2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) | 979-8-3503-3250-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/UPCON56432.2022.9986360

Number Plate Detection Using Drone Surveillance

1st Swati Jain
Dept. of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India
swati.jain@nirmauni.ac.in

2nd Shaili Patel
Dept. of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India
18bce168@nirmauni.ac.in

3nd Avi Mehta
Dept. of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India
18bce019@nirmauni.ac.in

4rd Jai Prakash Verma
Dept. of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India
jaiprakash.verma@nirmauni.ac.in

***Abstract*—Drones, or unmanned aerial vehicles, have a wide range of uses in a variety of sectors. Surveillance, photography, surveying physically difficult locations, and traffic patrols are some of the applications. License plate detection and identification by utilizing UAVs based video streaming is one such scenario on which we have engaged. Recognizing license plates in drone photographs is a difficult task since the images may be distorted, blurred, or contain background noise such as other cars, banners, or people. The automatic license plate detecting system is a well-established system in which the majority of research has been conducted. These approaches, on the other hand, concentrate on photographs taken from the front. The suggested study is limited to a subset of drone photos. We examined multiple state-of-the-art methods like Wpodnet and yolov5 for detecting license plates in subsets of drone images in this paper.**

***Index Terms*—Object Detection, Drone, License Plate, Text Recognition, Deep Learning**

I. Introduction

Due to several real world applications, including such automated toll booths, traffic rules and regulations compliance, private business access management, and traffic congestion surveillance, automatic license plate recognition (ALPR) has been a popular study area [1]. License Plate (LP) scanning, character segmentation, and character recognition are the three steps of ALPR technology [2]. Because inability to identify the LP would very certainly result in a failure in the future phases, the early steps demand better reliability or near-perfection. Numerous methods look for the vehicle foremost, then the LP, to decrease computational time and decrease false positives. Despite the fact that ALPR has been discussed extensively in academia, many research and approaches are still insufficiently resilient in realistic conditions. Several limitations, notably as various cameras or viewing angles, plain backdrops, acceptable illumination variations, scan in a defined zone, and particular types of vehicles, are usually used in these methods. In the past, numerous approaches based on sophisticated deep learning models for detecting license plate numbers were created to solve the issues of reduced contrast, complicated backgrounds, font size fluctuations, and other factors. These strategies, nevertheless, do not operate well with drone photographs.This seems to be due to the difference in elevation, proximity and oblique angle between head-on and skewed pictures. The camera's focus spans across cars since drones frequently contain many automobiles in a single shot. As a result, the proportion of cars in a single picture fluctuates in proportion to the altitude distance, influencing the effectiveness of each license plate number that is detected and recognized.

An ideal ANPR system passes through different steps: picture capture (input to the system), number plate extracting (NPE), character segmentation (CS), and character recognition (CR).The data may be retrieved and utilized for post-processing procedures after the vehicle has been successfully detected. The data from the automobile is delivered to the associated back office system software, which serves as a single repository for all data and includes capabilities for data analysis, inquiries, and monitoring. A technique known as Number Plate Extraction mainly a system employs plate localization capabilities to derive the license plate from the automobile picture when a car is spotted in the scene/image. Before the recognition procedure, the characters on the retrieved number plate are divided. Text segmentation is a technique for locating alphanumeric characters on a license plate. Optical character recognition (OCR) algorithms are then used to convert the fragmented characters into an alpha numeric text input. Algorithms like template matching and neural network classifiers are employed for character recognition. The efficacy per each phase of an ANPR system determines its overall efficiency. The outcomes or progress rate, which would be the measure of the number of number-plates correctly identified to the overall number of input photos obtained, which is a measure used to quantify the entire workflow. All three steps of the recognition process, number plate extraction, fragmentation, and character recognition, are included in the performance level.

II. Related work

In [3], several methods have been proposed previously. Here we have reviewed these methods for license plate detection. It aims at achieve higher accuracy for complex senarios and recognise characters in license plate. First, vehicle is extracted from the image and then characters are recognised using CNN model & Model's accuracy is 99% and has computing time of 30fps on test dataset. Limitations are Imbalance of chinese characters. In [4], A Single Neural Network for Mixed Style License Plate Detection and Recognition. In this research work, A novel model called ALPRnet is proposed which is used to detect multistyle license plate. Two convolution single-stage networks are used to detect the bounding box and recognise the characters in the license plate. The LP recognition accuracy rate is 98.21%. In this, manual annotation of license plate was a tedious job. In [5], V-LPDR: Towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos, authors proposed a deep learning based framework for detection of license plate in complex scenarios which is video based. The Plate recommender and recogniser obtained an accuracy of 95.44%. In [6], A new DCT-PCM method for license plate number detection in drone images, a new problem in license plate detection for drone based image is addressed. The proposed method uses combination of DCT and phase congruency to detect Number plate text in image. Further, uses fully connected deep learning networks to achieve better accuracy. In [7], ES-Yolov3-tiny is a novel end-to-end detection network is proposed. The secondary detection layer would process the results of first detection layer so that it gets successfully identified by the detector. Better accuracy and lower computational cost than other state-of-art models Vehicle position. In [8], Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion, authours propose a framework for resolving the challenges faces in scene text detection Integrated differential binarization module in binarization process. In [9], Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications. The system is divided into three part: Segmentation, detection and recognition. In detection phase, after CCA, RANSAC(random sampling consensus) is applied. RANSAC is used in stereo vision. The accuracy of 91% achieved on iranian dataset and 97% for english dataset.

III. Case Study based Impact Analysis

License plate recognition has many useful applications like traffic monitoring, parking management systems etc. With various applications comes different challenges that one might face while building this application. In [10], the authors have categorized challenges based on external and internal factors. Challenges due to external factors include plate variation, license plate color, camera mounting, License plate font, Lightning condition, surrounding effects and many more. Challenges due to internal factors include camera quality, camera shuttering speed, camera resolution etc. Rotation/ Skewness of License plates is a serious concern. Tilt occurs when the road is not straight, uneven or bumpy. Classifying characters from a skewed license plate is a difficult task and hence some techniques must be applied in order to correct the skewness. In order to solve this, the hough transformation can be used. By using the hough transformation method, we first find the angle by which our License plate rotates and then we rotate it in the opposite direction. A genetic neural network(GNN) as mentioned in is another way to tackle this problem especially when the borderlines of license plates are not correctly visible. The genetic algorithm used is written in a manner that it matches the classification thus saving computation time. For example, on the Hong-Kong - Macao highway, drivers integrate Macao LP over Hong-Kong LP which causes contour extraction problems [4]. Masked R-CNN are used for these multi-style license plate detection . They have proposed an end-to-end trainable method which performs three tasks: first is character instance segmentation, second is license plate instance segmentation and third one is assembly task. To build the feature pyramid, they have used ResNet as their backbone architecture [4].

At Barry University [11], due to illegal parking issues, the university has come up with a drone based license plate recognition for parking management. The drone moves through the parking lot and clicks pictures of the cars parked. The issue they faced was that they had to find a route for the drone that is safe. That means, the drone's path should not be hampered by any external object like birds, trees or humans. Furthermore, there are some places/corners where the angle of the camera in the drone needs to be changed or the height of the drone needs to be changed in order to have a car license plate in the line of sight. So randomly flying the drone and clicking images is not a good idea here. To automate the process, they used the concept of graphs [11]. A node point is the point from which the drone is released. The nodes cover a large view of parking lots. The weights of the undirected graph represents the walking distance between the nodes. The distances are programmed in the drone using GPS coordinates. The images from the drone are clicked in such a way that the efficiency of the image processing algorithm is improved. The license plate is positioned at the center of the image. OPENALPR, an open source automatic license plate detection algorithm is used to detect license plates. To overcome the issue of bad character recognition, map data structure is used. License plate string is used as the key, and the frequency of the key is the value. every

Fig. 1. Implementation of Hough Transformation

time the same string appears, value would be incremented by one. The more images the program analyzes, the better the result [11].

IV. Proposed Research Work

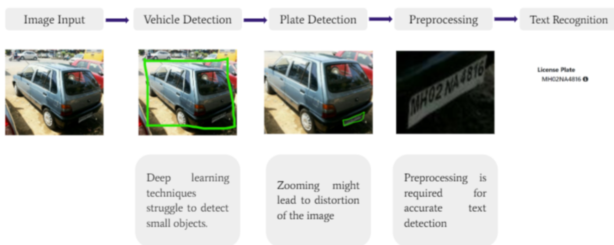

Fig. 2. Flow of the methodology

Figure 2 describes the flow of the implementation which has been further discussed in the paper. The Data set has annotations of the vehicle as well as the license plate, when fed into the deep learning model, the algorithm creates a bounding box around objects depending on the layers and filters which are implemented. This object detection phase is of utmost importance as it forms the foundation of further processes on image recognition and preprocessing. The objects are only bounded if the IOU produces a confidence of greater than 50 percent. Once the object of concern is detected, over here, being the license plate, we crop the image and store it in another location. The detected image is quite raw and cannot be fed to an OCR or text recognition algorithm. The second pass of the image is to run a text recognition algorithm on the preprocessed image. The goal of pre-processing is to increase the image's resolution so that we can objectively evaluate it. We can reduce unwanted distortions and boost some properties that are important for the application we're working on by preprocessing. These characteristics may alter depending on the application. As the images are quite difficult to detect mainly due to skewness of the license plates. The final output is received as a string output which can finally be used for further analysis.

A. Dataset Selection: A Novel Dataset

Literature survey was carried out before moving ahead with the methodology of the proposed work. The one unique thing which was noticed was that there had been no prior work carried out on images specially dedicated to Drones based imagery. The dataset, is customly tailored, which is a mix of roboflow, ALPR dataset and a few hand-clicked pics of some of the local cars in india. Natural scenes like pics, that have low clarity, obtuse angles, obscurities, illumination variation, night images etc.

The Number Plates dataset is a computer vision applications dataset that includes automobiles, vans, and other vehicles, as well as their license plates. Examples of "vehicle" and "license-plate" are also included in the annotations. The train/validation/test split for this dataset is 245/70/35, accordingly. The Open Images Dataset is a subset of this dataset. Google LLC has granted the annotations a CC BY 4.0 license. To better match the annotations with the aim of the dataset, Roboflow's annotation management tools were used to merge or eliminate some annotations.

V. Methodology and Concepts

First we transform our picture/image to gray scale to enhance identification and substantially limit the quantity of colors present in the image, which greatly aids in the recognition of license plates. Then we denoise the image. Gaussian Blur improves the clarity and smoothness of the borders, making the lettering more accessible. We then do canny edge detection to detect the edges in the images. To find the edge, canny uses a threshold value. If the pixel value is greater than the threshold then it is accepted as an edge. After edge detection, we need to find all those likes that join or are along the boundary of the image. These are known as contours. We will get a list of coordinates as our output. We will sort them according to their area. from the sorted area, we will find a shape that resembles a license plate. Once we find the location coordinates of the license plate, we will black out the rest of the background. This is called masking. After this, we will crop the image. This is how license plates are detected using OpenCV. WPODNET stands for warped planar object detection network. This approach has three steps: first is to detect the car from the input image, then we detect the license plate from the image and finally we rectify the license plate detected. Further, this rectified license plate is fed to OCR or any other text recognition system. In WPODNET [12], YOLOv2 is used to detect a car from the image. YOLOv2 was chosen as it was fast and gave good precision (around 76% mAP). WPODNET returns coefficients of affine transformations that would reorient the image into rectangle like shape. This model is developed using the insights from YOLO, STN (spatial transformation network) and SSD (single shot detector). While YOLO and SSD are good at fast detection of objects, STN would help us locate non-rectangular regions in the image [12].

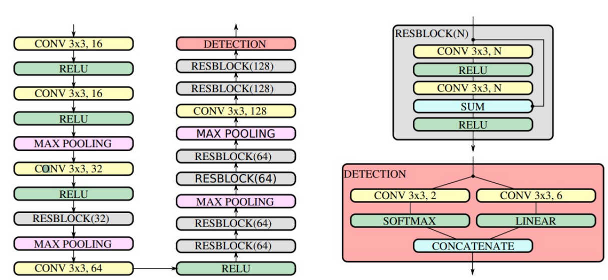

Fig. 3. WPOD-NET Architecture [12]

The detection block has 2 running layers. One is for finding probability and the other is for returning affine transformations [12]. In tesseract [**?**], processing is done in a step by step manner. The line detection technique is meant to detect a tilted page while not having to deskew it, preserving quality of the image. Blob filtering and line creation are critical steps in the process. And once text lines have now been identified, the baselines are more accurately conformed using a quadratic spline. Tesseract overcomes these kinds of issues by measuring gaps between the baseline and mean line. At this step, spaces around the threshold are made fuzzy so that a final judgment may be made following word recognition [13]. The next part of the step is word recognition. All the joined characters are chopped first. and these chop points are found using polygon approximation. Once all the chopping is done, The output is given to the associator. It does a best first search of chopped blobs to its characters. It is then sent to an adaptive classifier which also makes a difference between upper and lower case [13]. Object recognition is a broad word that refers to a group of computer vision tasks that include recognising things in digital pictures. Forecasting the class of one item in a picture is called image classification. Object localization refers to recognising the position of one or more items in a picture and creating a region of interest around their size. Object detection incorporates the following objectives of, locating and classifying one or more items in a photograph with high accuracy. The amount of things in the backdrop might change between photos, which complicates object recognition. If each image contains only one item, determining a bounding box and classifying the subject is a simple task. Because the bounding box is made up of four values, knowing its position is simply treated as a regression issue. The item is then classified, which is a classification challenge. For our constrained object detection task, the convolutional neural network (CNN) presented in Figure 2 solves the regression and classification difficulties. Our constrained object identification issue, like other classic computer vision tasks involving image recognition, key distinction detection, and semantic segmentation, has a set of possible targets. Designing the objectives as a set number of classification or regression tasks can help suit them.

VI. Execution and Implementation

The flow of implementation goes like this: first eliminate background noises like boards, humans, other cars, trees, dustbins etc manually or deep learning algorithms like YOLOv5. A license plate occupies less than 10% of the image. To find that very small portion in the image, we need to train deep learning models that would take hours and hours. So to improve or reduce the computation complexity, we can eliminate all redundant background noise and focus only on the vehicle we want to detect the plate from. In this section, we have discussed various process flows of the approaches we used.

A. Yolov5

The model is available in various configurations, ranging from small to large, each having a specific number of layers and connections between them. The new factor of this model is that it has a genetic algorithm clubbed with k- means to automatically create anchor Box.

A series of predetermined boundary boxes of a specific height and width are known as anchor boxes. Such boxes are often chosen depending on item dimensions in the training datasets to reflect the dimensions and aspect ratio of various object categories users wish to identify. A brief on the architecture is discuss below.

Model Backbone : The basic purpose of Model Backbone is to capture key characteristics from an initial picture. The CSP — Cross Stage Partial Networks — backbone is utilized in YOLO v5 to retrieve rich and advanced attributes from a source image.

Model Neck : The primary objective of Model Neck is to create feature pyramids. Feature pyramids aid model generalization when it comes to object scalability. It aids in the identification of the similar entity in various dimensions and scales.

The Model Head is mostly utilized for the penultimate detecting step. It produces final output vectors with class probabilities, object classes scores, and bounding boxes by applying anchor boxes to attributes. The Model by default has two optimization functions and those are Adam and Stochastic gradient descent(SGD).

In our training phase of the model , the SGD function has been used as generalization over new license plates is of more importance to us as to tackle the various challenges posed by our dataset. On diving deep into the architecture of the model , the YOLO v5 has been implemented : the Leaky ReLU activation function is employed in the middle/hidden layers, while the sigmoid activation function is used to create the data in the terminal classification layer.

The Model as first was trained for 10 epochs to test the run , later on the model was further trained on epochs 250 with a batch size of 2 and a dataset containing 400 images of vehicles. After successfully training the model, the below images show the successful detection of the number plate as well as the vehicle being identified and a bounding box is drawn around to mark the boundaries of the label class.

B. Pytesseract/OCR

Python-tesseract is a python-based optical character recognition (OCR) programme. [13] It will detect and ”read” text contained in photos, in other words. Python-tesseract will also output the recognised text rather than saving it to a disk when used as a script. To circumvent all of the potential problems with your tesseract result precision, make absolutely sure the picture is properly pre-processed.This can involve rescaling, binarization, noise reduction, deskewing, and other techniques. Use one of the many below python routines or the OpenCV manual to pre - process pictures for OCR. The electrical or mechanical translation of images of composed, handwritten, or printed text into machine-encoded data, whether from a pdf document or a picture of a page, is known as optical character recognition (OCR). OCR is a procedure that includes multiple sub-processes that must be completed as precisely as

Fig. 4. Results of using Pytesseract on our cropped license plate images

possible. The subprocesses are: a) Image to be pre processed. b) Localizing the text data. c) Segmenting the Characters. d) Character Recognition. e) Post Processing.

C. Scene Text Recognition(STR)

Scene text are of two types: one is scene text detection and the other is scene text recognition. In scene text detection, text from the external surroundings is detected. while in scene text recognition, text from the detected image is recognised. STR can be useful in license plate recognition. Since license plates have numbers and alphabets, it is easier to implement text based searching than finding rectangular shapes. We have implemented both of these frameworks to see the accuracy in our dataset. In scene text detection, we have used easyOCR. EasyOCR is an open source pretrained model for scene text recognition. It has the potential to detect text in 80+ languages. In scene text recognition, we implemented a four stage framework [8]. The framework was trained on multiple datasets like synthetic word dataset, character dataset etc.

VII. Results and Discussion

A. OpenCV + OCR based Implementation

Here, we tested our method using 120 images which were a subset of drone-based images. After the detection of the license plate, we gave the detected license plate image as an input to OCR. Tesseract is an optical character recognition engine which allows us to fetch text from the images. We looped through all the images in the test folder and stored the text of the license plate in a list. Later we compared the results of the OCR with an online OCR. We did this so that we can get to know the efficiency of the tesseract. In the preprocessing step, for the Open ALPR dataset, we achieved an accuracy of 30.88% for license plate detection. Out of 120 test images, only 37 of them were correctly identified. The rest of them were either incorrectly identified or were not at all identified. Since in this method we try to find a rectangular figure in the image, it may happen that due to the high degree of skewness of the camouflage colour of the license plate, the plate may not be detected. Other times it may happen that the headlights or the car bonnet may be detected as a license plate. All these types of false results may occur due to the shape edges of that region when compared to the license plate region.

Fig. 5. Results of using STR on our cropped license plate images

B. OpenCV + STR based Implementation

STR stands for scene text recognition. It detects text from the external environment. Usually scene text problems come with two terms: scene text detection and scene text recognition. In the former, we detect the text from the image and in the latter, we recognise the character in the detected text. We tried to implement scene text recognition directly into our cropped license plate image that we got after the LP detection step. The below figure shows the results.

In scene text recognition, the accuracy was even lower than pytesseract. The reason behind this is that, In scene text recognition architecture [14], there are 4 stages: transformation, feature extraction, sequence modelling and prediction. In the sequence modelling step, we predict any character based on the previous words. Since the below types of words are used more frequently, it is easy to predict what comes after “LONDO”. Hence, “London” is predicted with greater confidence than the license plate number.

C. WPOD-NET + OCR based Implementation

WPODNET stands for warped planer object detection network [12]. This model was created to overcome the issues of unconstrained scenarios in license plate detection. WPODNET return one affine transformation for every detection and does rectification like rotation, centring, cropping etc [12]. We tested this model on our dataset. The accuracy we achieved is 87%. The model was able to detect license plates very well in front as well as rearview. However, the OCR accuracy was less due to poor quality image. Successfully detected license plates are fed as an input to pytesseract- OCR engine. 55% of the positive license plates were correctly recognised.

D. YOLOv5 + OCR based Implementation

In the first pass of the image , the license plate is detected first , in order to test it we have utilized the detect.py file . To run the yolov5 on our custom images , we need to specify the paths and weights that are to be used.The annotators outline the objects in boxes as per the project requirements. When looking for a car the algorithm only searches in the bounding boxes labeled cars rather than looking for it in the whole image. The bounding box contains coordinates which has information about where exactly the object resides in the image.The image shows the coordinates of the bounding box annotation. To locate the automobile inside this image, the method directs the program to search exclusively inside these dimensions rather than through the entire image. Thereby easing the detection job of the model. However, just a single bounding box cannot enable a 100 percent prediction rate in the model.We need to provide the algorithms a higher volume of bounding boxes or basically ”training data” for improved feature recognition in the picture for this to work.

As seen from the above, the image used to test two license plates , one having a straightforward orientation and the other having a skewed orientation , both are detected with accuracy of 93 percent and 70 percent respectively.

The PR Curve that is obtained provides further information

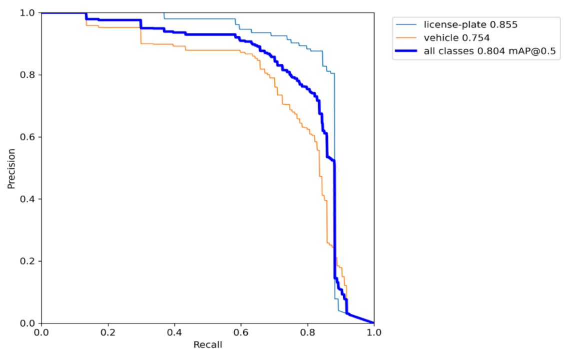

Fig. 6. PR curve

about the model’s projections as well as the threshold values, eventually providing us the mean average precisions (mAP). For various thresholds, the precision-recall curve depicts the trade off between precision and recall. A large area under the curve indicates good recall and precision, with high precision indicating a low false positive rate and high recall indicating a low false negative rate. A value of one precision and one recall value , that graphically is the top right corner of the curve , is known to be a perfect classifier . This is only hypothetically possible in a simulated environment. The cropped license plates were sent to OCR engine for character recognition. The below image shows the output.

Fig. 7. Results of using WPODNET with Pytesseract

VIII. Conclusion and Future Work

This paper explored different state-of-art algorithms for car license plate detection for drone based images. The YOLOv5 , WPODNET approach was successfully established in this study to recognize motor car license plates. Data preparation, preprocessing, data annotation, training, and testing are the stages done. We have evaluated the system using a benchmark dataset of drone images, which was customly edited for which we were able to achieve 85 percent mean accuracy. But it is examined that the dataset is low resolution so the quality of license plate is very low. Therefore, it is expected that a higher resolution drone data will have a much higher accuracy. Drones can be adjusted by height and angle; hence the images of license plates captured can be of good quality. One observation that we have made is that license plates in frontal view are accurately and efficiently detected when compared to skewed drone images. We can adjust the drone in a manner that the license plate is in the center of the image. This way we can reduce the processing time. For future scope, we aim to implement the model firstly on our android devices and later calibrate it with a Drone to test the model in real life scenarios and work.

References

[1] V. Lukic, A. Makarov, and M. Španović, “An application of image point operators for ocr improvement in alpr algorithm,” in *2012 20th Telecommunications Forum (TELFOR)*, 2012, pp. 697–700.

[2] A. Prayudi, I. A. Sulistijono, A. Risnumawan, and Z. Darojah, “Surveillance system for illegal fishing prevention on uav imagery using computer vision,” in *2020 International Electronics Symposium (IES)*, 2020, pp. 385–391.

[3] N. Duan, J. Cui, L. Liu, and L. Zheng, “An end to end recognition for license plates using convolutional neural networks,” *IEEE Intelligent Transportation Systems Magazine*, vol. PP, pp. 1–1, 04 2019.

[4] Q. Huang, Z. Cai, and T. Lan, “A new approach for character recognition of multi-style vehicle license plates,” *IEEE Transactions on Multimedia*, vol. 23, pp. 3768–3777, 2021.

[5] C. Zhang, Q. Wang, and X. Li, “V-lpdr: Towards a unified framework for license plate detection, tracking, and recognition in real-world traffic videos,” *Neurocomputing*, vol. 449, pp. 189–206, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221004951

[6] H. Mokayed, P. Shivakumara, H. H. Woon, M. Kankanhalli, T. Lu, and U. Pal, “A new dct-pcm method for license plate number detection in drone images,” *Pattern Recognition Letters*, vol. 148, pp. 45–53, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865521001690

[7] Y. Peng, H. Li, and Z. Qian, “A new end-to-end secondary network for high-efficient vehicles and license plates detection,” pp. 6–9, 2019.

[8] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, “Real-time scene text detection with differentiable binarization and adaptive scale fusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[9] R. Panahi and I. Gholampour, “Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.

[10] P. Mukhija and P. Dahiya, “Challenges in automatic license plate recognition system: An indian scenario,” 07 2021.

[11] J. Dasilva, R. Jimenez, R. Schiller, and S. Zivanovic, “Unmanned aerial vehicle-based automobile license plate recognition system for institutional parking lots,” *Journal of Systemics, Cybernetics and Informatics*, vol. 15, pp. 39–43, 10 2017.

[12] S. M. Silva and C. R. Jung, “License plate detection and recognition in unconstrained scenarios,” pp. 580–596, Sep 2018.

[13] R. Smith, “An overview of the tesseract ocr engine,” vol. 2, pp. 629–633, 2007.

[14] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, “What is wrong with scene text recognition model comparisons? dataset and model analysis,” pp. 4714–4722, 2019.