Single Shot Detector based MobileNet for Automatic Detection and Recognition of License Plates

D.C. Kiruthuka
Artificial Intelligence & Data Science, Dr.Mahalingam College of Engg.& Tech, Pollachi, India
kiruthikkadc@drmcet.ac.in

M. Prasanna Blessy
Computer Science & Engineering (CyberSecurity) Dr.Mahalingam College of Engg.&Tech, Pollachi, India
blessypaul90@gmail.com

M. Premalatha
Artificial Intelligence & Data Science, Kathir College of Engineering, Coimbatore, India
premalathaanandcse@gmail.com

S. Pooja
Computer Science & Engineering Sri Raganathar Institute of Engg.& Tech, Pollachi, India
poojacsecbe@gmail.com

***Abstract*— An Automatic motor vehicle Registration Number Recognition and Monitoring Solution is crucial for the shift towards technological advances in numerous everyday life features such as the management of parking spaces, tollbooth facilities, and roadway control situations. The proposed research concentrates on the databases for automobile registration numbers, regions, and forward images. The suggested approach concentrates on collecting deformed registration number plates caused by angled perspectives. The Single Shot Detector (SSD) is used to recognize as well as distinguish distorted registration plate numbers in frame by frame. Following license plate recognition, distortion on the registration plate is reduced using computerized processing methods. Additionally, it crops the text after retrieving them from the registration plate utilizing a sort outline technique. With the use of a lightweight in nature neural network such as MobileNets, the characters are subsequently turned into electronic characters through Optical Character Recognition (OCR). The suggested model also uses an evaluation measure, like stopping training when the value of the loss does not decrease after five epochs. As a result, it achieves 99.54% overall precision throughout the validation dataset and a precision of 98.26% after 23 training epochs. In order to prevent unauthorized entrance into susceptible sites, the suggested solution uses automated automotive vehicle identification validation through vehicle numbers.**

Keywords*— *single shot detector, mobilenet, deep learning, artificial intelligence, license plate detection, license plate recognition, opencv, optical character recognition, lightweight neural network

I. Introduction

Developers continue to work to develop an Automated License Plate Recognition (ALPR) technology that can quickly and accurately identify license plates from not high-quality or slanted images.

The primary uses of ALPR systems include law enforcement, traffic management, and municipal parking [1].

Typically, recognition of characters and cameras to detect automobiles are used by authorities and commercial businesses. However, in actuality, on the highways, notably, where there isn't an authorized camera, patrol cruisers' cameras are significantly impacted by the speed, perspectives, and visibility conditions, which means that they fail to offer adequate clarity under particular weather situations. Having a license plate system like this is essential since license plates reveal a lot about the cars. To name a few of the difficulties, we may highlight the poor quality, blurriness, and uneven lighting as well as the diverse font kinds, character counts, sizes, colors, directions, and complicated backgrounds in several nations or even distinct regions inside a single nation.

Owing to the impact caused by different atmospheric factors and lighting, LPR applications are rendered incapable of accurately interpreting the license plates sometimes [2]. For example, some systems are unable to read the characters effectively while utilizing a reflective vinyl automobile with writing to improve the characteristics and lessen the amount of contrast, resulting in incorrect identification.

Additionally, the clarity of the pictures captured by various cameras varies with respect to their position relative to the automobile, and the cameras' angles are highly diverse. Additionally, a few ALPR technologies can only detect specific kinds of cars, such as trucks and buses, and cannot detect other sorts of vehicles, such as SUVs, sedans, and motorcycles.

All of this makes one of the major challenges faced in developing a reliable model to recognize license plates is dealing with poor performance in conditions that are not ideal in terms of noise, angles and lighting. We utilize Hough Line transform and Canny Edge Detection to minimize this problem.

The proposed technology is capable of accurately recognizing and interpreting the characters in the license plate even when the conditions to be able to do so are not all perfect due to the image preprocessing techniques implemented to make the license plate more legible before the OCR is carried out.

This is done by employing a Single Shot Detector algorithm that utilizes the MobileNet architecture. This improves performance of the model even in small embedded systems. We chose to employ this technique as such models have been observed to perform well while requiring low resource overhead in such systems. [3]

This can be useful for security systems in residential areas and beyond that, as people living in cities also have an ever-growing requirement for private parking spaces that are managed by a computerized system. A parking administration system that registers the authorized car and its owner's details can be developed based on this technology. The registration plate information can be utilised to identify an authorized automobile automatically which makes it simple and safe for occupants of only a certain apartment block to gain entry to the parking spaces that are right for them.

II. Related Work

This section is an overview on some systems that have already been developed for the purpose of ALPR. With the objective of recognising objects, a lot of research has already been carried out over the previous several years on methodologies for image processing in addition to deep learning. Numerous detection and recognition techniques

2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE) | 979-8-3503-7156-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/AMATHE61652.2024.10582122

were designed for automobile surveillance. From a literature study, we can examine the many modern approaches that are being used.

Because methods of image processing may be utilised to determine all potential edges/borders (rectangles) of a vehicle picture, most LPs typically assume the form of a rectangular object with a set dimension ratio. The Sobel edge identification was used in [4] once the supplied pictures' distortion and unwelcome characteristics were eliminated. The study yielded a precision of 74.7% and was carried out on 300 automobile pictures from Malaysia. Additionally, [5] used a canny edge detection computational image analysis approach to marginally improve the recognition operation followed by improving the pictures and eliminating distortion employing median filtration and histogram adjustment. One hundred photos of Chinese vehicles made up the collection of images utilised in this study.

In order to identify the upward and horizontal contours of Indian cars, [6] combined clever edge recognition with the Hough transform (HT) and Euclidean distance. Nevertheless, the outcomes of this technology were subpar, with severely warped pictures. Since the HT approach is so dependent on movement, it is unable to identify twisted pictures effectively. On the other hand, movement may be handled via the connected component analysis (CCA) algorithm. When applying CCA to digital pictures having 4 or 8-pixel areas, each element in the picture is scanned, labelled, and then divided into chunks based on spatial connectivity—that is, the fact that the components share identical attributes. Because of that, 240 Jordanian automobiles were subjected to an 8-pixel area utilizing the CCA approach with canny edge detection in [7].

In 2017, Madhusree Mondal created an ALPR architecture that emphasized the convolutional neural network learning capabilities [8]. Here, CNN's self-synthesized function was used since it separates number plate states from vehicle states. The algorithm was constructed in this research as a columnar collection of detection mechanisms, processors for sensory data progressively in accordance with the visual cortex's predominant sensory background, which had an effect on the mathematical representation of the CNN. This research revealed a 90% improvement in accuracy, which was examined using less training samples.

2018 saw Andrew S. Agbemenu propose an ALPR method that takes into account the characteristics and variations of the plates inside [9]. In this paper, the writer proposes a transportation method that has been enhanced to work with Ghanaian license plates. In the generated framework, template-matching techniques and two possible identification of edges approaches were used. The gadget then used the character division technique, notably employing squared plates, to prevent noise impacts, text layout, and skewing. The Tesseract OCR algorithm was used to finish identification of characters. Despite the somewhat inferior pattern identification, 454 vehicles were correctly detected with an accuracy percentage of 90.8 percent and an overall time of 0.185 seconds per vehicle. The detection of optical characters procedure took an average time of 0.031 seconds, and sixty percent of the observed vehicles were successfully identified.

In their article titled "A Hierarchical Plate Recognition System using Supervised K-means and Support Vector Machine" published in 2017 [10], Weichen Liu et al. developed a brand-new hierarchical character identification method based on supervised K-means and SVM to identify tilted and blurred license plates. The primary goal of this suggested study was to decrease the number of SVMs and their complexity while also reducing the classes of characters in each subgroup. For tilt characters and obscured license plate information, our algorithm provides 98.89 percent accuracy. The average improvement is determined to be 3.6% when compared to state-of-the-art plate recognizing techniques.

Yongsheng Li et al. [2019] in their article "Vehicle License Plate Recognition Combining MSER and Support Vector Machine in a Complex Environment" propose a license plate identification method that utilises MSER and SVM. In this article two more widely used techniques—edge recognition and colour detection—are contrasted with the approach for identifying license plates. The advised approach performs effectively in challenging settings. The text in the region is instantly extracted by MSER. The symbols required for recognition of characters are immediately tested individually in MSER instead of segmenting and positioning license plates. The precision of this method when MSER and SVM are combined is over ninety percent. The precision of boundary identification and colour assessment is significantly poorer to that of the MSER and SVM alone. [11]

In their research titled "A Proposed License Plate Classification Method Based on Convolutional Neural Network" that was published in 2018, Kaili Ni et al. devised an original license plate categorization approach that utilises convolutional neural network. The seven components of the CNN in this work are made up of four layers of convolutional neural network, three layers with maximum pooling, and one final level. After that, the researchers employ the completely linked sections, which facilitate the mining and classification of features. The soft-max operation is lastly executed by the soft max layer. The faster-CNN is being used to localise the car plate. In this case, the success rate of classification for the training set is a full 100 percent. The proposed research is found to have the best precision score, coming in at 98.79%. [12].

III. METHODOLOGY

This section's goal is to provide thorough details on how a license plate in the picture that was recorded is detected. The ALPR system typically combines a monochrome camera with a color camera.

The detected license plate can be saved as an image in development mode, if necessary to better develop the application in the future. We have chosen to use the Single Shot Detector (SSD) based on MobileNet to train the dataset, so it remains a lightweight application.

To train a pre-trained model with a particularly created detector that will be used in the proposed system, such as a license plate detector, this application takes use of the concept of transfer learning. Transfer learning allows any pre-trained model to be used as the foundation for a model on a new task. Because the Single Shot Multibox Detector tensor flow model is the quickest and can function without stuttering with high frame rate video stream, this system employs it.

The Single Shot Multibox Detector simply employs a single convolutional network on the input image to produce a feature map. Additionally, after numerous convolutional layers, it predicts bounding boxes and uses anchor boxes with various aspect ratios. Convolutional layers operate at many scales, enabling them to recognize objects of various sizes. Finding the license plate region is an essential initial stage in achieving proper plate identification. Mixing the techniques for locating or segmenting the license plate in pictures might result in three different handling classes. To recognise distinct characters, several approaches employ structure pictures, black and white, and colour. Text segregation, that could be achieved by learning-based categorization or template matching, is crucial for character recognition. The numerous techniques used to identify the plate numbers are shown in the flow chart described in Figure 1. The following subsections provide an explanation of the procedures involved in the identification and detection of the license plate.

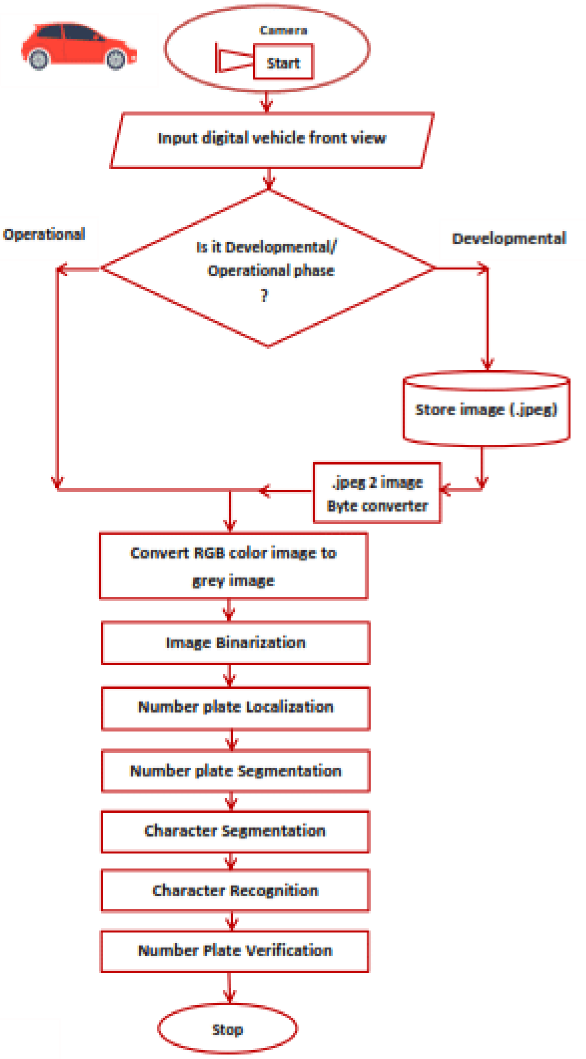

Fig. 1. ALPR Flow Diagram

This process is really similar to the heavier weight Convolutional Neural Network algorithm as it is also a CNN in its most basic sense. However, since the technique is optimized to work well on low-cost embedded systems with resouce constraints, it is able to perform without missing frames during the license plate detection process. The reason for this is that the method relies on a forward-looking CNN to produce an array of fixed-size bounding boxes as well as an assessment on the existence of object category cases within those containers. A non-maximal inhibition step is then added to produce the ultimate identifications without need for replication operations for the hypotheses.

A. Grayscale Image Processing

As shown in Figure 2, this technique, which combines edge statistics and morphological approaches, is used to extract license plate regions from background images. This method has a 98 percent recognition rate based on 4,000 images, supposing that the corners of the number plate frame are straight and exact. It also takes time to analyze all of the binary objects in this method of identifying the no-plate region by removing characters from the binary image. It also yields an incorrect result if there is extra text in the image.

Histogram equalization and color normalization are two approaches used in binary image processing. With its elegant mathematical design, histogram equalization redistributes pixel intensities to improve contrast and uncover hidden features in the picture. It's similar to a magician changing the lights to concentrate on the main performance. Color normalization, meanwhile, harmonizes the color distributions across pictures to prevent the neural network's perception from being distorted by changes in illumination.

The grayscale conversion is one of the venerable methods in this toolbox. Grayscale conversion turns the vivid range of colors into various degrees of grey, much like a traditional black and white image. The simplicity of this method allows the neural network to concentrate exclusively on the underlying structures and textures by reducing the chromatic complexity. This is particularly helpful in situations when the job at hand does not significantly depend on color, such as when object recognition or picture categorization are the main goals.

This makes it ideal for this application's purpose of locating the license plate's location and making it readable for optical character recognition (OCR), which makes it much simpler to complete.

Fig. 2. Grayscale Converted Image Example

B. Histogram Equalization

We must use the histogram equalization technique to increase the image's contrast. The contrast extension procedure improves the image's sharpness. The brightness of a pixel is shown by the image's grey-level histogram. Histogram equalization is used to enhance an image's quality when the contrast is relatively low. There are four stages in the whole process: (i) adding up each histogram's values. (ii) normalizing the values by dividing these values by the total number of pixels. Using the highest grey level value, (iii) expand these values. (iv) Graph the new value for the grey level.

C. Noise Removal

Unwanted sounds in the picture are removed using a median filter [13]. This function accepts an image as a 3x3 matrix. These proportions may be altered depending on the noise levels.

The first step in this method is to orderly sort all the pixel values, after which the median pixel value will be used to replace the pixel under consideration.

D. Optical Character Recognition

region to identify the characters in the license plate once the license plate has been identified and the frame has been processed.

We do this using the Python tool for extracting text from images known as EasyOCR. It is an all-purpose OCR that can read text found in documents as well as text found in natural scenes. It works incredibly well at identifying the characters from the processed part of the license plate in the natural scene in our application since it is an easy-to-implement Python library for OCR.

IV. MODULES

The modules used to build the application include the deep learning algorithm Single Shot Detector (SSD) which has been shown to be a much faster algorithm to detect objects even with very low computing power [13], which makes it an ideal algorithm to be used if the application is to be run on small devices such as a Raspberry Pi.

We have used the EasyOCR library in Python to do the optical character recognition (OCR). It is a lightweight module that can be used to do OCR on devices with low and high computing power alike [14], although the OCR process will take longer with less computing power.

The OpenCV Python library is used to get live input from the camera module. The frames are processed in real time in order to make the application as utilitarian as possible. The OpenCV module records the video live and allows for the extraction of all the frames in order to process them and allow for easy detection by the SSD MobileNet detector.

The frames are processed to make it easier for the EasyOCR module to correctly recognize the characters on the license plate and the recognized characters are displayed on the application and can be stored in a database for getting registration details of the vehicle, if needed. The modules used in the application are further explained in the following subsections.

A. Single Shot Detector (SSD) MobileNet

The SSD method uses a feed-forward convolutional network to generate a fixed-size set of box boundaries and scores for the existence of object-type members in those areas. A minimal suppression step is then used to get the final identifications. The base system, which we will refer to as the foundation of the early network layers, is based on a typical design for high-quality image classification (truncated before any of the classification layers). The network is then given an auxiliary structure, resulting in detections with the main characteristics listed below:

1) Multi-scale feature maps for detection: Convolutional layers of features are added as an extension to the trimmed core structure. Predictions of determinations at different sizes are made possible through these sections, which increasingly become thinner. For predicting identifications, every characteristic tier uses a different cognitive technique (cf. Overfeat [15] and YOLO [16], that operate on a single scale map).

2) Convolutional predictors for detection: Every characteristic tier which has been included (or, if desired, an existing characteristic tier of the original structure) can offer a specific set of recognition predictions using a set of convolutional filters. The design of the SSD mobile network depicts them in Figure 3. A 3 x 3 x p small core which produces maybe an assessment for an attribute or an outline departure in reference to the benchmark enclosed characteristics serves as the crucial foundation to forecast the characteristics of a potential identification of a characteristic layer having dimension m x n with p ports. Anywhere the kernel is employed, it produces a result score at every one of the m x n locations. The overall box's borders deviation results are established in respect to a baseline container placement for every feature map's location (cf. the architecture of YOLO[17], which accomplishes this step using a middle layer that is fully connected versus a convolutional filter).

3) Default aspect ratios and boxes: For each feature map cell throughout the network's numerous feature maps, we assign a set of standard boundaries. The location of each box in relation to its related cell is fixed since the default boxes tile the characteristic map in a neural fashion. We forecast the offsets from the cell's baseline boxed sizes in addition to actual per-class scores that show if a class instance is present in each of those boxes at every feature map cell. In further detail, we calculate the 4 offsets from the default box shape and the c class scores for all the boxes out of k at a given position. To produce (c + 4)kmn outputs for a m x n feature map, this leads to a maximum of (c + 4)k filters being applied all around each place in the feature map. Please see Figure 4 for an example of default boxes. We apply our default boxes to a variety of feature maps with various resolutions, but they are comparable to the anchor points utilised by Faster R-CNN [18]. We can discretize the universe of potential output box shapes effectively by allowing various baseline boxed sizes in a number of feature maps.

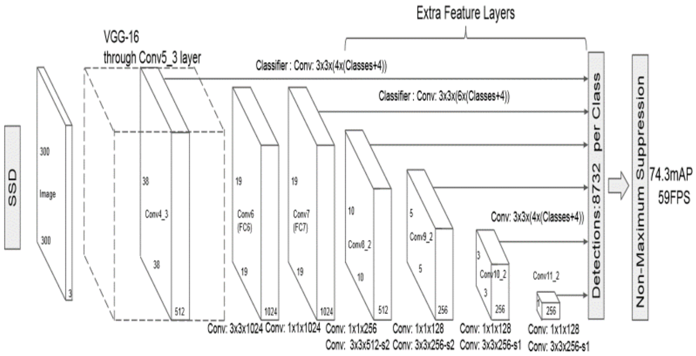

Fig. 3. SSD MobileNet Architecture

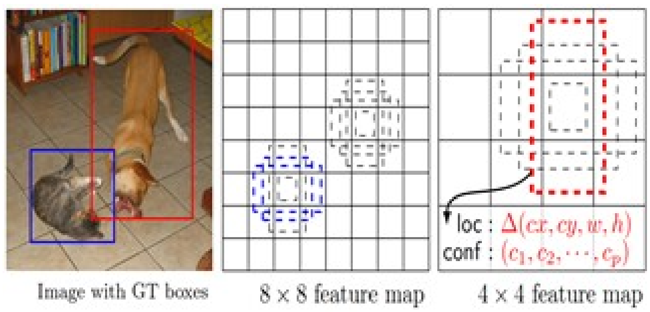

Fig. 4. Default Box Examples

B. OpenCV

The detection and the recognition of the license plates are both done in real-time. This is possible with the help of the Python library OpenCV which allows the program to access the camera module available on the device, and takes a stream of frames from the live video as the input, and allows us to do any kind of processing necessary to allow for easier detection or recognition of the license plates.

We segment the license plate from each frame when the confidence of the object being a license plate is more than 40% and process just the segmented part of the frame to allow for easier OCR of the characters in the license plate. The detection and bounding of the segmented license plate in the application are shown in Figure 5.

Fig. 5. Detection of License Plate in OpenCV

C. OCR of Characters

The license plate, once detected is segmented into a separate frame using OpenCV, and is then processed into a grayscale image, and histogram equalization and noise removal are done. After this, the frame is sent to the EasyOCR module, which recognizes the characters on the license plate by first detecting the text characters employing the Convolutional Neural Network (CNN). EasyOCR is able to do this for a multitude of languages and automatically recognizes the language before outputting the recognized characters.

Once it recognizes the text, it also does post-processing to refine the recognized text and correct errors. This ensures the best accuracy possible for a given input frame. This is however limited by the quality of the image frame and the resolution of the license plate detected and sent as the input to be recognized to the EasyOCR module.

D. Dataset

The dataset we have chosen to train our model with is the RodoSol-ALPR dataset with 20,000 images captured by static cameras located at pay tolls owned by the Rodovia do Sol (RodoSol) concessionaire, which operates about 67.5 kilometers of the highway ES-060 in the state of Espírito Santo in Brazil.

Following is how the 20,000 photos are divided: 5,000 photographs of vehicles with Brazilian license plates, 5,000 images of motorbikes with Brazilian license plates, 5,000 images of vehicles with Mercosur license plates, and 5,000 images of both. In this definition, "car" applies to all vehicles with four wheels or more (such as passenger cars, vans, buses, and lorries), while "motorcycle" includes both motorcycles and motorised tricycles for the purpose of clarity. Following the split technique of 40%/40%/20%, we randomly divided the RodoSol-ALPR dataset into 8,000 photos for training, 8,000 images for testing, and 4,000 images for validation.

V. Results

The model was trained using the SSD MobileNet algorithm with the RodoSol-ALPR dataset in Python and an accuracy of 99.54% was achieved during validation after training for 23 epochs, we had implemented early stopping to make sure that both the accuracy and performance would be optimized. The training was automatically early-stopped at 23 epochs out of the 80 epochs we had set in the Python code.

Similarly, the loss observed during validation was also very low and impressive at a mere 4.34% as seen in Figures 6 and 7.

Early stopping is a purposeful technique in Tensorflow which allows developers to save training time and also prevents overfitting, which would lead to increased training loss and reduced training accuracy. It allows developers to achieve greater performance even on unseen data. It has allowed us to achieve great model performance as well.

The dataset chosen seems to have been able to work really well with the SSD MobileNet algorithm and the dataset seems to be sufficient to detect license plates from vehicles in countries other than Brazil as well with our model used in the application.

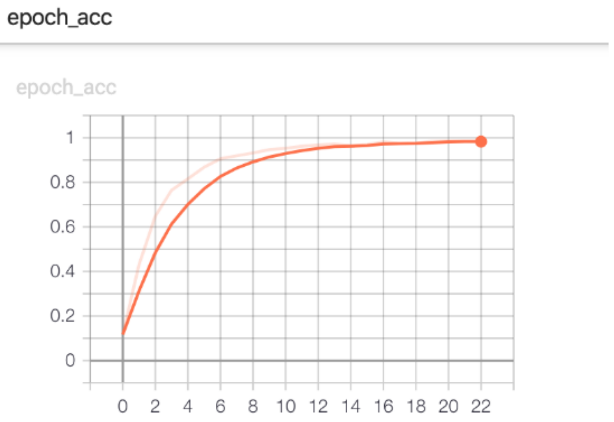

Fig. 6. Graph of Accuracy during Validation

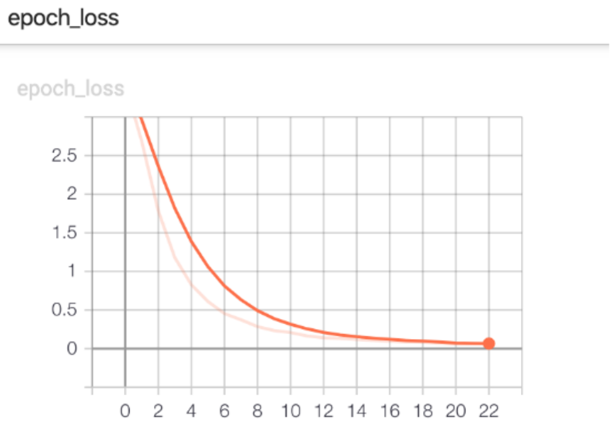

Fig. 7. Graph of Losses during Validation

The final application we have built works as expected and produces proper outputs at about 30 frames per second, as expected, although the EasyOCR module might fail to properly recognize all the characters in the license plate if the license plate is far away from the camera as seen in the sample outputs in Figures 8 and 9.

Fig. 8. Sample Output - 1

Fig. 9. Sample Output - 2

VI. CONCLUSION

The SSD MobileNet model we have trained and implemented in the application works really well in detecting license plates even for real-time video inputs as it is a lightweight model and the EasyOCR Python module is able to correctly recognize the characters in the license plate. The model we have trained has achieved a commendable accuracy of 99.54% during the validation process and this application is quite a practical option to be used for utilitarian purposes such as an apartment parking lot, home security, or institutional security. The application could be run even on smaller devices with lower computing power such as a Raspberry Pi with just a decent camera module.

This application could further be improved in the future with a better OCR module if there is a need to use the application on a larger scale level, and if there are vehicles moving at high speeds, the number of input frames could be increased, which should help capture the license plates on the vehicles when they are closer to the camera and help in the OCR of the characters on the license plate as well. Right now, the application has shown to be a viable option for utilitarian purposes by the masses.

REFERENCES

[1] J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," in IEEE Access, vol. 9, pp. 11203-11225, 2021, doi: 10.1109/ACCESS.2020.3047929.

[2] Pattanaik A, Balabantaray RC. Enhancement of license plate recognition performance using Xception with Mish activation function. Multimed Tools Appl. 2023;82(11):16793-16815. doi: 10.1007/s11042-022-13922-9. Epub 2022 Oct 14. PMID: 36258895; PMCID: PMC9560886.

[3] A. BOUGUETTAYA, A. KECHIDA, and A. M. TABERKIT, "A Survey on Lightweight CNN-Based Object Detection Algorithms for Platforms with Limited Computational Resources", IJIAM, vol. 2, no. 2, pp. 28–44, 2019.

[4] Marzuki, P.; Syafeeza, A.R.; Wong, Y.C.; Hamid, N.A.; Alisa, A.N.; Ibrahim, M.M. A design of license plate recognition system using convolutional neural network. Int. J. Electr. Comput. Eng. (IJECE) 2019, 9, 2196–2204.

[5] Li, M.; Miao, Z.; Wang, J.; Wang, S.; Zhang, Y. A Method of License Plate Recognition Based on BP Neural Network with Median Filtering. J. Physics Conf. Ser. 2019, 1229, 12033.

[6] Nasser, A.B.; Alsewari, A.; Zamli, K.Z. Smart Toll Collection Using Automatic License Plate Recognition Techniques; Springer: Singapore, 2018.

[7] Mustafa, H.; Hassanin, S.; Al-Yaman, M. Automatic Jordanian license plate recognition system using multistage detection. In Proceedings of the 2018 15th International Multi-Conference on Systems, Signals and Devices, SSD, Yasmine Hammamet, Tunisia, 19–22 March 2018; pp. 1228–1233.

[8] Mondal M, Mondal P, Saha N, Chattopadhyay P (2017) December. Automatic number plate recognition using CNN based self-synthesized feature learning. In 2017 IEEE Calcutta Conference 378-338.

[9] Agbemenu AS, Yankey J, Addo EO, (2018) An automatic number plate recognition system using opencv and tesseract ocr engine. International Journal of Computer Applications,180: 1-5.

[10] W. -C. Liu and C. -H. Lin, "A hierarchical license plate recognition system using supervised K-means and Support Vector Machine," *2017 International Conference on Applied System Innovation (ICASI)*, Sapporo, Japan, 2017, pp. 1622-1625, doi: 10.1109/ICASI.2017.7988244.

[11] Li, Y., Niu, D., Chen, X., Li, T., Li, Q., & Xue, Y. (2019, July). Vehicle License Plate Recognition Combing MSER and Support Vector Machine in A Complex Environment. In 2019 Chinese Control Conference (CCC) (pp. 7045-7050). IEEE.

[12] K. Ni, M. Fu, Z. Huang and S. Sun, "A Proposed License Plate Classification Method Based on Convolutional Neural Network," *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, Guiyang, China, 2018, pp. 344-347, doi: 10.1109/ICNIDC.2018.8525618

[13] S. Kumar, A. Kar and M. Chandra, "SVM based adaptive Median filter design for face detection in noisy images," *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2014, pp. 695-698, doi: 10.1109/SPIN.2014.6777044.

[14] D. Niranjan, B. VinayKarthik and Mohana, "Performance Analysis of SSD and Faster RCNN Multi-class Object Detection Model for Autonomous Driving Vehicle Research Using CARLA Simulator," *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Erode, India, 2021, pp. 1-6, doi: 10.1109/ICECCT52121.2021.9616712.

[15] Jain, P.H.; Kumar, V.; Samuel, J.; Singh, S.; Mannepalli, A.; Anderson, R. Artificially Intelligent Readers: An Adaptive Framework for Original Handwritten Numerical Digits Recognition with OCR Methods. Information 2023, 14, 305. https://doi.org/10.3390/info14060305.

[16] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: ICLR. (2014).

[17] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016).

[18] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015).