Smart Cities in Deep Learning : License Plate Recognition System*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Muhammad Iqbal Bin Mohd Fauzi
Hochschule Hamm-Lippstadt
Lippstadt, Germany
Muhammad-iqbal.bin-mohd-fauzi@stud.hshl.de

***Abstract*—The growing population of society will result in a greater use of vehicles. As a result, the implementation of a license plate recognition system is critical, especially when developing smart cities, as it is required for traffic control, toll management, security, and other purposes.Furthermore, the advancement of the LPRS is a well-known research topic in the field of image processing [1].There are several methods and techniques for detecting and recognizing license plates. However, it should be noted that there will be many difficulties in developing a license plate recognition system because the license plate will have different colors, fonts, languages, characters, and sizes.Furthermore, the entire vehicle will be moving at a high speed. As a result, much research will be conducted in order to develop this system in order to achieve maximum accuracy in license plate detection while also making this model reliable. The steps for developing the license plate recognition system are divided into three sections in this paper: LP detection, Crop boundary box, and character recognition using EasyOCR.For LP detection, a deep learning model based on Convolution Neural Network (CNN) will be used to determine the license plate's region of interest (ROI).The proposed system will make use of a dataset containing a large number of vehicles with license plates in various settings and environments.**

***Index Terms*—License plate recognition, Deep learning, CNN, EasyOCR**

I. Introduction

As vehicles are widely used, license plate recognition systems are one of the most important features when dealing with smart cities. Each vehicle will be distinct because it will have its own identification [1]. The use of the LPRS can be implemented everywhere, for example, for traffic, parking, and toll management, and it can also increase safety [2]. LPRS can increase safety by assisting in the monitoring and management of security at any location, as it can be used as a tracking tool for the security team in the event of an accident or theft [2].Furthermore, by implementing LPRS, it can be very helpful in easily directing vehicle identification as well as warning and speed control in a specific area [4]. The concept of deep learning must be clearly understood in order to implement an LPR system. Deep learning is a subclass of machine learning that employs artificial neural networks that mimic the behavior of the human brain. Deep learning will go through many layers in order to transform data, and the model's abilities, such as speech recognition and image classification, have already been demonstrated [4]. There are several techniques that can be used to achieve this LPR system while maintaining the system's accuracy and efficiency. During the extraction of Region of Interest, for example, various recognition algorithms have been proposed (ROI).Furthermore, various algorithms, systems, and techniques are used for LP detection and recognition.

Identify applicable funding agency here. If none, delete this.

II. Deep Learning Model

To increase the accuracy of a deep learning model, two major components are required: a dataset and a neural network.

A. Convolutional Neural Network (CNN)

The deep learning model is being used everywhere. CNN (convolutional neural network) is one of the most effective neural network deep learning models for image recognition [5]. The CNN will be able to recognize images by learning some features from the images provided. However, the input of the given images or data for CNN to learn must be extremely large in order to achieve human-like accuracy [5].Because of its exceptional performance, CNN will be used for license plate detection, in which the CNN will learn to find the license plate of the given images by analyzing the images' features. To understand how CNN works, the structure of the CNN, which consists of many layers, must be clearly understood. Convolution layer, pooling layer, fully connected layer, and output layer were among the layers.Figure 1 depicts a CNN structure example. Kernel will be used to filter the input image in the convolution layer. The most common filtering size is 5*5. The feature map will be obtained following the convolution layer. This feature map will then be filtered by a max-pooling layer or an average-pooling layer to produce the feature map. The feature map will then be flattened and used as input for the fully connected layer.

B. Dataset

It should be noted that the dataset used is one of the most important aspects of implementing deep learning. The dataset must be large enough to allow the model to be properly trained.The dataset can be easily used by anyone to build a

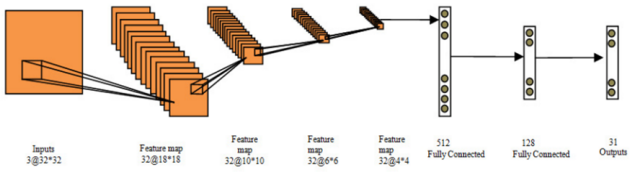

Fig. 1. Example of a figure caption.

specific deep learning model thanks to the assistance of many open sources. Imagenet, PASCAL VOC, COCO, and Kaggle are a few examples [15].

The Imagenet dataset contains over 14 million images organized into 20,000 categories [15]. Imagetnet also provides millions of images with explicit class annotations as well as annotations for the locations of the objects in the image [15]. As a result, the Imagenet dataset is one of the most commonly used datasets for deep learning models [15]. PASCAL VOC (pattern, analysis, statistical modeling, and computational learning visual object classes) will, on the other hand, need to provide a set of tools in order to access the data sets as well as the annotations. COCO (Common Objects in Context), a Microsoft-sponsored project, will have over 300,000 images in 80 object categories [15].

III. Related Work

A few steps will be required in the implementation of the LPRS in order to achieve this system correctly and efficiently. The system starts with LP detection, then character segmentation, and finally character recognition. The following steps are described in detail below.

A. LP detection

The primary goal of LP detection is to recognize the plate region over multiple objects. There are various types of systems for implementing LP detection, each with its own set of properties. This step requires a detection mechanism for the license plate in the given input image for LP detection. For example, the Gabor filter is used in the mechanism for plate detection as an input for RBG images, as well as gray scale transformation for binary images.Furthermore, the histogram technique can be used to detect plate region, which requires a histogram to be generated using horizontal and vertical projection on the input image. Hough transformation and edge-based methods are two other approaches. This technique is used in Hough transformation by locating the edges around the number plate.However, this Hough transformation technique will be less accurate than the edge-based method because it will use generalized symmetry transformation to identify the plate region [2]. There is also a method that combines a contour detector with morphological operations to find the rectangles of the plates.All of the preceding steps are algorithms for image processing. Deep learning is used to recognize license plates, also known as ROI (region of interest), by training the system with datasets. There are numerous datasets available on the internet that contain hundreds to thousands of license plate images. Kaggle is one example of a dataset that can be found online.Because different countries will have different types and designs of license plates, the deep learning model will be able to train to recognize all of the license plates. There are numerous deep learning models available. For example, inceptionResNetV2, MobileNetV2, InceptionV3, and so on. The majority of deep learning models will be based on CNN, which have many layers for training. This is accomplished by drawing boundary box of the license plate.

B. Character Segmentation

Following the completion of the LP detection step, the next step will be character segmentation, which will aid in detecting the region of character in the license plate image. A variety of approaches can be used in this step as well. For example, the You Only Look Once (YOLO) method divides a single frame into regions and then predicts the boundary box [4].In addition, the most commonly used methods for character segmentation are projection application, contours, and histogram treatment.However, the method for character segmentation will not be implemented in this paper because the system will only be used by cropping the boundary box that contains the character and number of the license plate. The proposed system will be discussed in greater detail later.

C. Character Recognition

After LP detection and character segmentation of the license plate, the final step is character recognition, which will be used to recognize each and every character of the number plate. Due to noise or a malfunctioning data acquisition process, it is very difficult to accurately detect the character due to the variety of character and number shapes such as, font style, font size, and font orientation [2].However, in most modern systems, these issues can be addressed by employing techniques based on raw data, extracted features, or neural networks [2]. EasyOcr will be used in this paper for character recognition to recognize the number plate, which includes numbers and characters. Following the implementation of EasyOcr, the result will be extracted and then rendered on the image.

IV. Proposed System

It is now time to talk about the proposed system for this paper as a way to implement the LPRS. The detection of LP will be discussed first. The LP detection will take place within the green boundary box. This boundary box will then be cropped. Following that, the extraction of the license plate's characters and numbers will begin. Finally, the result will be displayed in the image. Figure 1 depicts the flow of the system.

A. LP detection

As previously stated, the LP detection will require a system capable of detecting the license plate on the given input image of the vehicle. There are numerous license plate variations for this proposed system. For example,the position of the

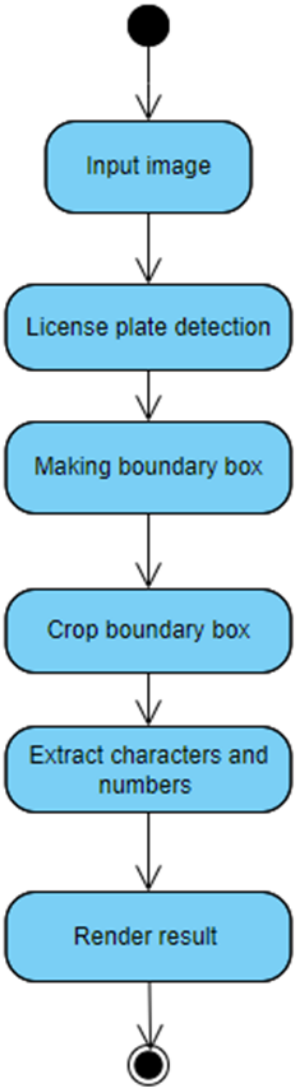

Fig. 2. Example of a figure caption.

license plate and the lighting of the license plate will always change.To detect the license plate in this system, a dataset as well as a CNN model are required.

1) dataset: The Kaggle dataset will be used in this system to obtain the dataset for the cars with license plates. This dataset contains license plates with various regions, colors, fonts, and so on. In addition, the dataset includes both the vehicle and the non-vehicle, such as a person. The images of the vehicles are also in front and back views. The deep learning model will be trained to recognize the license plate as a result of this. The dataset contains 432 images, with 80 percent of them classified as train images and the remaining 20 percent classified as test images. The descriptions of the train and test images are shown in Table 1. The images in the kaggle dataset are converted into XML files. These XML files

TABLE I
DESCRIPTION OF DATASET

Training Dataset	Test Dataset	Total
346	87	433

contains the position of the license plate's boundary box for each image. Following that, in order to train and test the data, these XML files are saved in a.csv file. Figure 3 depicts some examples from the Kaggle dataset of cars with license plates.

Fig. 3. Example of a figure caption.

2) Data Preprocessing: Before incorporating the dataset into a deep learning model, a few image preprocessing steps must be completed. The images will be converted to the target size of 350*350. This converted image will then be converted into an array for image normalization.

3) Faster R-CNN: Faster R-CNN inception ResNet V2 will be used to extract features from input images and for classification in this proposed system.The architecture of the Inception ResNet V2 must be well defined in order to fully understand how the Inception ResNet V2 will be able to extract the features of the given input images.Figure 7 depicts the network structure of the Inception-ResNet-V2 network. Inception ResNet V2 architecture consists of a combination of the Inception structure and a residual network (ResNet) connection [10].The main reason for employing the residual network is that it is extremely useful and efficient for training extremely deep architectures [11]. Furthermore, the residual network will be able to eliminate degradation issues as well as precisely extract features of the input images such as texture, size, color, and so on [10].For the inception module, which is well-known in multi-branch architectures, it will combine multiple convolutions, pooling layers, and feature maps, resulting in a single vector [10]. Typically, the sizes of the filter to extract the features for the inception module are

5*5, 3*3, and 1*1 [10]. The proposed system will be able to extract the license plate's boundary box by using the inception ResNet V2.For compiling the model, Adam optimizer will be used and 'mse' will be used as loss.The model will be trained by total of 300 epochs and the batch size of 10.The training and validation loss will be stored in Tensorboard. Figure 8 shows an example of the extracted license plate boundary box for the given input image. The green rectangle represents the boundary box.

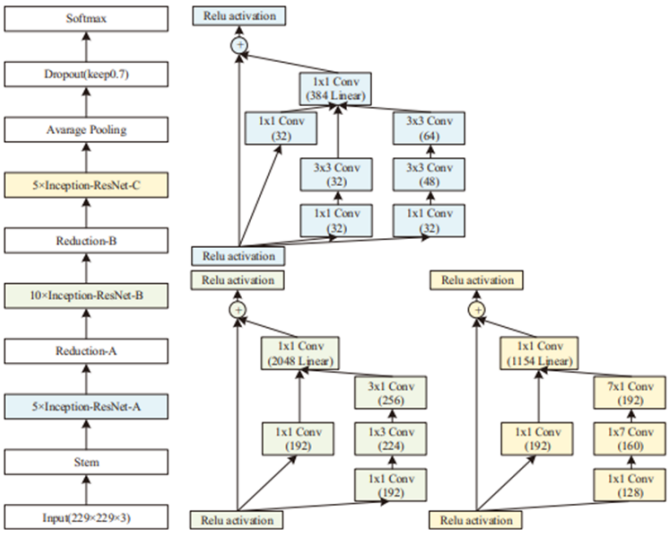

Fig. 4. Example of a figure caption.

Fig. 5. Example of a figure caption.

B. Crop Boundary Box

It is now time to crop the image within its boundary box after obtaining the boundary box of the input image. The cropped boundary box will be used as an input for license plate character recognition.

Figure 9 depicts a cropped image of the boundary box.

Fig. 6. Example of a figure caption.

C. Character Recognition by Using EasyOCR

In this paper, OCR engines will be used to extract the license plate's number and character. The OCR engines are used because of their ability to recognize the number plate and character in the license plate. TesseractOCR and EasyOCR are two well-known open source OCR engines. As EasyOCR can be used for many languages, including English, German, and Chinese, it will be used in this paper to extract the character and number of the license plate. Figure 7 depicts the workflow of the EasyOCr engine for extracting the number and character from a license plate [7]. When applying EasyOCR on the cropped image, the character of the license plate will be rendered. Figure 10 shows the rendered result.In this case, the output of the rendered result is "PL8 REC" which is the same as the actual identification of the license plate.

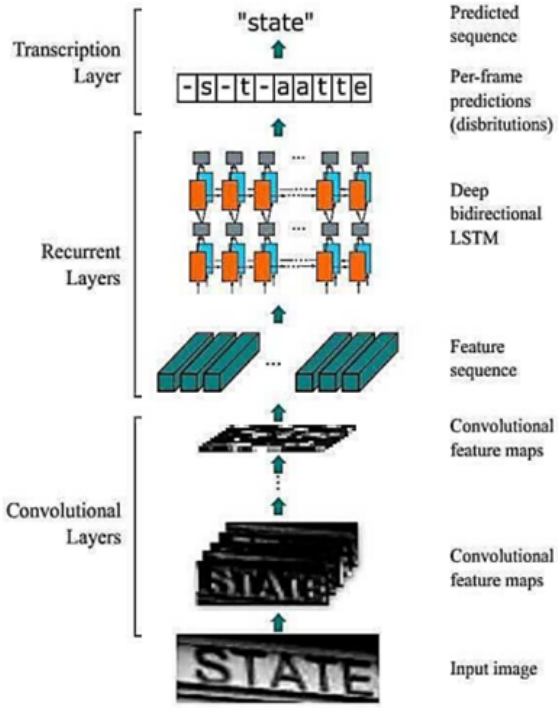

Fig. 7. Example of a figure caption.

Fig. 8. Example of a figure caption.

Fig. 10. Example of a figure caption.

V. Result And Evaluation

The model built is not accurate enough for the result and evaluation of the proposed system because there are images that will give a false boundary box for the license plate.Figure 9 depicts the example of error that occurred when determining the license plate's boundary box, where the created boundary box did not fit with the desired license plate segment.Following that, the output for the cropped image will not include the entire rectangle of the license plate, resulting in incorrect information about the license plate identification. The cropped image of the false boundary box is depicted in Figure 11.As a result, the proposed model will produce the incorrect result of the license plate, as shown in Figure 12.The actual identification of the license plate in this case is "CCC 444". However, the output of this model is "CCC 44," indicating that the model is not accurate enough.

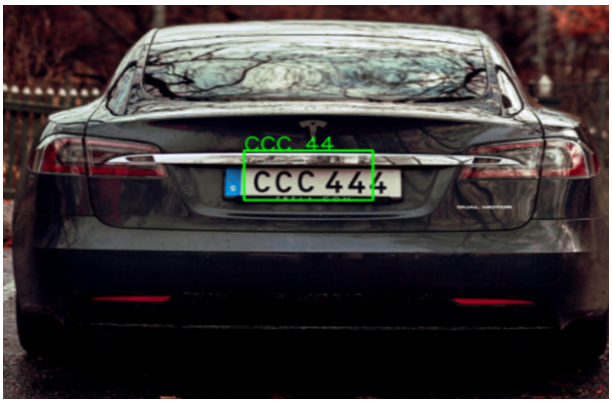

Fig. 11. Example of a figure caption.

Fig. 9. Example of a figure caption.

Fig. 12. Example of a figure caption.

After going through the tensorboard, the model built shows overfitting, which causes the error when detecting the license plate. Figures 10 and 11 show the overfitting of the graph during training and validation, respectively.

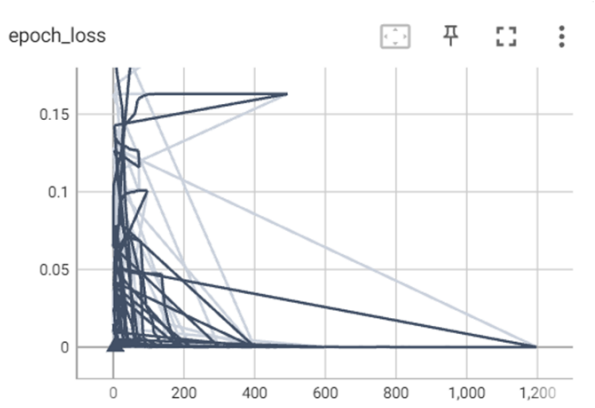

Fig. 13. Example of a figure caption.

VI. Analysis of Overfitting Problems

The analysis of overfitting and its solutions will be discussed in this section.

A. Definition of Overfitting

After obtaining the model's output, it is clear that the model is not accurate enough due to overfitting. The large gap between the loss of the training set and the loss of the validation set, as shown in figure 13, indicates an overfitting problem. One of the primary causes of overfitting is the complexity of the neural network combined with an insufficient number of training sets [16].

As a result, neural networks will over-learn the training data and will over-focus on specific features while ignoring common features [16]. Furthermore, if the features of the training set differ from those of the test set, the neural network will be unable to learn effectively.

B. Overfitting Solutions

The techniques listed below can help to solve the overfitting problem.

1) Data Augmentation: Data augmentation is one of the solutions as it will help on give the diverse dataset such as varying sizes, colors, lighting and condition [17]. The dataset image can also be rotated at different angles or at 90 degrees angles [17]. In addition, the image can be flipped in both horizontal or vertical directions . A lot more data augmentation can be done, such as changing the color space, translation, and noise injection [17].

2) Dropout: Dropout can be accomplished by randomly removing some of the units from the neural network during the training period [17]. Following that, the network will not be entirely reliant on those units [14].

VII. Conclusion

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.