iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VII Month of publication: July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36889

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Automatic Number Plate Recognition Using Image Processing

Hrithik Roshan Palampatla[1], Karthik Yeluripati[2], Koushik Choppakatla[3]
[1] Department of Electronics and Communication Engineering, CBIT, Hyderabad
[2, 3]Department of Electrical and Electronics Engineering, CBIT, Hyderabad

Abstract: Automatic Number Plate Recognition (ANPR) is a mass surveillance system that captures the image of vehicles and recognizes their registration number issued by government. ANPR is often used in the detection of stolen vehicles, traffic surveillance system. Our project presents a model in which the vehicle license plate image is obtained by the digital cameras and the image is processed to get the number plate information. A vehicle image is captured and processed using various methods. Vehicle number plate region is extracted using the deep neural networks. Optical character recognition is implemented using certain machine learning algorithms for the character recognition. The system is implemented using deep neural network model, machine learning algorithms and is simulated in python, and its performance is tested on real images. It is observed that the developed model successfully detects the license plate region and recognizes the individual characters. There are various recognition strategies that have been produced and number plate recognition systems are today used in different movement and security applications, such as access and border control, parking, or tracking of stolen vehicles.

Keywords: Machine Learning, Deep Learning, Image processing, Number plate recognition, You only look once (YOLO).

I. INTRODUCTION

The number of cars on the road has increased significantly in recent years, resulting in traffic congestion and fines. Every country's traffic management and car owner identification has become a key issue. It might be difficult to identify the owner of a car that breaches traffic laws. As a result, it is impossible to apprehend and penalize such individuals since traffic officers may be unable to acquire car identification numbers. As a result, one of the answers to this challenge is to design an ANPR system. The Automatic License Plate Recognition system can also be used to automate traffic management, improving traffic flow and enhancing access control systems. ANPR is a technique for extracting the licence plate number from a still image or video of a moving or stationary vehicle. Automatic Number Plate Recognition (ANPR) systems can perform a variety of traffic-related applications, including the detection of stolen vehicles, the classification of illegal vehicles, the capture of speed limit violators, toll control, and the expediting of parking by eliminating the need for human confirmation of parking passes. The ANPR was developed in 1976 at the Police Scientific Development Division in the United Kingdom. However, it grew in popularity during the previous decade as digital cameras improved and computational activity increased. It is essentially the capability to extract and recognise a car number plate character from a picture automatically.

II. LITERATURE REVIEW

Rahul R. Palekar, Sushant U. Parab, and Dhrumil P. Parikh suggested “Real Time License Plate Detection Using OpenCV and Tesseract” in a paper [1]. Python IDE is the programme used for license plate detection. The planned research is divided into many parts, the first of which is the collecting of photographs consisting of license plates. In the second step, several pre-processing methods such as RGB to grey, blurring, and thresholding are used to achieve high accuracy in recognizing the license plate area; the Histogram approach is used to acquire statistical data from an image and classify it as bad or excellent. In the third step, the processed picture is passed into Tesseract OCR Machine, which uses a command line interface to turn the detected license plate picture into text. The approach is also effective for identifying perpendicular pictures with high resolution, as shown by the trials and results. “License Plate Recognition Using Convolutional Neural Networks,” proposed Qinghong Wang. To find the license plate, an S.S.D. model trained on license plate photos is used in this study [2]. S.S.D. is one of the most sophisticated object detectors on the market. To recognize things in images, it just employs a single deep neural network. The SSD method makes use of a feed-forward CNN to create a fixed- size collection of bounding boxes and scores the existence of object class instances inside these boxes. The final detection is obtained after a non-maximum suppression (NMS) phase. A Deep neural network - CNN classifier is used for LP character recognition. The network performance was thoroughly analyzed and evaluated, yielding the best results. Mahesh Babu K and M.V Raghunand suggested a paper [3] titled "Vehicle Number Plate Detection and Recognition Using the Bounding Box Method."

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

The project was carried out using Matrix Laboratory (MATLAB), and the following technique was used. Preprocessing of recorded picture, extraction of license plate area, segmentation and recognition of license plate characters In pre-processing, the ideal vehicle picture is captured using a digital camera; for this purpose, they utilized a 13 MP camera. The picture's brightness is then adjusted, noise is removed using filters, and the picture is converted to grayscale. Exactions of the license plate area include locating the image's edges. The Sobel edge filter is used to determine picture boundaries. The bounding box approach is used to segment all of the characters in the picture (LP), and each segmented character is labelled. Finally, the template matching approach is used to recognize each character. K. Yogheedha, A. S. A. Nasir, H. Jaafar, and S. M. Mamduh offered a "Automatic Vehicle License Plate Recognition System Based on Image Processing and Template Matching Approach" article [4]. MATLAB is used to create the automated license plate recognition system. In the pre-processing step, they employed techniques such as rgb to gray conversion, edge detection using the Mexican operator, and median filtering to reduce undesired noise. In the segmentation step, local thresholding, contrast enhancement, and area expanding segmentation are utilized. Template matching occurs during the recognition step, when each character is verified against the templates in the database. The algorithm they created was primarily concerned with recognizing ambiguous characters based on their location.

III. METHODOLOGY

The basic technical approach or analysis of the project can be described in the four major factors. Firstly, vehicle(car) images i.e., data gathering need to be done. It can be said as data preparation. It is done by acquiring and augmenting the data. Secondly, deep learning model called yolov3 is designed and built to identify the region of interest i.e., number plate region. Thirdly, character segmentation is done to identify the character present in the number plate. Finally, Random Forest neural network is designed and built to classify the characters.

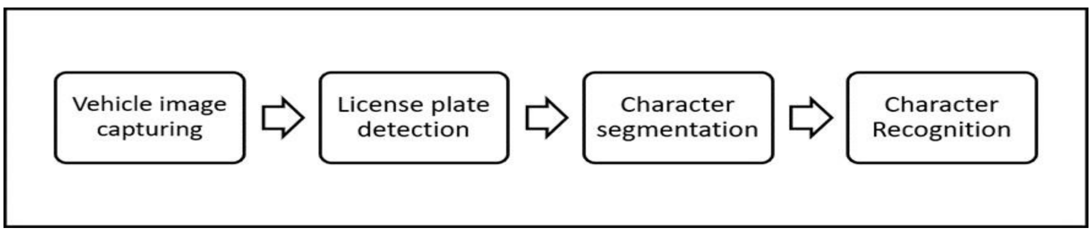


Fig. 1 Block diagram for Methodology

IV. EXPERIMENTAL SETUP

A. Dataset Description

We utilized a VOC dataset of Romanian (European Union) license plates. (The Pascal VOC file is an XML file.) In Pascal VOC, a file is created for each image in the dataset.) The dataset contains 524 photos, with 80 percent of them used for training and the remaining 20 percent used for validation.

The samples are gathered at various times, places, traffic concentrations, and meteorological conditions. In the collection including automobile photos, the camera is frequently located less than 5m from the plate, at ±30 degrees in the plane and 0 -60 degrees in the tilt. A tilt of 0 degrees is parallel to the ground. The width of a plate in the image is between the ratios of 0.20 and 0.25 of the pictures, and the orientation is less than 10 degrees. The total number of photos is 524, which were acquired under the previously specified conditions. The training set has 424 photos, whereas the test set has 100 photos.

B. Training

We replicated the photographs and classified the characteristics we wished to identify for the training. Depending on the elements on the license plate, a single image might be replicated numerous times. We manually spotted and identified the license plates for this pre-processing stage. To give coordinate location to the items we wished to detect and recognize, we utilized a labelling programme called Label-Img. The labelling tool will return four position coordinate points (xmax, ymax, xmin, ymin) that are recorded as Xml files. We then used the Darknet [8] script "XML to YOLOv3.py" to finish the conversion. In order to train our dataset effectively, we needed to create a special YOLO configuration file. Several configuration files were already included with the Darknet application. As a result, just a few items needed to be changed. There was just one class in the license plate detecting phase, which was number plate. The folders containing all training and validation photos related to the object data. The batch number specified the amount of photos processed at each training phase, which was further corrected by subdivisions based on computer performance.

C. Testing

After the model gained experience (Experience is obtained from data), the model analysis is done by testing it on testing data. The testing accuracy will be almost similar to the validation accuracy. Performance evaluation is measured on testing data which is considered as generalized performance.

D. Plate Character Segmentation Using Opencv

The ANPR system's next stage is number plate segmentation, which involves identifying and extracting each of the characters on the previously detected number plate. At this point, our computer system sees the licence plate image as nothing more than a dispersed amplitude of colour. As a result, we require a method for our system to "read" and "understand" it.

E. Image Pre-Processing

It is a critical stage since it tries to appropriately prepare the image so that the subsequent steps may be completed properly. All of the image's preceding preparations are discussed in the next paragraph.

Greyscale pictures are commonly utilised in image processing because they are more simple to manipulate than colour pictures [11]. Aside from that, photos normally include some unwanted noise that has to be eliminated. Furthermore, for this application, where number plates are the focus, it is useful to highlight within the image specific traits that characterise number plates so that they may be discovered afterwards. In this project we have used OpenCV methods to pre-process the image.

V. IMPLEMENTATION

We utilised a dataset of Romanian (European Union) VOC licence plates. We build a file for each picture in the dataset in Pascal VOC). The dataset is composed of 524 images of which 80% are for training and the rest 20% is for validation. We trained a Random Forest model for optical character recognition. This chapter presents a detailed analysis of the system from the perspective of Data flow from given image to getting the output.

A. Implementation of License Plate Detection

From the data we gathered from Kaggle and European dataset, we were able to collect a total of 540 images, which were then processed further for labelling. Labelling process was done by a tool known as Label-Img, for each image we manually located and labelled the licence plate region and saved in a folder named as “Number_plate”.

Now label-Img software converts all the selected license plate region images to a Pascal VOC (Visual object challenge) i.e. an XML file. Now using a python script XML labels were converted to a yolov3 labels and the model was trained.

For testing a dataset from Kaggle was used containing 200 images of cars from various angles and distances as shown in figure out of which our model was able to correctly detect license plates of 184 cars resulting in an accuracy of 92%.

The extracted license plates were stored in a folder as shown in figure-

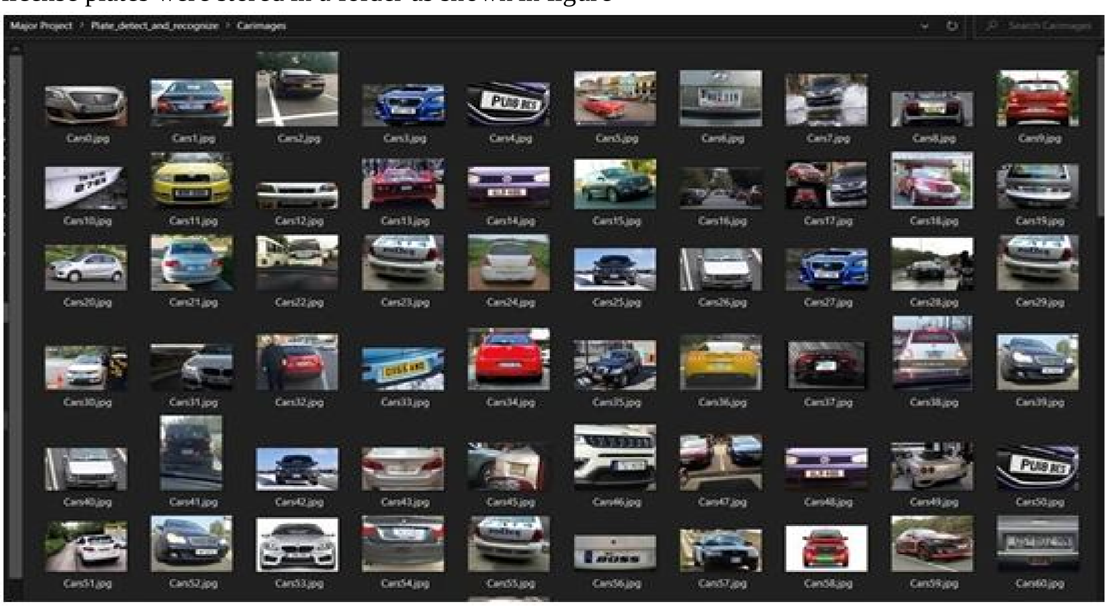


Fig. 2 Car images used for testing.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

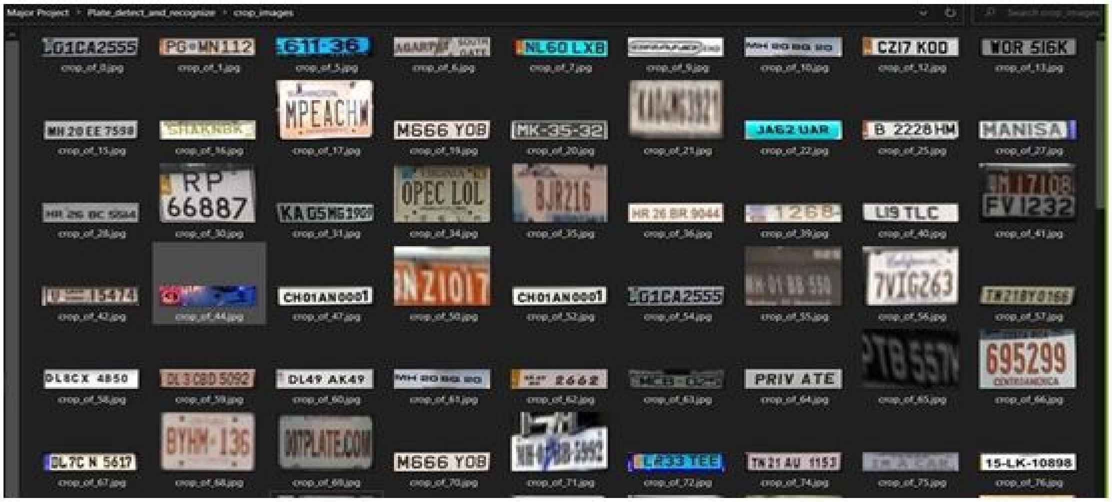


Fig. 3 Licence plates obtained from yolo model output

B. Implementation of Character segmentation

We have performed several pre-processing techniques such as

1) Converting to 8-bit image
2) Conversion to grey scale
3) Performing Blur operation
4) Converting to binary image
5) Image Dilation

The above operation were implemented by using OpenCV as shown in below codesnippet.

```
plate_image = cv2.convertScaleAbs(LpImg[0], alpha=(255.0))

# convert to grayscale and blur the image
gray = cv2.cvtColor(plate_image, cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(gray,(7,7),0)

# Applied inversed thresh_binary
binary = cv2.threshold(blur, 180, 255,
                       cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)[1]
## Applied dilation
kernel3 = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
thre_mor = cv2.morphologyEx(binary, cv2.MORPH_DILATE, kernel3)
```

Fig. 4 Operations performed in pre-processing.

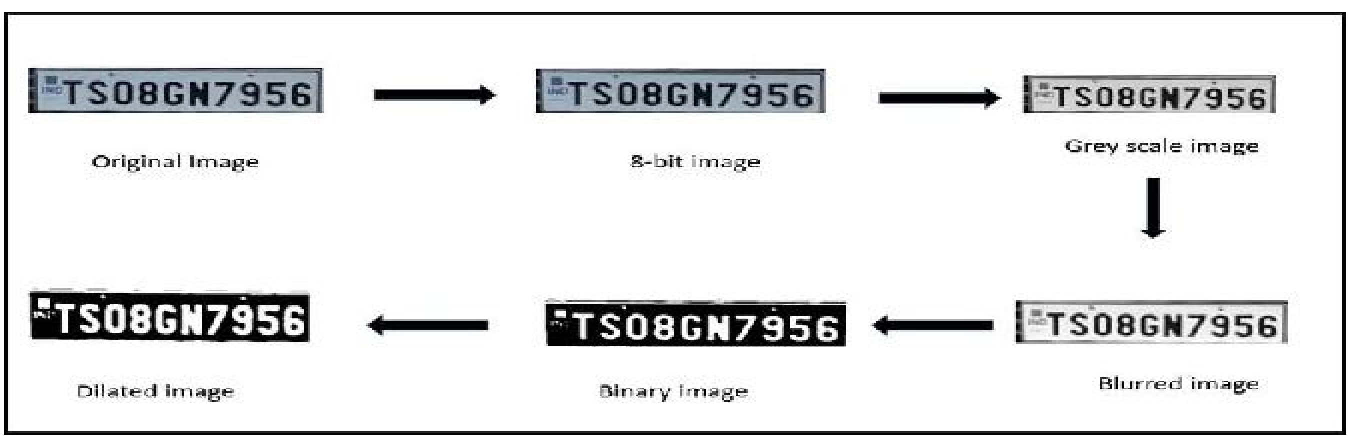


Fig. 5 Pre-processing operations carried out on an image

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

From the Dilated image contours were found by using OpenCV functions to recognise the characters and sort the contours for proper recognition of characters from left to right order.

Fig. 6 Characters identified from a given image

C. Implementation of Random Forest model

We have used Random forest classifier from Scikit learn library to train a random forest model. For training purpose, we have used a dataset of 5400 images in which each character (0 to 9 and A to Z) has 150 images each.

Fig. 7 Samples used for Random Forest model training

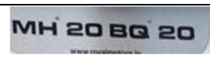
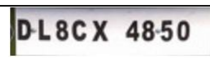
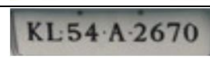
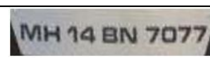
For testing the model, we have given all the correctly identified license plate images from yolo model output as the input to Random forest classifier.

We have found out that out of 184 images our classifier was able to recognise 151 images correctly which is 82.1% accurate and could not recognise low resolution images and images which were blurred out. Some of the recognised license plate text along with it are shown in below table.

VI. PERFORMANCE EVALUATION

The performance of the system has been evaluated by comparing the input images given and the output which is predicted by the system. Below table includes the recognized license plate and the predicted output.

TABLE I
PERFORMANCE EVALUATION OF ANPR ON CAR IMAGES

S.No	Cropped licesnce plate image	Predicted Output
1		PGMN112
2		MH20BQ20
3		DL7CN5617
4		N71AU1153
5		CH01AN0001
6		DL8CX4850
7		KL54A2670
8		KA05MG1909
9		MH14BN7077
10		MH15BD8877
11		MH20EE7598
12		MH14DX9937
13		TS08FN805
14		HP26CJ5779

VII. CONCLUSIONS

The results and discussions of all the steps involved in designing and implementing the YOLO model and Random Forest model are presented in this chapter. The model implemented has got the desired results as shown in this chapter. It describes the training and testing results and its accuracy. The predictions made by the build YOLO Model and Random Forest model is described briefly with images.

In our project we have successfully developed a model that can detect and recognize a number plate from a given input image. Our project is based on a deep learning model called YOLO which was used for plate detection. Image pre-processing is done using computer vision methods to improve the accuracy. We have used a machine learning algorithm called Random Forest classifier for the purpose of optical character recognition. We have observed that our trained yolo model is 92% accurate in license plate detection and Random Forest model is 82% accurate in optical character recognition. Further this project can be improved by training the model better with higher and more varied data sets and improved techniques. ANPR could also be used for identification of vehicle owners, identification of vehicle model, traffic management, speed control and monitoring of vehicle position.

REFERENCES

[1] R. R. Palekar, S. U. Parab, D. P. Parikh and V. N. Kamble, "Real time license plate detection using openCV and tesseract," 2017 International Conference on Communication and Signal Processing (ICCSP), 2017, pp. 2111-2115, doi: 10.1109/ICCSP.2017.8286778.

[2] Q. Wang, "License plate recognition via convolutional neural networks," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 926-929, doi: 10.1109/ICSESS.2017.8343061.

[3] K. M. Babu and M. V. Raghunadh, "Vehicle number plate detection and recognition using bounding box method," 2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2016, pp. 106- 110, doi: 10.1109/ICACCCT.2016.7831610.

[4] K. Yogheedha, A. S. A. Nasir, H. Jaafar and S. M. Mamduh, "Automatic Vehicle License Plate Recognition System Based on Image Processing and Template Matching Approach," 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), 2018, pp. 1-8, doi: 10.1109/ICASSDA.2018.8477639.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[6] M. T. Qadri and M. Asif, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition," 2009 International Conference on Education Technology and Computer, 2009, pp. 335-338, doi: 10.1109/ICETC.2009.54.

[7] Hendry and Rung-Ching Chen. (2019). Automatic License Plate Recognition Via Sliding-Window Darknet-Yolo Deep Learning. Image and Vision Computing 87,(2019), 47–56.doi: 10.1016/j.imavis.2019.04.007.

[8] M. B. Ullah, "CPU Based YOLO: A Real Time Object Detection Algorithm," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 552-555, doi: 10.1109/TENSYMP50017.2020.9230778.

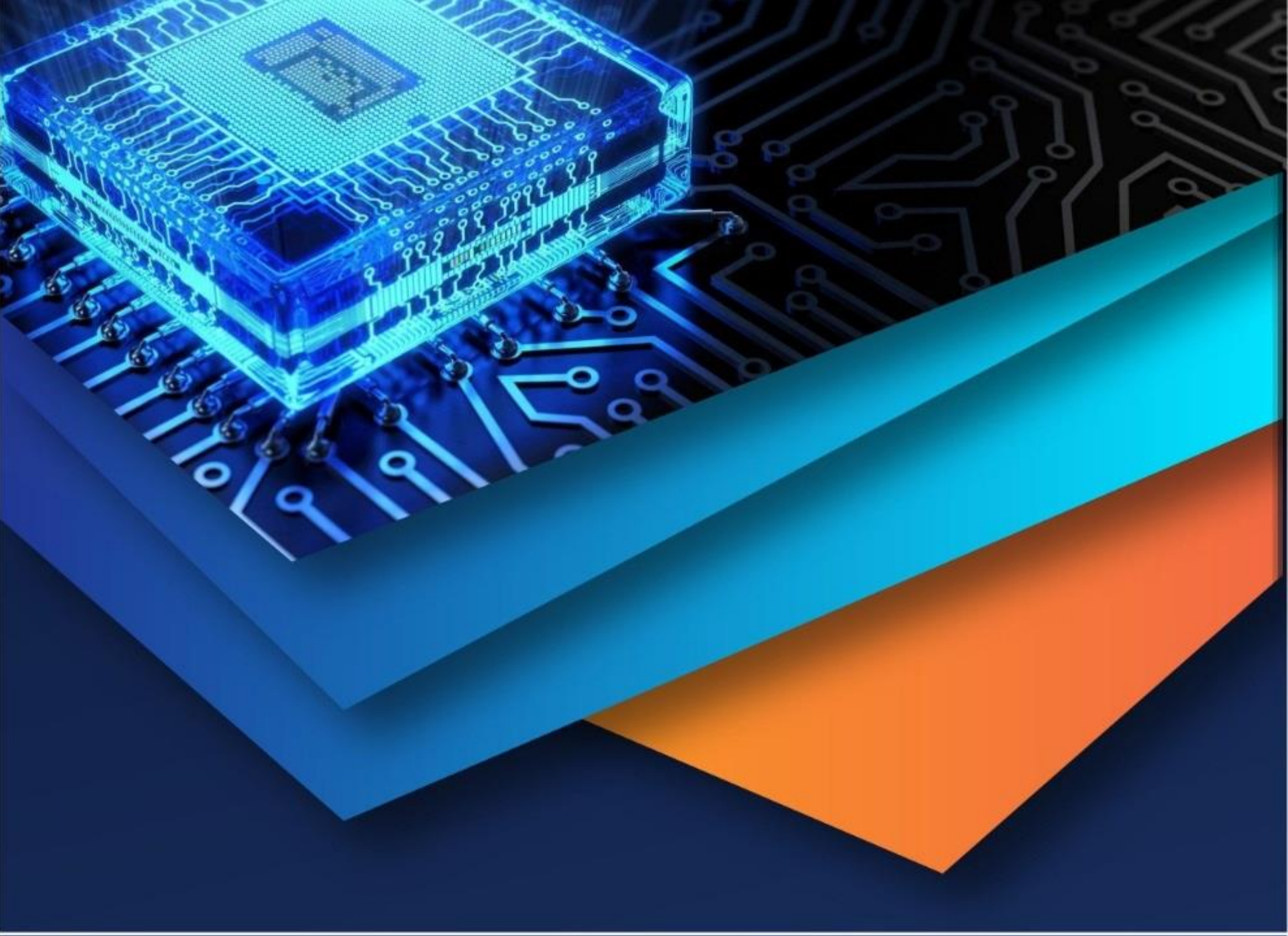
[9] Nawaf Hazim, Mohanad Nsaif Al-Mayyahi, Mohammed Aboalmaaly. (2018). An Efficient Character Recognition Technique Using K-Nearest Neighbor Classifier. International Journal of Engineering & Technology, 7 (4) (2018) 3148-3153. doi: 10.14419/ijet.v7i4.18952.

[10] S. N. H. S. Abdullah, K. Omar, S. Sahran and M. Khalid, "License plate recognition based on support vector machine," 2009 International Conference on Electrical Engineering and Informatics, 2009, pp. 78-82, doi: 10.1109/ICEEI.2009.5254811.

[11] Saquib Nadeem Hashmi, Kaushtubh Kumar, Siddhant Khandelwal, Dravit Lochan, Sangeeta Mittal. (2019). Real Time License Plate Recognition from Video Streams using Deep Learning. International Journal of Information Retrieval Research (IJIRR) 9(1),65-87. doi: 10.4018/IJIRR.2019010105.

[12] A. E. Rashid, "A fast algorithm for license plate detection," 2013 International Conference on Signal Processing, Image Processing & Pattern Recognition, 2013, pp. 44-48, doi: 10.1109/ICSIPR.2013.6497956.

[13] Zuhaib Akthar and Rashid Ali. (2020). Automatic Number Plate Recognition Using Random Forest Classifier. doi: 10.1007/s42979-020-00145-8.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY