Iraqi license plate recognition system using (YOLO) with SIFT and SURF Algorithm

Nada Hassan Jasem¹, Faisal Ghazi. Mohammed²

¹University of Baghdad, College of Science, Department of Computer Science, Baghdad, Iraq
²University of Baghdad, College of Science, Department of Remote Sensing and Geographic Information's, Baghdad, Iraq

¹Nadahassan57@yahoo.com, ²faisel@Scbaghdad.edu.iq

https://doi.org/10.26782/jmcms.2020.08.00048

Abstract

Automatic License Recognition (ALPR) has been considered significant in many applications in intelligent transport and monitoring systems. As in other tasks of the computer vision, deep learning methods (DL) were implemented recently in the ALPR context, with a focus on country-specific Iraqi councils, like German or Old and Northern. In this work, we proposed the DL-ALPR system from the beginning in the license plate detection phase of Iraqi plates according to the latest (YOLO) convolutional layers to detect single class. Utilizing a data set of Iraqi paintings collected by the researcher, and in the second stage, the detection plates are Recognition by extracting a set of license plate features using the SIFT and SURF algorithm, then using KNN to match the plates stored in the database to match them, the data is divided into two parts, part photos: 1300 pictures, And the second part, videos of the Iraqi vehicles in different environmental conditions, and the number is 35 videos. 1300 photos were divided 70% in the training phase and 30% in the testing phase and the results obtained in the testing phase were 99.2% for LP detection and 97.14% for recognition and the total accuracy of the system was 98.17%.

Keywords : Automatic License Recognition, deep learning methods, Iraqi plates, SIFT and SURF algorithm, training phase, testing phase.

I. Introduction

The recognition system of the license plates of the vehicles is a special field of interest in the area of the video surveillance for over 10 years. With the emergence of the advanced systems of the video vehicle detections for the applications of traffic regulation, the system of number plate identification can find many different places for fitting itself beyond the mere control of the access into a parking lot or a toll collection point. It may now be combined with video the systems of the vehicle detection that are typically installed in the places of interest for traffic surveillance, intersection regulation, and so on. The license plate properties differ from one country
to another. Such as numbering system, language used, colors, style (font), license plate sizes, and more researches are still required in this field (Marzuki., et al., 2019).

In this study, an innovative robust real-time LPR method has been suggested, consists of two parts: The first stage is the stage of identifying and detecting the license plate using convolutional neural networks based on “YOLO” algorithm (Redmon, Farhadi, 2017) and the second stage is the stage of excellence that is presented. The methods of matching the template are approved using the Scale-invariant Feature Transforms (SIFT) & Speed up Robust Features (SURF) algorithm (Mathew, 2016) through which the main features of the license plates are extracted and then matched with the LP stored in the database created using the oracle program. For matching, we use the K nearest neighbor search (Saharkiz., 2009). Besides necessary illustrations and numbers; moreover, procedures and algorithm. There are a small number of a few distinct algorithms for the object detection and they may be divided to 2 groups:

1. Algorithms that are based upon classification – which operate in 2 stages. In the first one, we select from the interest regions of the image. Then we perform the classification of these areas with the use of the CNNs. This solution may be quite slow due to the fact that we are required to run the prediction for each one of the chosen areas. The most common example of such type of the approaches is Region-based CNN (RCNNs) and their cousins Faster-RCNN and Fast-RCNN (Girshick, 2015).

2. The algorithms which are based upon the regression – rather than the selection of the interesting areas of the image, we predict the classes and the bounding boxes for entire image in a single run of algorithm. The most common one of the examples of such type of approaches is YOLO which is typically utilized for the real-time object detections (Geethapriya, et al., 2019).

Prior to going to the details of the YOLOs we must be aware what we will predict. The task is predicting a class of the object and bounding box which specifies the location of the object. Every one of bounding boxes may be characterized with the use of 4 descriptors (Marzuki., et al., 2019):

1. Width (bw)
2. Height (bh)
3. Center of a bounding box (bx,by)
4. C corresponds to an object class (e.g. traffic lights, car).

In addition to that, an additional predicted value \( p_c \) that is a likelihood that there’s an object inside bounding box, **object detection process is a single issue of regression, directly from the image pixels to the coordinates of the bounding box and class possibilities** as shown in the following figure (1) Image as input, passing it via an NN a normal CNN and obtain a bounding boxes vector and predictions of the classes in output (Redmon, Farhadi, 2017).
II. Related Work

In 2019, Ira Kusumadewi, et al, presented method for LP Number Recognition with the use of the Bounding Box and Template Matching Methods of the Indonesian vehicles. That includes three stages: the first stage pre-processing. Second stage Find the number-plate area using the bounding box method each character is then segmented separately. An LP is an image object which has several characters that typically include letters and numbers. The final step is the template matching method to identify each character of the plate with the correlation coefficient. According to the tests which were performed, the accuracy has reached 80%. In 2019, Jong Bae Kim, Suggested method for Automatic Vehicle License Plate Extraction Using Region-Based Convolutional Neural Networks and Morphological Operations. That allows detecting vehicles and discovering the vehicle license plate in real time. The proposed method takes advantage of the R-CNN method and morphological processes. To detect vehicles in the input image, a rectangular area containing the vehicle was detected using the R-CNN and SVM machines. The vehicle license plate is detected by a morphological process based on the allocation of the edge pixels of the vehicle area detected. Experimental results show that the vehicle detection rate is around 92% in real road environments, and that the vehicle license plate detection rate is about 83%. In 2019, Achraf Khazri, proposed Automatic License Plate Detection and Recognition system for Tunisian using deep learning with three steps: Detecting, Segmentation and Recognition, to detect (LP) use YOLO (You Only Look One) deep learning object detection architecture based on convolution neural networks. Predicting the surrounding squares using a convolutional grid and the likelihood of classification for these squares and image detection. Second step have Segmentation by projection method. The recognition phase is the last step based on multilayer perceptron (MLP). According to the tests which were performed, the accuracy has reached 98%. In 2019, Hendry, et al, proposed Automatic License Plate Recognition (ALPR) system for Taiwan. It consists of two basic phases: detection and identification of plates using the You Only Look Once (YOLO) panel - the deep learning framework. Detection method is a sliding window operation. Use the (AOLP) dataset containing 6-digit vehicle license plates. Each number on the license plate is detected by the sliding window, and then through one YOLO framework each window is discovered.

Fig. 1: Object in the bounding box.
The system achieves approximately 98.22% accuracy in detecting the license plate and 78% accuracy in identifying the license plate.

III. Architecture of the System

This system consists of two basic stages: the first stage, which is the stage where the license plate for Iraqi vehicles is decided, based on DL (deep learning) methods. The second stage is image recognition, where these methods, along with algorithms and diagrams, are described in depth for both stages. The general specification of (LP) recognition system shown as the following figure.

Fig. 2: Architecture of the system

IV. Proposed Methodology

This section includes a detailed description of the way of the detection of the plates from the images and utilization of the surf and sift in the recognition of the plates, which can be seen in Fig (2). The job has been divided to two basic parts and after that, we prepared approximately 1,300 images for the training of the CNN network based on (You only look once) algorithm for the identification of the position of number plate. Fundamentally, the YOLO completes a number of the algorithms at once which include orientation, localization, sizing and with the image processing, showing details in the figure below.

First Stage:

IV.i. CNN Construction

The LPRCNN includes nine convolutional layers and after 6-th max pool layer. Briefly, image following nine convolution and pooling layers, undergoes resizing to an SXS dimension grid, every one of the grids can predict the (B) bounding boxes (width, height, box centre x & y) and every one of the box probabilities of including a P (Object). Every one of the grids predicts C (which represents the entire number of the classes) as well. So, we classify one object and
take $B=5$ & $S=13$ then the final tensor’s dimension becomes $13\times13\times6$ as shown in the following.

![Fig. 3: The Final Tensor’s Dimension of Cell.](image)

In the case of wanting to categorize (1) objects and taking $B = 5$ and $S = 13$, this will give us $(13\times13 \times (1\times5+1)) = 1014$. All 9 convolutional layers illustrated in table (2), the input images undergo resizing into $416\times416$ with the 3 channels of the RGB. There are 16 filters of the feature, with a $3\times3$ size, applying the operation of the convolution process on every one of input image channels.

The moving stride was 1. The output feature maps’ size is $416\times416$ (input size / stride), and depth has been $48$ (3channels $\times$ 16 filters). After that, a $2\times2$ max pooling decreases size to $208\times208$, following 9 convolution layers and a single fully connected one, the output was a $7\times7\times30$ vector. The details of LPRCNN showing in the following:

![Fig. 4: LPRCNN Architecture Layers](image)

**IV.ii. Algorithm Operation**

Initially, an image will be taken and the algorithm is implemented. In this work, image is split as $13\times13$ grids matrices. The image may split to any number of the grids, based upon the image complexity. As soon as the image has been divided, every one of the grids’ objects is classified and localized. The confidence score’s
objectness of every one of the grids is obtained. In the case where there aren’t any suitable objects which are found in grid, after that, the objectness and the value of the bounding box of grid will be equal to 0 or in the case where there is an object in the grid then the objectness will be = 1 and the value of the bounding box is going to be the equivalent bounding values of obtained object. The prediction of the bounding box will be explained below. In addition to that, the Anchor boxes are utilized for increasing the precision of the object detection that is explained as well in detail below as a figure (5) (Redmon, Farhadi, 2017).

**Fig. 5: YOLO Working**

**IV.iii. The Predictions Vector**

YOLO is utilized to predict the precise bounding box from an image. By yolo image is divided to SxS grid of cells, for every one of the objects which are present on image, one grid cell will be “responsible” for its prediction. Which is a cell in which object centre falls in both object localization and image classification methods are applied for every one of the grids of the image and every one of the grids is given a label. After that, this algorithm performs the checking of every one of the grids, in a separate manner and marks the label that has an object in it and marks the bounding boxes as well (Redmon, Farhadi, 2017).

Every grid cell performs the prediction of the B bounding box (width, height, box centre x & y) and the likelihood of every box that contains an object P (Object), (x, y) coordinates denote the box’s centre, in relation to the grid cell location (in a case where the box’s centre doesn’t fall within the cell of the grid then this cell isn’t responsible for it).

Every one of the grids will predict C (which is the entire amount of classes) as well. The conditional probability values of the class P (Class (i) |Object) conditioned on grid which contains an object, only 1 group of class probabilities has been projected for each grid cell. In general, the network can predict an (SxSx (Bx5 + C)). Digit 5

*Copyright reserved © J. Mech. Cont.& Math. Sci.*

*Nada Hassan Jasem et al*
indicates that every one of the boxes has 1 object probability and 4 coordinates. Figure (6) shows that.

![Image: Fig. 6: object is detected with Bounding box and Class values of grid](image)

In the figure above for our work the dimension 13×13 comes from the size of grid cell that we have applied on our input image. In every grid cell, use 5 boxes vectors and each boxes vector is described with 6 parameters (numbers). These 6 parameters are showing as the following table:

| PC | Likelihood that there is object in that particular cell |
|----|-------------------------------------------------------|
| Bx, By | Position of centre of the object                     |
| Bh, Bw | The size of the bounding box                         |
| C1 | Labels for class                                      |

In this table (y) in the figure (6) denotes the existence of the object, bx, by, bh, bw represent the object bounding boxes in 2nd grid. And object in that grid is a plat so classes will be (0, 1, and 0).

If two grids or more include an identical object, in this case, the center point of the object has been found and grid that has that point will be considered. Therefore, for purpose of getting precise detection of objects two approaches can be utilized. Those are Non-Max Suppression as well as Intersection over Union. The IOU takes the predicted and the actual bounding box values and performs a calculation of $\text{IOU} = \frac{\text{Intersection Area}}{\text{Union Area}}$ (Redmon, Farhadi, 2017).

Formally the confidence defines as $\text{Pr (Object)} \times \text{IOU (pred, truth)}$. In the case where there aren’t any objects in that cell, the score of the confidence has to be equal.

*Copyright reserved © J. Mech. Cont.& Math. Sci.*

*Nada Hassan Jasem et al*
to 0. Otherwise the score of the confidence for equalling the IOU between projected box and ground truth.

Pred which represents the predicted bounding box from model.

Truth which represents the ground-truth bounding box.

**Fig. 7:** shows the ground-truth in addition to projected bounding box. (Redmon, Farhadi, 2017).

With the YOLO method we do not search for the interest areas on the image which may be including an object. Rather than that, the image is split to cells, usually it is a SxS grid. In this work one of the cells is responsible for the prediction of 5 bounding boxes (in case there’s over one object in that cell). Most of these boxes and cells do not have an object in them, which is why, it is required for the prediction of the pc. In the following step, the boxes with the low likelihood of the object are removed and the bounding boxes with maximum shared area in procedure are referred to as the non-max suppression **non-max suppression** (Redmon, Farhadi, 2017). That is showing in the figure below.

**Fig. 8:** removing boxes by non-max suppression

V. **Network Training in the Current Search Proposal**

Ution module. In case of LPRCNN, it is recommended that the algorithm be run for **500 epochs** for at least **6 days**. Approximately the execution speed for LPRCNN algorithm. In this context, a checkpoint file of the trained variables has been used after being trained for **54750** steps, error rate was $$\alpha = 2.4$$. The table (3.2) & (3.3) illustrated training step and Parameter of the CCN neural network. The images

*Copyright reserved © J. Mech. Cont.& Math. Sci.
Nada Hassan Jasem et al*
dataset was 1300 divided into 910 images (70%) for training and 390 images (30%) for testing. The training is performed on Windows machine which has a CPU Intel Core i7-8565, 1.99GHz and GPU Nvidia Ge-Force GTX1080-Ti, with a RAM of 8 GB.

Table 2: LPRCNN Layers Detailed Information

| Layer        | Filters | Size / Stride | Output         |
|--------------|---------|---------------|----------------|
| Convolution  | 16      | 3 x 3/1       | 416x416 x 16   |
| Maximum pooling | 2 x 2/2 | 208x208 x 16  |
| Convolution  | 32      | 3 x 3/1       | 208x208 x 32   |
| Maximum pooling | 2 x 2/2 | 104x104 x 32  |
| Convolution  | 64      | 3 x 3/1       | 104x104 x 64   |
| Maximum pooling | 2 x 2/2 | 52x52 x 64    |
| Convolution  | 128     | 3 x 3/1       | 52x52 x 128    |
| Maximum pooling | 2 x 2/2 | 26x26 x 128   |
| Convolution  | 256     | 3 x 3/1       | 26x26 x 256    |
| Maximum pooling | 2 x 2/2 | 13x13 x 256   |
| Convolution  | 512     | 3 x 3/1       | 13x13 x 512    |
| Maximum pooling | 2 x 2/1 | 7 x 7 x 1024  |
| convolution  | 1024    | 3 x 3/1       | 7 x 7 x 1024   |
| convolution  | 1024    | 3 x 3/1       | 7 x 7 x 1024   |
| convolution  | 30      | 1 x 1/1       | 7 x 7 x 30     |

VI. Testing Time

At testing time, it can compute the scores which are class-specific for every one of the boxes with the use of the Eq. (1). For a certain value of the threshold Pmin, the system will output the objects of the detection whose P Class ≥ Pmin. In the phase of the post-processing, the Non-maximum suppression has been utilized for the elimination of the duplicated detecting of same object.

\[ P(\text{Class } i) = P(\text{Class } i \mid \text{Object}) \times P(\text{Object}) \]  

(1)

P (Class i) represents the i-th class probability. P (Object) represents likelihood of including one of the objects in the grid and P (Class i | Object) represents conditional likelihood of class of the ith class that has been conditioned on the P (Object) (Redmon, Farhadi, 2017).
Second Stage:

VII. Feature Extraction

This method can be viewed as a significant stage in any system of pattern recognition, it indicates the detection of some features of certain objects in the image, the object may be found in a variety of the images based on those characteristics, the feature extraction method has to have efficiency and robustness for the sake of finding the objects in a variety of the scales or rotations (Pedersen, 2011) such robustness is not available in all methods of feature extraction. The Scale-invariant Feature Transforms (SIFT) The basic computational processed which were utilized for generating the group of the image characteristics: 1- Constructing the Scale Space 2- Key point Localization 3- Orientation Assignment 4- The Description of the Key-points. And Speed up Robust Features (SURF) is a sufficient feature detection approach; it has been developed from the SIFT. This procedure may be split to 3 steps:

1- The Construction of the Integral Image 2- Detection of the SURF Key points 3- Key point descriptions (Bay, 2006)

VII.i. Applying SIFT and SURF Algorithm

This is first step in the 2nd stage of the suggested system. It is implemented to test and train the images for the sake of detecting and extracting the characteristics, and this step’s input is a single merging image whereas its output is a floating vector containing key points which have been found with their positions, with the dimension (128 for the recipes in SIFT and (64) for the SURF descriptor. Table (2) lists the main points positions which have been found in the margins image, the total of key points for every one of the images differs from the other one, and that procedure has been performed in an automatic manner for each image in the data-base, the main points are found via the use of SIFT & SURF algorithms, the number of points varies Key discovered from one image to another.

VII.ii. Matching Strategy

The 128–64-dimensional descriptors by using (sift and surf), for each major point looks great, but is actually useless without a way to decide whether the query descriptor matches the training descriptor or not. For matching, the K-NN search has been utilized (Mathew, 2016).

The K-NN algorithm essentially calculates “distance” between the descriptor of the query and all of the training descriptors, and outputs K pairs with the minimum distance. Here we will keep K=2. Then we now have two pairs of matches for each query descriptor. So, the KNN search gave us two matches for each query descriptor. In the table below, the number of key points in which an actual match occurred is determined by the number of key points for both LPS that have been matched and the time that the matching took. Table (3) showing Details of Number of Key Points for sample pictures executed.
Table 3: Details of Number of Key Points for sample pictures executed

| Car Image | Detection | Reco. Using Surf | Nu. Points | Tim/Ms |
|-----------|-----------|-----------------|------------|--------|
| ![Image 1](image1.png) | ![Image 2](image2.png) | ![Image 3](image3.png) | 1746 | 234 |
| ![Image 4](image4.png) | ![Image 5](image5.png) | ![Image 6](image6.png) | 360 | 31 |
| ![Image 7](image7.png) | ![Image 8](image8.png) | ![Image 9](image9.png) | 1103 | 127 |

VIII. Dataset Description

In this research, dataset consisted of the Iraqi vehicles that were collected by the researcher from various places, including the University of Baghdad, Imam Jaafar al-Sadiq University, and general speed-specific traffic method of capturing images and videos based on using the **mobile phone iPhone camera** (7-megapixel, **Resolution1080x1920 pixels**) and/or **flash-Dual LED has been used**, also using other camera is Longs with specification are : 5-megapixel, **Waterproof IRHD-IP Camera with Internal POE, Lens:2.7-13.5 mm, Power: Dc12V/950Ma**. These datasets were taken in **different conditions**, the dataset consist (1,300) images and (35) videos that shown in the following Table (4). This data-set has been obtained from the acquisition of the images and videos of a car it the difficult conditions. Table 4 showing Image size and division of tested datasets.

*Copyright reserved © J. Mech. Cont. & Math. Sci.  
Nada Hassan Jasem et al*
Table 4: Image size and division of tested datasets

| Dataset | Image size (Pixel) | Image type and format | Division dataset |
|---------|--------------------|-----------------------|------------------|
| CNN     | 416X416            | Colour Scale(png)     | Total       | Training 70% | Testing 30% | Video testing |
|         |                    |                       | 1300         | 910          | 390         | 35            |

IX. Results
IX.i. Control of False Negatives / Positive

The Real-world, applications of the ALPR are not capable of, entirely focusing on the general, accuracy of the prediction with no identification between negatives, positives and false positives. In, the studied case, positivity is a plate that is either found or read accurately, and negative can be defined as a panel that is not found or read, and the false positives are an object other than a detected plate or incorrectly read a plate. Both negative and false, positive cases are counted as failed, projection, however, based, upon module, it is preferred, one over another.

Table 5: Result of measure the accuracy in 1st iteration training

| Measuring the Accuracy 1st iteration | false, positive Error rate | false negative Error rate | Accuracy     |
|-------------------------------------|-----------------------------|---------------------------|--------------|
| CNN                                 | 0.6%                        | 2.9%                      | (435 / 451) 96.5% |

Table 6: Result of measure the accuracy in 2nd iteration training

| Measuring the Accuracy 2nd iteration | false, positive Error rate | false negative Error rate | Accuracy     |
|-------------------------------------|-----------------------------|---------------------------|--------------|
| CNN                                 | 0.43%                       | 2.19%                     | (886 / 910) 97.4% |
### IX.ii. Evaluation Measures of LPR system

This is a metric which is utilized to evaluate the models of the classification. Informally, it can be defined as the fraction of the predictions that our model has gotten right. Which are defined as:

\[
\text{Accuracy} = \frac{\text{Number of Correct Prediction}}{\text{Total Number of Prediction}} \times 100
\]  \hspace{1cm} (2)

![Table 7: Result of recognition](image)

| Cars Video | Recognized | Accuracy |
|------------|------------|----------|
|            | Yes        | No       |
| Total videos (35) | 34         | 1        | 4        |
| Images (390)   | 387        | 3        | 99.2%    |

### IX.iii. Experience the Value of the Threshold of Confidence

The accuracy of the projection largely depends upon the amount of the training data and in the ideal case, a huge amount of the LPs has been utilized as samples of training. The proposed system has been trained in two iterations. The network training time was approximately one week for each one of the iterations.

For 1\textsuperscript{st} iteration, a training, group of (451) license, plate images were utilized, for network. Plates which, have been discovered in training, set have been only the plates which have been available at early phase of project, afterwards in project, a bigger data-set, of the license plates, has been, provided. Testing set, which has been utilized in the 1\textsuperscript{st} and 2\textsuperscript{nd} iteration includes (910) LPs which have been, chosen in a, random manner from the, bigger data-set, which, keeps the remaining for the additional testing and enhancements. The optimal results, have been accomplished with, network that trains on training group that spans in a range of the threshold ranging from 70 to 80. Table (8) lists the obtained results at testing time at 30% of the total number of images, for the 1\textsuperscript{st} iteration and 2\textsuperscript{nd} iteration respectively.
Table 8: Experimental results’ summary for different threshold

| Describe                   | Testing dataset | Threshold | Mistakes | False positives | Precision |
|----------------------------|-----------------|-----------|----------|-----------------|-----------|
| 1st iteration              | 190             | 70%       | 2        | 1               | 99.2%     |
| 2nd iteration              | 390             | 70%-80%   | 0        | 1               | 99.7%     |
| Avoid false positives      | 390             | 90%       | 1        | 0               | 99.7%     |

Fig. 9: Examples of Plate Detection Final System in Challenging Conditions for videos.

IX.iv. Metric of the Intersection Over Union (IOU)

Since the main goal of our work is to attach omni-directional LPs in a more compact manner, an IOU standard is also taken for measuring interference between the obtained areas and areas of true license plates (Hoiem, 2012). IOU can be characterized as follows:

\[
IOU = \frac{S_{\text{intersection}}}{S_{\text{union}}}
\]

\(S_{\text{intersection}}\) represents common area between discovered areas and terrestrial areas of reality, and \(S_{\text{union}}\) represents total area of both regions. Bigger value of the IOU will indicate a tighter container. Table (9) illustrates average IOU and accuracy for the plate class which is produced with every one of the epochs. Following 500 of the epochs, each one of the epochs with the maximum mean value of the IOU has been chosen as weights which require the implementation in suggested real-time applications. From Table (9), weights can be selected from 500th iteration due to the fact that it has 84.99% overlapping of the projected bounding box with ground truth and 99.52% predicted plates with the training loss of 0.0231.
Table 9: Predicted Plates and mean value of the IOU for every one of the Epochs

| Epoch | Plate (%) | Avg. IOU |
|-------|-----------|----------|
| 100   | 99.25     | 82.75    |
| 200   | 99.51     | 84.49    |
| 300   | 99.04     | 84.63    |
| 400   | 99.52     | 84.92    |
| 500   | 99.52     | 84.99    |
| 600   | 99.52     | 84.94    |
| 700   | 99.52     | 84.93    |

X. The Main Steps of User Interface

Fig. 10: Detecting an LP

Fig. 11: Retrieving all information related to the LP
XI. Conclusion

In this paper, the application program is designed to get to know the car license plate. First, we discovered the placement of the plate using CNN, then the plate was individually extracted in order to obtain the features of each plate using the surf algorithm in the images phase and the sift algorithm in the video phase, and finally we applied the template matching using KNN. This system has been designed for the identification of Iraqi LPs and the system has been tested in more than 1300 images. The 70% images are fragmented in the training phase 30% in the testing phase, as well as 35 videos were tested and finally, it proved 99.2% to detection plate and 97.14% for the plate recognition is accurate, which gives the overall system average performance 98.17%.

References

I. A. Khazri, "Automatic License Plate Detection & Recognition using deep learning", towards data science, 2019. web: https://towardsdatascience.com/automatic-license-plate-detection-recognition-using-deep-learning-624def07eaaf?gi=fc80f0526b7.

II. Saharkiz, "Nearest Neighbor Algorithm Implementation and Overview ", Code project, 2009. Web: https://www.codeproject.com/Articles/32970/K-Nearest-Neighbor-Algorithm-Implementation-and-Ov.
III. D. Hoiem, Y. Chodpathumwan, Q. Dai, "Diagnosing Error in Object Detectors", Computer Vision – ECCV, Springer Berlin Heidelberg, Berlin, Heidelberg, Vol. 1, No. 1, Pp. 340-353, 2012.

IV. H. Bay, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features", Computer Vision – ECCV, Springer Berlin Heidelberg, Berlin, Heidelberg, Vol. 5, No. 1, Pp. 404-417, 2006.

V. H. A.-H. Kahdum, "Leukocytes Image Segmentation and Classification Based on Geometrical Features and Naive Bayes Classifier," Master Degree, College of Science, University of Baghdad, Baghdad - Iraq, 2019.

VI. Kusumadewi, C.A. Sari, E.H. Rachmawanto, "License Number Plate Recognition using Template Matching and Bounding Box Method", Journal of Physics: Conference Series, IOP Publishing, Vol. 1, No. 1, Pp. 012067, 2019.

VII. J. Kim, "Automatic Vehicle License Plate Extraction Using Region-Based Convolutional Neural Networks and Morphological Operations", Symmetry, Vol. 11, No. 7, Pp. 7263-7271, 2017.

VIII. J. Redmon, A. Farhadi, "YOLO9000: better, faster, stronger", Proceedings of the IEEE conference on computer vision and pattern recognition, Vol. 1, No. 1, Pp. 7263-7271, 2017.

IX. J.T. Pedersen, "Study group SURF: Feature detection & description", Department of Computer Science, Aarhus University, Pp. 1-12, 2011

X. Mathew, Sheena S, “A Comparison of Sift And Surf Algorithm For The Recognition of An Efficient Iris Biometric System”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 1, Pp. 37–42, 2016.

XI. P. Marzuki, F. Radzi, Y.C. Wong, N. Abdul Hamid, N. Ali, M. Mat ibrahim, "A design of license plate recognition system using convolutional neural network", International Journal of Electrical and Computer Engineering (IJECE), Vol. 9, No. 2196, Pp.2, 2019.

XII. R. Girshick, "Fast r-cnn", Proceedings of the IEEE international conference on computer vision, Vol. 1, No. 1, Pp. 1440-1448, 2015.

XIII. S. Geethapriya, N. Duraimurugan, S. Chokkalingam, "Real-Time Object Detection with Yolo", International Journal of Engineering and Advanced Technology (IJEAT), Vol. 8, No. 1, Pp. 1440-1448, 2019.

XIV. S. Geethapriya, N. Duraimurugan, S. Chokkalingam, "Real-Time Object Detection with Yolo", International Journal of Engineering and Advanced Technology (IJEAT), Vol. 8, No. 1, Pp. 1440-1448, 2019.