A Real-Time Automatic Kurdistan Numberplate Recognition System

Abubakar M. Ashir[1]

[1]Department of Computer Engineering, Faculty of Engineering, Tishk International University, Erbil, Iraq
Correspondence: Abubakar M. Ashir, Tishk International University, Erbil, Iraq.
Email: abubakar.ashir@tiu.edu.iq

Doi: 10.23918/eajse.v8i1p126

Abstract: The current paper presents a real-time implementation of a numberplate recognition system for Kurdistan region of Iraq. Automatic numberplate recognition systems (ANPR) have played a key role in many places in enforcing regulations on safe driving and preventing vehicular theft, identity establishment and many other applications. Though there are number of such systems available, their accuracy and speed of execution at inference time is still a challenging issue. To detect a numberplate from a fast-moving vehicle or from a high-resolution camera input, an extremely fast algorithm is required capable of processing tenth of frames per second. This is a huge challenge for many systems and often a compromise is made between the accuracy of detection and execution speed. This work proposes and implement Automatic numberplate recognition system with high accuracy and capable of processing over 50 frames per second at image resolution of 1920x1080 on a raspberry pi B 4 processor. The proposed approach has two major parts: numberplate region localization and character recognition or extraction from the localized region. We used standard machine-learning approach to detect the region of interest using Haar-like features algorithm as feature extractor and Adaptive Boosting (AdaBoost) algorithm to train a cascade of weak learner's classifiers for classification. After detection of the numberplate region from the input image, an optical character recognition algorithm (Tesseract) is used to extract the characters from the image for display and other use. Tesseract is a machine-learning based OCR algorithm which was pretrained with many languages and made available by google. To increase the detection accuracy, we proposed a masked training approach. The masked training approach uses masked positive samples as negative samples during the training. We also investigated the effect of using different boosting optimization techniques on the overall accuracy of the system. The overall accuracy and inference speed has greatly been improved when tested on a raspberry pi 4 B hardware.

Keywords: Numberplate, Haar-like Features, Cascaded Classification, AdaBoost, OCR

1. Introduction

Automatic Number Plate Recognition (ANPR) is a system capable of reading vehicle number plates from a moving vehicle without human intervention. The entire systems may consist of so many parts such as the high-speed image capture sensors with supporting illumination, character recognition, verification of the character sequences from a local or remote vehicular license database. The character recognition part converts an extracted image of the numberplate from the image frame into text. Generally, the system outputs a set of metadata that identifies an image containing a vehicle license plate and the associated decoded text of that plate. ANPR is therefore the underlying technology used to find a vehicle license/number plate and it, in turn, supplies this information to a next stage of computer processing through which the information can be interpreted, stored or matched to create an

Received: March 20, 2022
Accepted: May 28, 2022
Ashir, A. M. (2022). A Real-Time Automatic Kurdistan Numberplate Recognition System. *Eurasian Journal of Science and Engineering*, *8*(1), 126-138.

ANPR based application.

ANPR systems is used by many security and government agencies to track down criminal behavior and is also seen on many cities motorways as a method of detecting over speeding through average speed calculation. However, ANPR is used in a variety of other ways to support the security and safety of the public as well as supporting efficiencies in the way we interact with transportation and vehicle-based infrastructure (Singh, Kaur, et. al. 2016). In addition, ANPR is sometimes known by various other terms such as Automatic (or automated) license-plate recognition (ALPR), Automatic (or automated) license-plate reader (ALPR), Automatic vehicle identification (AVI), Car-plate recognition (CPR), License-plate recognition (LPR), Mobile license-plate reader (MLPR), and so on (Li-Sang et al., 2020).

The History of ANPR is considerably longer than most people realize. Because of its prolific use in more recent years for a broad range application such as traffic studies, access control and parking, many people, if asked, would guess at it being an invention belonging to this millennium. Surprising to most people, the history of ANPR stretches into the last century as it was invented in 1976 in the UK at what was then known as the Police Scientific Development Branch (PSDB) (now titled Home Office Scientific Development Branch) and early systems were developed for use from 1979. Early trial systems were deployed in the UK on the A1 Road and at the Dartford Tunnel crossing on the M25 motorway and the first arrest that was credited to ANPR detection of a stolen car did not come until 1981 (Byung-Gil et al., 2020). Since its inception, ANPR Technology has evolved and adapted with the times, finding new outlets and applications taking it beyond the boundaries of just policing and security. These are a few notable milestones along our journey to date. For instance, in 1993 – ANPR was deployed for the first time as part of a "Ring of Steel" camera network around the City of London. In 1997- The Police National ANPR Data Centre (NADC) formed as an extension to the Police National Computer service. In 2003 - The London Congestion Charge scheme is introduced which was aimed at reducing traffic in central London for several reasons. In 2006 - ANPR International deployed its first static camera system for Parking Management - bodyguarding which is the first module built for the eye TRAFFIC back-office system. In 2009 - ANPR International develops its first mobile ANPR product - street SWEEPER, which is designed for multiple applications including Traffic surveys, Mobile surveillance and untaxed vehicle enforcement for the Driver and Vehicle Licensing Agency (DVLA). In 2016 - Vehicle Damage Recording System (DRS) extended to include dual lane capture, allowing for multiple lanes and up to 14 cameras to be integrated with ANPR data and vehicle pre-booking software to speed up the customer experience and increase convenience. The first dual lane system goes live at Doncaster Sheffield Airport with average times for visitors to drop-off their vehicle recorded at 32 seconds.

2. Literature and Theoretical Background

2.1 Literature

In an article “A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector”, authors in (Laroca et al., 2018) proposed a method for Automatic License Plate Recognition (ALPR) that can be used in real time based on YOLO Detector. The trained Convolutional Neural Networks (CNNs) and finetuned each stage of the ALPR stage to adopt to different conditions (e.g., variations in camera, lighting, and background). The authors designed two-stage approach for character segmentation and recognition, using inverted License Plates (LPs) and flipped characters. Once the plate is detected, they deployed CNN proposed by Montazzolli and Jung (Silva & Jung, 2018)

(CR-NET) for character segmentation and recognition. However, instead of performing both stages at the same time through an architecture with 35 classes (0-9, A-Z, where the letter O is detected jointly with the digit 0) (Shehu et al., 2015). They reported significantly improved performance of 93.53% and 47 Frames Per Second (FPS) when tested on SSIG dataset and achieved a recognition rate of 93.53% and 47 Frames Per Second (FPS), on streaming videos (Laroca et al., 2018).

In their submission, (Al Nasim et al., 2021) proposed a method for Automatic Recognition of Bengali License Plates using a hybrid method for detecting license plates using characters from them. A capture input image is preprocesses and then YOLO object detection model was used to locate the position of the plate in the input image. The detected for license plate detection, and then, for license plate segmentation, Otsu's Thresholding was used and eventually, for character recognition, the CNN model was applied. This model will allow the vehicle's automated license plate detection system to avoid any misuse of it. The author reported the prediction accuracy of around 81% during testing (Al Nasim et al., 2021).

In (Singh et al., 2016) International Journal of Computational Vision and Robotics is about "Automatic number plate recognition system by character position method" is about Plate localization and character recognition with two stages of ANPR. A new algorithm has been proposed for number plate localization which is based on character positioning method. The system has been tested on 419 sample images from various countries with various variations in viewing angles, illuminations, and distances. The work proposed in the paper presents a new NPL algorithm with a higher detection rate. A new concept called character position method (CPM) has been used in NPL which checks whether given candidate region is number plate or not. SVM classifier with the features extracted with recursive subdivisions of character image has been used. For increasing the robustness of the system, the possible fonts for number plates have been considered. For experiments, a dataset consisting of 419 sample images containing 431 license plates from various countries with various variations in viewing angles, illuminations and distances has been prepared. Further to increase the accuracy of proposed system, syntactic analysis of number plate format based on geographical regions performed (Ashir, 2022). Experimental results show that the proposed system detects number plates and recognize characters successfully. The overall success rate of plate localization is 97.21% and recognition of number is 95.06%. A new algorithm for the NPL is proposed in which character positioning method has been used to locate number plate (Singh et al., 2016).

IN an article "Development of New ANPR Dataset for Automatic Number Plate Detection and Recognition in North of Iraq" is about an automatic number plate detection (ANPD) and automatic number plate recognition (ANPR) (Yaseen et al., 2019). The article is for improving dataset, which is called North Iraq-Vehicle Images (NI-VI) of three provinces (Duhok, Erbil, and Suleimani) for vehicle images, is presented. There are 1500 images in this dataset. The main contribution of this work is the creation of a new dataset for license plate of vehicles in north Iraq with Arabic fonts in different and difficult conditions. Moreover, some images created for bad weather conditions, such as snowy, dusty, and low lighting (Anwar & Ashir, 2020). The flow of the traffic is controlled by many cameras that are spread around the streets inside and outside of cities, as these cameras acquire instantaneous image of the vehicles that are on the roads, intelligent software is needed to detect vehicles as well as their license plates. The purpose of introducing dataset is to provide data for testing ANPD and ANPR algorithms by researchers to increase methods performance. The NI-VI dataset was systematically presented in this paper. All works in the ANPD and ANPR approaches, in north of Iraq, should be tested and experimented by some dataset that contains standardized images of vehicles and related to

north Iraq location. The limitation in this work is that this dataset related to vehicle license plates of only north Iraq (Yaseen et al., 2019).

In an article "An Automatic Recognizer for Iraqi License Plates Using ELMAN Neural Network" is proposed in this manuscript. The processing procedures are developed in several stages (Mohsin et al., 2010). Experimental Explanation of results to illustrate the performance of the proposed method. LPR problem classified into two stages, the first one is plate image processing (including detection of plate from vehicle image, plate numbers and characters segmentation), and second is recognition of the isolated plate number. In the field of isolated plate number recognition, a lot of conventional methods are used, for instance template matching, MPL neural networks, mathematical morphology, and image frequency analysis approach (such as DFT). Also, it's about introducing a new algorithm for Iraqi license plate number recognition. The test images were taken under various illuminations, size, and types of license plate conditions. The experimental results for the LPR system were performed under different illuminations, different distances, and various types of license plate. The experimental results that the algorithm achieved about 85% correct segmentation and 76% correct recognition for 21 samples (Mohsin et al., 2010).

3. Proposed Method

The general layout of the proposed numberplate recognition system for Kurdistan region of Iraq is shown in Figure 1. In the training phase, a database of positive and negative samples is required. The positive samples consist of images containing cars with numberplate visible on them whereas the negative samples contain images of cars, peoples, objects, or anything without numberplate visible in them. The positive samples were then annotated to indicate the region which enclose the numberplate in them using a special annotation software. The annotation information is usually extracted to text file known as positive sample description file which are used during training to extract Haar-like feature from them. Two files are prepared (positive and negative description files) which provides information about the positive and negative sample images. To investigate the efficacy of the feature extraction algorithms, two different types of feature extractors were used (Haar-like and Local Binary Pattern LBP) and different boosting techniques in case of classification. The extracted features are used to train cascades of weak learners (decision tree Haar-like features) classifiers using AdaBoost algorithms. The trained classifier is tested on still images and live streaming video and then the whole process is fine-tuned until the desired accuracy and inference speed is achieved. We then applied OCR on the extracted region of interest (numberplate) convert the numberplate region from image to characters using Tesseract. A graphical user interface was also created using Pyqt6 libraries in python to provide a common interface for the user camera and the hardware (Raspberry 4 B) upon which the algorithm is implemented. The flowchart of the proposed system architecture is shown in Figure 1.

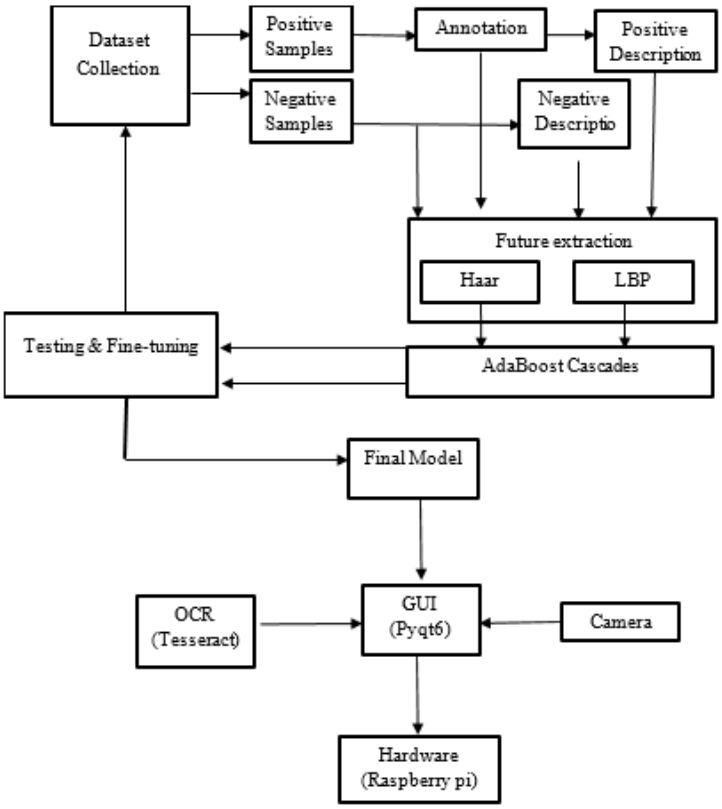

Figure 1: Flowchart of the proposed approach

3.1 Numberplate Dataset

The dataset was collected from within and around Erbil city with camera and also from an online used-car selling website (https://kurdshopping.com/ , http://kurdsale.com/ and https://www.dasy2.com/). The positive samples consist of 1443 images of cars with visible numberplate on them. The negative samples consist of 2135 images of scenes, objects and people without visible numberplate in them. We also masked out all the region of numberplate in the positive samples and then used as negative samples as shown in Figure 2.

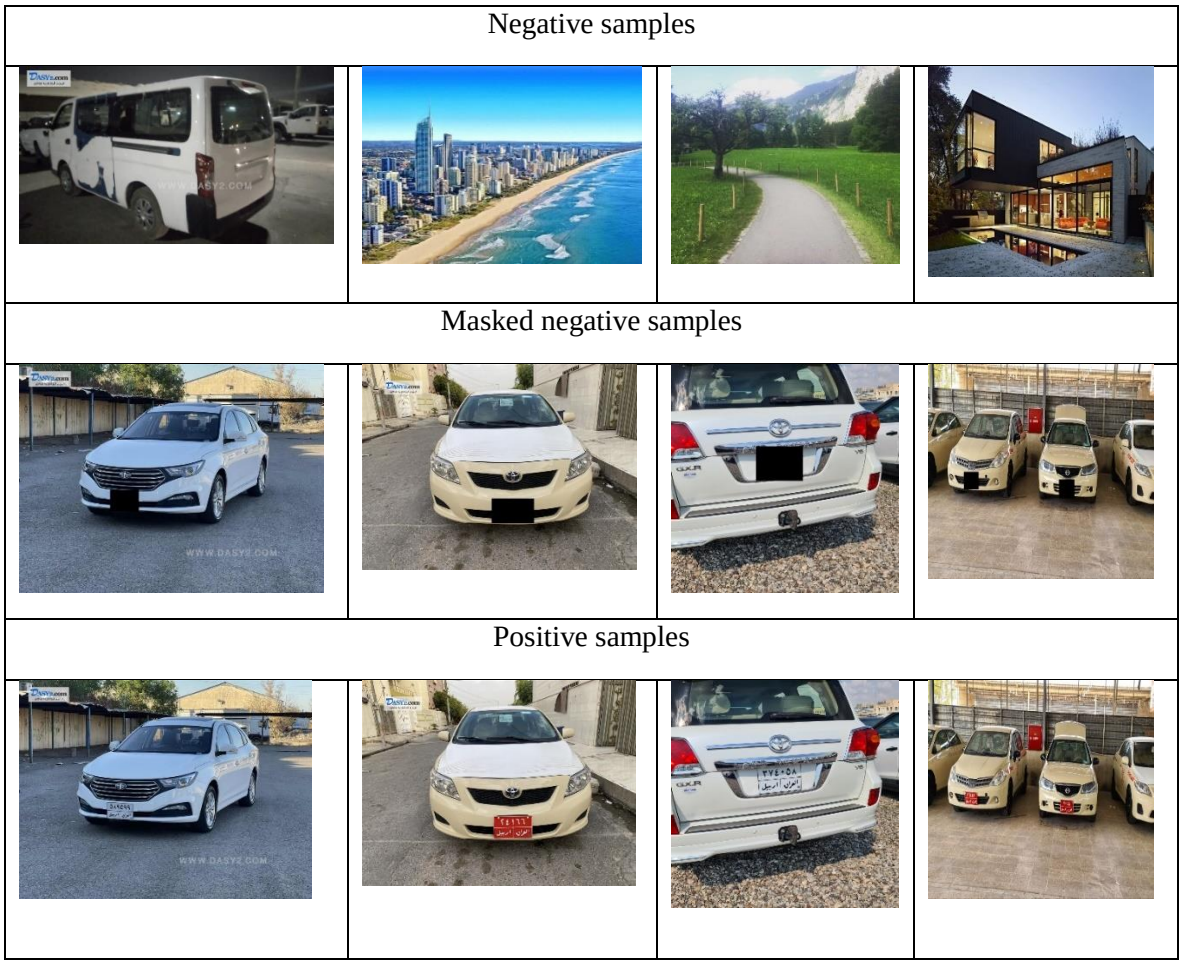

Figure 2: Samples of positive, negative, and masked positive images

3.2 Data Annotation

To train supervised learning algorithm such as AdaBoost cascaded classifiers for numberplate detection, it requires positive sample description files which will contain the spatial coordinates of the region of interest (numberplate centers, width, and height) within the image. A special graphical object annotation application was used to create the bounding boxes around the object and extract both the spatial coordinates and class of the object in a process referred to as object annotation as shown in Figure 3.

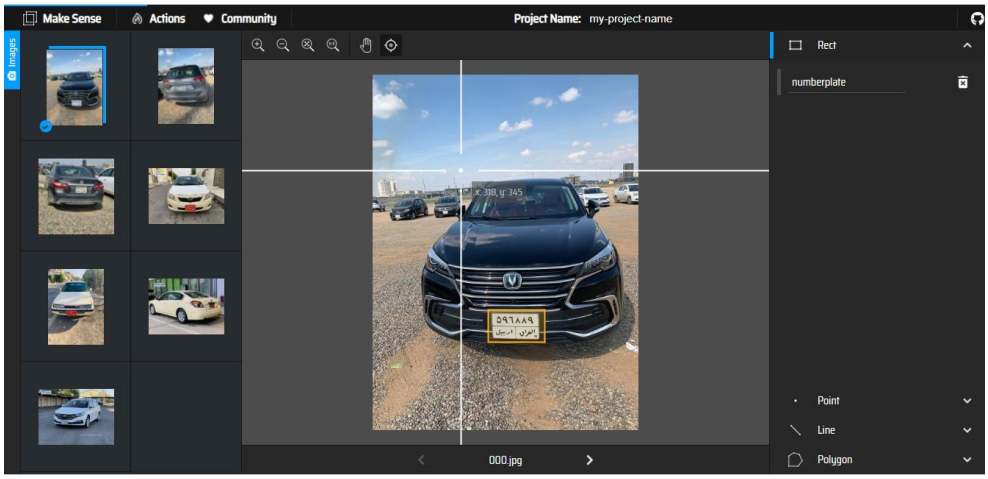

Figure 3: Numberplate annotation

3.3 Feature Extraction

The aim of feature extraction stage is to extract the most useful information or data from the image that will best describe that image or that differentiate it most from other similar images. There is plethora of feature extraction algorithms in the literature but in this context, we used the Haar-like features proposed by (Viola & Jones, 2001) and in addition we used different types of booting optimization techniques to compare the effectiveness of the process.

3.3.1 Haar-Like Features

Haar-like features is a feature extraction technique popularly used with images for object detection and was proposed in 20001 by (Viola & Jones, 2001)and was initially used for face detection. The algorithm owes its name to Haar wavelets features with striking similarities. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. The algorithm needs a lot of positive images (images of numberplates) and negative images (images without numberplate) to train the classifier. The Haar-like features shown in figure 4, are used just like a convolutional kernel with each kernel responsible for computing a single value feature over a given window. This single valued feature is obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle (D.N. et al., 2012). The Haar-like feature extraction is obviously computationally expensive and to mitigate these effects, the authors used the concept of integral image to reduce the number of computations required to extract one feature from the image. The number of features return by this method is still enormously large considering that in each window, different sizes, types, and locations of a Haar kernel are used to compute unique features. Even for a window of size 24x24, over 160000 Haar-like features can be extracted which is still a huge number.

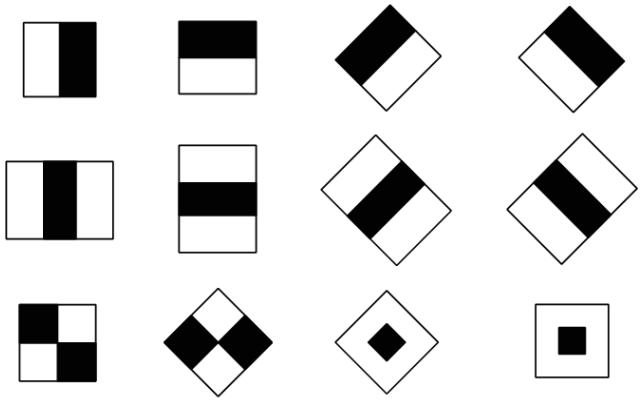

Figure 4: Haar-like feature kernel types

3.4 Cascade Classification with AdaBoost

One of the major issues with the extracted Haar-like features are that they are enormously large and most of them do not contribute to the detection process. One way to address this problem is through feature selection process whereby only the most relevant features are retained, and the redundant or irrelevant ones are discarded. This is achieved using the AdaBoost classification method. The adaptive boosting technique apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the image window to positive (numberplate) and negative (non-numberplate). Obviously, there will be errors or misclassifications. Therefore, features with minimum error rate are selected, which means they are the features that most accurately classify the numberplate and non-numberplate images. After each classification, weights of misclassified images are increased. Then the same process is repeated to find a new error rates and weights. The process is continued until the required accuracy or error rate is achieved or the required number of features are found. In the end, out of 160,000+ features for 24x24 the most highly ranked features are selected which might just be between 0.2%-0.4% (3,200-6,400) of the original features.

Another improvement in the classification stage is the use of cascade classification. Instead of applying all 16000 + selected features on a window, the features are grouped into different stages of classifiers (weak learners) and applied one-by-one. If a window fails the first stage, it is discarded and the rest of the features in that window are never considered. If it passes, it advances to the next stage of features and repeat the test as was in the previous stage. The window which passes all stages is considered as a numberplate region. The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. According to the authors by (Viola, Jones, 2001) on average 10 features out of 6000+ are evaluated per sub-window.

4. Experimental Result and Discussion

In this chapter we present the results from the experiments based on the proposed approach. The dataset described in section 3.1 was used to create the positives, negatives and masked negative samples. For each category, 90% of the samples were used for training and the remaining 10% were used for testing. The experiment was conducted under different set-ups and conditions. Firstly, we considered training with regular positives and negative samples with varying stages of the cascades. Secondly, we repeated the experiment but with negative samples as the mixture of masked negatives and unmasked negative samples at various level of cascades. We also consider the effect of using various types of boosting optimization techniques in Adaboost.

4.1 Experimental Setup

We implemented the proposed method in python environment with opencv-3.4.11 library application for training of the Haar cascade classifiers with a windows 11 Operating System environment. Some of the hardware resources used in the experiment are listed in table 1 below.

Table 1: Hardware resources

Desktop Computer	Dell XPS 8930 Tower
CPU	8th Generation Intel Core i7-8700k, 6 cores processor @3.7 GHz
RAM	16 GB, 2666 MHz
Storage	SSD KXG50ZNV 512G NVMe TOSHIBA DDR4 515GB
Graphics card	NVIDIA GeForce GTX 1080 8 GB
Operating System	Windows 10 Home 64-bit

4.2 Training and Validation Results

Training the Haar cascade classifiers using OpenCV software is quite easy and straightforward. In general, it consists of five stages. (1) annotating the positive samples (which can be accomplished using opencv_annotation.exe software or any other annotation application). (2) generating positive and negatives samples text files which contains the paths of the samples and also the location of the region of interest (e.g., numberplate) in case of positive samples. (3) creating a positive sample vector files (using opencv_createsamples.exe software) which basically is a file with extracted region of interest (e.g., numberplate) from the positive samples. (4) train the cascade classifier with desired feature extractor (Haar-like feature or LBP) and Adaboost with a chosen minimization technique (using opencv_traincascade.exe). (5) Lastly the trained cascade classifier can be visualized with opencv_visualisation.exe software.

In Table 2, we present some of the required setup parameters for the training. In our experiment we considered a window of size 24x30 that will most likely match the aspect ratio of the numberplate used in this experiment.

Table: Haar cascade training parameters

OpenCV software	Input parameters	Description
opencv_createsamples.exe	-info	Full path name to the positive sample description file.
	-w, -h	Width and height of the window to be used for feature extraction
	-num 1000	Number of positives samples to be used for training from *-info*
	-vec	Full path name of the "*.vec" file* (to be created) where the extracted region of interest from positive samples will be stored.
opencv_traincascade.exe	-data	Path of the folder where the trained classifer will be written to
	-vec	Full path name to "*.vec" file*
	-bg	Full path name to the negative sample description file.
	-numPos, -numNeg	Number of positives and negative samples to be used for training
	-numStages	Number of stages of weak learners to be used in the cascade
	-featureType	Types of feature extraction algorithm e.g Haar or LBP
	-stageType, -bt	Boosted classifier and its type e.g., Discrete AdaBoost, RAB - Real AdaBoost, LB - LogitBoost, GAB - Gentle AdaBoost.
	-w , -h	Width and height of the window to be used for feature extraction

4.3 Evaluation Metrics

To further examine the performances of the trained models we generated different models under various circumstances (e.g., different boosting types and number of cascade stages) and recorded the False Rate and Hit Rate of each one of those circumstances. Figure 5 present the plot of the False Rate and Hit Rate against the number of stages of the cascade classifier when regular (unmasked) positive samples were used. In a similar way Figure 6 depicted the False Rate and Hit Rate against the number of stages of the cascade classifier with masked negative samples. Figure 7 represents the performance comparison between the different boosting types schemes (i.e., Discrete AdaBoost-DAB, Real AdaBoost- RAB, LogitBoost- LB, and Gentle AdaBoost- GAB) with fixed number of stages (i.e., 28 stages) of the cascade classifier.

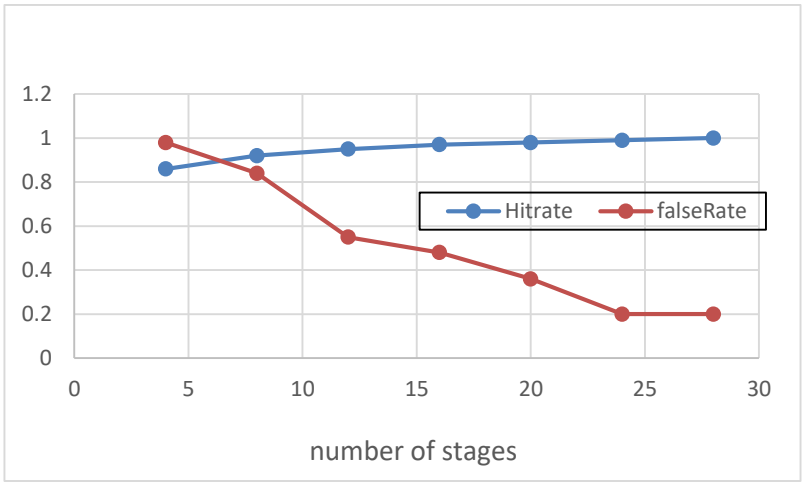

Figure 5: Unmasked negative sample model -plot of cascade stages vs Hit-Rate and False-Rate

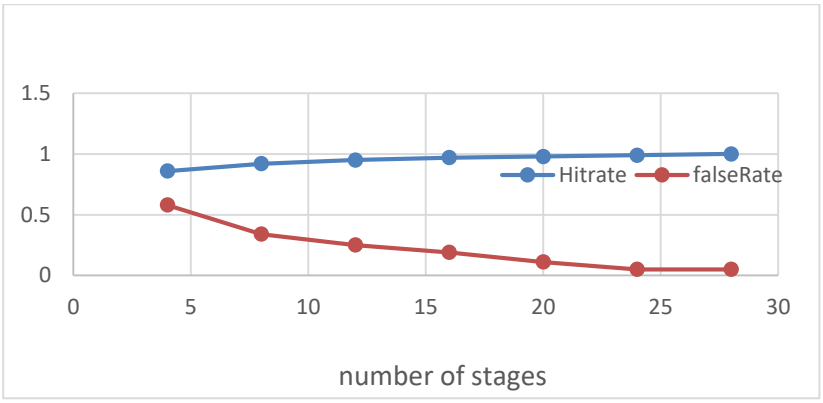

Figure 6: Masked negative sample model -plot of cascade stages vs Hit-Rate and False-Rate

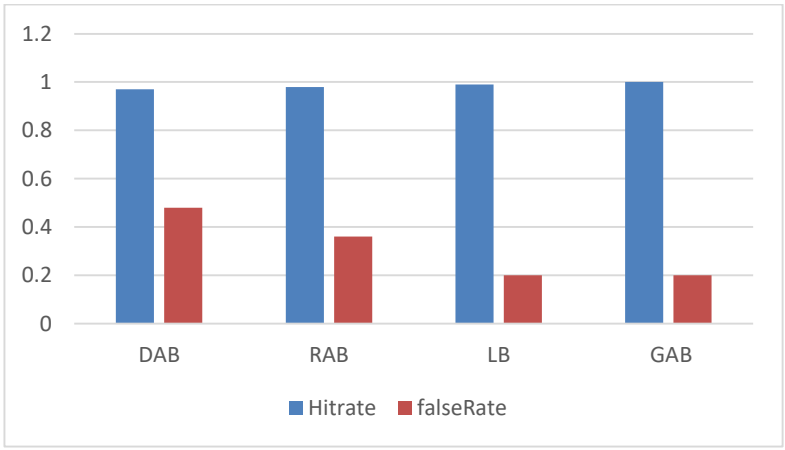

Figure 7: Comparison of the performance of boosting types vs Hit-Rate and False-Rate at fixed number of stages

4.4 Testing and Graphical Interface

Apart from the regular testing which follows model training, a graphical user interface was implemented using a PyQt6 library in python. The GUI provides three key options for testing new

data. These options include collection of images from a folder location, a video file or a stream of video from a live camera feed for real-time detection. Figure 8 shows the GUI.

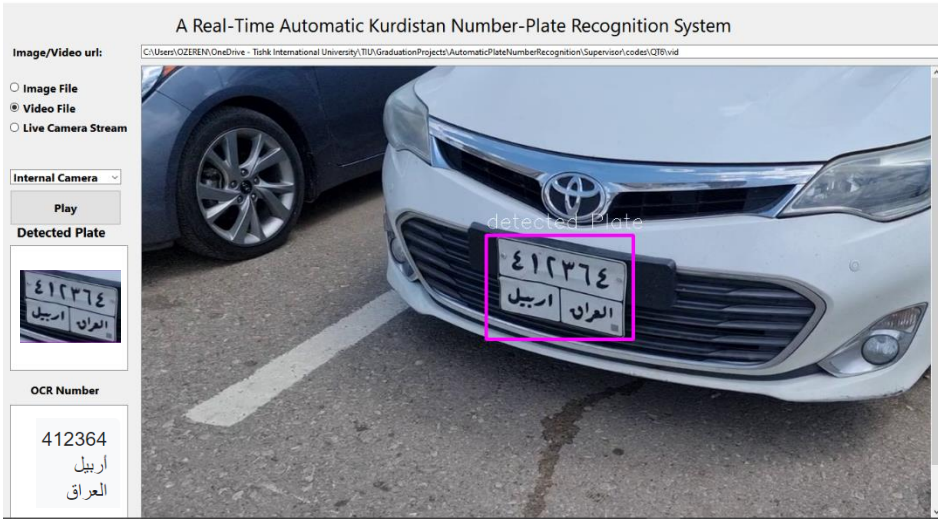

Figure 8: Graphical user interface for implementing the proposed method running on from raspberry pi 4 B

5. Conclusion

A new approach for automatic numberplate recognition in Kurdistan region has been proposed. One of the objectives of the approach is to improve on the accuracy and speed of the process during execution time in order to be applicable in real-time applications. The use of masked positive samples during the training has significantly improve the accuracy of the system by enormously mitigating the effect of false positives detection. In turn, less false positive samples do not only improve the detection accuracy but also helps to reduce the number of cascade stages to navigate through during the execution or inference time which increases the speed of the algorithm. The presented results from the experiments and the system implementation provides a clear support to the efficacy of the proposed approach.

References

Al Nasim, M. A., Chowdhury, A. I., Muna, J. N., & Shah, F. M. (2021). An automated approach for the recognition of bengali license plates. *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, 1–4. https://doi.org/10.1109/ICECIT54077.2021.9641214

Anwar, M., & Ashir, A. M. (2020). An efficient image de-blurring technique using point spread function in high definition medical image. *Eurasian Journal of Science & Engineering*, *6*(1), 27–38.

Byung-Gil, H., Lee, J. T., Lim, K.-T., & Choi, D.-H. (2020). License plate image generation using generative adversarial networks for end-to-end license plate character recognition from a small set of real images. *Applied Sciences*, *10*(8), 1–16.

D.N., C., G., A., & M., R. (2012). Face detection using a boosted cascade of features using openCV. In K. R. Venugopal & L. M. Patnaik (Eds.), *Wireless Networks and Computational Intelligence* (pp. 399–404). Springer Berlin Heidelberg.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Goncalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the {YOLO} detector. *2018 International Joint Conference on Neural Networks ({IJCNN})*.

https://doi.org/10.1109/ijcnn.2018.8489629

Li-Sang, L., Dong-Wei, H., Ying, M., Zhang, X.-Z., Jing, H., Li1, J.-N., & Yao, J. (2020). A novel license plate location method based on deep learning. *Journal of Network Intelligence*, *5*(3), 93–101.

Mohsin, A. H., Hassin, A. H., & Jaleel, I. Q. A. (2010). An automatic recognizer for iraqi license plates using ELMAN neural network. *Journal of Software Engineering and Applications*, *3*(12), 1163–1166.

Shehu, G. S., Ashir, A. M., & Eleyan, A. (2015). Character recognition using correlation & hamming distance. *2015 23rd Signal Processing and Communications Applications Conference, SIU 2015 - Proceedings*. https://doi.org/10.1109/SIU.2015.7129937

Silva, S. M., & Jung, C. R. (2018). License plate detection and recognition in unconstrained scenarios. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision -- ECCV 2018* (pp. 593–609). Springer International Publishing.

Singh, B., Kaur, M., Singh, D., & Singh, G. (2016). Automatic number plate recognition system by character position method. *International Journal of Computational Vision and Robotics*, *6*(1), 94–112.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, *1*, I–I. https://doi.org/10.1109/CVPR.2001.990517

Yaseen, N. O., Ganim Saeed Al-Ali, S., & Sengur, A. (2019). Development of new anpr dataset for automatic number plate detection and recognition in North of Iraq. *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 1–6. https://doi.org/10.1109/UBMYK48245.2019.8965512