An Improved Visual Attention Model for Automated Vehicle License Plate Number Recognition Using Computer Vision

Ejiofor Martins Ugwu, Onate Egerton Taylor, and Nuka Dumle Nwiabu

ABSTRACT

The role of an automatic licensed plate detection system (ALPDS) cannot be over-emphasized in the world today. The need for an automated system for vehicle license plate number recognition is important for security challenges. Therefore, this paper provides a smart system for vehicle license number recognition using Computer Vision. The system was trained using images of vehicles license numbers as training data. The training images were first annotated using the Visual Graphic Generator (VGG) annotation tool, after the annotation process, the trained images were pre-processed using the OpenCV library for conversion and masking of images. TesseractOCR was then used in extracting just texts from the images. The pre-processed and segmented images were then used in training the Mask R-CNN from a pre-trained weight. The result of the proposed system shows how the Mask R-CNN model was trained in ten training steps. The mask R-CNN model obtained accuracy and a loss value for each training step. The mask R-CNN model was evaluated using both training and test data. For the training and testing data, the Mask R-CNN was evaluated in terms of accuracy and loss. The evaluation was done using graphs. The results from the graph show that the Mask R-CNN had a better accuracy result in both training and testing data. The accuracy for training data was that of 95.25% and the accuracy for the testing data was 97.69%. For real-time vehicle license plate number recognition, we deployed our proposed model to the web. Here, we built a web application that allows real-time surveillance video. Our model was tested on different vehicles in the car park. The result of the mask R-CNN on the test shows how the Mask R-CNN model was used in not just capturing and extracting the vehicle’s license plate number but predicting the characters that appeared on the vehicle’s license plate number. We also compared our proposed system with another existing system. The comparison was done in terms of accuracy, loss, and precision. The result of our proposed model gave us an accuracy of 97.69%, which is higher than the existing system (85%). This study can further be improved by using the Internet of Things in performing live video streaming and also providing a database system that will be storing the predicted vehicle numbers for cars that are detected.

Keywords: Computer Vision, Mask R-CNN, Vehicle License Plate Number, Visual Attention Model.

I. INTRODUCTION

The role of an automatic licensed plate detection system (ALPDS) cannot be over-emphasized in the world today. This is due to the important role it plays in intelligent transport systems with the development of smart city. License plate number detection has attracted good research interest in academic and industrial communities. Although great progress has been made during the past decades. The first automatic licensed plate recognition system was established in 1976 at the police scientific development branch in the UK (Attah, 2016). However, the functionality of a license plate recognition system was very limited. The initial intention of number plate recognition in the Police Force is to prevent unlicensed and auto thefts. Police forces were provided with vehicles mounted with car plate recognition technology. In 2007, the United States incorporated the automatic license plate recognition into the red-light camera network technology to apprehend drivers whose vehicles drove passed the red traffic lights (Yuan et al., 2017). However, despite many algorithms that have been proposed for license plate detection systems, there are need to develop more complex algorithms to tackle some challenges associated with license plate detection such as plate variation, size, color, font,
occlusion, inclination angle, and environmental variation which includes a change in illumination and background, weather conditions, lighting conditions, and even camera conditions may contribute to this problem.

VLPR (vehicle license plate recognition) has sparked a lot of scientific interest. Government agencies can utilize it to track or detect stolen automobiles, as well as collect data for traffic control and development. Because of its close linkages to public security, VLPR requires generalization and high accuracy in real-world applications. Intelligent Transport Systems (ITS) play a significant role in aiding smart cities because of their many applications, including electronic toll collection, highway surveillance, unattended parking lots and security control of restricted areas, urban logistics, and traffic law enforcement. One of the necessary aspects of ITS is the recognition of vehicle license plates, which identifies each vehicle by recognizing the characters on its license plate using image processing and computer vision techniques (Zainal et al., 2017).

Computer vision is an interdisciplinary scientific field that investigates how computers see high-resolution digital images or movies. It seeks to comprehend and automate processes that the human visual system is capable of from an engineering approach. Computer vision tasks include methods for capturing, processing, analyzing, and understanding digital images, as well as the extraction of high-dimensional data from the real world in order to provide numerical or symbolic information, such as judgments (Girshick et al., 2014). Understanding, in this context, refers to the transformation of visual representations (retinal input) into world descriptions that make sense to brain processes and can motivate appropriate action. Image comprehension is the disentangling of symbolic information from visual input utilizing models created with the help of geometry, physics, statistics, and learning theory. Computer vision is a scientific field that studies the science behind artificial systems that extract information from images. Picture data includes video sequences, diverse camera viewpoints, multi-dimensional data from a 3D scanner, and medical scanning data. Computer vision is a branch of technology that tries to apply its theories and models to the creation of computer vision systems (Milan et al., 2008).

II. RELATED WORKS

The processes for vehicle management employing automatic plate number recognition were given by Saqib et al. (2019). By matching different criteria such as aspect ratio, area, and height, plate number width, the distance between letters, and angles between changes in the area, the matching technique was utilized to create a list of probable plates in an image. In a comparison between K Nearest Neighbour and Convolutional Neural Network, they found that Convolutional Neural Network surpasses K Nearest Neighbour in terms of vehicle plate number detection, with an accuracy of about 85%.

Jessie et al. (2017) provided a framework for recognizing the plate number of a color-coding scheme by utilizing character recognition and image processing. The authors accomplished this by constructing a unified vehicular volume program that detects the coding of a vehicle's license plate number in an image and notifies authorities. The identified vehicle's image will be converted to grayscale and then subjected to morphological processing. They used a convolutional neural network technique for segmented character recognition. In summary, their findings reveal that in order for an image of a vehicle's license plate number to be identified and recognized, the image must be processed using appropriate algorithms (Convolutional Neural Network Algorithm), and the processed image's result must be saved in cloud storage.

On a live stream video, Sachin et al. (2017) used automatic license plate number recognition. They used two techniques to train their model: the convolutional neural network algorithm and OpenALPR. OpenALPR is a free and open-source automatic license plate recognition library based on the local binary pattern idea. They used OpenALPR to train their model using an image of an Indian license plate number by adjusting the pattern of the license plate number, and they achieved a minimum recognition rate of 30%. They achieved an accuracy of nearly 90% with a minimal recognition rate of roughly 50% using the convolutional neural network algorithm. Their results reveal that OpenALPR fails to recognize license plate number typefaces, whereas the convolutional neural network method recognizes the license plate number font. This demonstrates that the OpenALPR algorithm is outperformed by the convolutional neural network algorithm.

Sanghyeop et al. (2017) suggested a deep-learning network for recognizing automobile registration plates. They trained a deep learning model using 500 photos of car license plate numbers. The license plate part was extracted. AlexNet learns and recognizes the car number from the retrieved vehicle license plate. The vehicle number recognizes two or more consecutive numerals. The input image is 1392 × 1040 pixels in size, with 20% of the horizontal and vertical planes deleted and the remainder treated as an area of interest. They also used DIGITS to recognize digits and letters first, then increased each recognized area by 40% to find things in their immediate vicinity. The number plates are four consecutive numbers and two consecutive numerals that identify these newly discovered artifacts. The proposed model has a recognition rate of 95.24% and an error rate of 4.76%, according to their testing results.

Abayomi et al. (2020) presented an automatic vehicle plate recognition system based on the Raspberry Pi. A camera was employed to help take plate number photos, and it was linked to a Raspberry Pi processor for authentication. Using Open Computer Vision (Open CV) and Optical Character Recognition, the system can extract numbers from the obtained plate image and completely automate license plate recognition. The system exceeded the majority of the baseline studies analyzed, according to the findings of many testing in diverse locations and conditions.

Oluchi et al. (2019) proposed a method for detecting license plates using existing image processing techniques. Their proposed technique made use of photos from the Caltech collection as well as a dataset from Ahmadu Bello University (ABU). To lessen the processing requirements of the devised system, the obtained photos were pre-processed to detect the license plate. The edge of the pre-processed
photos was detected using the Canny operation, and the contrast of the image spread out using histogram equalization. Edged information was used to extract the region that made up the license plate number, and then Support Vector Machine was used to separate the real license plate from the other regions. On the Caltech and FZU datasets, the performance of the developed technique was assessed. The experimental results demonstrated that their model outperformed certain current methods in terms of detection rate accuracy.

For accurate Chinese Vehicle License Plate Recognition, Shiming et al. (2018) presented an attention-enhanced ConvNet-RNN (AC-RNN). In the stage of recognition, the attention mechanism aids in the location of key occurrences. Recurrent neural networks (RNN) with connectionist temporal classification (CTC) were utilized for sequence labeling while the ConvNet was employed to extract features. The suggested AC-RNN was trained using a huge, generated dataset containing several sorts of Chinese license plates. Even in the presence of light changes, spatial distortion, and partly blurring, the AC-RNN was able to determine the car license. When compared to the standard ConvNet-RNN, their research demonstrated that the AC-RNN works better on real-world picture testing, boosting accuracy by roughly 85%.

Teik et al. (2017) propose a unified ConvNet-RNN model to recognize real-world obtained license plate photographs. They solved the problem of sliding window techniques being unable to access the context of the complete image by supplying the entire image as input to the ConvNet and employing a Recurrent Neural Network (RNN) for sequencing. Using this strategy, they were able to train the complete model from start to finish using labeled, full license plate photographs. Experiments showed that the ConvNet-RNN architecture outperformed a sliding window-based strategy by about 76.53%.

Sweta et al. (2017) proposed using OpenCV to recognize plate license numbers automatically. They began by capturing an image from a camera and loading it into the system; then they utilized OpenCV library capabilities to create a training set of several characters of various sizes. They retrieved the character from photos using these training sets. The digits of the license plate were recognized and approximated location of the number plate. The accuracy can be improved by changing the exact size, color and approximate location of the number plate. Normally the detection algorithm is trained based on the position of the camera, and the type of number plate used in that particular country. In other to find a better training accuracy, the detected images will need to be filtered. Below is pseudocode to filter the image.

```python
for c in cnst:
    # approximate the contour
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.018 * peri, True)
    # if approximated contour has four points, then
    # one can assume that the screen have been found
    if len(approx) == 4:
        screenCnt = approx
        break
```

Mathematical Representation of image filtering can be seen below.

\[
w = \frac{(w-f+2p)}{x+1}
\]

where \(w\) is the width of the resulting image, \(f\) is filter size, \(P\) is padding size, and \(x\) is stride.

The Gaussian function will be used to refine the image for better results; the mathematical expression is given below:

\[
G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}
\]

where \(G\) is the Gaussian function with \(x\) and \(y\) as the input data, \(\sigma\) is the learning rate, and \(e\) is the edge of the object. The
input x and y represent the x-coordinates and y-coordinates of the image (Vehicle license plate number).

D. Haar Cascade Classifier

This is a computer vision class that teaches you how to detect objects in a video stream. A cascade classifier is a machine learning strategy that involves training a cascade function using a large number of positive and negative images. It’s used to find things in other photos. To detect several automobiles in an image, we will use cascade classifiers. A cascade classifier’s architecture is shown in Fig. 1. The pseudocode for the Haar cascade classifier is shown below.

Step 1: Pick f (maximum and minimum false positive rate per layer and d (minimum acceptable detection rate per layer).

Step 2: Let \( F_{target} \) be target overall false positive rate.

Step 3: Let P be the set of all positive examples.

Step 4: Let N be a set of all negative examples.

Step 5: Let \( F_0 = 1 \), and i=0 (\( F_i \): overall false positive rate at layer 0, \( D_i \): Acceptable detection rate at layer 0, and i: is the current layer).

While \( F_i > F_{target} \) (\( F_i \): Overall false positive rate at layer i).

Step 6: i++ (layer increasing by 1).

Step 7: \( n_i = 0 \); \( F_i = F_{i,ini} \) (\( F_{i,ini} \): negative example i).

Step 8: While \( F_i > f^*F_{i+1} \):

Step 9: ni++ (check a next negative example).

Step 10: Use P and N to train with AdaBoost to make xml (Classifier).

Step 11: Check result of new classifier for \( F_i \) and \( D_i \).

Step 12: Decrease threshold for new classifier to adjust detection rate \( r_i = d_i^*F_{i+1} \).

Step 13: N= empty.

Step 14: if \( F_i > F_{target} \), use the current classifier and false detection to set N.

E. Character Recognition

TesseractOCR is an optical character recognition (OCR) engine that is free and open source. It is widely regarded as one of the most widely used and accurate open-source OCR engines. This engine was originally developed by Hewlett Packard as proprietary software, but it was later open sourced in 2005, and its development has since been supported by Google.

F. Mask R-CNN

Mask R-CNN is a state-of-the-art model for instance segmentation, developed on top of a faster R-CNN (Kaiming et al., 2017). Mask R-CNN is structured with different stages. The stages can be described as follows:

i. Stage 1: The primary stage comprises two networks, spine (ResNet, VGG, Inception, and so forth) and region proposition network. These networks run once per picture to give a bunch of region recommendations. Region proposition are areas in the component map, which contain the object.

ii. Stage 2: In the subsequent stage, the organization predicts bounding boxes and object class for every one of the proposed areas acquired in the stage1. Each proposed region can be of various sizes while completely associated layers in the organizations consistently require fixed-size vectors to make expectations. Size of these proposed districts is fixed by utilizing either RoI pool (which is very much like MaxPooling) or RoIAlign technique.

IV. RESULTS AND DISCUSSION

This paper presents a smart system for automatic vehicle license plate number recognition. The system starts by acquiring a dataset. The dataset used here in this work comprises 66 vehicle license plate numbers for training and 20 vehicle license number for testing. The dataset was pre-processed into stages. The first stage of the pre-processing has to do with the annotation of the images. The annotation of the image was done using the VGG annotation tool. The annotation tool was used in creating a bounding box for the license plate number. Fig. 2 shows the annotated image with a bounding box on the license plate number. The second phase of the pre-processing has to do with the conversion of the images to grayscale. Finally, the last stage of the pre-processing has to do with applying of filter and finding the localization of the vehicle license plate number and extracting just the information needed. Fig. 3 shows the masked image. The vehicle license plate number was detected using a green rectangular box. This can be seen in Fig. 4. The extracted license plate number was further extracted by performing some segmentation, grayscale conversion, and extracting each of the numbers in the vehicle license plate number. This was achieved by putting a rectangle on each of the text written on the vehicle license plate number. This can be seen in Fig. 5. The experimental result shows the building of a smart system for recognizing vehicle license plate numbers using computer vision. The computer vision algorithm used in this work is Mask-R-CNN. The experiment was carried out in two phases. The first phase has to do with the image segmentation using the VGG annotation tool, pre-processor the annotated images by transforming them into grayscale so that the edges of the vehicle license plate number can be seen clearly by the
computer vision model. In the second phase, the pre-processed images were used as input for the Mask R-CNN. Table I shows the performance measure the Mask R-CNN obtained during training. The performance measure is evaluated in terms of accuracy, loss, and the runtime the system took in completing each training step. This can be seen in Fig. 6 and 7. Fig. 8 shows the predicted characters on the plate number.

**TABLE I: EVALUATION MATRIX OF THE FIRST TEN TRAINING STEPS**

<table>
<thead>
<tr>
<th>Training Step</th>
<th>Time/step (secs)</th>
<th>Training Accuracy (%)</th>
<th>Validation Accuracy (%)</th>
<th>Training Loss (%)</th>
<th>Validation Loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58</td>
<td>11.92</td>
<td>3.19</td>
<td>35.19</td>
<td>2.13</td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>54.40</td>
<td>1.49</td>
<td>73.61</td>
<td>0.93</td>
</tr>
<tr>
<td>3</td>
<td>55</td>
<td>74.07</td>
<td>0.75</td>
<td>87.04</td>
<td>0.39</td>
</tr>
<tr>
<td>4</td>
<td>57</td>
<td>85.65</td>
<td>0.48</td>
<td>87.84</td>
<td>0.49</td>
</tr>
<tr>
<td>5</td>
<td>55</td>
<td>88.66</td>
<td>0.35</td>
<td>92.59</td>
<td>0.19</td>
</tr>
<tr>
<td>6</td>
<td>55</td>
<td>92.82</td>
<td>0.23</td>
<td>93.98</td>
<td>0.12</td>
</tr>
<tr>
<td>7</td>
<td>56</td>
<td>93.06</td>
<td>0.20</td>
<td>96.30</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td>56</td>
<td>94.10</td>
<td>0.17</td>
<td>96.30</td>
<td>0.08</td>
</tr>
<tr>
<td>9</td>
<td>57</td>
<td>94.44</td>
<td>0.16</td>
<td>94.91</td>
<td>0.09</td>
</tr>
<tr>
<td>10</td>
<td>55</td>
<td>95.25</td>
<td>0.13</td>
<td>97.69</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Fig. 2 shows a data augmentation process that was used in drawing a bounding box for each of the training images that were used in training a Mask R-CNN model that was used in training a computer vision model for vehicle license recognition. The images were annotated using the VGG annotation tool. After the process of annotation, the images were then pre-processed by reading the training images into a directory, converting the images to gray scale, and drawing a mask on the images.

Fig. 3 shows the masked result of the training image. The image was masked so as to identify the locations (regions or coordinates) of the training images. This was done so as to enable the Mask R-CNN to quickly identify just the vehicle license plate number.

Fig. 4 shows the rectangular box that was used in recognizing the vehicle license plate number. After the masking of the training images, it was easy in extracting just the vehicle license number instead of capturing the whole image of the vehicle, with this, the extracted license number can further be divided into various segments.

Fig. 5 shows are the extracted vehicle license plate number was divided into various segments.
The accuracy (Fig. 6) shows the performance of the model for both training and testing data. The training accuracy can be represented using a blue line whereas the testing accuracy can be represented using an orange line. Both training and testing performance achieved almost accuracy of about 97.6%.

![Fig. 6. Model accuracy versus the number of epochs (The number of training steps).](image)

The graphical representation (Fig. 7) shows the loss of the model for both training and testing data. The training loss can be represented using a blue line whereas the testing loss can be represented using an orange line. Both training and testing had loss values of less than 0.5%.

![Fig. 7. A graphical representation of the Model loss for both training and testing.](image)

Here, the proposed model predicted each of the text written on the plate number correctly.

![Fig. 8. Predicted Result.](image)

V. CONCLUSION

The need for having an automated system for vehicle license plate number recognition is an important one for security challenges. Therefore, this dissertation provides a smart system for vehicle license number recognition using Computer Vision. The system was trained using images of vehicles’ license numbers as training data. The training images were first annotated using the VGG annotation tool, after the annotation process, the trained images were pre-processed using the OpenCV library for conversion and masking of images. TesseractOCR was then used in extracting just texts from the images. The pre-processed and segmented images were then used in training the Mask R-CNN from a pre-trained weight. The result of the proposed system shows how the Mask R-CNN model was trained in ten training steps. The mask R-CNN model obtained accuracy and a loss value for each of the training steps. The mask R-CNN model was evaluated using both training and test data. For the training and testing data, the Mask R-CNN was evaluated in terms of accuracy and loss. The evaluation was done using graphs. The results from the graph show that the Mask R-CNN had a better accuracy result in both training and testing data. The accuracy for training data was 95.25% and the accuracy for the testing data was 97.69%. For real-time vehicle license plate number recognition, we deployed our proposed model to the web. Here, we build a web application that allows real time surveillance video. Here, we tested our model on different vehicles in the car park. The result of the mask R-CNN on the test shows how the Mask R-CNN model was used in not just capturing and extracting the vehicle’s license plate number but predicting the characters that appeared on the vehicle’s license plate number. We also compared our proposed system with another existing system. The comparison was done in terms of accuracy, loss, and precision. The result of our proposed model gave us an accuracy of 97.69% which is higher than the existing system (85%). This study can further be improved by using the Internet of Things in performing live video streaming and also providing a database system that will be storing the predicted vehicle numbers for cars that are detected.
REFERENCES


