ONLINE ISSN: 3007-3197 | PRINT ISSN: 3007-3189

http://amresearchreview.com/index.php/Journal/about

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 3, Issue 7 (2025)

Implementation of YOLO Through DevOps for Automated Vehicle Number Plate Recognition

Malik Muhammad Afaq Haidari, Muhammad Fawad Nasim, Saleem Zubair, Abdul Rehman, Mudassir Ali

Article Details

Keywords: YOLO, vehicle number plate recognition, DevOps, real-time applications, continuous integration, continuous delivery, rapid deployable, containerization, IaC, smart cities, intelligent transport systems.

Malik Muhammad Afaq Haidari
Department of Computer Science, Faculty of Computer Science & Information Technology the Superior University Lahore, Pakistan
afaq.malik@superior.edu.pk

Muhammad Fawad Nasim
Department of Computer Science, Faculty of Computer Science & Information Technology the Superior University Lahore, Pakistan
fawad.nasim@superior.edu.pk

Saleem Zubair
Department of Computer Science, Faculty of Computer Science & Information Technology the Superior University Lahore, Pakistan
saleem.zubair@superior.edu.pk

Abdul Rehman
Department of Computer Science, Faculty of Computer Science & Information Technology the Superior University Lahore, Pakistan
abdulrehman.cs@superior.edu.pk

Mudassir Ali
Department of Computer Science Information Technology the Superior University Lahore, Pakistan
mudassir.ali9522@gmail.com

ABSTRACT

The use of automatic number plate recognition (ANPR) systems has gained popularity in recent years including traffic control, law enforcement, toll collection as well as management of parking. However, older ways always have problems with scale, deployment and support. In this paper, we describe how to implement the You Only Look Once (YOLO) object detection model with the help of DevOps principles in order to increase the automation, accuracy and scale of ANPR systems. In this study, the authors relied on the YOLO real-time detection algorithm to implement the core machine learning model for detecting and reading number plates of vehicles with high accuracy. As a result of DevOps practices, the use of YOLO in a DevOps pipeline provides the ability to continuously integrate, develop, test deploy and update the model as new datasets are provided to the system. This approach uses containerizing (Docker) and orchestration (Kubernetes) for scalable application deployment and used CI/CD tools including Jenkins and GitLab CI for automated build and deployment processes. The authors also integrated system monitoring tools on the system to receive timely information, improve performance, and resolve adverse events to ensure the desired level of system reliability and availability.

1. Introduction

In various works of employment, including monitoring of traffic, parking management, and administration, computerized license number plate identification has grown more notable in recent years. Regarding exactness, speed, and adaptability, the normal number plate recognition approach regularly experiences facing barriers [1]. The YOLO technique has displayed development in improving the overall working of these given systems due to its real-time entity identification capabilities. The implementation and use of this YOLO model can be tough and more time. This research aims to facilitate the creation, deployment, and preservation of computerized number plate identification systems by using YOLO with DevOps methods. The ANPR (Automatic Number Plate Recognize) system go well with rapid increase is more important for competent traffic management, traffic flow control, law enforcement & security apps [2, 3].These given systems manage the automate process of verifying LPs (Vehicle License plates) that is much important for observing and handling vehicle traffic, particularly in faster development cities [4, 5].

Automated recognition of blockchain payment systems has become increasingly important today in urban settings that are interconnected due to the growing need to maintain traffic, uphold law and order and facilitate smart transportation systems. Commonly used number plate recognition systems tend to provide poor accuracy in real time and more so in areas of high traffic where everyone is looking for easy and cost-effectiveness in scalable solutions [6]. However, recent developments in deep learning and the application of some models for object detection like YOLO have shown improvements in both accuracy and speed. Nonetheless, there is need for an effective model deployment strategy to cope with such demands because the production environments are appealing and entails processing of streaming data. This is where the principles of DevOps come in and help in building smooth processes for integrating, deploying and keeping an eye on YOLO-based license plate recognition systems. Organizations that embrace the integration of two advanced technologies, machine learning and software engineering are able to implement YOLO through DevOps [7,8].

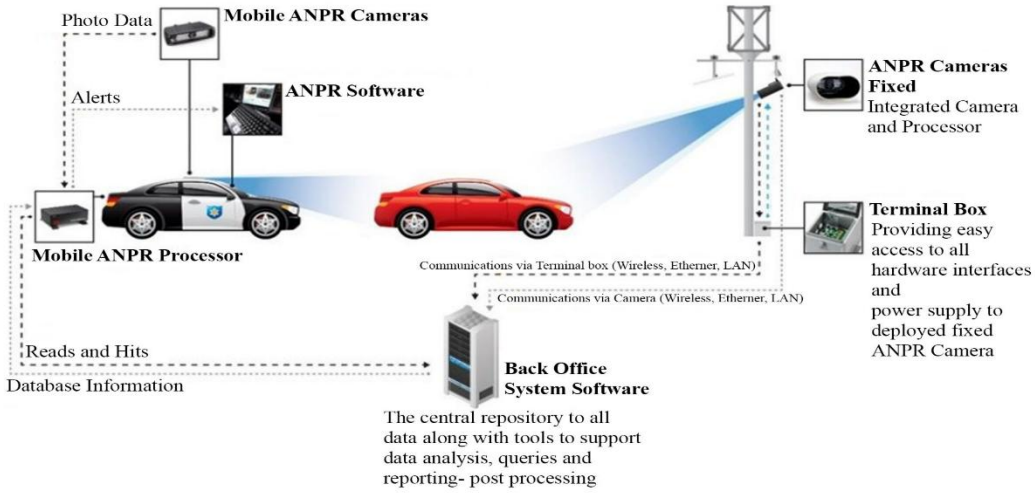


Figure 1: The generalized model for data fetching through DevOps [9].

This integration delivers a continuous pipeline that not only automates model deployment but also adds the aspect of scalability and continuous improvement to the pipeline thus optimizing it for the realities of deploying YOLO in practice. Differently from classic detection models that perform the localization and classification tasks separately, YOLO does both at once during a single processing of the complete image [10, 11]. This is what allows the authors to name it "one-stage" architecture and will in some cases using YOLO suit the best for applications such as vehicle number plate recognition, where speed and accuracy are crucial. Since its original concept was launched, numerous enhancements in the design and implementation of system characteristics have been implemented into YOLO, which made it more effective and stronger for various uses than its predecessors [12,13].

Table 1: Summary Of given tech that is used in YOLO Based ANPR (Automate Number Plate Detection with DevOps combination

Component	Technology/Tool	Role in the System	Ref
Object Detection	YOLO (You Only Look Once)	Detects automatic vehicle number plate in image and video stream and processes them in real-time	[14]
Version Control	Git, GitHub/GitLab	Controls the versioning history of the source code manages code branching and merging and supports collaboration for multiple developers	[15]
Continuous Integration	Jenkins / GitHub Actions	Actions test and integrate the changes done to the codebase automatically	[16]
Containerization	Docker	Ensure the apps and all its dependencies are included and can be deployed in a consistent manner across multiple environments	[17]
Orchestration	Kubernetes (Optional)	Handles the deploying, scaling and container availability	[18]
Cloud Platform	AWS /Microsoft Azure / GCP	Provides computing and storage resources and hosts the apps	[19]
Monitoring and Logging	Prometheus + Grafana / ELK	Provides logging of events for the system and application for the purpose of system health monitoring performance tracking and event logging for future debugging and analysis	[20]

Automated Deployment	Ansible / Terraform	Provide the capability to implement infrastructure as code (IaC) and provide better budget and cost control for managing the system infrastructure	[21]
Dataset	OpenALPR / Custom Dataset	Trains YOLO so that it can recognize number plates with greater accuracy	[22]
Programming Language	Python	Integration of DevOps tools with YOLO was done using Python as well as implementation of YOLO itself.	[23]

Recently the DevOps approach has become widespread in order to improve and simplify the processes related to the development, delivery and support of the software. Utilizing DevOps concepts such as CI/CD, containers, and IaC, organizations can deliver software, ease scaling, and provide solid coverage of deployments. Especially when paired with ML models, DevOps gives the ability to "bring them production-ready" as applications like YOLO in real time need active changes and don't stop after they've been deployed. Incorporating YOLO in the DevOps pipeline for automated vehicle number plate recognition is advantageous in that it optimizes the precision of the models and the performance requirements. With the help of cloud and container management, it is possible to effortlessly scale this system up, handle numerous data, and work under high volume of traffic [24]. This application has great prospects in intelligent transport systems and public safety systems enhancing vehicle surveillance and control functionalities.

Ongoing computerized methods for identifying license plates constantly have speed & correctness issues, which makes them unproductive for a variation of uses. Researchers & developers have faced problems since experiencing algorithms e.g., YOLO are tough to create & deploy. While keeping adaptability & responsibility, a more automated & productive method is essential to improve the overall performance of number plate identification systems [25].

2. Literature View:

In different sections, like toll gathering, parking administration & law enforcement organization, automated vehicle number plate identification systems are necessary. For a safe future, productivity, & regulatory conformity to be ensured, these given systems must be accurate & quicker. The standard methods for identifying license plates normally include human participation & are prone to error & delays. The implementation of deep learning methods, e.g. YOLO, has changed the section by making it possible to identify the objects in real time with a high level of accuracy.

It is widely known that the You Only Look Once (YOLO) method can rapidly and correctly recognize objects in images or video frames[26,27]. For entity observation tasks, YOLO is a constructive election since it cracks the input picture into a framework network and creates direct predicting about joining boxes & class possibilities. In the sector of motorcar number plate identification, YOLO has given a demonstration of

promoting results in exact recognizing & locating license plates in complex environments[28]. Conversely, DevOps is the gathering the approach that automates & simplifies the procedure of generating, testing, transferring, and monitoring applications by combining software evolution (Dev) with IT functioning (Ops). Observers & innovators may deliver the implementation process, enhance the association, and ensure the safety of the identification system by consolidating DevOps conventions into the constitution & implementation of machine learning algorithms like YOLO. Although, there is not a lot of research on the topic of YOLO & DevOps coordinate for automated vehicle number plate identification. Most of the midway research points on one of the excellent DevOps methods in software evolution or the practical feature of YOLO deployment[6]. A unique chance to enhance the productiveness & achievement of number plate identification systems appears by the merging of these two areas[29].

Table 2: Relative Analysis of Similar Studies in the Detection of Car Identification Plates

Ref	Procedure	Approach	Leverage	Controls
[30]	Image Analysis with OCR	OpenCV, Tesseract OCR	Very easy to deploy	Low correctness in low light
[31]	CNN-based Plate verification	TensorFlow, CNN	High level correctness under manage situation	Little slow inference time, not real-time
[32]	Hybrid ML Model	SVM + HOG Features	Good in detecting static images	Not acceptable for video streaming or real-time work
[33]	YOLOv3 for Real-time Detect	YOLOv3, Python	Real-time working, better correctness	Specific DevOps pipeline collaboration
[34]	YOLOv5 with DevOps Pipeline collaboration	YOLOv5, Docker, Jenkins, AWS	Real-time detect, automate implement, scalable	First setup difficulty, need DevOps expertise

To facilitate a final framework for computerized vehicle number plate reorganization, this research tries to cover the gap between DevOps techniques & deep learning methods like YOLO.

This research will decide the current level of awareness, analysis gaps, & possible offering in the given topic by accomplishing a comprehensive literature study. Further we recognize vehicle number plates in foggy environment using fog-hazing algorithm[35]. The study proceeds & method to be working in implementation YOLO through DevOps for enhanced number plate identification accomplishment will be knowledgeable by the awareness collected from the figure of existent literature.

3. Method & Material:

3.1 Problem Definition and Requirements Gathering

The grand plan for deployment of the YOLO method through DevOps for computerized vehicle number plate identification is report in the strategy sector portion of this study proposal. The methodology's main points, the study plan, data gathering techniques, sample choosing & data examinations methods—will be enclosed in given information below. This procedure fused the YOLO technique implementation into a DevOps source to make sure uninterrupted incorporation implementation, & observing of computerized car number plate identification system. The System Requirements are Identify and comprehend vehicle number plates in an identified video or image stream or a series of said streams. Specify data sources like CCTVs or traffic cameras. Choose the YOLO version according to the performance needs to the application (for instance, the versions can range from YOLOv5 to YOLOv8). Determine the type of deployment platform that is suitable (i.e. cloud-based, stand alone, or hybrid). The application must be able to process incoming data almost instantaneously. The application can dynamically adjust to the needs of the users. The application will adhere to security measures, and this is consistent with compliance requirements when dealing with vehicle sensitive data.

3.2 Data Collection and Preprocessing

A vehicle images database with different number plate's models, reflectivity, fonts, light conditions, camera tilts, and so forth is required. Annotate the dataset manually or automatically using the available tools such as labeling which have the necessary features for annotating license plates. Data Preprocessing: Alter images to suit the intended purpose and thus perform different operations like resizing and even normalization. Convert the dataset into three segments such that there is a training and validation and a testing segment. Include other related structures such as data augmentation into the algorithm to make the model less sensitive to the overfitting problem.

3.3 Proposed Technique for Processing Pipeline Using (DevOps Cycle)

The research will engage a hybrid-approach technique, blended informal & assessable method to achieve its aim. Model coaching, software development, evaluating & implementing will all be a constituent of the DevOps deployment of the YOLO method. Since the research plan will be insistent, the number plate identification system may be consistently promoted and enhanced. The proposed Technique for Processing Pipeline Using (DevOps Cycle) is explained in steps below:

Step 1: Collecting Data and Preparation of Making benefit of a wide database of vehicle number plates. To make use of image enhancement (occlusion, haze, motion blur) while Applying software including the LabelImg. Roboflow, identify plates.

Step 2: Train the Yolo Dataset

Use datasets with annotations to teach YOLOv8. Evaluate mAP, efficiency, and remembering to judge

progress. Exports the loads which you recently worked .

Step 3: Development of an application programming interface

Embed the newly acquired method into an isolated service which utilizes the FastAPI. Capture image/video transmissions via terminals and give back information and plate locations.

Step 4: Containerizing

Packaging applications with the language Python, the YOLO OpenCV, & FastAPI with Docker itself. Add the container version to the GitHub as a Containers Directory or Docker Hub.

Step 5: Combination of both the CD and CI

Jenkins workflows & GitHub activities to: Lint and verify the code. Patch pushing trigger deployments. Deploy the Docker picture onto the Kubernetes clusters.

Step 6: Deployment

Take advantage of Kubernetes for: Management of scaling (HPA CPU/traffic). Pod distribution load balancing. Deployment with no downtime.

Step 7: Systems For Logging & Monitoring

Systems for monitoring and visualization of insight. CPU, memory, inference latency. Grafana and Prometheus. For log storage and analysis of logs for APIs, performance monitoring, ELK stack can be utilized. Monitoring and alerting for stale models, degraded unknown metrics, failed predictions.

3.3.1 Managing Complications

Foggy conditions. Having to respond is the issue.

The employing of methods for preprocessing (such as using a dark channels prior) is the correct approach. Plate limited viewpoints. Concern: Replying

Solution: Training models with datasets of partially filled license plates and applying OCR filtering. Responses in multiple languages. Problems: Responding. Implementation of multilingual OCR engines such as EasyOCR or Tesseract with language packs. Low light conditions or at night. Problem: Responding. Histogram equalization / datasets enhanced with IR.

Compliant & Protection

While in transmission as well as rest, footage or pictures has to be encoded. Access control based upon the roles of the DevOps tools. Confirm compliance towards the GDPR's collection requirements.

3.3.2 Description of assessment criteria

Active metrics of the model: mAP (IoU=0.5), FPS, latency, and other relevant measures. Active metrics of the System: Deployment time, uptime percentage, and performance in scalability tests. Active metrics of CI/CD: Average time for builds and tests, rollback rate, and success rate.

3.3.3 Expected Program Outcomes

A workflow which employs a DevOps-ready the YOLO framework. An architectural framework for a deployment template that can be utilized and modified for open source.

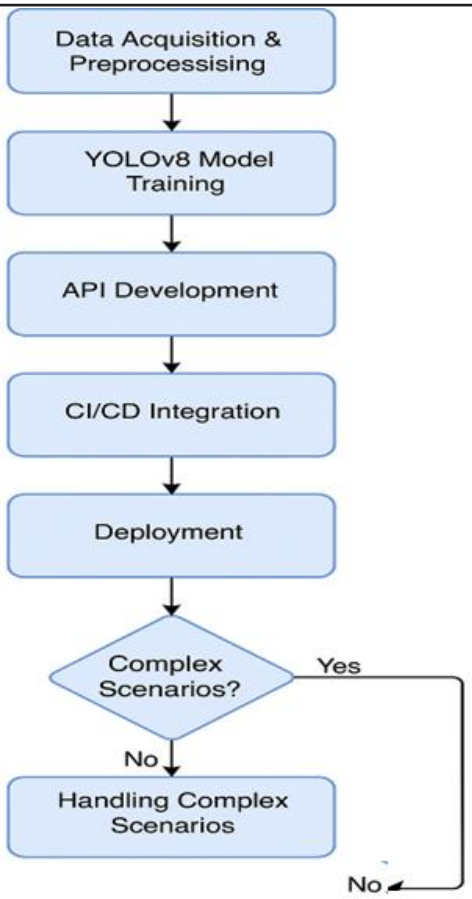


Figure 2: Flow Chart for Proposed Model

The proposed framework will be able to allow variability and scalability to different configurations and scenarios. Owing to its modular nature, which is facilitated by DevOps practices, the system can be relatively easily tailored to fit and augment existing infrastructure making it suitable for being transferred to a wide variety of markets and sectors. This research will contribute to the knowledge by exploring the relationship between DevOps processes and deep learning algorithms and their application within the specific domain of vehicle number plate recognition systems. The insights gained from this research will be very beneficial to practitioners, scholars, and developers in the different fields such as software engineering, machine learning and computer visualization.

4. Model Training and Evaluation (YOLO)

The model's level of accuracy in object detection should be evaluated with a set evaluation Metric that measures its average precision level such as map for instance.
Tests of generalization will be conducted on the out of sample held out test dataset held for the purpose.

4.1 Model Training and Evaluation (YOLO)

4.1.1 Model Selection and Configuration:

License plates recognition and detection must include configuration of the specified Yolo model by attaching license plates and specifying appropriate detection targets.

4.1.2 Developing application in Docker Containers Containerization:

Deploy the learned model in a Docker container so that the model usage is uniform even in different environments. Pack dependencies needed like python, OpenCV, and yolo inside the container. Implement container orchestration technology, for example, Kubernetes, for scalability if required.

4.1.3 API Development of Applications in REST

Write or otherwise implement an API with Flask, Fast API or any other framework for prediction. The mentioned API should be used to take pictures or video frames, call the yolo model and return the coordinates of bounding boxes around license plates as detection output.

4.1.4 DevOps Pipeline Setup at Work Integration Implementing Continuous Integration (CI):

Automated testing and building of the CI pipelines is possible upon configuration. Include a version control system such as git and integrate it a continuous integration and deployment tool such as Jenkins, GitLab CI or GitHub action. Unit tests and integration tests are initiated automatically to help foster good code practices during programming. Enable the automatic promotion of releases to be staged to the Staging and production environments by configuring deployment pipelines. Infrastructure installations should not involve running commands on dynamic resources, but rather use IaC tools such as Terraform, AWS CloudFormation. As pointed out by the proposition, there are a number of subtasks that need to be looked after under the functional relation of cloud infrastructure development.

4.2 Scalability and Performance Measurement:

Hosting the containerized application in the cloud using AWS, Azure or GCP container services. Implement auto-scaling groups and load balancers as interfaces to various levels of workload distribution. Ownership and Responsibility for Storage and Data Use a flexible storage resource to manage videos or pictures of the automobile (such as Amazon S3).Use appropriate databases like Amazon DynamoDB or PostgreSQL for storing image data, information about when and where images were taken and registered plate numbers.

4.2.1 Policy and Procedures of Data Management in Respect of Eventual Consistency:

Monitor and keep records of application performance indicators using Prometheus, Grafana or AWS cloud watch. Utilize ELK Stack (that is, Elasticsearch, Logstash, Kibana) or Amazon CloudWatch Logs to monitor for errors and then conduct analysis on them. Validation Activity and Assurance of Design and Implementation Validation Integration Testing: The focus of the test will be on the specific instance of logging an automobile's plate number, with the outcome expected that the vehicle will be captured incorporating the system in its uses. Performance testing is Attempt to deploy the solution to high resource concurrent users in order to stress the system and evaluate its behaviors.

4.2.2 Configuration Security Testing:

Conduct configuration security scans in order to detect security weaknesses but take into consideration the APIs used to provide public services.

To attain its intention, the study will use a hybrid-approaches technique that integrates informal & assessable methods. This DevOps deployment of YOLO technique will involve representation coaching, software evolution, demanding, and implementation. The vehicle number plate reorganization system perhaps frequently improved & clarified attributable to the insistent environment of the study plan. Figure for given research will approach from accessible-source software machine learning substructure, publicly approachable images set of information, & fulfillment evaluation tools. Exactly to collect & examine the data, images set of information will be preprocessing, the YOLO technique will be coaching, demanded on secret information, & completion standards will be evaluated. Different tools e.g. Docker, Source code management (Git), & Jupiter Notebooks will be employed for source control, data collecting & model coaching. Coaching the YOLO technique on the selected set of information using machine learning frames e.g. Azure Machine Learning or PyTorch at hand the data interpretation procedure for this research. The correctness of car number plate identification in test image at hand is used to approach the model's work. Comparing the system to traditional procedures & approaching how DevOps proceed towards influence the implement procedure are more distant features of the examination.

4.3 Validity and Reliability:

To guarantee the correctness & stability of facts & investigation, the research will defend to production better exercise for evolving deep learning imitation. Optimization parameters at hand modified by rotation estimation methods, fact effluent process will be noted, & model demanding at hand finish on various test sets. As comparison with traditional study & observed facts will be nearly new to verify the discovery.

5. Results and Contributions:

This study scheme is expected to prosperously combine the YOLO technique with DevOps procedure for computerized number plate recognition. The YOLO prototype at hand deployed using DevOps, assembling a decent & successful structure that can exactly discover & identified license plates in real-time. The experimentation benefaction is various & may be added up along these lines:

5.1 Improve Correctness: The computerized number plate identification system is forecast to achieve big time of correctness in finding & identifying number plates below a diversity of framework by usage the YOLO technique's abilities. The number plate recognition system from any chaotic scenarios is going to be made possible through the real time object detection by YOLO. The appropriate number plate recognition system will be developed, tested, deployed and monitored using DevOps processes. The model integration level is supposed to increase thus speeding up the outcomes of model training, deployment and maintenance. The results of the research are applicable to the environment of such areas as the smart city, parking control, transport, and security. An automatic identification of vehicles by their number plates, which is the subject of the study, might enhance security, efficiency, and adherence in many spheres where rapid and correct identification of vehicles is critical. The present research project, through the achievements of these expected outcomes and contributions, aims at enhancing the developed automated car number plate recognition system and demonstrating the applicability of deep learning algorithms such as YOLO in the DevOps environment.

The principles of conduct of the people involved in the research are of great importance for the content of the study, the rights of the participants, the ethical frameworks, standards, and laws. There are a number of ethical considerations that ought to be taken into account when applying DevOps with the YOLO algorithm for automatic vehicle plate recognition system:

5.2 Informed Consent: Informed and voluntary consent is important especially where human subjects or their personal information are involved. Participants should be provided with requisite details such as the risks and methodologies to be employed as well as the objectives of the study and what their rights are prior to enrollment.

5.3 Data Privacy and Prejudice and Equity: For whatever reasons, it is paramount to protect the confidentiality and privacy of the information solicited for the study. Of course, necessary measures must be taken to ensure the confidentiality of sensitive information, such as the images of license plates, and that such information be appropriately de- identified and secured.

Prejudice and discrimination are not to be in existence when designing or using the automated number plate recognition system. In order to attain equitable and just results, there is a need to take steps towards elimination of any sympathies that can be embedded in the data, algorithms or even in the decision-making frameworks. Adherence to the relevant laws, regulations and ethical standards that govern the research activities is very important. Research participants have to be mindful of the compliance requirements of the institutions, GDPR and other moral principles at all times during the investigation. It is very important not to lose transparency in the research processes and results. Procedures, outcomes, and relations regarding other parties should be honestly reported by researchers to retain credibility and responsibility.

The research team may conduct the study in an active responsible way, taking the ethical and legal aspects into consideration at all stages of the research.

Research studies always seem to create dilemmas especially when they deal with ethical issues such as developing human embryos and stem cells. Therefore, when handling ethical dilemmas, it would be beneficial to seek the assistance of an ethics committee, regularly implement ethical assessments of the study and continuously strive to develop solutions that would uphold the mission of the project as well as its most profound ethical standards.

Moreover, to enhance the study's credibility and trustworthiness, the research will be performed in line with ethical principles. Such ethical conditions will ensure that the research team focuses on the ethical and responsible approaches of the study and thus, enhance integrity of and trust in the research process.

5 Conclusion

Through the use modern methods for image processing, this research examines the establishment of an efficient car license plate recognition system optimized for hazy conditions. Through the use modern methods for image processing, this research examines the establishment of an efficient car license plate recognition system optimized for hazy conditions.

To enhance readability while minimizing obstructions, the study participants merged flexible filtering using OCR-based imagine noise elimination approaches. There are sharpened algorithms aimed to trim number plates that account for background fog and complex features. The alphanumeric character extraction processes were fortified using OCR algorithms and machine learning models to enhance the integrity of number plate recognition even in challenging weather conditions. Evaluating accuracy, recall, and F1 metrics showed much greater results when compared to traditional systems. This research addressed a critical need to improve detection systems for number plates to work in challenging conditions, thus operational fortitude. Additional work is absolutely needed to.

References:

[1] D Islam, T Mahmud, T Chowdhury , "An efficient automated vehicle license plate recognition system under image processing" Indonesian Journal of Electrical Engineering and Computer Science, 2023 model for smart parking systems"- Ain Shams Engineering, 2023 2023

[2] MM Abdellatif, NH Elshabasy, AE Elashmawy ,"A low-cost IoT-based Arabic license plate recognition

[3] A Mhatre, P Sharma ,"Deep Learning Approach for Vehicle Number Plate Recognition System with Image Enhancement Technique". International Journal of Intelligent Systems and Applications in Engineering,

[4] C Wei, Z Tan, Q Qing, R Zeng, G Wen Sensors, "Fast helmet and license plate detection based on lightweight YOLOv5",2023

[5] AJA Tahseen, S Sotnik, T Sinelnikova, V Lyashenko, "Binarization Methods in Multimedia Systems when Recognizing License Plates of Cars" ,2023

[6] RA EKE, SI ORAKWUE, "Automatic License Plate Recognition System", 2023

[7] SS Tippannavar, Y Deekshitha, M Shivakumar, KG Kanvihalli "License Plate Recognition for Indian Vehicles based on Image Processing using OpenCV" 2023

[8] An Ammar, A Koubaa, W Boulila, B Benjdira, Y Alhabashi Sensors, "A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference" 2023

[9] MM Khan, MU Ilyas, IR Khan, SM Alshomrani , "A review of license plate recognition methods employing neural networks" , - IEEE, 2023 IJCSP24C1046 International Journal of Current Science (IJCSPUB) www.ijcspub.org 414 www.ijcspub.org © 2024 IJCSPUB | Volume 14, Issue 3 July 2024 | ISSN: 2250-1770

[10] H Shi, D Zhao , "License Plate Recognition System Based on Improved YOLOv5 and GRU" IEEE Access, 2023

[11] Xianli Jin, Ruocong Tang, Linfeng Liu, Jiagao Wu ,"Vehicle license plate recognition for fog-haze environments" , 2023

[12] Zhang, X., Yu, X., & Zheng, S,"Foggy Environment Vehicle Number Plate Detection Based on Deep Learning". IEEE Transactions on Intelligent Transportation Systems, (2023).

[13] Brownlee, J. Deep Learning for Computer Vision, "Image Classification, Object Detection, and Face Recognition in Python (2nd ed.). Machine Learning Mastery", (2023).

[14] Januszewski, M., & Zalewski, L, "A comprehensive guide to image processing and computer vision with OpenCV 5 and Python 3.9", . (2023).

[15] Abid, M. R., & Khan, M. A, "Explore the new features of TensorFlow 3.0 to build powerful machine learning models" (2023).

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Dec. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[17] X. Jin, R. Tang, L. Liu, and J. Wu, "Vehicle license plate recognition for fog-haze environments," IET Image Process, vol. 15, no. 6, pp. 1273–1284, May 2021, doi: 10.1049/ipr2.12103.

[18] Y. W. Gustian, B. Rahman, D. Hindarto, and A. B. P. B. Wedha, "Detects Damage Car Body using YOLO Deep Learning Algorithm," SinkrOn, vol. 8, no. 2, pp. 1153–1165, May 2023, doi: 10.33395/sinkron.v8i2.12394.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497

[20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.05587

[21] A. Vikram, J. Akshya, S. Ahmad, L. Jerlin Rubini, S. Kadry, and J. Kim, "Deep Learning Based Vehicle Detection and Counting System for Intelligent Transportation," Computer Systems Science and Engineering, vol. 48, no. 1, p. `115-130, 2024, doi: 10.32604/csse.2023.037928.

[22] H. I. Nasim et al., "Fog-Resilient Bangla Car Plate Recognition using Dark Channel Prior and YOLO. "An Integrated Approach for Traffic Scene Understanding from Monocular Cameras Towards Resource-constrained Perception of Environment Representations with Multi-task Convolutional Neural Networks."

[23] A. A. Ravindran, "Internet-of-Things Edge Computing Systems for Streaming Video Analytics: Trails Behind and the Paths Ahead," Dec. 01, 2023, Multidisciplinary Digital Publishing Institute (MDPI). doi: 10.3390/iot4040021.

[24] R. Gupta, A. Jain, and S. Garg, "The Vehicle Number Plate Detection in Foggy Environments Using Image Processing," 2024. [Online]. Available: www.ijcspub.org

[25] Al-Hasan, T.M.; Shibeika, A.S.; Attique, U.; Bensaali, F.; Himeur, Y. Smart speed camera based on automatic number plate recognition for residential compounds and institutions inside Qatar. In Proceedings of the 2022 5th International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 7–8 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 42–45.

[26] Ahmad, M.B.; Musa, U.F.; Dahiru, M.; Abimbola, M.B. Advantages of Automated License Plate Recognition Technology. Eng. Technol. 2024, 4, 10–15.

[27] Job, R.; Truong, J.; Sakashita, C. The ultimate safe system: Redefining the safe system approach for road safety. Sustainability 2022, 14, 2978.

[28] Khalai, I. Road safety performance monitoring practices: A literature review. Eurasia Proc. Sci. Technol. Eng. Math. 2023, 22, 99–110.

[29] Lubna; Mufti, N.; Shah, S.A.A. Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms. Sensors 2021, 21, 3028. https://doi.org/ 10.3390/s21093028

[30] Amitava Choudhary, Alok Negi "A New Zone based algorithm for Detection of License plate From Indian Vehicle" International Conference on Parallel, Distributed and Grid Computing December 2016.

[31]. Reji P. I., Dharun V. S "An Efficient Algorithm for Real Time License Plate Localization" ARPN Journal of Engineering and Applied Sciences December 2016.

[32]. Joseph Redmon, Ali Farhadi "YOLO9000: Better, Faster, Stronger" December 2016.

[33]. Rahim Panahi, Iman Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications" IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS August 2016.

[34]. Joseph Redmon, Ali Farhadi "YOLO: Unified, Real-Time Object Detection" May 2016.

[35]. Ketan S. Shevale "Automatic number plate Recognition System for Toll Booth Application" Internation Journal of Engineering Research and Applications October 2014.