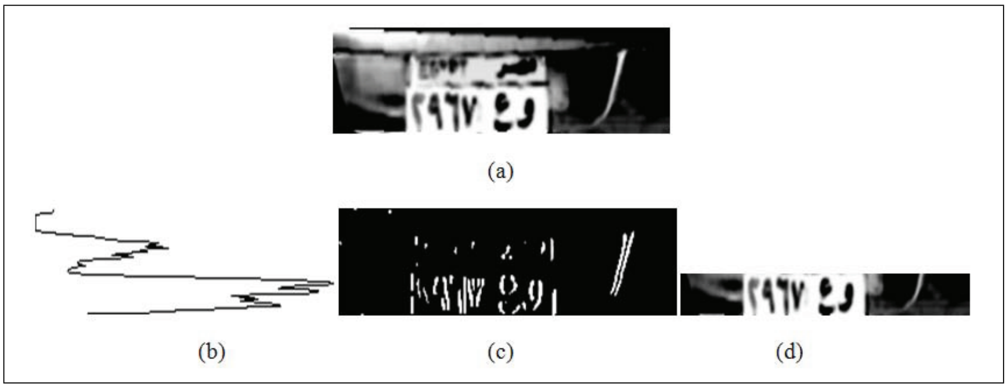

FIGURE 6. (a) an image after before the step of cutting top and bottom extra parts; (b) the vertical projection of image (c); (c) Sobel vertical edge detection for image (a); (d) the exact band of the plate after cutting the top and bottom extra parts

We got rid of the top and bottom extra parts. But we still have extra parts from left and right that have to be cut to end up with an exact rectangle around the plate. So, a stamp filter is applied to the sub-image we got from the previous step. This filter is just a blurring followed by a soft threshold operation. Now the white color will dominate the plate area (see Fig 7 and Fig 8). After this a horizontal projection is done then smoothed using average filter with width equals 40. Then we get the strongest peak from this projection. This peak corresponds to the plate range on the X-axis (see Fig 7 and Fig 8). So, a sub-image is cut using the peak range. In many cases when the color of the vehicle is bright, the previous operation is not sufficient to cut all the extra pieces from left and right. So, this is followed by getting Sobel horizontal edge detection, applying a threshold, then getting the horizontal projection, then smoothing this projection with average filter of size 40. Then we will get two points that will define range of the peak. The first point is the point with least x coordinate that has a value (from the smoothed projection) greater than or equal the average value. The second point is the point with maximum x coordinate that has a value greater than or equal to the average. We will cut the image again using these two points we got. And this is the final plate that the next processing stages will work on.

At the current moment we have a "candidate" final plate. The next processing stages are computationally expensive. Also using the fact that all the plates have a very similar (if not exact) values for some measures like aspect ratio, contrast, average brightness, average saturation in both the colored and grayscale plate images. We can begin to reject the plates based on the previous measures, such that, If we found that the current candidate plate for any measure has a very far value from the ranges of values for the true plates, It's simply rejected and the processing continues on the next candidate plate. But a false plate may pass these tests, and it will be rejected in subsequent stages. The next stage is to segment characters from the plate that passed all the measures tests.

3.3. Character Segmentation. This stage is meant for segmentation of the characters from the plate. The output of this stage is a set of monochrome images for

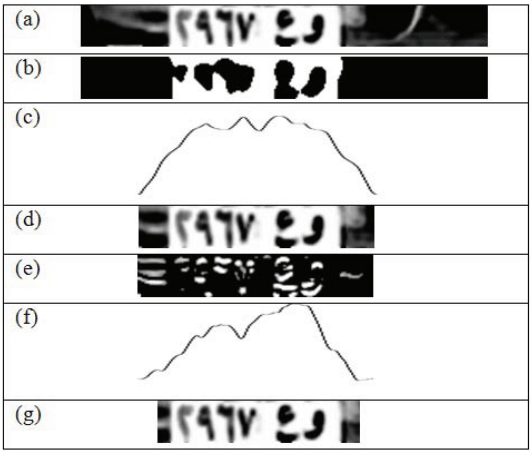

FIGURE 7. (a) an exact plate band; (b) after applying the stamp filter which just blurs the images then apply a threshold to convert it into binary image; (c) A horizontal projection of the binary image after the stamp filter; (d) the plate after cutting extra parts from left and right according to the peak got from the horizontal projection; (e) applying Sobel horizontal edge detection to cut the remaining extra parts from left and right; (f) horizontal projection; (g) cutting according to the peak of the horizontal projection (In this case the second step in cutting extra parts from left and right approximately has no effect since the plate is well cut from the previous step)

each candidate character in plate. The first step in this stage is to convert the plate image to a binary image. This is done using adaptive threshold with a window of size 11 (This is selected using trial and error). Then a process of noise removal is applied. This is done by getting the connected components from the binary image based on the 8-neighbourhood using flood fill. For every component, we decide if it's a noise or not based on the aspect ratio of the component and based on the number of pixels in that component. This is based on the fact that the characters of the plate have a certain range of aspect ratio and a certain range of number of pixels. After removing the noise components a maximum filter is applied to make the effect of thinning the characters to make sure that no two components are merged. This is followed by a horizontal projection, to detect the boundaries between the characters to be able to cut them individually. The peaks in this projection correspond to the gaps between the characters. So, we get all of these peaks and a rejection process is applied also, since a true plate has a fixed range of gaps between characters. So, any plate that has number of peaks that do not fit in that range, will be rejected. Also, there is a powerful rejection measure; it is the variance of the characters width (the variance of the spaces between peaks). After this the characters are cut according to the peaks of the previous projection. Then another set of measures are computed to reject the false characters that may still exist after the noise removal operation. These measures are aspect ratio, deviation from average height test, deviation from average contrast, deviation from average brightness, deviation from hue, deviation from average saturation. After rejecting false characters, if the number of characters is not located in a predefined range, then the plate is rejected. Otherwise, the processing is continued

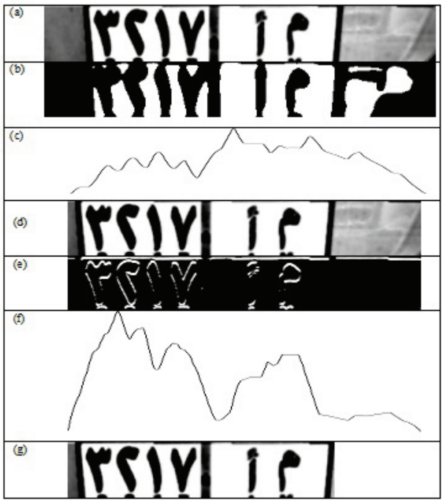

FIGURE 8. (a) an exact plate band; (b) after applying the stamp filter which just blurs the images then apply a threshold to convert it into binary image; (c) A horizontal projection of the binary image after the stamp filter; (d) the plate after cutting extra parts from left and right according to the peak got from the horizontal projection; (e) applying Sobel horizontal edge detection to cut the remaining extra parts from left and right; (f) horizontal projection; (g) cutting according to the peak of the horizontal projection (In this case the second step in cutting extra parts from left and right is crucial note the difference between the first and second steps in cutting extra parts from left and right in image (d) and (g))

and for every character a copy of its corresponding location in the grayscale is got. The gray level histogram is computed for the sub-image of each character, This gray level histogram will have a standard shape which is one peak at the dark values (this corresponds to the character's pixels) and another peak at the bright values (this corresponds to the background) and some small values between them. So, this gray level image is converted to binary using the following procedure. First, we find two peaks in the histogram then we find the minimum value in between, this will be the value of the threshold (thus, every pixel that has a gray level value less than the mentioned value, will be converted to black, every other value will be converted to white). This way for converting the grayscale image that contains only a character to binary one proved to be effective. At this point we have a set of binary images each contains one character and this is the output of this stage and the input to the next (see Fig 9).

3.4. Character recognition. The goal of this stage is to recognize and classify the binary images that contain characters received from the previous one. After this stage every character must have a label and an error factor, and this error factor if greater than a predefined value will be used to reject false characters accidently passed from the previous steps. For the sake of classification, some features must be collected from the characters. The feature we work with in this system is the chain code of the contour of the image after dividing it into four tracks then into four sectors [10].

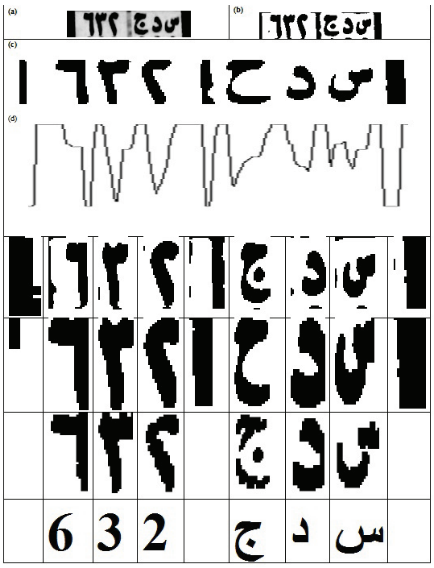

FIGURE 9. (a) Final located plate; (b) After applying adaptive threshold; (c) After removing components that has small area and wrong aspect ratio, this is done using flood fill; (d) Horizontal projection to separate the characters (Note that the gaps between characters create peaks in the projection); After this we split the characters according to the previous peaks (we can notice that dots of the characters are back since we just use the flood fill to remove the noise and to make the peaks clear then we cut the characters from the image before noise removal); Then another process of noise removal is done, it's consists of removing all components except the largest one so that we can test the candidate characters and reject the ones that have features too deviated from the characters features; Finally for candidate characters that passed the measures tests, we cut them again from the original gray scale image, Then a gray scale threshold is applied to convert it to binary image. This way we can keep the dots of the characters for a successful recognition.

Also we used a feed forward artificial neural network trained with back propagation with sigmoid activation function and the ANN is trained on the chain code feature of the optimal characters images [8][5]. The neural network has 4X4X8=128 input neuron, it also has 37 output neurons corresponds to the Arabic alpha-numeric set of characters except zero, it also ceil (37+128)/2)=83 hidden neurons.

So, for every character we get the chain code feature and do a feed forward on the trained FFNN (Feed Forward Neural Network) then the class the corresponds to the neuron with the maximum value will the predicted class of that character. If the error exceeds a predefined value then the character is considered a false one and rejected. The plate is known to have a fixed range of characters that may appear in it, so if the total number of passed characters does not match this range, then the plate is rejected. Otherwise, the license plate number is found.

4. Conclusion

The objective of this paper was to study and resolve algorithmic and mathematical aspects of the automatic number plate recognition systems, such as problematic of machine vision, pattern recognition, OCR and neural networks. The problematic has been divided into several chapters, according to a logical sequence of the individual recognition steps. Even though there is a strong succession of algorithms applied during the recognition process, chapters can be studied independently.

ANPR solution has been tested on static snapshots of vehicles, which has been divided into several sets according to difficultness. Sets of blurry and skewed snapshots give worse recognition rates than a set of snapshots which has been captured clearly. The objective of the tests was not to find a one hundred percent recognizable set of snapshots, but to test the invariance of the algorithms on random snapshots systematically classified to the sets according to their properties.

References

[1] B.D. Acosta, Experiments in image segmentation for automatic US license plate recognition, M.Sc. thesis, Department of Computer Science, Faculty of the Virginia Polytechnic Institute and State University (2004).

[2] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos and E. Kayafas, License Plate Recognition From Still Images and Video Sequences: A Survey, *IEEE Transactions on Intelligent Transportation Systems* **9** (2008), no. 3, 377–391.

[3] C.N.E. Anagnostopoulos, I.E. Anagnostopoulos, V. Loumos and E. Kayafas, A License Plate-Recognition Algorithm for Intelligent Transportation System Applications, *IEEE Transactions on Intelligent Transportation Systems* **7** (2006), no. 3, 377–392.

[4] S.-L. Chang; L.-S. Chen; Y.-C. Chung; S.-W. Chen, Automatic License Plate Recognition, *IEEE Transactions on Intelligent Transportation Systems* **5** (2004), no. 1, 42–53.

[5] S. Draghici, A neural network based artificial vision system for licence plate recognition, *International Journal of Neural Systems* **8** (1997), no. 1, 113–126.

[6] B. Enyedi, L. Konyha, C. Szombathy and K. Fazekas, Strategies for fast license plate number localization, *Proceedings of the 46th International Symposium Electronics in Marine, Elmar 2004*, Zadar, Croatia, June 16-18, 2004, IEEE Press (2004), 579–584 .

[7] J.-W. Hsieh, S.-H. Yu and S.-H. Yu, Morphology-based License Plate Detection from Complex Scenes, *IEEE Proceedings of the 16th International Conference on Pattern Recognition*, Qubec City, Canada, August 11-15, 2002, Vol. 3 (2002), 176-179.

[8] O. Martinsky, Algorithmic And Mathematical Principles Of Automatic Number Plate Recognition Systems, B.Sc. thesis, Department of Intelligent Systems, Faculty of Information Technology, Brno University of Technology (2007).

[9] J.R. Parker and P. Federl, An Approach To Licence Plate Recognition, University of Calgary 1996.

[10] H. Sarukhanyan, S. Alaverdyan, and G. Petrosyan, Automatic Number Plate Recognition System, *Proceedings of the 7th International Conference on Computer Science and Information Technologies*, Yerevan, Armenia, September 28-2 October, 2009, Electronic Copy of the CSIT 2009 Proceedings, 347-350.

[11] V. Shapiro, D. Dimov, S. Bonchev, V. Velichkov and G. Gluhchev, Adaptive License Plate Image Extraction, *Proceedings of the 5th international conference on Computer systems and technologies*, Rousse, Bulgaria, June 17-18, 2004 (K. Boyanov), ACM New York, NY, USA (2004), 1-7.

(Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, Ahmed M. Abdelsadek) Department of Computer Science, Cairo University, 5 Dr. Ahmed Zewail Street, Cairo, 12613, Egypt
E-mail address: a.badr.fci@gmail.com, wahab@writeme.com, amthabet@gmail.com, ahmed_sadek_1989@yahoo.com