

Article

An Automated Precise Authentication of Vehicles for Enhancing the Visual Security Protocols

Kumarmangal Roy ¹, Muneer Ahmad ², Norjihhan Abdul Ghani ¹, Jia Uddin ³ and Jungpil Shin ^{4,*}

¹ Department of Information Systems, Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur 50603, Malaysia; roy.kumarmangal@gmail.com (K.R.)

² Department of Human and Digital Interface, Woosong University, Daejeon 34606, Republic of Korea

³ Artificial Intelligence and Big Data Department, Woosong University, Daejeon 34606, Republic of Korea; jia.uddin@wsu.ac.kr

⁴ School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Fukushima, Japan

* Correspondence: jpshin@u-aizu.ac.jp

Abstract: The movement of vehicles in and out of the predefined enclosure is an important security protocol that we encounter daily. Identification of vehicles is a very important factor for security surveillance. In a smart campus concept, thousands of vehicles access the campus every day, resulting in massive carbon emissions. Automated monitoring of both aspects (pollution and security) are an essential element for an academic institution. Among the reported methods, the automated identification of number plates is the best way to streamline vehicles. The performances of most of the previously designed similar solutions suffer in the context of light exposure, stationary backgrounds, indoor area, specific driveways, etc. We propose a new hybrid single-shot object detector architecture based on the Haar cascade and MobileNet-SSD. In addition, we adopt a new optical character reader mechanism for character identification on number plates. We prove that the proposed hybrid approach is robust and works well on live object detection. The existing research focused on the prediction accuracy, which in most state-of-the-art methods (SOTA) is very similar. Thus, the precision among several use cases is also a good evaluation measure that was ignored in the existing research. It is evident that the performance of prediction systems suffers due to adverse weather conditions stated earlier. In such cases, the precision between events of detection may result in high variance that impacts the prediction of vehicles in unfavorable circumstances. The performance assessment of the proposed solution yields a precision of 98% on real-time data for Malaysian number plates, which can be generalized in the future to all sorts of vehicles around the globe.

Keywords: security protocol; automated data collection; computer vision; vehicle identification; deep learning



Citation: Roy, K.; Ahmad, M.; Ghani, N.A.; Uddin, J.; Shin, J. An Automated Precise Authentication of Vehicles for Enhancing the Visual Security Protocols. *Information* **2023**, *14*, 466. <https://doi.org/10.3390/info14080466>

Academic Editors: Danilo Avola, Daniele Pannone and Alessio Fagioli

Received: 21 July 2023

Revised: 8 August 2023

Accepted: 12 August 2023

Published: 18 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the ever-growing vehicle population and growth in auto sectors, managing and monitoring vehicles manually is tiresome, and costly building solutions are also not acceptable [1]. Thus, moving towards an intelligent transport system is essential. Smart vehicle management systems and intelligent transportation systems require automated vehicle identification architectures. This is a powerful tool for traffic management, electronic authentication at toll gates, vehicle monitoring for law enforcement, commercial transportation, access control, etc. The principal working architectures for all these applications require uniquely identifying each vehicle. To uniquely identify each vehicle there are a few components, like the number plate, vehicle identification number (VIN), and owner's details. Every vehicle should be equipped with a device that can emit all this necessary information and feed it to different checkpoint receivers. However, identifying

and reading the owner's details raises concerns about data privacy. Furthermore, it will not be practically feasible for every vehicle to carry such an information-emitting device.

Manual inspection of vehicles and managing a database of such vehicles is tedious. This kind of vehicle management is inconvenient and time-consuming, and also requires the physical intervention of security personnel. A contactless option for a smart vehicle management system is more suitable [2]. License plate recognition systems are an important part of an intelligent transportation system and are used in traffic management, electronic toll collection, access control, security purpose, managing parking, etc. [3].

There has been quite a good advancement in object detection algorithms and processes due to enhancements in the field of artificial intelligence and deep learning [4]. The most common issue that exists in the automated number plate detection domain is the detection of the candidate region. Determining the region of interest at a fast pace to keep up with traffic/vehicle movement is always a concern. Most classical detection systems work by repurposing classifiers to detect desired object classes. This means that the model is applied at multiple locations across the image and varied scales. Most high-scored values are returned as detection. This results in slower processing. Multi-oriented, multi-directional detection and recognition fail in many cases [5]. Low light and poor resolution also lower the detection accuracy rate [6]. Furthermore, most of the existing ANPR systems suffer similarity issues between certain set characters during optical conversion [7].

For real-life object detection, speed is of the utmost importance. In real-life scenarios, number plate identification should be performed quickly as it is impractical to have a system that may hinder normal traffic flow. Slower processing will lead to more power consumption and added cost. Low light and poor resolution may lead to unsuccessful number plate detection [8,9] as well. Additionally, incorrect alphanumeric representations of number plates also raise security concerns [10]. There can be a possibility of incorrect detection of characters in this situation.

To overcome the issue of applying the model at multiple locations of the image with multiple scales across the image, the Haar-like features are computationally cheap due to the exploitation of line features, edge features, and rectangle features as the area of interest [11]. The features are processed through a convolution network and Gentle AdaBoost (GAB). MobileNet is an architecture that uses depthwise separable convolutions to construct a light CNN, which works well for embedded vision applications [12]. The SSD algorithm, instead of using a sliding window like in the case of a convolution network, divides each frame into a grid, and every individual grid cell is made responsible for detecting the class of objects in that region of the frame [13]. Image processing, contour finding, edge detection, and masking techniques are applied to candidate retrieval [14]. These steps enable the system to process images/frames of poor resolution under low light.

Mostly, the automatic recognition of license plates is employed for security risk mitigation to ban non-registered vehicles from entering the safer zones. Machine learning and deep learning methods have been found to be quite useful in the identification and recognition of objects in certain scenarios. Automatic identification and recognition of number plates of vehicles make the process hassle-free, save time and fuel, and help in the prediction of other side measures e.g., traffic management and counts of incoming and outgoing vehicles at periodic cycles. The manual inspection of vehicles in a predefined area, like an academic campus, is tedious. The campus security officials in most cases manually manage and monitor the vehicles at checkpoints and entry gates. This research proposes a smart vehicle authentication system based on automatic scanning and recognition of licensed plates of vehicles entering the campus. We mount security cameras at the security checkpoints of the campus to help the security officers register new legitimate vehicles, update the data of existing vehicles, assign warnings to illegitimate or suspect vehicles, and calculate the statistics related to the vehicles' access history. These statistics help the authorities estimate the carbon emissions to reduce carbon pollution for the pursuit of a sustainable campus.

This research study has the following contributions:

- We developed a single-shot detection system to enhance the real-time detection of vehicle objects, enabling us to obtain predictions as per the global context of the image-frame.
- We integrated lightweight Haar cascade and MobileNet-SSD to reduce the computational complexity involved in the real-time detection and identification of vehicles.
- We adopted a new character recognition approach to reduce incorrect recognition of characters.
- We reduced the percentage of error by adopting regex matching to reduce characters' dissimilarity to seek exact alphanumeric patterns.
- We significantly achieved 98% prediction accuracy under even unfavorable weather conditions.

2. Literature Review

Automatic license plate recognition uses two fundamental parts, namely license plate extraction and optical character recognition on frames to fetch license plate numbers. It has a wide range of applications and is used in traffic management, electronic toll collection, access control, security purpose, vehicle identification, managing parking, etc. [15]. With the upgrade of systems and the use of AI in our natural surroundings, automatic toll collection, automated parking systems, and security checks, researchers have dived in to find easy and possible techniques to smoothen automotive traffic while at the same time prioritizing security instances. Useful knowledge extraction can support decision-making and cost reduction in intelligent transportation systems.

With a growing number of vehicles around the globe [16], an automatic and robust authentication system would be required to manage and monitor vehicles. An intelligent traffic system can help us to curb wrong-side driving, reckless driving, and vehicle authorization [17–22]. With the advancement of technology, both in hardware and software, it is now possible to make smart, automated, intelligent traffic systems. There are several use cases for an intelligent traffic system that includes ANPR systems, car-to-car communication, GPS-based systems, smart traffic lights, etc. In particular, the key part of any intelligent traffic system is license plate recognition [23–27].

Thanks to deep learning and artificial intelligence, there has been some serious growth [28]. With good growth in computer vision, we can achieve a better intelligent transport system. Refs. [15,29] adopted the deep learning approaches, such as SSD-mobile. Efficient models for mobile and embedded vision applications, such as MobileNet, are applicable for intelligent traffic management [12]. In deep learning, SSD-MobileNet achieved high average precision of 99.76% for a car class, 97.76% for a person class, and 71.07% for a class chair [30]. A comparison of different object detection techniques on the car number plate, namely SSD-Mobile, Resnet, a faster regional convolution neural network (R-CNN), and a region-based fully convolutional network (RFCN) was studied by [13]. ANPR with convolution networks on a single pass is another method for ANPR [5]. Different approaches, such as YOLO, K-means with segmentation, CNN with graphic boards, and ant colony optimization, were studied for car plate recognition [6,23,31,32], respectively.

This research focuses on the performance comparison of deep learning models, such as SSD-MobileNet and the Haar cascade classifier along with the OCR module for character recognition. Table 1 presents a summary of selected 40 prior studies. For a fair comparison, different technological adoptions along with research outcomes and limitations of the respective studies were noted.

Table 1. Summarization of selected prior studies.

Ref.	Technology Adoption	Research Outcomes	Limitation
Darapaneni, et al. [1]	YOLO V3, OpenCV, PyTesseract	Accuracy: 100% for NPR	Training data: 300 images/videos Test data: 20 images
Mangal, et al. [2]	ANPR, Aurdino Uno, Raspberry Pi, IR sensors, QR code reader	Accuracy: 70%	Low accuracy
Wang, et al. [3]	Cascaded CNN for object detection, recurrent NN, connectionist temporal classification (CTC) for character recognition	Accuracy: 98%	Limited to the Chinese number plates
Sharma, et al. [4]	Haar cascade classification for region detection. Py-Tesseract as the optical character module	Accuracy: 90%	Certain characters read incorrectly during the character recognition phase
Li, et al. [5]	Convolution network, pooling	Detection rate: 99.73%	Problems with multi-oriented license plates
Xie, et al. [6]	Convolution neural network, YOLO algorithm for detection.	Accuracy: 98.6%	Low light and poor resolution.
Babu, et al. [7]	LPD, character segmentation and recognition	Accuracy: 93.3%	Not able to detect blurry images and broken number plates
Yulianto, et al. [8]	Haar cascade classifier, scale invariant feature transformation (SIFT), RFID-E-KTP	Accuracy: 89%	Processing time at entry and exit seemed a bit long
Shariff, et al. [9]	Bilateral filtering, canny edge detection, Py-Tesseract	Accuracy: 88%	The detection worked for number plates with white backgrounds
Cheon, et al. [11]	Haar cascade, DoG filter, connected component labeling, histogram, color quantization	Precision: 96%	A real-time performance study was not conducted
Peker [13]	SSD with MobileNet and Resnet50 features, inception layer features on faster R-CNN, R-FCN with Resnet101 features	Accuracy: 97.9%	Small-size number plates were not detected with SSD architecture. This study was specific to Turkey
Aggarwal, et al. [14]	The smart detection system, image segmentation, edge processing, region of interest	Detection rate: 93.34%	Camouflage stickers around the number plate
Zhang, et al. [15]	CNN, R-CNN, bi-directional LSTM	Average precision: 97.11%	Degradation for tilt angle of the characters
Varkentin, et al. [16]	YOLO, CNN	Recognition rate: 86%	The model was trained for Russian number plates
Goyal, et al. [18]	IR proximity sensors, Arduino Uno Controller (ATmega328P), Raspberry Pi, IP camera, DMD display, SSD-MobileNet, optical character recognition	Accuracy: 74.3%	IR proximity sensors have a detection range of 85 cm maximum. Any vehicle which moves beyond this range will not be detected
Srivastava, et al. [19]	Deep learning architecture, ResNet, feature pyramid network	Accuracy: 92.6%	A slight dip during the live case scenario
Mandi, et al. [20]	Optical character recognition (OCR), ANPR, character segmentation, character recognition	Accuracy was not given	OCR may not convert characters to very large or small font sizes
Babu, et al. [21]	YOLO algorithm with a combination of CNN	Recognition rate: 91.0%	Problem between characters, like 0 and O
Sasi, et al. [23]	Ant colony optimization, Kohonen neural network, SVM for character classification	Accuracy: 94%	NN with other classification algorithms for character detection was not tested

Table 1. Cont.

Ref.	Technology Adoption	Research Outcomes	Limitation
Zhao, et al. [24]	Haar cascade classifier for detecting license plate location, Adaboost classifier to train from several features	True positive rate: 85.54%.	The study was conducted on still images and only for Chinese license plates
Agarwal, et al. [25]	Edge detection, plate extraction, character segmentation	Accuracy was not given	Limited to Indian car number plates
Lib, et al. [26]	CNN, plate/non-plate CNN classifier, RNN with LSTM	Precision: 97.56%	Slow detection speed of multi-scale sliding window compared to real-life application
Silva, et al. [27]	License plate, deep learning, CNN, OCR	Average accuracy: 89.33%	Cannot detect motorcycle license plates
Zhang, et al. [29]	MobileNet with single-shot detection, image augmentation	Accuracy: 95%	Low-level feature maps are usually not that good for predicting small targets of high-level features
Younis, et al. [30]	MobileNet, single-shot detection	Average precision: 99.76%	The dataset had limited object classes
Stefanović, et al. [31]	Data partitions, digital image, K-means algorithm, edge detector	Accuracy was not given	The model performance was judged based on restricted parameters that are specific to Serbia
Lee, et al. [32]	NVIDIA Jetson TX1 boards, CNN, embedded System	Recognition rate: 95.24%	Fairly small dataset was used. The model was unable to detect broken and reflective number plates
Viola et al. [33]	Integral image, AdaBoost, complex cascading	Detection rate: 15 frames per second	Limitatons were not given.
Nguyen, et al. [34]	Convolution neural network	Precision: 98.5%	Only works on Vietnamese number plate fonts
Omran, et al. [35]	OCR, template matching, character segmentation	Recognition Rate: 85.7%	Fairly small dataset was used and the model was region-specific to Iraq
Balaji, et al. [36]	License plate region recognition, segmentation, noise removal, bounding box, and filter process	Accuracy was not given	Not able to detect blurry images, broken number plates, and character similarities
Tiwari, et al. [37]	ANPR, computer recognition system, OCR, vehicle number plate, Matlab	Accuracy was not given	The system was designed for a specific region
Yogheedha, et al. [38]	Image processing, image segmentation, optical character recognition	Accuracy: 92.85%	Small datasets
Soon, et al. [39]	Adaboost and connected component analysis for plate detection. KNN for character recognition	Accuracy: 96.84%	KNN character classifier is not accurate and robust enough
Zakaria, et al. [40]	Template matching, top hat filtering, contrast correction method, color information method	Accuracy: 97.1%	Character segmentation and character recognition were not included
Priya, et al. [41]	Edge detection, character segmentation, filtering techniques, optical character recognition	Accuracy: 75%	A very small dataset was tested
Du, et al. [42]	Image acquisition and processing, license plate segmentation with edge detection algorithm, NN or template matching for character recognition	Accuracy was not given	Limitations in detecting multi-style plate design and video-based feed

The existing literature shows a variety of approaches which include mostly CNN-based approaches, template matching techniques, computer recognition systems, Adaboost, and connected component analysis for license plate detection. For character recognition, OCR is used in most cases. Similarly, neural networks have been combined with template matching, K-nearest neighbor has been adopted, and connectionist temporal classification for character recognition has also been used. Most of the prior studies result in moderate prediction accuracy, especially under unfavorable weather conditions, and computational complexity. A few studies also adopted different variations of the YOLO algorithm to speed up the processing power capabilities of authentication systems. In addition, earlier convolution models were also applied to multiple locations and scales of images that consumed much of the processing time. Moreover, in most cases, the existing research focused the prediction accuracy, which in most state-of-the-art methods (SOTA) is very similar. Thus, the precision among number of use cases is also a good evaluation measure that was ignored in the existing research. It is evident that the performance of prediction systems suffer due to adverse weather conditions, e.g., foggy and rainy seasons, inadequate light exposure, distance of vehicle from cameras, and other similar limitations. In such cases, the precision between events of detection may result in high variance that impacts the prediction of vehicles in unfavorable circumstances.

3. Methodology

We are proposing a novel hybrid approach for automated number plate recognition (ANPR) that combines Haar cascade object detectors, MobileNet-SSD, and a new optical character reader mechanism for character identification on number plates. There are three broad segments when it comes to segmenting ANPR.

- Object Detection using a Haar cascade and MobileNet-SSD:
 - In this segment, the proposed approach utilizes both Haar cascade object detectors and MobileNet-SSD as the base network for object detection in the ANPR system.
 - Haar cascade object detectors are used for their ability to detect specific patterns and features in images, while MobileNet-SSD is employed for its lightweight design and fast inference.
 - The fusion of these two detectors provides improved accuracy and speed in locating the license plates in the input images.
- Character identification using optical character recognition (OCR):
 - After locating the license plate in the previous segment, this segment focuses on character identification and segmentation from the license plate region.
 - The proposed approach introduces a new optical character reader mechanism that efficiently extracts characters from the license plate using OCR techniques.
 - The OCR mechanism is designed to handle variations in fonts, styles, and image conditions for robust character recognition.
- Automated number plate segmentation:
 - This segment involves the final step of the ANPR process, where the individual characters recognized in the previous step are segmented to obtain the complete license plate number.
 - The proposed approach uses various techniques, such as contour analysis, bounding box extraction, or character grouping to achieve accurate segmentation of the license plate number.

By explicitly defining and describing the three segments, we provide a clear structure for our proposed hybrid approach, as shown in Figure 1.

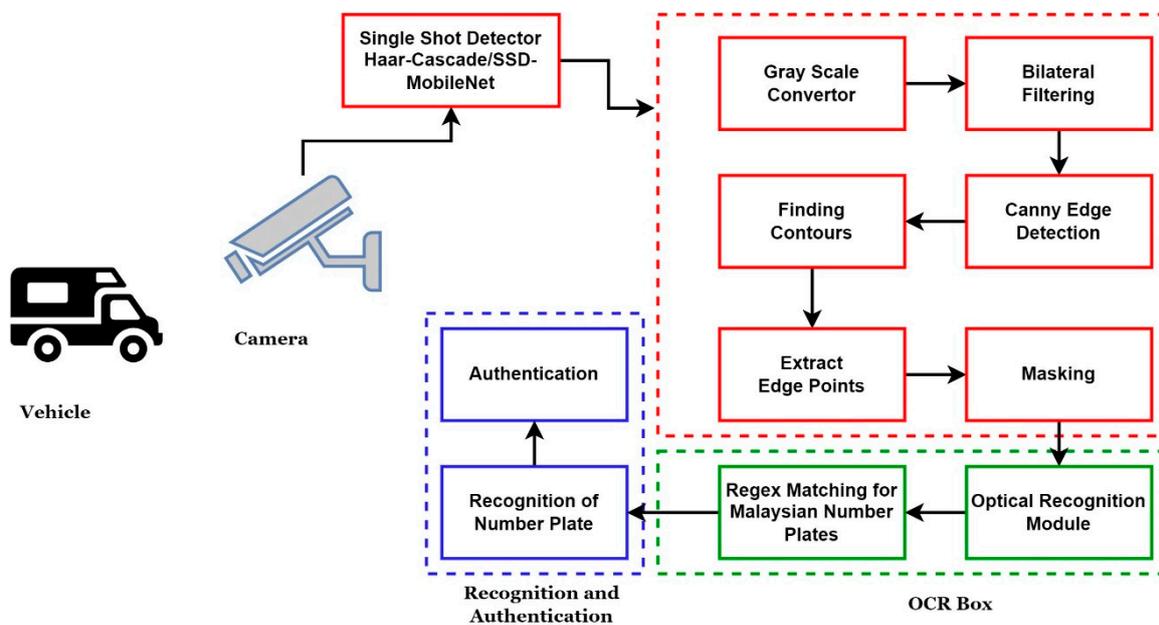


Figure 1. Single-shot detector.

The details of each component are given below.

3.1. Haar Cascade

The main object of the ANPR system is to detect candidate regions, and the utmost requirement to do so is processing speed. As we are aiming to detect number plates from moving vehicles, the proposed system should be able to perform this at a great speed. We also need to focus on the complexity involved with moving vehicles, since the frame size will keep changing with a lot of variations in the candidate region, as well as changes in direction.

In a Haar cascade, the orientation may change as needed but the base features remain the same. Each of the rectangles can be a pixel or even part of the frame, which thus becomes scalable. Haar features are computed using Haar filters. Haar filters are again based on Haar wavelets, which are the square function. Since in Haar, all the filters have only two values, it is either black or it is white, and this makes it computationally cheap during operations. For each one of the Haar filters, we apply it as a correlation on the image, which provides us with a number at each pixel, as shown in the following Equation (1):

$$[H]a = v_a[i, j] \tag{1}$$

where $[H]a$ represents the Haar filter. Every scale will have its feature vectors. Every white is equal to +1 and every black is equal to -1. Equation (2) shows the response to the Filter H_a at the location, pixel value (i, j) , as follows:

$$v_a[i, j] = \sum_m \sum_n I[m - i, n - j] H_a[m, n] \tag{2}$$

Equation (2) can be simplified by writing it like Equation (3) as shown:

$$v_a[i, j] = \sum(PW) - \sum(PB) \tag{3}$$

where PW and PB represent the pixel intensities in the white and black areas, respectively.

3.2. Mobile Architecture

MobileNet object detection component provides us with a class index of the objects along with their respective prediction probabilities and bounding box edges. We filter

out our desired class objects with a more than 0.65 prediction probability. For object detection, we are implementing SSD-MobileNet. SSD single-shot multi-box detection is a neural network architecture designed to detect object classes, which means the extraction of bounding boxes and object classification occurs in one go. The author of this novel study has mentioned the use of the VGG neural network as the base network, which is the feature extractor for detection [43], on top of SSD architecture. Therefore, two types of deep neural network, namely a combination of a base network and detection network, are being used here. The high-level features for classification objects come from the base network. To reduce the model size and complexities, depthwise separable convolution is used. It is used for classification purposes, to extract features for detection. As shown in Figure 2, we can see that depthwise convolution is followed by pointwise convolution.

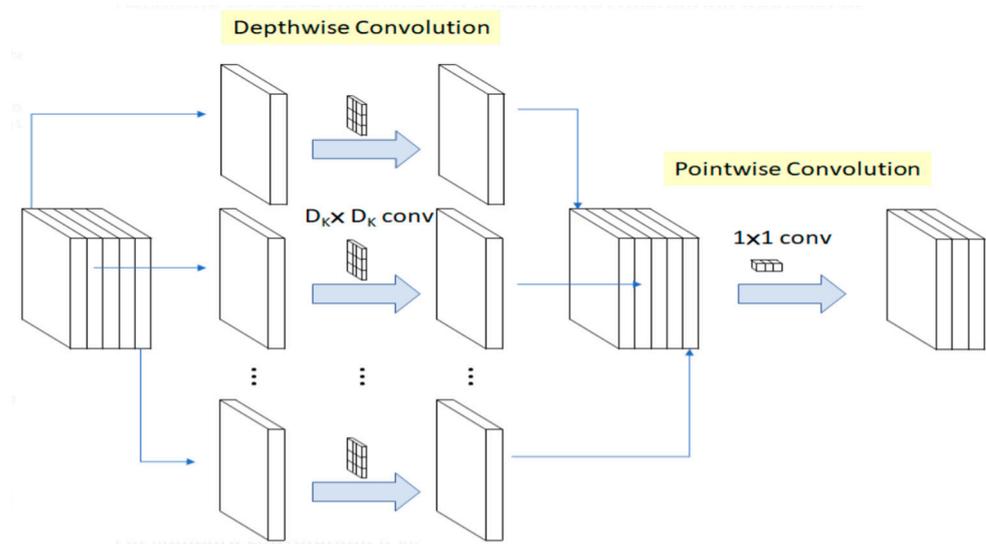


Figure 2. Depthwise separable convolution.

Depthwise convolution is the channel-wise $D_k \times D_k$ spatial convolution. For 5 channels, we have $5 \times D_k \times D_k$ spatial convolution. Pointwise convolution is always 1×1 Conv. This is to change the dimension of the channel. Convolution is always a product of two functions that produces a third function which can be seen as an altered version of the first function. Let us suppose that we have two functions, j , and k . The second function, k , is considered the filter. A spatial convolution operation is when the f function is defined on a variable (spatial), like x , rather than a time (t). Convolution for functions $j(x)$ and $k(x)$ on a variable x is shown in Equation (4).

In the case of images, the functions are always of two variables. In image processing, the image made by the lens is always a continuous function. Thus, the function happens to be $j(x, y)$. In those cases, a smoothing filter is applied, $k(x, y)$ and we obtain Equation (5) as follows:

$$j(x) * k(x) = \int_{-\infty}^{\infty} j(\tau) \cdot k(x - \tau) d\tau \tag{4}$$

$$j(x, y) * k(x, y) = \int_{\tau_1 = -\infty}^{\infty} \int_{\tau_2 = -\infty}^{\infty} j(\tau_1, \tau_2) \cdot k(x - \tau_1, y - \tau_2) d\tau_1 d\tau_2 \tag{5}$$

where $*$ represents the convolution operation.

By referring to Figure 2, we can find the equation of depthwise convolution and pointwise convolution. As shown in Expression (6), the first part is referring to depthwise operation and the second part is referring to pointwise operation. Here, M indicates the number of in channels, N represents the number of out channels, D_k is the kernel size and D_f is the feature map size [12].

$$D_k \cdot D_k \cdot M \cdot D_f \cdot D_f + M \cdot N \cdot D_f \cdot D_f \tag{6}$$

$$D_K.D_K.M.N.D_F.D_F \tag{7}$$

Expression (7) denotes standard convolution operation. When we divide Expression (6) by Expression (7), we obtain the total computational/operational reduction. When we apply architecture for a 3×3 kernel size, we can achieve 8–9 times less computation by this method.

Single-shot detection has two components, including a base network, which in our case is MobileNet, for extracted feature mapping. Another component is to apply a convolution filter to detect objects [43].

MobileNet provides us with all the required feature maps. The six-layer convolution network performs the classification detection of the object class. The architecture can make 8732 predictions per object class. The architecture checks the confidence score of each box of 8732 predictions per object class and will pick only the top 200 predictions from the image. This is handled with the help of non-max suppression which selects a few entities out of overlapping entities based on probability. The calculation for reference is provided. At Conv4_3, it is of size $38 \times 38 \times 512$.

A 3×3 kernel convolution is applied. There are 4 bounding boxes, and each bounding box will have (classes + 4) outputs. Classes cannot be zero. In at least one class, the background is always included. Thus, at Conv4_3, the output is $38 \times 38 \times 4 \times (\text{class} + 4)$. Let us suppose that there are ten object classes. By default, one background class is always inclusive. Then, the output will be $38 \times 38 \times 4 \times (10 + 1 + 4) = 86,640$. Thus, for the bounding box calculation, Table 2 presents the total number of bounding boxes (that add up to 8732) which is our prediction probability per object class. It means that we require 4 to 6 bounded boxes for the detection of the object in the image.

Table 2. Prediction probability per object class.

Convolution Type	Dimension	No. of Bounding Boxes
Conv4_3	$38 \times 38 \times 4 = 5776$	4 boxes for each location
Conv7	$19 \times 19 \times 6 = 2166$	6 boxes for each location
Conv8	$10 \times 10 \times 6 = 600$	6 boxes for each location
Conv9	$5 \times 5 \times 6 = 150$	6 boxes for each location
Conv10	$3 \times 3 \times 4 = 36$	4 boxes for each location
Conv11	$1 \times 1 \times 4 = 4$	4 boxes for each location

In object detection, the architecture not only predicts object classes but also locates their bounding boxes. The subtle difference between object classification and object detection is that in the case of object classification, the prediction is about if the class of object is present or not. But for object detection, not only does it provide prediction probability of the object class but also the boundary location of the object. Instead of using normal sliding window operation for convolution networks, a single-shot detector divides the image into multiple regions, like a grid. Every individual grid is tasked with detecting objects in that particular region of an image. If there is no object detected in a particular grid, then we consider it null, and that particular location is ignored.

There are possibilities of certain scenarios where there are many objects in a single grid or there are multiple objects of different sizes and shapes that need to be detected. To handle this, we have an anchor. These are also known as ground truth boxes. While training, the matching phase anchors boxes are connected to the bounding boxes of every ground truth object within a frame. Anchor boxes are predefined, precalculated, fixed regions of probable space, and approximate box predictions. The highest region of overlap with the anchor box determines the object’s class, its probability, and its location. This is the base principle of training the network and predicting the detected class of objects and their locations. In practice, each anchor box is specified by an aspect ratio. Aspect

ratios of anchor boxes are pre-defined in single-shot detection architecture. This allows for accommodating different sizes and shapes of objects.

Once we have the desired frame with our intended class object, we have to find out our candidate region and number plate section. The series of steps that are required to extract the number plate region is shown in Figure 3.

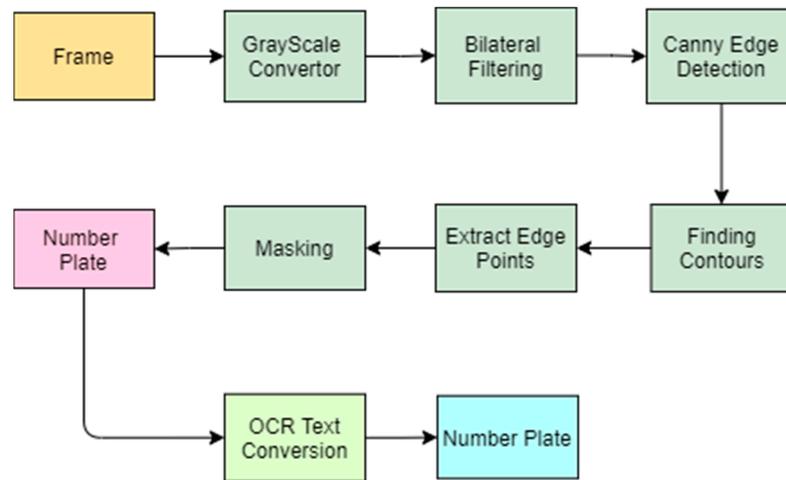


Figure 3. Number plate extraction.

In Figure 4, we can see our reference image. We need to convert the frame to grayscale, as shown in Figure 5. This is very important as the consequent steps cannot be performed without this.



Figure 4. Original car image.

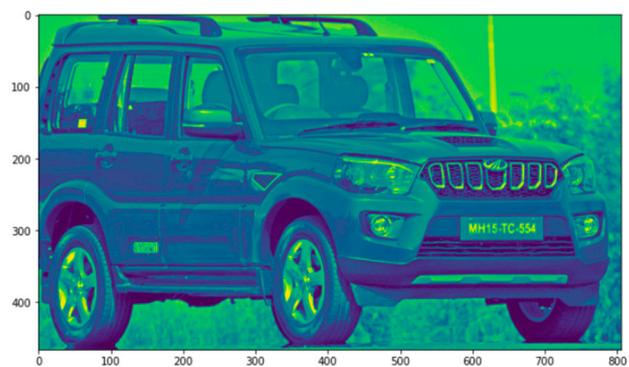


Figure 5. Grayscale converted frame.

Next, we proceed with the bilateral filtering process as shown in Figure 6. A bilateral filter is used for reducing noise and smoothing images. Other denoising filters are present, like the average filter, median filter, and Gaussian filter. The bilateral filter retains the edges while the Gaussian filter uniformly blurs the content and edges together.



Figure 6. Bilateral filtered frame.

A canny edge detector is used for detecting edges as shown in Figure 7. A prerequisite for a canny edge is to have a noise-reduced image, which we already have from bilateral filtering. The canny edge detector points out all the edges with the help of non-maximum suppression as an edge is a sharp difference and change in a frame's pixel values.

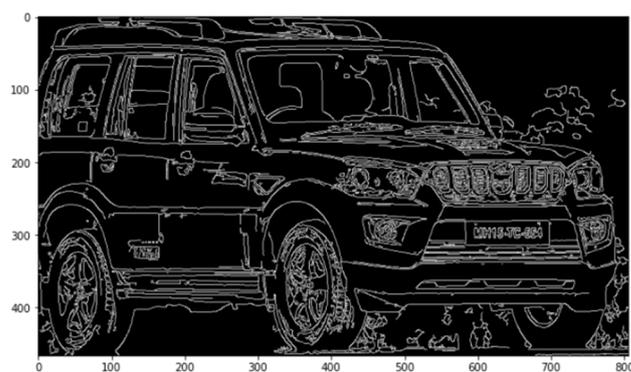


Figure 7. Canny edge detection.

We find contours which are curves that are joining all the points, continuous in nature, along the edges of the same color intensity. When we apply a find-contour operation the original frame will become impacted. A copy of the image should be processed during this phase. This function is to detect objects in an image. Sometimes objects are separated and located at different places without any overlapping. But in some cases, many of the objects overlap with one another. The outer edge is the parent, and the inner edge is referred to as the child.

The relationship between contours in an image is established like this. Due to the relationship between parent and child, we can call this a hierarchy. Retr_tree is passed as contour retrieval mode which provides the total hierarchy of all the edges in a tree hierarchy-like structure. The contour approximation method needs to also be specified in this step, otherwise it will not be able to judge the (x, y) coordinates of the boundary points of any shape. We use chain_approx_simple to return only the two endpoints of a line; otherwise, it will return all the point coordinates in a line.

Now, we approximate polygons. The target here is to fetch the number plate region, which is always rectangular. This step helps us to fetch the polygonal-shaped edges or contour points. The rectangle is a four-sided polygon. Thus, any approximation which

returns four sides can be our possible edge point for the number plate. Edge points returned are the (x, y) coordinates of four corners of a rectangle.

Masking is the process applied before we draw the contours of the new edge points that were approximated. Once the original image is masked and only the candidate region is passed, we obtain our desired number plate output, as shown in Figure 8.

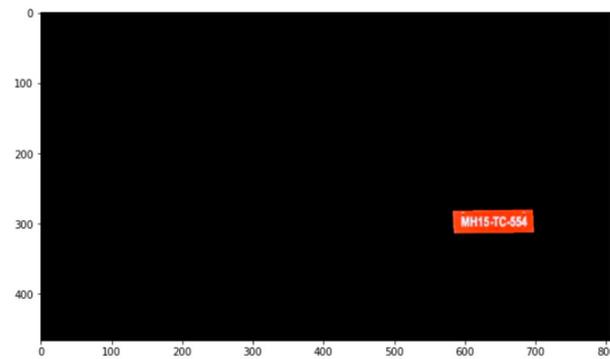


Figure 8. Masked edge points.

Now we have our desired area. We segregate the number plate region from the image for further processing, as shown in Figure 9.



Figure 9. Number plate.

For character recognition from the segregated license plate, we use EasyOCR. This is a library for optical recognition. EasyOCR is based on deep learning models which involve text recognition and detection models. This optical character recognition process returns text along with its prediction probability. We have deployed a regex conditioning which helps us to format all the text as per Malaysian car number plate formats. Different car number plate formats were designed into regex formats to validate the text conversion. Most Malaysian number plates follow a format of Sxx @@@@, where:

S—The state or territory prefix. (e.g.,: W = Kuala Lumpur, A = Perak etc.);

X—The alphabetical sequences. (e.g.,: A, B, C, . . . , X, Y);

@—The number sequence (0, 1, 2, 3 to 9999).

Please find below Table 3 for state references.

Table 3. State prefix for Malaysia.

State	Prefix	State	Prefix
Kuala Lumpur	W	Kedah	K
Penang	P	Pahang	C
Terengganu	T	Johor	J
Malacca	M	Perak	A
Negeri Sembilan	N	Selangor	B
Perlis	R	Kelantan	D

The number plate format being used for Sarawak is QDx @@@@ x, where:

Q—The constant prefix for all Sarawak number plates;

D—The division prefix. (e.g.,: A = Kuching, M = Miri);

x—The alphabetical sequences. (e.g.,: A, B, C, . . . , X, Y, except Q and S are restricted);

@—The number sequence (0, 1, 2, 3 to 9999).

Refer to Table 4 for the division prefix of different regions of Sarawak.

Table 4. Registration plates for the Sarawak region.

Division	Prefix	Division	Prefix	Division	Prefix
Bintulu	QT	Samarahan	QC	Kuching	QA/QK
Sarikei	QR	Limbang	QL	Sibu and Mukah	QS
Miri	QM	Kapit	QP	Sri Aman and Betong	QB

The number plate format being used for Sabah is SDx @@@@ x, where:

S—The constant prefix for all Sabah number plates;

D—The division prefix. (e.g.,: A = West Coast, T = Tawau;

x—The alphabetical sequences. (e.g.,: A, B, C, . . . , X, Y, except Q and S are restricted);

@—The number sequence (0, 1, 2, 3 to 9999).

Refer to Table 5 for the division prefix of different regions of Sabah.

Table 5. Registration plates for region SABAH.

Division	Prefix	Division	Prefix	Division	Prefix
Sandakan	SS	Kudat	SK	Beaufort	SB
Tawau	ST	Labuan (replaced)	SL	Lahad Datu	SD
Sabah Government	SG	West Coast	SA, SAA-SAB	Keningau	SU

The number plate format being used for Taxi is HSx @@@@, where:

H—The constant prefix for all taxi number plates;

S—The state or territory prefix. (e.g.,: W = Kuala Lumpur, P = Penang);

X—The alphabetical sequence. (e.g.,: A, B, C, . . . , X, Y);

@—The number sequence (0, 1, 2, 3 to 9999).

Refer to Table 6 for the division prefix for different regions.

Table 6. Taxi license plates prefixes.

State	Prefix	State	Prefix
Kuala Lumpur	HW	Selangor	HB
Johor	HJ	Penang	HP
Pahang	HC	Malacca	HM
Kelantan	HD	Sarawak	HQ
Sabah (replaced)	HE	Perlis	HR
Perak	HA	Sabah	HS
Kedah	HK	Terengganu	HT
Labuan	HL	Negeri Sembilan	HN

Certain taxis around Shah Alam use a somewhat different format of HB ##### SA. Certain limo service from the airport also use a different format of LIMO ##### S. Military services use the format ZB #####, where:

Z—Malaysian Armed Forces vehicles;

B—Segment prefix. (e.g.,: A = prior use before the division of different services, D = Malaysian Army, L = Royal Malaysian Navy, U = Royal Malaysian Air Force, Z = Ministry of Defense);

#—The number sequence. (e.g.,: 1, 2, 3 to 9999).

All the mentioned regex formats help to make sure that the optical recognized character is in format, because there are times when the converted texts are incorrect with a high confidence probability, while at times texts are correct with a low confidence probability.

4. Experimental Results

We discuss the performance of the pipeline and the performance of individual components in the pipeline. Based on multiple use cases of a large number of vehicles' authentication, since a significant precision is essentially required in most of cases, we focused on measuring the average precision to reveal the relevance in misidentification of cases. The precision is the ratio between true positive to the total number of true positive and false positive cases. The proposed smart authentication method achieved a precision of 98% s shown in Table 7 below.

Table 7. Region-wise precision of different use cases.

State	Use Cases	No. of Images	Precision	Mean Precision
Kuala Lumpur	1	500	98.3	98.28
	2	500	98.5	
	3	500	97.9	
	4	500	99	
	5	300	97.7	
Penang	1	300	97.5	98.14
	2	300	98.2	
	3	300	98.6	
	4	300	97.6	
	5	300	98.8	
Terengganu	1	450	99	98.16
	2	450	96.9	
	3	450	98	
	4	450	97.7	
	5	450	99.2	
Malacca	1	500	98	98.28
	2	500	98.1	
	3	500	98.4	
	4	500	97.5	
	5	500	99.4	
Negeri Sembilan	1	300	97.4	97.96
	2	300	97.2	
	3	300	98.3	
	4	300	98.7	
	5	300	98.2	
Perlis	1	450	98.1	97.88
	2	450	97.8	
	3	450	97.5	
	4	450	98.2	
	5	450	97.8	
Mean precision of all use cases				98.1

Precision–recall tradeoff is a good measure to signify the performance of different use cases to reveal how precisely the experiments were performed and the impact of gap (errors) between authenticated cases. In most of the existing research, we could not find the precision–recall tradeoff.

Figure 10 presents the precision and error comparison. We can see a high precision value as compared to error in estimation between different experiments of use cases. A slight difference in error between the state results is negligible.

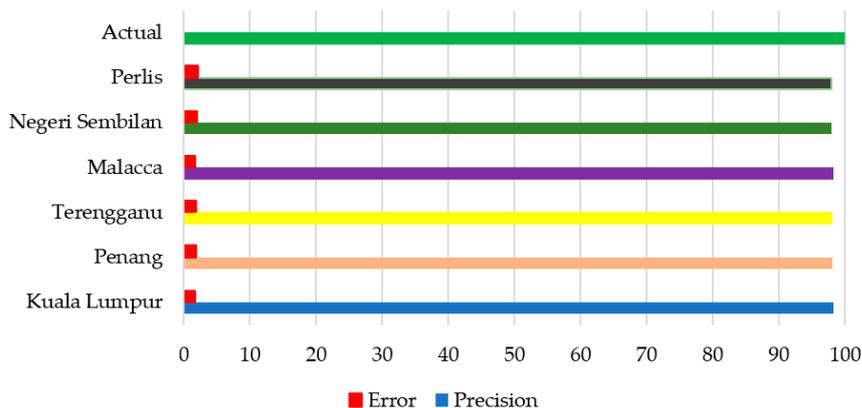


Figure 10. Precision and error.

We achieved a high precision value with a moderate recall value. A precision–recall curve will show us the true summarization of the trade-off between true positives and predicted values which are from the positive class. As shown in Figure 11, we see the precision–recall trade-off. We can see that as the recall increases, the precision decreases. This is so because as the number of positive samples (in our case, vehicles) increases, classification accuracy for each sample decreases.

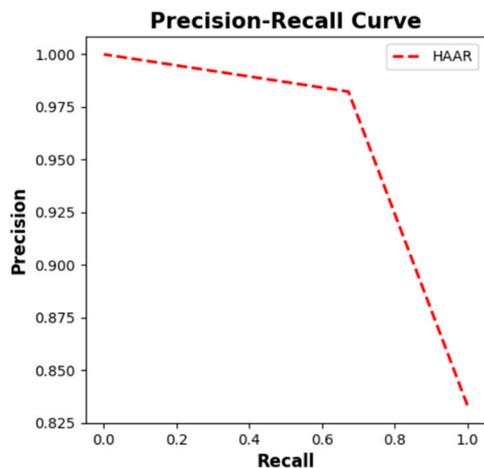


Figure 11. Precision–recall curve.

Additionally, to reduce the probability of bias in results, we performed the analysis of variance (ANOVA) test based on the accuracy and precision. Since accuracy, by default, in most research is very similar, we focus on precision to signify the investigation errors in use case experiments. The ANOVA test is based on the F-test to reveal the variability between group means (within and between the groups). The outcomes are presented in Table 8 below.

Table 8. Single factor ANOVA.

SUMMARY				
Groups	Count	Sum	Average	Variance
Accuracy	6	594.8	99.13333	0.014667
Precision	6	588.7	98.11667	0.027267

Table 8. Cont.

ANOVA						
Source of Variation	SS	df	MS	F	<i>p</i> -value	F crit
Between groups	3.100833	1	3.100833	147.8935	2.58×10^{-7}	4.964603
Within groups	0.209667	10	0.020967			
Total	3.3105	11				

Chosen on a 95% confidence interval ($\alpha = 0.05$), the F value is 147.89, which is significantly larger than the F-critical value of 4.96, achieving a *p*-value of 0.00, which is significantly smaller than 0.05.

5. Conclusions

We proposed a single-shot and smart vehicle authentication approach based on Haar cascade object detectors and MobileNet-SSD for better precision and reduction in time complexity. The hybrid model performed well for number plate detection in live video feeds. Since the speed of the car has a large impact on the number plate detection, the model performs best by overcoming the limitations of existing work. Moreover, the existing research focused on the prediction accuracy, which in most state-of-the-art methods (SOTA) is very similar. Thus, the precision among a number of use cases is also a good evaluation measure that was ignored in the existing research. It is evident that the performances of prediction systems suffer due to adverse weather conditions, e.g., foggy and rainy seasons, inadequate light exposure, distance of vehicle from cameras, and other similar limitations. In such cases, the precision between events of detection may result in high variance that impacts the prediction of vehicles in unfavorable circumstances. The performance assessment of the proposed solution yields a precision of 98% on real-time data for Malaysian number plates, which can be generalized in the future to all sorts of vehicles around the globe. Furthermore, for the future scope of the study, this research recommends, (1) adopting a high-definition camera to capture images at high pixel and FPS rates and (2) ensuring the clock synchronization of all cameras.

Author Contributions: Conceptualization, M.A. and K.R.; data curation, M.A., N.A.G., and J.U.; formal analysis, K.R.; funding acquisition, J.S.; investigation, N.A.G.; methodology, K.R.; resources, J.S.; software, J.U.; visualization, K.R.; writing—review and editing, K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the University Malaya, UM Living Labs, under university Grant LL055-2021. This research was also supported by the Competitive Research Fund of the University of Aizu, Japan, and Woosong University Academic Research in 2023.

Data Availability Statement: Data will be available upon request.

Conflicts of Interest: The authors declare no conflict of interest in this research.

References

1. Darapaneni, N.; Jillela, R.; Reddy, A.; Rao, P.C.R. Computer Vision based License Plate Detection for Automated Vehicle Parking Management System. In Proceedings of the 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 28–31 October 2020; IEEE: Piscataway Township, NJ, USA, 2020.
2. Mangal, A.; Sharma, P.; Kumar, S.; Suhane, S.M. Design and Development of IOT Based Prototype Police Barricade. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 9–12 September 2020; IEEE: Piscataway Township, NJ, USA, 2020.
3. Wang, W.; Li, J.; Wang, Y.; Wang, Q. A Light CNN for End-to-End Car License Plates Detection and Recognition. *IEEE Access* **2019**, *7*, 173875–173883. [[CrossRef](#)]
4. Sharma, P.S.; Raturi, A.; Pandey, S.; Sharma, S. Localisation of License Plate and Character Recognition Using Haar Cascade. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 3–15 March 2019; IEEE: Piscataway Township, NJ, USA, 2019.

5. Li, H.; Liu, Z.; Doermann, D. Toward End-to-End Car License Plate Detection and Recognition with Deep Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1126–1136. [[CrossRef](#)]
6. Xie, L.; Wu, J.; Xiong, Z. A New CNN-Based Method for Multi-Directional Car License Plate Detection. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 507–517. [[CrossRef](#)]
7. Kumar, M.B.K.A.; Chatterjee, A.; Kumar, P. Vehicle Number Plate Detection and Recognition using Bounding Box Method. In Proceedings of the 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), Ramanathapuram, India, 25–27 May 2016.
8. Yulianto, R.F.; Nugraha, A.C.; Bhawiyuga, E.S.; Dewantari, H.K.N. Smart Parking System Based on Haar Cascade Classifier and SIFT Method. In Proceedings of the 2021 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2021; IEEE: Piscataway Township, NJ, USA, 2021.
9. Shariff, A.S.M.; Shukor, M.B.M.; Ali, M.A.I.; Razali, M.F.M. Vehicle Number Plate Detection Using Python and Open CV. In Proceedings of the 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 4–5 March 2021; IEEE: Piscataway Township, NJ, USA, 2021.
10. Nandy, T.; Tasnim, S.M.; Das, S.; Chakraborty, S. A Secure, Privacy-Preserving, and Lightweight Authentication Scheme for VANETs. *IEEE Sens. J.* **2021**, *21*, 20998–21011. [[CrossRef](#)]
11. Cheon, Y.; Kim, J.; Kim, T.H. License Plate Extraction for Moving Vehicles. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019; IEEE: Piscataway Township, NJ, USA, 2019.
12. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
13. Peker, M.; Karagul, Y.; Sonmez, O. Comparison of Tensorflow Object Detection Networks for Licence Plate Localization. In Proceedings of the 2019 1st Global Power, Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 6 December 2019; IEEE: Nevsehir, Turkey, 2019.
14. Aggarwal, A.; Anand, M.; Garg, S. A robust method to authenticate car license plates using segmentation and ROI based approach. *Smart Sustain. Built Environ.* **2019**, *9*, 737–747. [[CrossRef](#)]
15. Zhang, J.; Ouyang, W.; Yang, J.; Sindagi, V.A. License Plate Localization in Unconstrained Scenes Using a Two-Stage CNN-RNN. *IEEE Sens. J.* **2019**, *19*, 5256–5265. [[CrossRef](#)]
16. Varkentin, V.; Shabalina, E.; Salgansky, M. Development of an Application for Car License Plates Recognition Using Neural Network Technologies. In Proceedings of the 2019 International Conference “Quality Management, Transport and Information Security, Information Technologies” (IT&QM&IS), Sochi, Russia, 23–27 September 2019.
17. Nandy, T.; Tasnim, S.M.; Das, S.; Chakraborty, S. An enhanced lightweight and secured authentication protocol for vehicular ad-hoc network. *Comput. Commun.* **2021**, *177*, 57–76. [[CrossRef](#)]
18. Goyal, A.; Das, A.K.; Tiwari, S. Wrong Side Vehicle Detection. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Karnataka, India, 25–27 June 2021; IEEE: Piscataway Township, NJ, USA, 2021.
19. Srivastava, D.; Bhushan, S.; Gupta, S. Automatic traffic surveillance system Utilizing object detection and image processing. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 27–29 January 2021; IEEE: Piscataway Township, NJ, USA, 2021.
20. Mandi, M.S.; Roy, A.; Das, A. An automatic number plate recognition system for car park management. *Int. J. Comput. Appl.* **2017**, *175*, 0975–8887.
21. Babu, R.N.; Kishore, A.N.; Patra, A.; Saha, K.K. Indian Car Number Plate Recognition using Deep Learning. In Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur Kerala, India, 5–6 July 2019; IEEE: Piscataway Township, NJ, USA, 2019.
22. Montazzolli, S.E.; de Souza, A.F.; de Quadros, C.B.; Oliveira, L.S. Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images, Niteroi, Brazil, 17–20 October 2017. [[CrossRef](#)]
23. Sasi, A.; Krishnan, S.R.K.; Kannan, S.S.; Subathra, M.P. Automatic car number plate recognition. In Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 17–18 March 2017; IEEE: Piscataway Township, NJ, USA, 2017.
24. Zhao, Y.; Cheng, F.; Luo, L. License Plate Location Based on Haar-Like Cascade Classifiers and Edges. In Proceedings of the 2010 Second WRI Global Congress on Intelligent Systems, Wuhan, China, 16–17 December 2010; IEEE: Piscataway Township, NJ, USA, 2010.
25. Agarwal, A.; Gupta, R.; Singh, S.; Agarwal, A. An Efficient Algorithm for Automatic Car Plate Detection & Recognition. In Proceedings of the 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 12–13 February 2016; IEEE: Piscataway Township, NJ, USA, 2016.
26. Li, H.; Wang, G.; Wang, S.; Tang, Z. Reading car license plates using deep neural networks. In *Image and Vision Computing*; Elsevier: Amsterdam, The Netherlands, 2018; Science Direct; Volume 72. [[CrossRef](#)]
27. Silva, S.E.M.; Santos, J.A.D.; da Silva Torres, R.; de Carvalho, J.M. License Plate Detection and Recognition in Unconstrained Scenarios. In *Computer Vision Foundation*; ECCV: Glasgow, UK, 2018.
28. Noor, R.M.; Ilias, I.A.F.M.; Anuar, A.F.M.; Ilias, N.F.M. Campus Shuttle Bus Route Optimization Using Machine Learning Predictive Analysis: A Case Study. *Sustainability* **2021**, *13*, 225. [[CrossRef](#)]

29. Zhang, J.; Ouyang, W.; Yang, J.; Sindagi, V.A. An improved MobileNet-SSD algorithm for automatic defect detection on vehicle body paint. *Multimed. Tools Appl.* **2020**, *79*, 23367–23385. [[CrossRef](#)]
30. Younis, A.; AboulSeoud, M.A.; Tolba, M.F. Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD. In Proceedings of the ICCDE 2020: Proceedings of 2020 the 6th International Conference on Computing and Data Engineering, Sanya, China, 4–6 January 2020; ACM Digital Library: New York, NY, USA, 2020.
31. Stefanović, H.; Radosav, D.; Miletić, P. An Adaptive Car Number Plate Image Segmentation Using K-Means Clustering. In Proceedings of the International Scientific Conference On Information Technology and Data Related Research, Belgrad, Serbian, 20 April 2018.
32. Lee, S.; Lee, H.; Song, J. Car plate recognition based on CNN using embedded system with GPU. In Proceedings of the 2017 10th International Conference on Human System Interactions (HSI), Ulsan, Republic of Korea, 17–19 July 2017; IEEE: Piscataway Township, NJ, USA, 2017.
33. Viola, P.; Jones, M.J. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 8–14 December 2001; IEEE: Piscataway Township, NJ, USA, 2001.
34. Nguyen, T.-N.; Dang, P.M.; Pham, H.D. A New Convolutional Architecture for Vietnamese Car Plate Recognition. In Proceedings of the 2018 10th International Conference on Knowledge and Systems Engineering (KSE), Ho Chi Minh City, Vietnam, 1–3 November 2018.
35. Omran, S.S.; Salman, A.T.; Zuhair, M.H. Iraqi Car License Plate Recognition Using OCR. In Proceedings of the Annual Conference on New Trends in Information & Communications Technology Applications-(NTICT'2017), Baghdad, Iraq, 7–9 March 2017; IEEE: Piscataway Township, NJ, USA, 2017.
36. Balaji, G.N.; Keerthika, K.B.; Shantharaj, P. Smart Vehicle Number Plate Detection System for Different Countries Using an Improved Segmentation Method. *Imp. J. Interdiscip. Res. (IJIR)* **2017**, *3*, 263–268.
37. Tiwari, B.; Thakore, P.; Tiwari, R.; Dwivedi, S. Automatic Vehicle Number Plate Recognition System using Matlab. *IOSR J. Electron. Commun. Eng. (IOSR-JECE)* **2016**, *11*, 10–16. [[CrossRef](#)]
38. Yogheedha, K.; Ulagamuthalvi, S.A.R.A.M.; Kavitha, R. Automatic Vehicle License Plate Recognition System Based on Image Processing and Template Matching Approach. In Proceedings of the International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), Kuching, Malaysia, 15–17 August 2018.
39. Soon, C.K.; Idris, N.R.N.; Ghazali, R.M.S. Malaysian Car Number Plate Detection and Recognition System. *Aust. J. Basic Appl. Sci.* **2012**, *6*, 49–59.
40. Zakaria, M.F.; Nor, N.M.; Nordin, A.N.K. Malaysian Car Number Plate Detection System Based on Template Matching and Colour Information. *Int. J. Comput. Sci. Eng.* **2010**, *2*, 1159–1164.
41. VPriya, L.; Seshadri, R.R.; Vijayalakshmi, M.S. Detecting the Car Number Plate Using Segmentation. *Int. J. Eng. Comput. Sci.* **2014**, *3*, 8823–8829.
42. Du, S.; Ibrahim, M.; Wang, Z.; Lu, Z. Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 311–325. [[CrossRef](#)]
43. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016.

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.