Article

Not peer-reviewed version

Automatic Jordanian License Plate Detection and Recognition System Using Deep Learning Techniques

Tharaa Aqaileh * and Faisal Alkhateeb

Posted Date: 9 August 2023

doi: 10.20944/preprints202308.0747.v1

Keywords: automatic license plate detection and recognition; automatic vehicle logo detection and recognition; deep learning; transfer learning; convolutional neural network

Article

Automatic Jordanian License Plate Detection and Recognition System Using Deep Learning Techniques

Tharaa Aqaileh 1 and Faisal Alkhateeb [1,2]

[1] Department of Computer Science, Yarmouk University, Irbid, Jordan
[2] Department of Computer Science, University of Bahrain, Bahrain
* Correspondence: tharaa.aqaileh@gmail.com, falkhateeb@uob.edu.bh

Abstract: Recently, the number of vehicles on the road, especially in urban centers, has increased dramatically due to the increasing trend of individuals towards urbanization. As a result, manual detection and recognition of vehicles (i.e., license plates and vehicle manufacturer) become an arduous task and beyond human capabilities. In this paper, we have developed a system using transfer learning-based DL techniques for automatic identification of Jordanian vehicles. The YOLOv3 (You Only Look Once) model was re-trained using transfer learning to accomplish the license plate detection, character recognition, and vehicle logo detection. While VGG16 (Visual Geometry Group) model was retrained to accomplish the vehicle logo recognition. To train and test these models, four datasets have been collected. The first dataset consists of 7,035 Jordanian vehicle images, the second dataset consist of 7,176 Jordanian license plates, and the third dataset consists of 8,271 Jordanian vehicle images. These datasets have been used to train and test the YOLOv3 model for Jordanian license plate detection, character recognition, and vehicle logo detection, respectively. While the fourth dataset consists of 158,230 vehicle logo images used to train and test the VGG16 model for the vehicle logo recognition. Text measures were used to evaluate the performance of our developed system. Moreover, mean average precision (mAP) measure was used to evaluate the YOLOv3 model of the detection tasks (i.e., license plate detection and vehicle logo detection). For license plate detection, the precision, recall, F-measure, and mAP were 99.6%, 100%, 99.8%, and 99.9%, respectively. While for character recognition, the precision, recall, and F-measure were 100%, 99.9%, and 99.95%, respectively. The performance of license plate recognition stage was evaluated by evaluating these two sub-stages as a sequence, where the precision, recall, and F-measure were 99.8%, 99.8%, and 99.8%, respectively. Furthermore, for vehicle logo detection, the precision, recall, F-measure, and mAP were 99%, 99.6%, 99.3%, and 99.1%, respectively, while for vehicle logo recognition, the precision, recall, F-measure were 98%, 98%, and 98%, respectively. The performance of vehicle logo recognition stage was evaluated by evaluating these two sub-stages as a sequence, where the precision, recall, and F-measure were 95.3%, 99.5%, and 97.4%, respectively.

Keywords: automatic license plate detection and recognition; automatic vehicle logo detection and recognition; deep learning; transfer learning; convolutional neural network

1. Introduction

The Automatic License Plate and Vehicle Logo Detection and Recognition (ALP&VLDR) is an advanced computer vision-based task used to identify the vehicles without any human intervention (Sachin Prabhu, Kalambur and Sitaram, 2017). Moreover, the License Plate (LP) is considered as the identifier (ID) for each vehicle, which means that it is the primary information used to distinguish between vehicles (Rahman *et al.*, 2018). Moreover, the Vehicle Logo (VL) is used to complete identification information about the vehicles. Recently, the number of vehicles has increased dramatically on urban roads due to the increasing trend of urbanization, so ALP&VLDR system is an essential technology for intelligent transport system, surveillance system, and security. The main applications of ALP&VLDR in these systems are accident monitoring (Lu *et al.*, 2019), detecting stolen vehicles (Ramazankhani and Yazdian-Dehkordi, 2019), traffic control (Izidio *et al.*, 2019), and so on.

Automatic License Plate Detection and Recognition is sometimes known task by various other terms as Automatic Number Plate Recognition or Automatic License Plate Recognition. Moreover, Automatic Vehicle Logo Detection and Recognition task is known by another term is the Automatic Vehicle Manufacture Detection and Recognition. Recently, ALP&VLDR has become a vital topic and has attracted the attention of many researchers. However, they are still considered to be challenging tasks due to several reasons (Chai and Zuo, 2019) and (Yang *et al.*, 2020) such as the following:

- **LPs Variations:** there are many variations in LPs in terms of size, orientation, location, font, style, color, languages, etc.
- **VLs Variations:** there are many variations in VLs in terms of size, orientation, location, style, color, shape, etc.
- **Environmental Conditions Variations:** the vehicles are captured with different environmental conditions such as illumination (Lighting), variance (contrast), shadows, etc.

In addition, there are many factors that have made ALP&VLDR a challenging research topic such as the dirt and oblique LPs and VLs, the quality of the camera that captures the vehicles, and the distance between camera and vehicles.

The structure of the LPs is determined by the government of the country to which the LP belongs. In Jordan, the LP consists of maximum seven digits where the first number or the first two numbers determine the use of the vehicle (code number(s)) while the remaining numbers determine the vehicle ID. Also, it contains the name of the kingdom in English and Arabic, which are "الأردن" and "JORDAN", respectively. The numbers and letters of LP appear in black color and the background in white color. Further, the Jordanian LP consists of four-color codes: red, yellow, white, and green. In addition, there are two types of the Jordanian LPs: American and European, which differ in shape. American LP takes the square shape with $34 * 22$ cm size and consist of one or two numbers followed by the ID of the vehicle in a new line. European LP takes the rectangle shape with $52 * 11.4$ cm size and consists of one or two numbers followed by dash symbol and the ID of the vehicle (Alhaj Mustafa, Hassanin and Al-Yaman, 2018).

The main goal of this research is to develop a system for identifying Jordanian vehicles by detecting and recognizing the LP and logo of these vehicles. we apply Deep Learning (DL)-based Convolutional Neural Network (CNN) to accomplish the developed AJLP&VLDR system. In other words, all stages of the AJLP&VLDR system are mainly based on DL techniques that have achieved the state-of-the-art performance in most Computer Vision (CV) tasks. To our knowledge, the current systems depend upon datasets, where the plate coordinates are given. Moreover, there is no available dataset for Jordanian vehicles and existing VL datasets do not cover all VLs included in our research. So, we collected our own datasets for training and testing the proposed system. This, in turn, allows automatic traffic control, detecting traffic violations, detecting stolen vehicles, etc.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 shows the research methodology. The collected dataset is presented in Section 4. Section 5 presents the results, the evaluation of its efficiency, and discusses the findings. Finally, Section 6 presents the conclusions and future work.

2. Related Work

Over the last few decades, ALP&VLDR has become a vital research topic and attracted the attention of many researchers. A lot of techniques can be found in the literature that have been used to achieve the stages of the ALP&VLDR system. In this section we discuss the approaches that have been proposed for each stage in the ALP&VLDR system.

2.1. License Plate Recognition

This stage is responsible for recognizing LPs from the input images, as it passes through two other sub-stages to perform this task, namely are License Plate Detection (LPD), and Character Recognition (CR). Several methods have been proposed for these sub-stages as follows:

1.2.1.1. License Plate Detection

The responsibility of this sub-stage is to detect the place of the LP from the vehicle. Several methods have been used to successfully achieve this stage.

Generally, most LPs take a rectangular shape with a fixed aspect ratio, as image processing techniques can be used to detect all possible edges/borders (rectangles) of a vehicle image. (Radzi *et al.*, 2019) applied Sobel edge detection operator after removing noise and undesirable features from the input images. The experiment was conducted on 300 Malaysian vehicles images and achieved an accuracy 74.7%. Also, (Li *et al.*, 2019) applied canny edge detection image processing technique after enhancement the images and removing the noises from them using histogram equalization and median filtering respectively, which in turn slightly enhanced the detection process. The dataset used in this research consisted of 100 Chinese vehicles images.

(Nasser, Alsewari and Zamli, 2018) used canny edge detection along with Hough Transform (HT) and Euclidean distance to detect vertical and horizontal edges of Indian vehicles. But this method produced poor results with extremely distorted images. HT technique cannot detect efficiently rotated images because it is very sensitive to rotation. Whereas Connected Component Analysis (CCA) technique can deal with rotation. CCA is applied to binary images with 4- or 8-pixels neighborhood by scanning all pixels in the image, labeling them, and dividing them into blocks according to pixel connectivity (i.e., they have the same properties). For this reason, (Alhaj Mustafa, Hassanin and Al-Yaman, 2018) applied 8-pixels neighborhood CCA technique with canny edge detection for 240 Jordanian vehicle images. (Rabbani *et al.*, 2019) applied CCA technique based on the aspect ratio of LP for 100 Bangladeshi vehicle images, as the accuracy was 93.78%. In (Pustokhina *et al.*, 2020), Improved Bernsen Algorithm and CCA models were applied for three datasets: FZU cars, Stanford cars, HumAIn 2019. (Yousef *et al.*, 2020) applied fast marching segmentation method to detect Jordanian LPs. The dataset used to conduct the experiment consisted of 100 vehicle images, where the accuracy was 95%.

Edge-based image-processing techniques are usually easy to implement and have low time complexity, but require clear, continuous edges to work efficiently. Morphological operations (MOs) such as opening, closing, dilation, erosion, etc. have been used to filter the image by removing undesirable edges. These operations greatly improved the detection process of LPs. In (Ahmed, 2020), (Chai and Zuo, 2019), (Jagtap and Holambe, 2018), (Wang, 2017), and (Aung, Nwe and Yoshitaka, 2019), edge detection with MO have been proposed. The datasets have been used were 100 Egyptian car images, 100 Western Australia vehicle images, 100 Indian vehicle front or rear images, 200 Indian LP images, and 40 four-wheel Myanmar vehicles, respectively. The accuracy of LPs detection for this research was 82%, 97%, 92%, and 96%, respectively. The color of LPs in each country is usually specific, so (Khan *et al.*, 2021a) applied HSI color space with morphological open and close operations to localize Pakistan LPs. The dataset used consisted of 5,543 images where the accuracy was 96.72% accuracy.

Recently, DL techniques based CNNs models have been introduced. These models achieved remarkable results in most CV tasks. Moreover, unlike Machine Learning (ML) techniques, DL techniques do not require features extraction step as these features are automatically extracted without introducing hand-coded rules or human domain knowledge.

So, in (Ying, Xin and Wanxiang, 2018), a sliding window based selective search algorithm has been applied to generate all possible candidate areas that may contain a LP. Then Intersection over Union overlap was applied to evaluate these candidate regions based on the ground-truth bound/ annotated bounding box. Finally, Support Vector Machine (SVM) was applied to make the final judgment (i.e., for classification). This method was examined on 1,000 Chinese vehicle images and achieved 98.81% accuracy. Nevertheless, selective search algorithm is slow and take a long time to generate proposed regions. So, (Huang *et al.*, 2018) adapted Faster Region-CNN (R-CNN) based on the Region Proposal Network (RPN) algorithm to generate the proposed regions for Chinese LPs. Two datasets were used to examine this method, standard dataset, and real scene dataset, where the standard dataset contains 10,873 images and real scene dataset contains 4,711 images. In (Lin and Li, 2019), Mask R-CNN model was applied for LPD. This model can identify every pixel within the bounding box, which in turn improves the segmentation process accuracy. The experiment was conducted on 5,000 Taiwanese LP images and achieved 91% accuracy. (Selmi *et al.*, 2020a) mainly applied the Mask-RCNN model to detect Tunisian LPs. But the performance of this model was evaluated using four datasets, namely Caltech, AOLP, PKU, and Tunisian datasets, where the accuracy was 97.7%, 97.8%, 98.8%, and 97.4%, respectively.

Despite the regional algorithms are accurate in detecting objects, they are very slow and have high computational cost. For these reasons, proposal-free algorithms have been proposed in the literature, which directly detect the objects without generate any proposed regions. Single Shot MultiBox Detector (SSD) has been used by (Wang, 2018), which is considered faster than Faster R-CNN. The experiment was conducted for Chinese LPs using two datasets: 1,700 low-resolution images and 3,584 high-resolution images. Also, (Yao *et al.*, 2019) applied SSD CNN for LPs detection and achieved 98.3% accuracy for 50K Chinese LPs. (Kessentini *et al.*, 2019) applied You Only Look Once (YOLOv2) model to detect Tunisian LPs. Two datasets were used for the experiment, namely, GAP-LP consisting of 9,175 images and Radar datasets consisting of 6,448 images. The accuracy of the two datasets was GAP-LP 100% and 99.09%, respectively. Another proposal-free algorithm has been applied is YOLOv3 by (Saif *et al.*, 2019), which is considered extremely fast and real time object detection algorithm. The dataset used consisted of 200 images of Bangla vehicles where the accuracy was 100% accuracy. (Shafi *et al.*, 2021) also applied YOLOv3 to detect Pakistani LP, which achieved an accuracy of 97.82% for 2,131 vehicle images. In (Tourani *et al.*, 2020), YOLOv3 model was applied to detect Iranian LPs, where the dataset used consisted of 5,719 images and the accuracy was 97.8%. (Slimani *et al.*, 2020) applied both CCA with Inception-v3 model to detect the American LPs, where CCA was used to detect candidate regions and Inception-v3 model was used to classify these regions. The dataset used to conduct the experiment consists of 126 Caltech cars, which achieved an accuracy of 96.72%. In (Omar *et al.*, 2020), the SegNet model (semantic image segmentation) was applied to detect northern Iraq LPs. The dataset used to conduct the experiments consists of 600 vehicle images, where the accuracy was 91.01%.

1.1.12.1.2. Character Recognition

The last sub-stage of LP recognition is CR, which gives the result by recognizing the characters of the LPs that have been segmented and extracted from the previous stage. Several methods have been proposed in the literature for this purpose.

Recently, DL-based CNNs model have been used in most CV tasks, where achieved the stat-of-the-art accuracy. (Pham, Dinh and Nguyen, 2018) modified Lenet-5 CNN model by removing the last convolutional layer in order to increase the processing time. The experiment was conducted on 112,000 US vehicles, where the accuracy was 93.001%. (Yao *et al.*, 2019) modified SSD CNN model by removing the fully connected layer to increase the efficiency of CR. The efficiency of this model was measured through the experiment conducted on 90,213 Chinese LPs, where the accuracy was 99.1%. (Lu *et al.*, 2019) applied Visual Geometry Group (VGGNet) model for Chinese vehicles. The experiment was conducted on 1,403 vehicle images and achieved 97% accuracy. (Saif *et al.*, 2019) applied YOLOv3 to recognize 200 Bangladesh vehicle images and achieved 99.5% accuracy. (Shafi *et al.*, 2021) also applied YOLOv3 to recognize Pakistani LP, which achieved an accuracy of 96% for 571 plate character images. In (Tourani *et al.*, 2020), YOLOv3 model was applied to recognize Iranian LPs, where the dataset used consisted of 5,719 images and the accuracy was 95.05%. (Samadzadeh and Nickabadi, 2020) applied a residual network with 18 layers (ResNet18) for Iranian vehicles. The experiment result was conducted on tow dataset: IRCP dataset consists of 200 images and the collected dataset consists of 350 images. The accuracy was 95% and 81%, respectively. In (Raza *et al.*, 2020), AlexNet CNN model was applied as feature extraction and SVM was applied to classify these features. A multi-style dataset was used to evaluate this method, which consists of 3,718 vehicle images from eight different countries and the recognition accuracy was 96.04%. (Slimani *et al.*, 2020) applied the Inception-v3 model to recognize Taiwan LPs. The application-oriented license plate (AOLP) dataset, consisting of 2,049 images, was used, where the accuracy was 98.9%.

2.2. Vehicle Logo Recognition

This stage is responsible for recognizing VLs from the input images, as it passes through three other sub-stages to perform this task, namely are VLD, and VLR. Several methods have been proposed for these sub-stages as follows:

2.2.1. Vehicle Logo Detection

The responsibility of this sub-stage is to detect the place of the VL from the vehicle. Several methods have been used to successfully achieve this task.

(Yu *et al.*, 2018) applied overlapping enhanced patterns of oriented edge magnitudes method for VLD. The HFUT-VL dataset, consisting of 32,000 images belonging to 80 auto manufacturers, was used, where the accuracy was 96.3%. (Zhang *et al.*, 2017) applied bottom-up visual saliency to locate the VL for 11,500 images belonging to 10 vehicle manufacturers, where the accuracy was 96.07%. (Ansari and Shim, 2019) applied perspective transformation to detect 5 VLs with 2,042 images.

As mentioned earlier, all image processing techniques can achieve acceptable results under specific conditions. So, ML techniques have been used along with image processing techniques for detecting VLs. In (Tang *et al.*, 2017), Haar-like features and AdaBoost algorithm were applied to locate the VL from the input image. A dataset of 20,000 images belonging to 8 vehicle manufacturers with an accuracy of 97 % was used. Moreover, (Chen *et al.*, 2017) applied Adaboost-Based learning method to detect the VL. A dataset of 1,436 images belonging to 15 vehicle manufacturers with an accuracy of 93.94% was used. In (Yousef *et al.*, 2020), the VL is determined based on the locating of the LP using the fast-marching segmentation method. In other words, VL is detected based on prior knowledge that the VL is located on top of the LP. A dataset of 100 images belonging to 5 vehicle manufacturers was used.

Despite the robustness of ML techniques and its ability in object detection. However, it has a high computational cost due to features extraction step that is required to train a ML technique to apply the classification step based on these features. So, DL techniques have been for detecting VLs. (Yu *et al.*, 2021), applied the RPN to detect proposed regions that might contain VLs. A dataset of 225,000 images belonging to 15 vehicle manufacturers with an accuracy of 98.7% was used.

Despite the regional algorithms are accurate in detecting objects, they are very slow and have high computational cost. For these reasons, proposal-free algorithms have been proposed in the literature, which directly detect the objects without generate any proposed regions. In (Yang *et al.*, 2020), the VGG16 and YOLOv2 model was applied, where VGG16 was used for feature extraction and YOLOv2 was used for classification. A dataset of 2,065 images belonging to 30 vehicle manufacturers was used. The experiment was conducted on single-scale and multi-scale images, where the accuracy was 74.4% and 78.5%, respectively. In (Zhang *et al.*, 2021), a multi-scale vehicle logo detector based on the SSD model was applied. The VLD-45 dataset, consisting of 45,000 images belonging to 45 auto manufacturers, was used, where the accuracy was 84.8%. Whereas (NGUYEN, 2020) applied the ResNet-50 model to 8,000 images belonging to 13 vehicle manufacturers, where the accuracy was 90.5%.

2.2.2. Vehicle Logo Recognition

The responsibility of this sub-stage is to recognize the detected VL from the previous sub-stage. Several methods have been used to successfully achieve this task.

(Meethongjan, Surinwarangkoon and Hoang, 2020) applied Histogram of Oriented Gradients (HOG) descriptors with a feature selection method recognize VLs. A dataset of 4,000 images belonging to 40 vehicle manufacturers was used, where the accuracy was 75.25%. As mentioned earlier, all image processing techniques can achieve acceptable results under specific conditions. So, ML techniques have been used along with image processing techniques for recognizing VLs. (Du *et al.*, 2018) applied Scale Invariant Feature Transform descriptor with the SVM classifier, using 200 images belonging to 5 vehicle manufacturers, where the accuracy was 84.32%. But this method fails to detect the true VL of images with low contrast and high occlusion. In (Chen *et al.*, 2017), the SVM classifier with pyramid of HOG and multi-scale block local ternary patterns features for VLR was used. A dataset of 1,436 images belonging to 15 vehicle manufacturers was used, where the accuracy was 98.2%. In (Ding and Wu, 2020), a multi-feature fusion approach based on a two-level hierarchical classifier based was applied. Which means, three descriptors, namely, HOG, Curvature histograms, and GIST, were used for feature extraction with the SVM classifier. Moreover, the grey wolf optimize method was used to improve recognition accuracy by optimizing the kernel function. A dataset of 1,072 images belonging to 8 vehicle manufacturers was used, where the accuracy was 99.77%. In (Yu *et al.*, 2018) collaborative representation-based classifier was used for VLR. A dataset of 32,000 images belonging to 80 vehicle manufacturers was used, where the accuracy was 99.1%. (Tang *et al.*, 2017) applied principal components analysis (PCA) to 20,000 images belonging to is 8 vehicle manufacturers, where the accuracy was 91%. Whereas, in (Albera, 2017), gray level co-occurrence matrix PCA with K-Nearest Neighbor classifier was applied. A dataset of 1,000 images belonging to 10 vehicle manufacturers was used, where the accuracy was 88.5%.

Despite the robustness of ML techniques and its ability in object detection. However, it has a high computational cost due to features extraction step that is required to train a ML technique to apply the classification step based on these features. So, DL techniques have been for recognizing VLs. (Zhao, 2017) applied LeNet-5 model to identify VLs, using a dataset of 400 images belonging to 10 manufacturers was used, where the accuracy was 90.94%. (Zhang *et al.*, 2017) applied autoencoder pre-training deep neural network to identify VLs. 11,500 images belonging to 10 vehicle manufacturers, where the accuracy was 99.20%. (Ansari and Shim, 2019) applied Faster R-CNN and YOLOv2 models to 2,042 images belonging to 5 classes. (Huang *et al.*, 2019) applied Faster R-CNN model with two CNN models, namely, VGG16 and ResNet-50 to 4,000 images belonging to 8 vehicle manufacturers. The accuracy was 94.33% and 87.71%, respectively. (NGUYEN, 2020) applied multi-scale feature fusion YOLOv3 model to 8,000 images belonging to 13 vehicle manufacturers.

According to the above reviewing of the literature, and up to our knowledge, the works that have been applied for automatic Jordanian license plates detection and recognition depend on traditional techniques (image processing techniques). Furthermore, VLD and recognition methods do not cover all VLs included in our research. For these reasons, and to benefit from the advantages of all above mentioned methods, we develop an end-to-end AJLP&VLDR system based on DL-based CNNs using Transfer Learning (TL). The proposed system use image processing techniques for images enhancement to increase the accuracy of our system in detection and recognition of LPs and VLs as an initial stage. Moreover, DL-based CNN model will be re-trained using TL for LPs and VLs detection and recognition.

3. Methodology

In this research, we propose a system for automatic detection and recognition of Jordanian LPs and VLs using DL-based CNNs models. The proposed approach consists of three phases as shown in Figure 1: image pre-processing techniques, dataset preparation, license plate recognition using TL and vehicle logo recognition using TL.

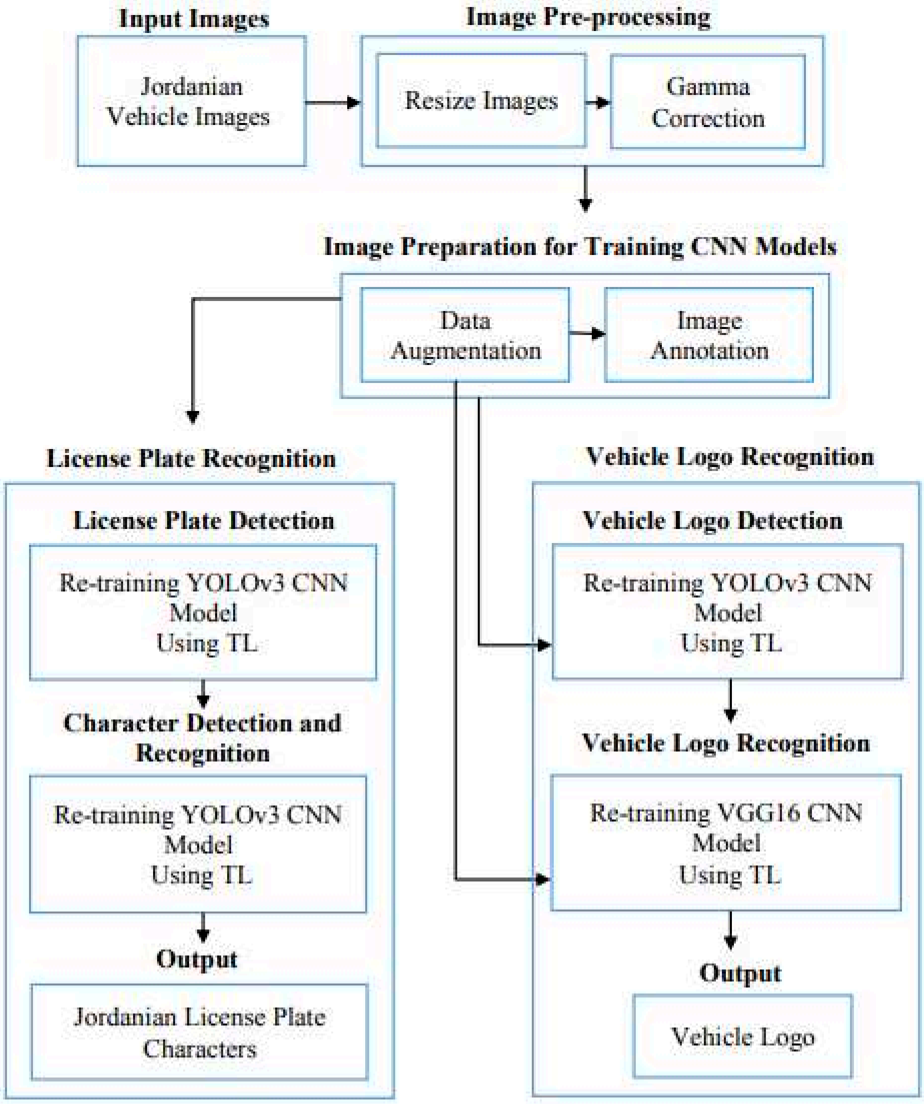

Input Images
Jordanian Vehicle Images
Image Pre-processing
Resize Images
Gamma Correction
Image Preparation for Training CNN Models
Data Augmentation
Image Annotation
License Plate Recognition
License Plate Detection
Re-training YOLOv3 CNN Model Using TL
Character Detection and Recognition
Re-training YOLOv3 CNN Model Using TL
Output
Jordanian License Plate Characters
Vehicle Logo Recognition
Vehicle Logo Detection
Re-training YOLOv3 CNN Model Using TL
Vehicle Logo Recognition
Re-training VGG16 CNN Model Using TL
Output
Vehicle Logo

Figure 1. Proposed AJLP&VLDR System Framework.

3.1. Image Pre-processing Techniques

The accurate of the other stages of the AJLP&VLDR system depends mainly on the quality of the input images. Where images are affected by several factors and conditions, the most important of which are environmental conditions such as illumination, variance, weather, etc. Several image processing techniques were used to improve the input images and make their details clearer.

3.1.1. Resize Images

Since CNN models are affected by large size images, which require too much memory and take a lot of computational time to process, train and test (output the result) (Huang *et al.*, 2017) and (Shafi *et al.*, 2021). Therefore, the Jordanian vehicle images were resized by downsampling their dimensions by YOLOv3 model automatically while maintaining the same aspect ratio and keeping the LP and logo region still predictable and recognizable. After resizing these images, their dimensions will be $416 * 416$ pixels. In addition, the VL images were resized by downsampling their dimensions by VGG16 model automatically. After resizing these images, their dimensions will be $224 * 224$ pixels. As a result, the process will be speed up.

3.1.2. Gamma Correction

The brightness of the input images may affect the performance of the proposed system. Whereas, images may be captured in an unrestricted environment, and capture devices may not capture lighting/ luminance properly. Moreover, the devices that display the input images may not display the original brightness correctly. Typically, the brightness/luminance of each pixel is a value between 0 and 1, where 0 indicates to darkness/black image and 1 indicates to brightness/white image. The gamma function will be used to correct the luminance of images as follows (Visa Sofia, 2011):

$$Otput\_luminance = gammaCorrectionFunction\,[input\_luminance] \tag{1}$$

Where $luminance$ is γ that indicates to the value of gamma. The higher the gamma value, the higher the brightness of images (the image tend to be white) and vice versa. The best value of gamma is 1.8 for outdoor images, which has been proven and achieved the best results (States, 2006). Furthermore, this step was applied to all our datasets.

3.2. Dataset Preparation

Two methods were used to prepare the datasets for the CNN model training process as follows:

3.2.1. Data Augmentation

Data Augmentation (DA) is a technique used to improve the performance of the model being trained (Lata, Dave and K.N., 2019) and (Selmi *et al.*, 2020b). As mentioned earlier, small dataset is not sufficient to build more skilled and accurate DL model (Barman and Patil, 2021). Therefore, DA is an option to automatically and artificially expand the size of the dataset (Shorten and Khoshgoftaar, 2019). The main objective of DA is to improve the model generalization and avoid overfitting by increasing the quantity and diversity of the dataset (Ying, 2019). Overfitting generally happens when a model trained using a small dataset, where the features that are extracted from that dataset are not sufficient to be generalized to new data (unseen data) (Ying, 2019), (Shorten and Khoshgoftaar, 2019), and (Garbin, Zhu and Marques, 2020). In our work, several but realistic transformations were applied to our datasets as follows:

- Normalizes images by 1./255, which transforms image pixels in a range between 0 and 1.
- Random rotation in range 0 and 45.
- Random zoom with a range equal to 0.2.
- Width and height shift with a range equal to 0.2.
- Random shear with a range equal to 0.2.
- Average blurring using a $5\ x\ 5$ average kernel.
- Random translation with a range equal to 0.5.
- Random horizontal flip with a range equal to 0.5.
- Random crop with a range equal to 0.5.

These transformation methods have been applied to our datasets interchangeably. The first six transformations were applied to our VL dataset as they were all applied together for the first time. After that, these six methods were divided into groups and each group was applied separately. Further, the last three transformations were applied by the YOLOv3 model automatically during the training process.

3.2.2. Image Annotation

As is known, object detection models require annotation file for training. Since YOLOv3 is considered an object detection model, text annotation files are required. Typically, the annotation file consists of information about the bounding box (ground-truth) that surrounds the object. LabelImg is a graphical annotation, free and open-source tool for manually annotating and labeling objects in images. This annotation tool is written in python and uses Qt for developing the graphical interface. Moreover, it provides several annotation formats that are supported by many ML models. These formats are VOC PASCAL for XML format and text files for YOLO. Figure 25 shows the Graphical

User Interface (GUI) of LabelImg tool (Kilinc and Uludag, 2012) and (Prusty, Tripathi and Dubey, 2021).

Moreover, the name of each annotation file must be the same as the corresponding image name. In addition, annotation information about bounding boxes of each object is saved in a text file for YOLO, each on a separate line. Each line has following format (Kilinc and Uludag, 2012):

<object = class> <x-center> <y-center> <width> <height>

- <object = class>: it is a number ranging from 0 to (number of classes - 1), which represents the class (label) of the object.
- <x-center> and <y-center>: x and y are ranging from 0.0 to 1.0, which represent the center of the bounding box.
- <width> and <height>: width and height are ranging from 0.0 to 1.0, which represent the width and height of the bounding box.

3.3. License Plate Recognition Using Transfer Learning

Object detection is a CV technique that consists of two tasks: object detection (localization) and object classification (recognition/identification). Object localization is responsible to identify the location of objects in the input image by drawing a bounding box around them. Whereas object classification is responsible for classifying the localized object into a specific class (Liu, 2019), (Pal and Chawan, 2020), and (Prusty, Tripathi and Dubey, 2021). In our work, license plate recognition (LPR) stage is responsible to recognize the character of the Jordanian LPs from the input vehicle images. To accomplish this task, it passes into two other sub-stages: LPD and CR.

3.3.1. License Plate Detection

Detecting the location of the LP from the input vehicle image is one of the most important stages because the accuracy of the CR stage depends on it. YOLOv3 model is one of the most accurate and fast CNN models for object detection. This model was trained on a large-scale object detection dataset, which is Common Objects in Context consisting of 330K images belonging to 80 different classes (labels). The model achieved a mAP of 57.9% in 51ms for object detection (Lin *et al.*, 2014), (He *et al.*, 2016), and (Redmon and Farhadi, 2018).

TL is, as it is known, one of the most effective ways to improve training of DL CNN models on a small dataset, since training such models from scratch requires a huge amount of dataset (Garbin, Zhu and Marques, 2020). Therefore, The YOLOv3 model was adapted to fit our task of Jordanian LPD using TL. In general, YOLOv3 model consists of two chained networks: the backbone (feature extractor) and the feature detector. The layers of feature extractor network have no constraints on the number of filters (i.e., channel size/depth). In simple words, backbone layers have an arbitrary number of filters. Whereas, in a feature detector network, the number of filters for all three detector layers (i.e., trainable layers) is constrained by the number of classes. Thus, if the number of classes changes, the number of filters (output size) in the detector network will change (Redmon and Farhadi, 2018). In our work, the number of categories for this sub-stage (Jordanian LPD) is one, which is the LP. Consequently, only trainable layers (i.e., detector layers) were re-trained on the labeled Jordanian LPs from our own Jordanian vehicle dataset to fit the task of Jordanian LPD, while all other layers were frozen. The output of the fitted YOLOv3 model is the Jordanian LP location (the bounding box containing the LP) with the “License_Plate” label and its confidence score. Figure 2 shows the output of the re-trained YOLOv3 model for the Jordanian LPD task.

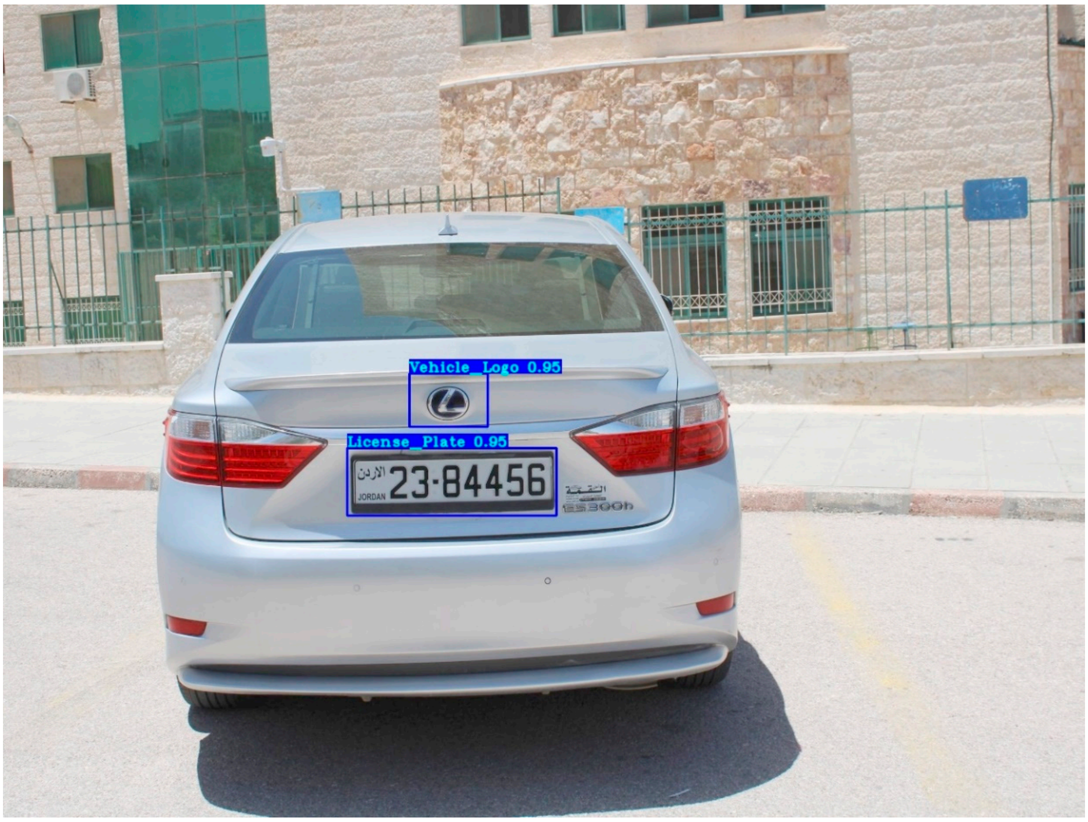
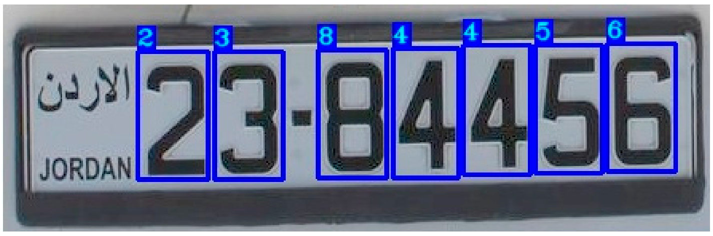


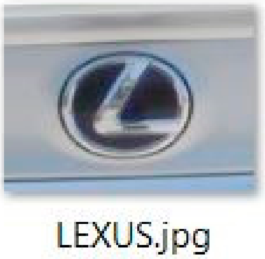
Figure 2. Outputs of the Re-trained YOLOv3 Model for the Jordanian LPD and VLD Tasks.

3.3.2. Character Detection and Recognition

This sub-stage is responsible for giving the result by recognizing the characters from the previously detected Jordanian LP. Since YOLOv3 model is very effective in detecting and recognizing small objects, it has been re-trained to achieve this task using TL. In general, YOLOv3 has only three trainable layers (i.e., detector layers), which are constrained by the number of classes (Redmon and Farhadi, 2018). Since Jordanian LP consists of a maximum of seven characters (numbers), the number of classes is seven. Therefore, these trainable layers were re-trained on the labeled Jordanian LP characters from our own dataset and all other layers were frozen. Figure 3a shows the output of the re-trained YOLOv3 model for the Jordanian LPR (character) task of the previously detected LPs presented in Figure 2.



(**a**) Outputs of the Re-trained YOLOv3 Model for the Jordanian LPR (CR) Task.



(**b**) Outputs of the Re-trained VGG16 Model for the VLR (Classification) Task.

Figure 3. LPR and VLR Stages Outputs for the ALP&VLDR System.

3.4. Vehicle Logo Recognition Using Transfer Learning

In this stage the vehicle manufacturer (i.e., VL) is recognized from the input vehicle images. To achieve this task, this stage goes through two other sub-stages, namely, VLD and VLR.

3.4.1. Vehicle Logo Detection

This sub-stage is responsible for detecting the logo from the input vehicle images. As mentioned earlier, YOLOv3 model outperforms other object detection models in detecting small objects (Redmon and Farhadi, 2018). So, it was fitted to achieve the VLD task by re-training its trainable layers (i.e., detector layers) on the labeled VLs from our Jordanian vehicle dataset and all other layers were frozen. The output of the fitted YOLOv3 model is the VL location (i.e., the bounding box containing the VL) with the “Vehicle_Logo” label and its confidence score. Figure 2 shows the output of the re-trained YOLOv3 model for the VLD task.

3.4.2. Vehicle Logo Recognition (Classification)

VLR (i.e., vehicle logo classification) is the last stage of our developed system, which is responsible for recognizing (classifying) the previously detected logo. VGG16 model is considered as an off-the-shelf classification model that achieved 92.7% accuracy in ImageNet dataset, which consists of fourteen million images belonging to thousand different classes (Simonyan and Zisserman, 2015). Since the domain and task of the VGG16 model are similar to our domain and task, we used it as a starting point for solving our task. In other words, we used the VGG16 model as a feature extraction and then added and re-trained our dense layers (i.e., fully connected layers and classification layer).

The process of retraining the VGG16 model using TL to fit the task of recognizing the VL (classification) is as follows:

1. At first, we instantiated a base model and then loaded the weights of the pre-trained VGG16 base layers (backbone/feature extractor) to it.
2. Dense layers have been cropped, leaving only the base layers (i.e., convolutional, and pooling layers) by setting include_top = False.
3. Layers in the instantiated base model have been frozen by setting trainable = False to avoid destroying any of the features (weights) during training rounds.
4. New trainable layers have been added (dense layers) on the top of the frozen layers and have been re-trained on our VL dataset to fit the VLR task.
5. Dropout layer have been added after the activation function (full-connected layers) to reduce the overfitting.

Figures 3b shows the output of the re-trained YOLOv3 model for the VLR (classification) task of the previously detected VLs presented in Figure 2.

4. Data Sample

In this work we collected four different datasets to accomplish the AJLP&VLDR system (i.e., to train and test CNN models). Up to our knowledge, there is no publicly available dataset for Jordanian vehicles. So, we collected our own dataset by take photos of the Jordanian vehicles from different places, which are garages, public streets, malls, universities, and hospitals. The vehicle images were captured from front and rear ate different angles from -35° to 35° at various distances from 2m to 8m using a Canon digital camera. Moreover, these images were captured in different illumination conditions and environments (i.e., day, sunny weather, and cloudy weather). The total number of captured images is 8,271 with $6000 x 4000$ resolution and JPG format. The collected dataset consists of the following attributes:

- 7,102 single vehicles.
- 1,169 multiple vehicles.
- 2,290 front vehicles.
- 7,645 rear vehicles.
- 7,265 vehicles were captured with sunny weather.
- 1,006 vehicles were captured with cloudy weather.
- 8,271 vehicles were captured during the day.
- 3,853 vehicles with American LPs.
- 6,082 vehicles with European LPs.
- LPs color codes:
    - 9,838 vehicles with white color code.
    - 81 vehicles with green color code.
    - 8 vehicles with yellow color code.
    - 8 vehicles with red color code.

As mentioned earlier, the LPR stage consists of two other sub-stages: LPD and CR. Therefore, two different datasets were used to accomplish this stage. For the first sub-stage (i.e., LPD), we used a dataset consisting of 7,035 Jordanian vehicle images taken from the images we captured. Whereas, for the second sub-stage (i.e., CR), we manually cropped Jordanian LPs from the vehicle images used in the first sub-stage. In total, the number of cropped Jordanian LPs was 7,176. As a result, these cropped LPs formed a new dataset.

Furthermore, two other different datasets were used to accomplish the second stage (i.e., VLR) of our developed system. This stage consists of two other sub-stages: VLD and VLR. For the first sub-stage (i.e., VLD), we used a dataset consisting of all 8,271 Jordanian vehicle image we captured. Whereas, for the second sub-stage (i.e., VLR), we collected 152,572 VL images. These images were collected as follows:

- We cropped some of these images from the Jordanian vehicles images we captured, which equals 11,147 images.
- We collected some of them from different websites, which equals 36,620 images.
- We generated some of them by applying DA techniques, which equals 104,805 images.

In addition, we used some images from the VL dataset created by Kuba Siekierzynski after filtering, which equals 5,658 images belonging to 28 VLs (siekierzynski, 2017). Eventually, the dataset used for VL recognition sub-stage consists of 158,230 images distributed over 39 VL classes, which covers all the logos in the Jordanian vehicle images we captured. These classes are AUDI, BMW, BRENTHON, BUICK, C_EMBLEM, CHEVROLET, CITROEN, DAEWOO, DAIHATSU, DODGE, FIAT, FORD, GENESIS, GMC, GT_EMBLEM¸ HONDA, HYUNDAI, ISUZU, JEEP, KIA, LANCIA, LAND_ROVER, LEXUS, LINCOLN, M_EMBLEM, MAZDA, MERCEDES_BENZ, MERCURY, MITSUBISHI, NISSAN, OPEL, PEUGEOT, PORSCHE, RENAULT, RENAULT_SAMSUNG, SUZUKI, TESLA, TOYOTA, and VOLKSWAGEN.

5. Results and discussion

In this section, we present and discuss the experimental results which were carried out based on the proposed methodology.

5.1. Data Sample Partitioning

The previously mentioned datasets used to accomplish the developed AJLP&VLDR system were divided into three partitions (i.e., training, validation, and testing) with specific ratios as shown in Table 1.

Table 1. Datasets Partitioning.

Stage	Training Dataset	Validation Dataset	Testing Dataset
LPD	70%	15%	15%
Character Detection and Recognition	70%	15%	15%
VLD	70%	15%	15%
VLR	70%	20%	10%

5.2. System Validation

In this sub-section, we conducted experiments to validate the performance of our developed AJLP&VLDR system. The implementation (training) and experiments were carried out on an Intel (R) Xeon ®, 2.79 GHs CPU, 32 GB RAM, and 64-bit Windows 10 operating system with NVIDIA Tesla k40 GPU using python programming language with TensorFlow and Keras libraries. Moreover, the input size and hyper-parameters of all the CNN models were configured as shown in Table 2 to perform the training (i.e., TL).

Table 2. Input Size and Hyper-parameter Configuration.

Stage	CNN Model	Input Size	Batch Size	Number of Epochs
LPD	YOLOv3	416	32	100
Character Detection and Recognition	YOLOv3	416	32	100
VLD	YOLOv3	416	32	100
VLR	VGG16	224	32	10

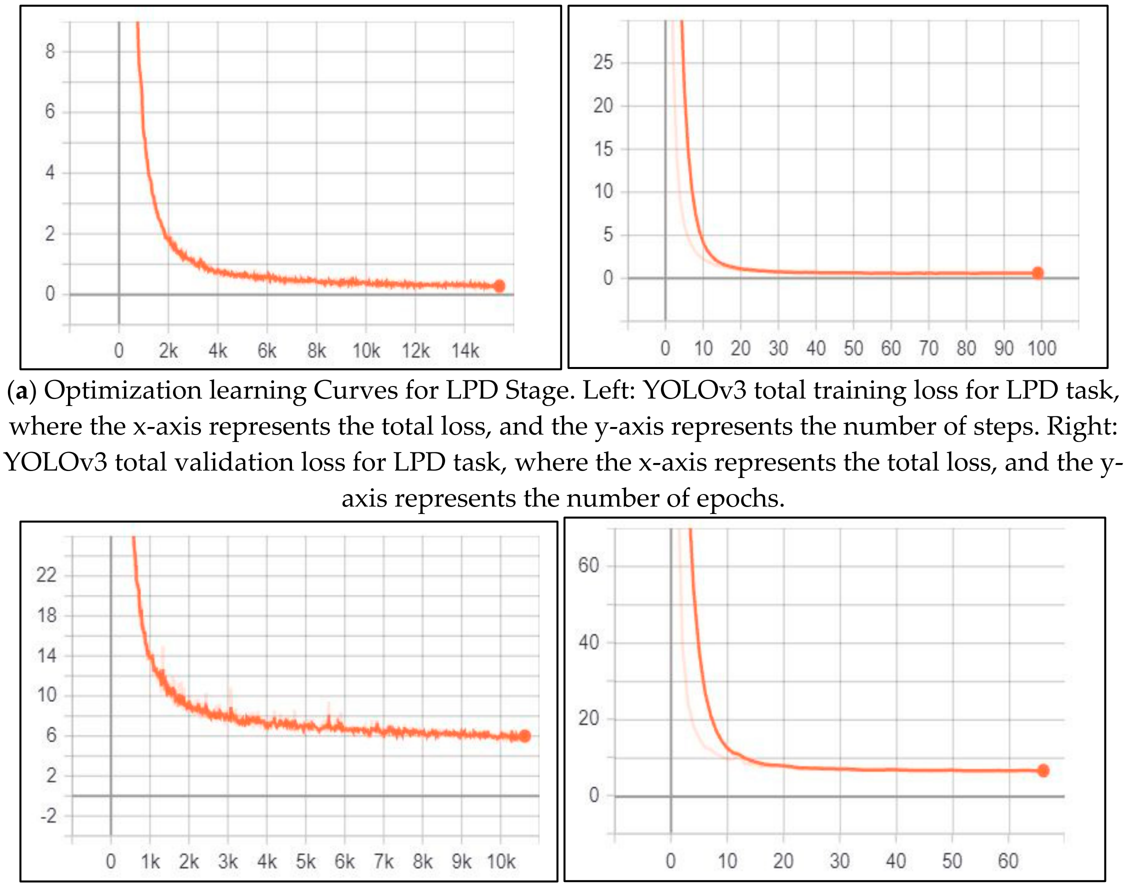
The performance of the CNN models used to achieve the developed system was monitored and examined during the training process using learning curves. Learning curves belong to two types: optimization learning curves and performance learning curves. Optimization learning curves are charts that are calculated using optimization matrices, by which model parameters are optimized such as loss. Whereas performance learning curves are also charts that are calculated using performance metrics, by which the model is evaluated and selected such as accuracy. Furthermore, these curves are calculated for both the training and validation dataset during the training process. Thus, two curves are generated: train learning curve and validation learning curve. Train learning curve is calculated on the training dataset which gives an idea of how well the model has learned. Whereas validation learning curve is calculated on the validation dataset which gives an idea of how good the model is to generalize (Anzanello and Fogliatto, 2011). In our work, optimization learning curve was calculated for YOLOv3 model, while performance learning curve was calculated for VGG16 model as follows:

5.2.1. License Plate Recognition

During the YOLOv3 training process, the optimization learning curve is calculated using loss function for both the training and validation dataset in each batch. As result, two charts are generated: training loss and validation loss. The training loss chart gives an idea of how well the model fits the training dataset, while the validation loss chart gives an idea of how well the model fits the unseen dataset (new dataset). The lower the loss, the better the performance of the model (Smith, 2018). In our work, training loss and validation loss were stored in a log file during training process and then visualized using the TensorBoard provided with TensorFlow. Generally, YOLOv3 loss function is the sum of box regression loss (coordinates loss), confidence loss (objectness loss), and classification loss (Redmon *et al.*, 2016).

5.2.1.1. License Plate Detection

While training YOLOv3 model on the Jordanian LPD task, its performance was monitored and validated using the loss function. Figure 4a illustrates the training and validation loss for this task, respectively.



(**a**) Optimization learning Curves for LPD Stage. Left: YOLOv3 total training loss for LPD task, where the x-axis represents the total loss, and the y-axis represents the number of steps. Right: YOLOv3 total validation loss for LPD task, where the x-axis represents the total loss, and the y-axis represents the number of epochs.

(**b**) Optimization learning Curves for Character Detection and Recognition Stage. Left: YOLOv3 total training loss for character detection and recognition task, where the x-axis represents the total loss, and the y-axis represents the number of steps. Right: YOLOv3 total validation loss for LPD task, where the x-axis represents the total loss, and the y-axis represents the number of epochs.

Figure 4. Optimization learning Curves for LPD and Character Detection and Recognition stages for the ALP&VLDR System.

According to the above figure, both training loss and validation loss are gradually decreasing, which tend to coincide after epoch 40. The training loss was 0.24 and the validation loss was 0.60. Moreover, the training loss indicates that the model fits the training dataset, converges rapidly after epoch 40 and remains stable. Also, the validation loss indicates that the model fits the new dataset (i.e., the model is generalizable).

5.2.1.2. Character Detection and Recognition

The performance of the YOLOv3 model for the character detection and recognition task was monitored and validated during the training process using the loss function. Figure 4b illustrates the training loss and the validation loss for this task, respectively.

According to the figure 4b, both training loss and validation loss are gradually decreasing, which tend to coincide after epoch 50. The training loss was 6.086 and the validation loss was 6.745. Moreover, the training loss indicates that the model fits the training dataset, converges rapidly after epoch 50 and remains stable. Also, the validation loss indicates that the model fits the new dataset (i.e., the model is generalizable). Initially, we set the training epochs to 100, but the model stopped improving (i.e., converged) after the epoch 67. So, we stopped the training process in this epoch.

5.2.2. Vehicle Logo Recognition

In this stage, the performance was monitored and validated using both types of learning curves mentioned earlier. For the YOLOv3 model dedicated for VLD tasks, the optimization learning curve was used, which is the loss. Whereas, for the VGG16 model dedicated for VLR (classification), the performance learning curve was used, which is the accuracy.

5.2.2.1. Vehicle Logo Detection

The performance of the YOLOv3 model for the VLD tasks was monitored and validated during the training process using the loss function. Figures 5 illustrates the training loss and the validation loss for this task, respectively.

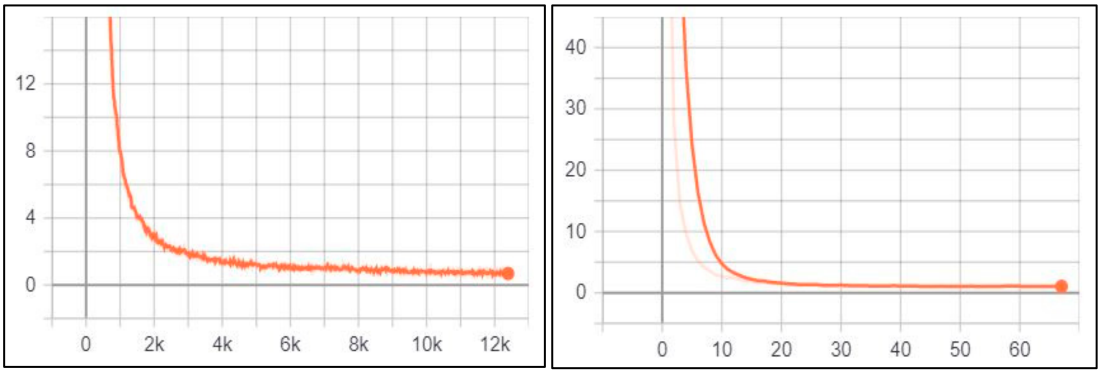


Figure 5. Optimization learning Curves for VLD Stage. Left: YOLOv3 total training loss for VLD task, where the x-axis represents the total loss, and the y-axis represents the number of steps. Right: YOLOv3 total validation loss for LPD task, where the x-axis represents the total loss, and the y-axis represents the number of epochs.

According to the above figure, both training loss and validation loss are gradually decreasing, which tend to coincide after epoch 50. The training loss was 0.726 and the validation loss was 1.002. Moreover, the training loss indicates that the model fits the training dataset, converges rapidly after epoch 50 and remains stable. Also, the validation loss indicates that the model fits the new dataset (i.e., the model is generalizable). Initially, we set the training epochs to 100, but the model stopped improving (i.e., converged) after epoch 68. So, we stopped the training process in this epoch.

5.2.2.2. Vehicle Logo Recognition (Classification)

The performance of the VGG16 model for the VLR (classification) task was monitored and validated during the training process of using an accuracy matrix. The training accuracy and the validation accuracy were visualized using the matplotlib library provided by the python programming language as deomstrante in Figure 6.

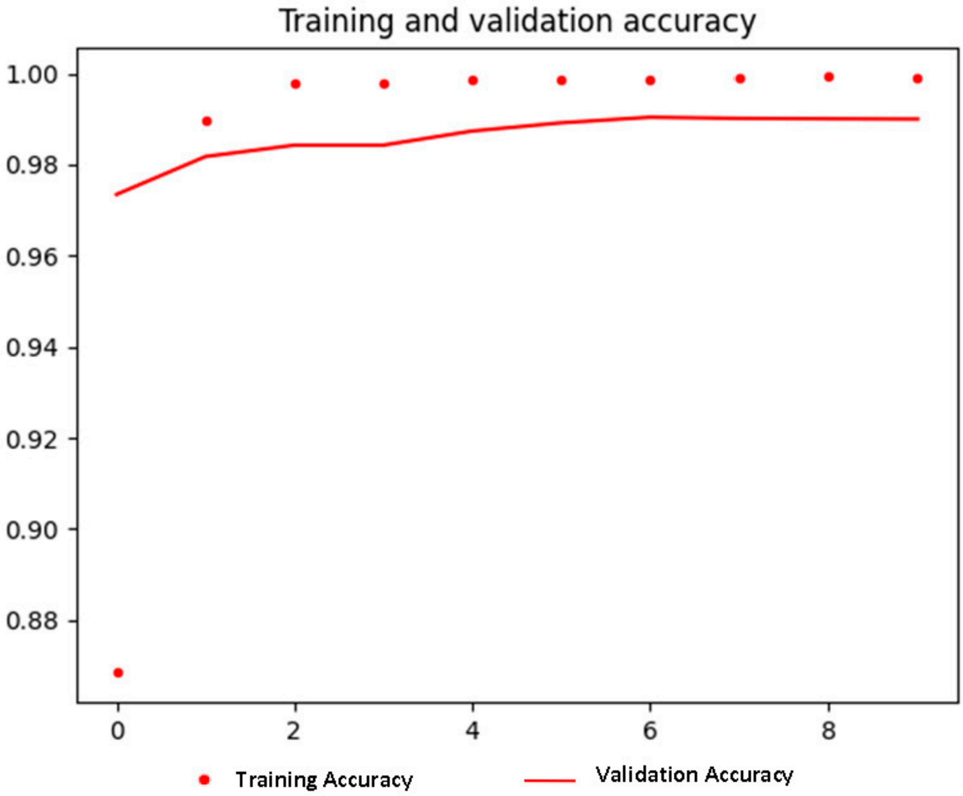


Figure 6. Training and Validation Accuracy for VGG16 Model for VLR (Classification) Task. The x-axis represents the Accuracy, and the y-axis represents the number of epochs. Also, the dotted line represents the training accuracy, and the solid line represents the validation accuracy.

According to the above figure, the training accuracy and validation accuracy are gradually increasing and tend to coincide. The training accuracy was 0.9991 and the validation accuracy was 0.99. Moreover, the training loss and validation loss were 0.0041 and 0.0604, respectively. The training loss indicates that the model fits the training dataset, converges rapidly after epoch 3 and remains stable. Also, the validation loss indicates that the model fits the new dataset (i.e., the model is generalizable).

5.3. System Evaluation

Several metrics have been conducted to evaluate the performance of the proposed approach. These matrices are precision, recall, and F-measure, which were applied to all test datasets. Moreover, mAP measure was used to evaluate the YOLOv3 model of the detection tasks (i.e., LPD and VLD). In other words, each stage in the developed system was evaluated by calculating the test metrics for its dataset. Table 3 and Figure 7 show the performance of the AJLP&VLDR stages on test set.

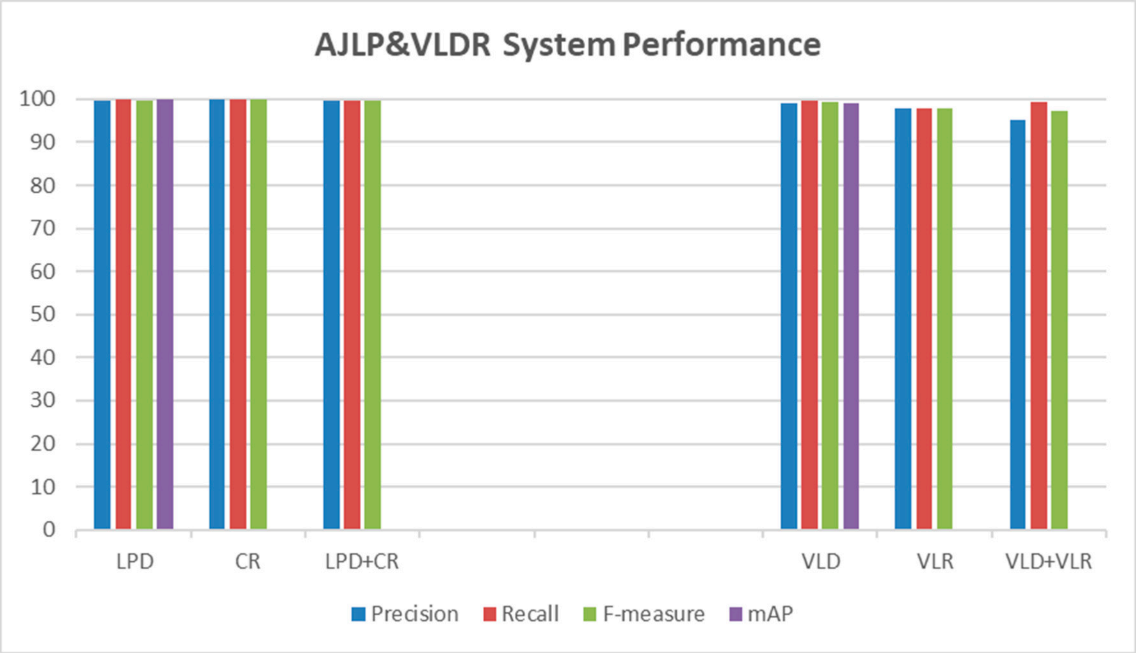


Figure 7. Performance of AJLP&VLDR Stages on Test Set.

Table 3. Performance of AJLP&VLDR Stages on Test Set.

Stage	Precision	Recall	F-measure	mAP
LPD	99.6%	100%	99.8%	99.9%
Character Detection and Recognition	100%	99.9%	99.95%	____
Overall License Plate Recognition	99.8%	99.8%	99.8%	____
Vehicle Logo Detection	99%	99.6%	99.3%	99.1%
Vehicle Logo Recognition (Classification)	98%	98%	98%	____
Overall Vehicle Logo Recognition	95.3%	99.5%	97.4%	____

5.3.1. License Plate Recognition

As described in the methodology chapter, the LPR stage consists of two other sub-stages: LPD and CR. Therefore, the effectiveness and robustness of this stage in Jordanian license plate recognition was evaluated by evaluating each sub-stage separately. The overall performance is then evaluated by evaluating the two sub-stages together as a sequence.

5.3.1.1. License Plate Detection Evaluation

The performance of the YOLOv3 model that was trained to fit the Jordanian LPD task was evaluated using the previously mentioned text measures (metrics), which were defined for this task as follows (Khan *et al.*, 2021b):

$$Precision = \frac{the\ number\ of\ correctly\ detected\ license\ plates}{the\ total\ number\ of\ license\ plates\ positively\ detected} \tag{2}$$

$$Recall = \frac{the\ number\ of\ correctly\ detected\ license\ plates}{the\ total\ number\ of\ ground\ truths\ (vehicles)} \tag{3}$$

The precision and recall were 99.6% and 100%, respectively. Meanwhile, the F-measure reached 99.8%. Also, the mAP was 99.9%.

5.3.1.2. Character Detection and Recognition Evaluation

The performance of the YOLOv3 model that was trained to fit the Jordanian LP character detection and recognition task was evaluated using the previously mentioned text measures, which were defined for this task as follows:

$$Precision = \frac{the\ number\ of\ correctly\ recognized\ license\ plates}{the\ total\ number\ of\ license\ plates\ positively\ recognized} \tag{4}$$

$$Recall = \frac{the\ number\ of\ correctly\ recognized\ license\ plates}{the\ total\ number\ of\ ground\ truths\ (license\ plates)} \tag{5}$$

Moreover, the recognition result is valid when all LP characters are recognized correctly. The precision and recall were 100% and 99.9%, respectively. Meanwhile, the F-measure reached 99.95%.

5.3.1.3. Overall License Plate Recognition Evaluation

The overall performance of the developed system for Jordanian license plate recognition was evaluated by evaluating the two sub-stages (i.e., LPD and character detection and recognition) together as a sequence. The test set from the first dataset was used for evaluation. The text measures for this task were defined as follows(Zhang *et al.*, 2018):

$$Precision = \frac{the\ number\ of\ correctly\ recognized\ license\ plates}{the\ total\ number\ of\ license\ plates\ positively\ detected} \tag{6}$$

$$Recall = \frac{the\ number\ of\ correctly\ recognized\ license\ plates}{the\ total\ number\ of\ ground\ truths\ (vehicles)} \tag{7}$$

The precision and recall were 99.8% and 99.8%, respectively. Meanwhile, the F-measure reached 99.8%.

5.3.2. Vehicle Logo Recognition

As described in the methodology chapter, the VLR stage consists of two other sub-stages: VLD and VLR (classification). Therefore, the effectiveness and robustness of this stage in vehicle recognition was evaluated by evaluating each sub-stage separately. The overall performance is then evaluated by evaluating the two sub-stages together as a sequence.

4.3.2.1. Vehicle Logo Detection Evaluation

The performance of the YOLOv3 model that was trained to fit the vehicle logos detection task was evaluated using the previously mentioned text measures, which were defined for this task as follows:

$$Precision = \frac{the\ number\ of\ correctly\ detected\ vehicle\ logos}{the\ total\ number\ of\ vehicle\ logos\ positively\ detected} \tag{8}$$

$$Recall = \frac{the\ number\ of\ correctly\ detected\ vehicle\ logos}{the\ total\ number\ of\ ground\ truths\ (vehicles)} \tag{9}$$

The precision and recall were 99% and 99.6%, respectively. Meanwhile, the F-measure reached 99.3%. Also, the mAP was 99.1%.

5.3.2.2. Vehicle Logo Recognition (Classification) Evaluation

The performance of the VGG16 model that was trained to fit the vehicle logos recognition (classification) task was evaluated using the previously mentioned text measures. At first, these metrics were calculated for each class (VL) which were defined as follows:

doi:10.20944/preprints202308.0747.v1

$$Precision_{class\ i} = \frac{the\ number\ of\ correctly\ recognized\ (classifed)\ vehicle\ logos}{the\ total\ number\ of\ vehicle\ logos\ positively\ recognized} \tag{10}$$

$$Recall_{class\ i} = \frac{the\ number\ of\ correctly\ recognized\ (classified)\ vehicle\ logos}{the\ total\ number\ of\ ground\ truths} \tag{11}$$

Where $class\ i$ refers to the VLs mentioned in Section 4. After that, the micro-average precision and micro-average recall were calculated as follows (Asch, 2013) and (Grandini, Bagli and Visani, 2020):

$$Micro - Average\ Precision = \frac{\sum_{i=1}^{n} TP_{class\ i}}{\sum_{i=1}^{n} TP_{class\ i} + \sum_{i=1}^{n} FP_{class\ i}} \tag{12}$$

$$Micro - Average\ Recall = \frac{\sum_{i=1}^{n} TP_{class\ i}}{\sum_{i=1}^{n} TP_{class\ i} + \sum_{i=1}^{n} FN_{class\ i}} \tag{13}$$

Where n is the number of classes. After that, the Micro F-measure based on average precision and recall was calculated as follows (Asch, 2013) and (Grandini, Bagli and Visani, 2020):

$$Micro\ \ F - measure = \frac{2 * Micro - Average\ Preccision * Micro - Average\ Recall}{Micro - Average\ Precision + Micro - Average\ Recall} \tag{14}$$

The micro-average precision and micro-average recall were 98% and 98%, respectively. Meanwhile, the micro F-measure reached 98%.

5.3.2.3. Overall Vehicle Logo Recognition Evaluation

The overall performance of the developed system for VLR was evaluated by evaluating the two sub-stages (i.e., VLD and VLR (classification)) together as a sequence. The test set from the first dataset was used for evaluation. The text measures for this task were defined as follows:

$$Precision = \frac{the\ number\ of\ correctly\ recognized\ vehicl\ logos}{the\ total\ number\ of\ vehicle\ logos\ positively\ detected} \tag{15}$$

$$Recall = \frac{the\ number\ of\ correctly\ recognized\ vehicel\ logos}{the\ total\ number\ of\ ground\ truths\ (vehicles)} \tag{16}$$

The precision and recall were 95.3% and 99.5%, respectively. Meanwhile, the F-measure reached 97.4%.

5.4. Discussion

The proposed AJLP&VLDR system achieved high performance in terms of text measures (i.e., precision, recall, and F-measure) and mAP measure for detection tasks as shown in Figure 7. But there are some cases where our system went wrong. For LPD task, the LP is not detected for some vehicles. While, for the task of character detection and recognition, there are some vehicles whose LP characters are not all recognized and in other cases in some vehicles the characters have been incorrectly recognized. For both stages together (i.e., LPD and character detection and recognition), in some vehicles the LP was not recognized due to undetected, while in others the LP was detected but incorrectly recognized. For the VLD task, the logo of some vehicles was not detected, while in others the logo was not present but was detected as a logo. For VLR task, some VLs were incorrectly recognized. For both stages together (for example, VLD and VLR), on some vehicles the logo was detected but incorrectly recognized, in other cases the logo was not recognized because it was not detected, and on some vehicles also the logo was not present, but it was detected as a logo and incorrectly recognized.

Moreover, in general, the ALP&VLDR system can be applied for both still images and video streams. Also, it can be applied to images in PNG or JPG formats. Further, in object detection tasks, the true negative from the confusion matrix represents all predicted bounding boxes that do not include objects (i.e., a corrected misdetection). This case is not useful for object detection because of there are a lot of potential true negative in the image which should not be detected (Padilla, Netto and Da Silva, 2020). Our research focuses on applying the proposed algorithm for still images from the collected dataset. Also, it will be applied to images in JPG format. Moreover, the proposed algorithm will be applied to Jordanian vehicles only. Further, in our Jordanian vehicle dataset, all vehicles have LPs. So, true negative metric was ignored.

6. Conclusion

In this research, an efficient ALP&VLDR system for Jordanian vehicles has been developed. As mentioned earlier, AJLP&VLDR system mainly consists of two stages: LPR and VLR. Moreover, these two main stages also consist of two other sub-stages, for the first stage are LPD and character detection and recognition, while for the second stage are VLD and VLR (classification). DL-based CNN using TL was used to accomplish the proposed system. YOLOv3 model was re-trained using TL to accomplish the LPD task, character detection and recognition task, and VLD task. Whereas VGG16 model also was re-trained to accomplish the VLR (classification) task. Since these sub-stages have different tasks, four datasets were collected for training and testing. The first dataset consists of 7,035 Jordanian vehicle images used to train and test the YOLOv3 model for the Jordanian LPD task. The second dataset consists of 7,176 Jordanian LPs used to train and test the YOLOv3 model for CR of Jordanian LPs. The third dataset consists of 8,271 Jordanian vehicle images used to train and test the YOLOv3 model for the VLD task. The fourth dataset consists of 158,230 VL images used to train and test the VGG16 model for the VLR task. The implemented experimental results proved a high performance of our proposed system in terms of precision, recall, F-measure, and mAP.

As a future work, we will develop a system to detect and recognize Jordanian LPs and VLs from a real-time video stream. Moreover, we will expand the VL dataset to include new VLs.

Acknowledgments: The authors would like to thank the Nvidia Learning Center at Yarmouk University (YU), represented by Dr. Ahmad Alomari, for providing access to the GPU Unit for training and testing the CNN Models on all image datasets and getting the results.

Declaration of Study Funding: This work was supported by YU [grant number 2020/39].

References

Ahmed, A.M. (2020) 'Egyptian License Plates Recognition System Using Morphologial Operations and Multi Layered Perceptron', (January).

Albera, S. (2017) 'METHODS A MASTER ' S THESIS IN by SUMIA A . A ALBERA', (April).

Alhaj Mustafa, H., Hassanin, S. and Al-Yaman, M. (2018) 'Automatic Jordanian license plate recognition system using multistage detection', *2018 15th International Multi-Conference on Systems, Signals and Devices, SSD 2018*, pp. 1228–1233. Available at: https://doi.org/10.1109/SSD.2018.8570408.

Ansari, I. and Shim, J. (2019) 'Brief Paper: Vehicle Manufacturer Recognition using Deep Learning and Perspective Transformation', *Journal of Multimedia Information System*, 6(4), pp. 235–238. Available at: https://doi.org/10.33851/jmis.2019.6.4.235.

Anzanello, M.J. and Fogliatto, F.S. (2011) 'Learning curve models and applications: Literature review and research directions', *International Journal of Industrial Ergonomics*, 41(5), pp. 573–583. Available at: https://doi.org/10.1016/j.ergon.2011.05.001.

Asch, V. Van (2013) 'Macro-and micro-averaged evaluation measures [[BASIC DRAFT]].', *Belgium: CLiPS*, pp. 1–27.

Aung, K.P.P., Nwe, K.H. and Yoshitaka, A. (2019) 'Automatic License Plate Detection System for Myanmar Vehicle License Plates', *2019 International Conference on Advanced Information Technologies, ICAIT 2019*, pp. 132–136. Available at: https://doi.org/10.1109/AITC.2019.8921286.

Barman, M. and Patil, A. (2021) 'Transfer Learning for Small Dataset', (March 2019).

Chai, D. and Zuo, Y. (2019) *Anonymization of System Logs for Preserving Privacy*. Springer International Publishing. Available at: https://doi.org/10.1007/978-3-030-03405-4.

Chen, C. *et al.* (2017) 'An effective vehicle logo recognition method for road surveillance images', *2016 2nd IEEE International Conference on Computer and Communications, ICCC 2016 - Proceedings*, pp. 728–732. Available at: https://doi.org/10.1109/CompComm.2016.7924798.

Ding, S. and Wu, H. (2020) 'A Multi-Feature Fusion Based Vehicle Logo Recognition Approach for Traffic Checkpoint', *IOP Conference Series: Earth and Environmental Science*, 440(2), pp. 0–8. Available at: https://doi.org/10.1088/1755-1315/440/2/022071.

Du, K.L. *et al.* (2018) 'Vehicle Logo Recognition Using SIFT Representation and SVM', *Lecture Notes in Networks and Systems*, 16, pp. 928–935. Available at: https://doi.org/10.1007/978-3-319-56991-8_68.

Garbin, C., Zhu, X. and Marques, O. (2020) 'Dropout vs. batch normalization: an empirical study of their impact to deep learning', *Multimedia Tools and Applications*, 79(19–20), pp. 12777–12815. Available at: https://doi.org/10.1007/s11042-019-08453-9.

Grandini, M., Bagli, E. and Visani, G. (2020) 'Metrics for Multi-Class Classification: an Overview', pp. 1–17.

He, K. *et al.* (2016) 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. Available at: https://doi.org/10.1109/CVPR.2016.90.

Huang, D.-S. *et al.* (2018) *Automatic License Plate Recognition Based on Faster R-CNN Algorithm, 14th International Conference, ICIC 2018, August 15-18, 2018*. Springer International Publishing. Available at: https://doi.org/10.1007/978-3-319-95957-3.

Huang, J. *et al.* (2017) 'Speed/accuracy trade-offs for modern convolutional object detectors', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua, pp. 3296–3305. Available at: https://doi.org/10.1109/CVPR.2017.351.

Huang, Z. *et al.* (2019) *Recognition of vehicle-logo based on faster-RCNN*, *Lecture Notes in Electrical Engineering*. Springer Singapore. Available at: https://doi.org/10.1007/978-981-13-1733-0_10.

Izidio, D.M.F. *et al.* (2019) 'An embedded automatic license plate recognition system using deep learning', *Design Automation for Embedded Systems* [Preprint]. Available at: https://doi.org/10.1007/s10617-019-09230-5.

Jagtap, J. and Holambe, S. (2018) 'Multi-Style License Plate Recognition using Artificial Neural Network for Indian Vehicles', *2018 International Conference on Information, Communication, Engineering and Technology, ICICET 2018*, pp. 1–4. Available at: https://doi.org/10.1109/ICICET.2018.8533707.

Kessentini, Y. *et al.* (2019) 'A two-stage deep neural network for multi-norm license plate detection and recognition', *Expert Systems with Applications*, 136, pp. 159–170. Available at: https://doi.org/10.1016/j.eswa.2019.06.036.

Khan, K. *et al.* (2021a) 'Performance enhancement method for multiple license plate recognition in challenging environments', *Eurasip Journal on Image and Video Processing*, 2021(1). Available at: https://doi.org/10.1186/s13640-021-00572-4.

Khan, K. *et al.* (2021b) 'Performance enhancement method for multiple license plate recognition in challenging environments', *Eurasip Journal on Image and Video Processing*, 2021(1). Available at: https://doi.org/10.1186/s13640-021-00572-4.

Kilinc, M. and Uludag, U. (2012) 'Gender identification from face images', (Icoei), pp. 1–4. Available at: https://doi.org/10.1109/siu.2012.6204517.

Lata, K., Dave, M. and K.N., N. (2019) 'Data Augmentation Using Generative Adversarial Network', *SSRN Electronic Journal*, pp. 1–14. Available at: https://doi.org/10.2139/ssrn.3349576.

Li, M. *et al.* (2019) 'A Method of License Plate Recognition Based on BP Neural Network with Median Filtering A Method of License Plate Recognition Based on BP Neural Network with Median Filtering'. Available at: https://doi.org/10.1088/1742-6596/1229/1/012033.

Lin, C.H. and Li, Y. (2019) 'A License Plate Recognition System for Severe Tilt Angles Using Mask R-CNN', *International Conference on Advanced Mechatronic Systems, ICAMechS*, 2019-Augus, pp. 229–234. Available at: https://doi.org/10.1109/ICAMechS.2019.8861691.

Lin, T.Y. *et al.* (2014) 'Microsoft COCO: Common objects in context', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), pp. 740–755. Available at: https://doi.org/10.1007/978-3-319-10602-1_48.

Liu, Y. (2019) 'DigitalCommons @ University of Nebraska - Lincoln Amodal Instance Segmentation and Multi-Object Tracking with Deep Pixel Embedding'.

Lu, Q. *et al.* (2019) 'License plate detection and recognition using hierarchical feature layers from CNN', *Multimedia Tools and Applications*, 78(11), pp. 15665–15680. Available at: https://doi.org/10.1007/s11042-018-6889-1.

Meethongjan, K., Surinwarangkoon, T. and Hoang, V.T. (2020) 'Vehicle logo recognition using histograms of oriented gradient descriptor and sparsity score', *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(6), pp. 3019–3025. Available at: https://doi.org/10.12928/TELKOMNIKA.v18i6.16133.

Nasser, A.B., Alsewari, A. and Zamli, K.Z. (2018) *Smart Toll Collection Using Automatic License Plate Recognition Techniques*. Springer Singapore. Available at: https://doi.org/10.1007/978-981-13-0755-3.

NGUYEN, H. (2020) 'Vehicle logo recognition based on vehicle region and multi-scale feature fusion', *Journal of Theoretical and Applied Information Technology*, 98(16), pp. 3327–3337.

Omar, N. *et al.* (2020) 'Cascaded deep learning-based efficient approach for license plate detection and recognition', 149. Available at: https://doi.org/10.1016/j.eswa.2020.113280.

Padilla, R., Netto, S.L. and Da Silva, E.A.B. (2020) 'A Survey on Performance Metrics for Object-Detection Algorithms', *International Conference on Systems, Signals, and Image Processing*, 2020-July(July), pp. 237–242. Available at: https://doi.org/10.1109/IWSSIP48289.2020.9145130.

Pal, S. and Chawan, P.P.M. (2020) 'Object Detection and Recognition in Satellite map images using Deep Learning', (August), pp. 3–7.

Pham, V.H., Dinh, P.Q. and Nguyen, V.H. (2018) 'CNN-Based Character Recognition for License Plate Recognition System', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10752 LNAI, pp. 594–603. Available at: https://doi.org/10.1007/978-3-319-75420-8_56.

Prusty, M.R., Tripathi, V. and Dubey, A. (2021) 'A novel data augmentation approach for mask detection using deep transfer learning', *Intelligence-Based Medicine*, 5, p. 100037. Available at: https://doi.org/10.1016/j.ibmed.2021.100037.

Pustokhina, I.V. *et al.* (2020) 'Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems'. Available at: https://doi.org/10.1109/ACCESS.2020.2993008.

Rabbani, G. *et al.* (2019) 'Bangladeshi License Plate Detection and Recognition with Morphological Operation and Convolution Neural Network', *2018 21st International Conference of Computer and Information Technology, ICCIT 2018*, pp. 1–5. Available at: https://doi.org/10.1109/ICCITECHN.2018.8631937.

Radzi, F. *et al.* (2019) 'A design of license plate recognition system using convolutional neural network', (June), pp. 2196–2204. Available at: https://doi.org/10.11591/ijece.v9i3.pp2196-2204.

Rahman, M.M.S. *et al.* (2018) 'Bangla License Plate Recognition Using Convolutional Neural Networks (CNN)'.

Ramazankhani, F. and Yazdian-Dehkordi, M. (2019) 'Iranian License Plate Detection using Cascade Classifier', *ICEE 2019 - 27th Iranian Conference on Electrical Engineering*, pp. 1860–1863. Available at: https://doi.org/10.1109/IranianCEE.2019.8786468.

Raza, M.A. *et al.* (2020) 'applied sciences An Adaptive Approach for Multi-National Vehicle License Plate Recognition Using Multi-Level Deep Features and Foreground Polarity Detection Model'.

Redmon, J. *et al.* (2016) 'You only look once: Unified, real-time object detection', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 779–788. Available at: https://doi.org/10.1109/CVPR.2016.91.

Redmon, J. and Farhadi, A. (2018) 'YOLO v.3', *Tech report*, pp. 1–6.

Sachin Prabhu, B., Kalambur, S. and Sitaram, D. (2017) 'Recognition of Indian License Plate number from live stream videos', *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017*, 2017-Janua(06), pp. 2359–2365. Available at: https://doi.org/10.1109/ICACCI.2017.8126199.

Saif, N. *et al.* (2019) 'Automatic License Plate Recognition System for Bangla License Plates using Convolutional Neural Network', *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2019-Octob, pp. 925–930. Available at: https://doi.org/10.1109/TENCON.2019.8929280.

Samadzadeh, A. and Nickabadi, A. (2020) 'RILP : Robust Iranian License Plate Recognition Designed for Complex Conditions'.

Selmi, Z. *et al.* (2020a) 'DELP-DAR system for license plate detection and recognition', *Pattern Recognition Letters*, 129, pp. 213–223. Available at: https://doi.org/10.1016/j.patrec.2019.11.007.

Selmi, Z. *et al.* (2020b) 'DELP-DAR system for license plate detection and recognition', *Pattern Recognition Letters*, 129, pp. 213–223. Available at: https://doi.org/10.1016/j.patrec.2019.11.007.

Shafi, I. *et al.* (2021) 'License plate identification and recognition in a non-standard environment using neural pattern matching', *Complex & Intelligent Systems* [Preprint]. Available at: https://doi.org/10.1007/s40747-021-00419-5.

Shorten, C. and Khoshgoftaar, T.M. (2019) 'A survey on Image Data Augmentation for Deep Learning', *Journal of Big Data*, 6(1). Available at: https://doi.org/10.1186/s40537-019-0197-0.

Simonyan, K. and Zisserman, A. (2015) 'Very deep convolutional networks for large-scale image recognition', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14.

Slimani, I. *et al.* (2020) 'An automated license plate detection and recognition system based on wavelet decomposition and CNN', *Array*, 8(September), p. 100040. Available at: https://doi.org/10.1016/j.array.2020.100040.

Smith, L.N. (2018) 'A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay', pp. 1–21.

States, U. (2006) '(12) Patent Application Publication (10) Pub. No.: US 2006/0284895 A1', 1(19).

Tang, Y. *et al.* (2017) 'Vehicle detection and recognition for intelligent traffic surveillance system', *Multimedia Tools and Applications*, 76(4), pp. 5817–5832. Available at: https://doi.org/10.1007/s11042-015-2520-x.

Tourani, A. *et al.* (2020) 'A robust deep learning approach for automatic Iranian vehicle license plate detection and recognition for surveillance systems', *IEEE Access*, 8, pp. 201317–201330. Available at: https://doi.org/10.1109/ACCESS.2020.3035992.

Visa Sofia, D. (2011) 'Confusion Matrix-based Feature Selection Sofia Visa', *Visa,Sofia Ramsay,Brain Ralescu,Anca van der Knaap,Esther*, 710(January), p. 8.

Wang, Q. (2018) 'License plate recognition via convolutional neural networks', *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 2017-Novem, pp. 926–929. Available at: https://doi.org/10.1109/ICSESS.2017.8343061.

Wang, W. (2017) 'License plate recognition system based on the hardware acceleration technology on the ZYNQ', *Proceedings of 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2017*, pp. 2679–2683. Available at: https://doi.org/10.1109/IAEAC.2017.8054512.

Yang, S. *et al.* (2020) 'Vehicle Logo Detection Based on Modified YOLOv2', pp. 75–86. Available at: https://doi.org/10.1007/978-3-030-17763-8_8.

Yao, L. *et al.* (2019) 'Research and Application of License Plate Recognition Technology Based on Deep Learning Research and Application of License Plate Recognition Technology Based on Deep Learning'. Available at: https://doi.org/10.1088/1742-6596/1237/2/022155.

Ying, T., Xin, L. and Wanxiang, L. (2018) 'License plate detection and localization in complex scenes based on deep learning', *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, pp. 6569–6574. Available at: https://doi.org/10.1109/CCDC.2018.8408285.

Ying, X. (2019) 'An Overview of Overfitting and its Solutions', *Journal of Physics: Conference Series*, 1168(2). Available at: https://doi.org/10.1088/1742-6596/1168/2/022022.

Yousef, K.M.A. *et al.* (2020) 'Automatic license plate detection and recognition for jordanian vehicles', *Advances in Science, Technology and Engineering Systems*, 5(6), pp. 699–709. Available at: https://doi.org/10.25046/aj050684.

Yu, Y. *et al.* (2018) 'Vehicle logo recognition based on overlapping enhanced patterns of oriented edge magnitudes', *Computers and Electrical Engineering*, 71(July), pp. 273–283. Available at: https://doi.org/10.1016/j.compeleceng.2018.07.045.

Yu, Y. *et al.* (2021) 'A Cascaded Deep Convolutional Network for Vehicle Logo Recognition from Frontal and Rear Images of Vehicles', *IEEE Transactions on Intelligent Transportation Systems*, 22(2), pp. 758–771. Available at: https://doi.org/10.1109/TITS.2019.2956082.

Zhang, J. *et al.* (2017) 'Visual saliency-based vehicle manufacturer recognition using autoencoder pre-training deep neural networks', *IST 2017 - IEEE International Conference on Imaging Systems and Techniques, Proceedings*, 2018-Janua, pp. 1–6. Available at: https://doi.org/10.1109/IST.2017.8261506.

Zhang, J. *et al.* (2021) 'Multi-Scale Vehicle Logo Detector', *Mobile Networks and Applications*, 26(1), pp. 67–76. Available at: https://doi.org/10.1007/s11036-020-01722-0.

Zhang, X. *et al.* (2018) 'Vehicle license plate detection and recognition using deep neural networks and generative adversarial networks', *Journal of Electronic Imaging*, 27(04), p. 1. Available at: https://doi.org/10.1117/1.jei.27.4.043056.

Zhao, W. (2017) 'Research on the transfer learning of the vehicle logo recognition', *AIP Conference Proceedings*, 1864(August). Available at: https://doi.org/10.1063/1.4992875.

siekierzynski, k., 2017. *CarL-CNN.* [Online] Available at: https://github.com/kuba-siekierzynski/CarL-CNN [Accessed 3 April 2021].

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.