International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 1, February 2025, pp. 365~373
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i1.pp365-373

Enhancing automatic license plate recognition in Indian scenarios

Abhinav Samaga[1], Allen Joel Lobo[1], Azra Nasreen[1], Ramakanth Kumar Pattar[1], Neeta Trivedi[2], Peehu Raj[2], Koratagere Sreelakshmi[3]
[1]Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, India
[2]Inferigence Quotient Private Limited, Bengaluru, India
[3]Department of Telecommunication Engineering, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Apr 24, 2024
Revised Sep 21, 2024
Accepted Oct 1, 2024

Keywords:

Darknet
License plate detection
Object detection
Object tracking
Optical character recognition
YOLO

ABSTRACT

Automatic license plate recognition (ALPR) technology has gained widespread use in many countries, including India. With the explosion in the number of vehicles plying over the roads in the past few years, automating the process of documenting vehicle license plates for use by law enforcement agencies and traffic management authorities has great significance. There have been various advancements in the object detection, object tracking, and optical character recognition domain but integrated pipelines for ALPR in Indian scenarios are a rare occurrence. This paper proposes an architecture that can track vehicles across multiple frames, detect number plates and perform optical character recognition (OCR) on them. A dataset consisting of Indian vehicles for the detection of oblique license plates is collected and a framework to increase the accuracy of OCR using the data across multiple frames is proposed. The proposed system can record license plate readings of vehicles averaging 527.99 and 2157.09 ms per frame using graphics processing unit (GPU) and central processing unit (CPU) respectively.

Corresponding Author:

Azra Nasreen
Department of Computer Science and Engineering, RV College of Engineering
Mysore Rd, RV Vidyaniketan, Post, Bengaluru, Karnataka 560059, India
Email: azranasreen@rvce.edu.in

1. INTRODUCTION

The onset of artificial intelligence and machine learning has revolutionized the entire society. It has made its way into a wide range of domains from education to agriculture to manufacturing. In recent years, the number of vehicles on Indian roads has increased dramatically, leading to a significant challenge for law enforcement agencies and traffic management authorities in keeping track of vehicle license plates. Automatic license plate recognition (ALPR) technology has emerged as a potential solution to this problem. ALPR systems use cameras and advanced software to automatically detect and read license plates on vehicles, allowingfor more efficient and accurate tracking of vehicles on the road.

In vehicle detection, an early significant use of deep learning was the implementation of region-based convolutional neural networks (R-CNN) on the KITTI dataset [1]. Despite its advancement, R-CNN suffered from slow computation due to the neural network's size. A major breakthrough came with you only look once (YOLO) [2] in 2016, which used a simple, efficient neural network for object detection, enabling real-time image and video analysis. YOLO surpassed R-CNN in sensitivity and processing speed, though both had similar precision [3]. The YOLOv4 model further improved accuracy, achieving 93% in vehicle model recognition, outperforming faster R-CNN and single shot detector (SSD) [4]. While most work has focused on YOLO, other methods, such as a scale-intensive convolutional neural networks (CNN) with a context-aware region of interest (RoI), have shown promise for highway scenarios but struggle with small, crowded vehicle detections [5].

Vehicle tracking is crucial for ALPR, relying heavily on multiple object tracking (MOT) to re-identify vehicles across frames. Using orientation-invariant feature embeddings and spatial-temporal regularization enhances performance but slows computation [6]. A key challenge is object occlusion, where obstacles hinder complete feature extraction. Predicting an object's position in the next frame via bounding box regression aids tracking small or occluded objects [7]. Tracking by association, which uses low-confidence bounding box similarities with track lets, helps recover true objects and filter background detections [8]. An advancement over DeepSORT called StrongSORT, improves on object detection, feature embedding, and trajectory association, excelling in scenarios with severe occlusion and making it ideal for vehicle tracking [9], [10].

The use of CNNs for ALPR has shown high accuracy on datasets like the Caltech and application-oriented license plate (AOLP) license plate datasets [11], but they struggle with rotated images. Modifying YOLO improved detection accuracy to 78% [12], yet it failed with oblique angles. Uthaib *et al.* [13] proposed a deep CNN for multiclass classification of license plates by country, which can help classify single and multi-line plates for better optical character recognition (OCR). Enhancing this model with a prepositive CNN before YOLO to handle angle rotation [14] allows for detection in unconstrained scenarios and better character segmentation [15]. However, this approach has only been trained on European and Brazilian vehicles, limiting its effectiveness on Indian vehicles.

Text recognition, crucial for applications like document digitization and number plate reading, traditionally uses CNNs for image understanding and recurrent neural networks (RNNs) for char-level text generation. One approach replaces RNNs with two fully convolutional one-stage object detectors for simultaneous license plate (LP) and character detection and classification [16]. Another employs a transformer architecture for both image understanding and text generation, showing promise in various text recognition tasks [17]. Bilingual license plates are addressed using a deep learning-based recognition system for dual language detection [18]. Post-correction and error analysis significantly enhance OCR accuracy. A voting mechanism with five similar mixed models improves OCR accuracy [19], and a semi-supervised, lexically aware method with a count-based language model reduces errors by 15% to 29% [20].

Several end-to-end ALPR systems have been developed for various scenarios. An ALPR system based on YOLOv5 trained on Indian datasets for internet of things (IoT) platforms struggles with low-light conditions [21]. Using multiple CNNs for image preprocessing in low-light conditions achieves 98.13% accuracy in vehicle recognition but fails to detect multiple license plates in a single image [22]. Another approach uses RCNN Inception V2 COCO for license plate detection and tesseract long short term memory (LSTM) RNN for text recognition [23]. A system utilizing faster RCNN and image preprocessing techniques before tesseract for OCR has also been proposed [24]. Combining multiple CNN models to extract and label license plate features with an RNN model yields 92% accuracy on a custom Algerian license plate dataset [25]. Additionally, character recognition network (CR-NET) employs post-processing for improved character accuracy within an ALPR system [26].

Despite the various improvements in ALPR technology, the implementation of such systems in Indian scenarios has been limited. While there have been various advancements in object detection, tracking, and optical character recognition, the development of integrated pipelines for ALPR that take advantage of multiple frames have yet to be seen. This paper proposes an architecture for ALPR in Indian scenarios that aims to bridge this research gap. The main contributions of the proposed technology are: i) a novel architecture for ALPR that can track vehicles across multiple frames, detect number plates, and perform OCR on them; ii) the entire pipeline is enhanced for Indian scenarios by applying transfer learning techniques using a custom Indian vehicle dataset consisting of 2000 images; and iii) the system also applies a set of majority pooling techniques utilizing string distance measures that are aimed at improving the accuracy of the OCR results.

The paper outlines the current technologies involved in ALPR, followed by a brief description of the vehicle detection and tracking module. Further it provides a detailed description of the license plate detection, its training and functioning along with the dataset collected. The framework used in conjunction with OCR to improve the accuracy is presented and finally it concludes with experimental results and future scope.

2. METHOD

The proposed pipeline seeks to enhance existing open-source modules used for vehicle tracking and license plate recognition. It builds upon the methodology proposed by Tu and Du [27] by utilizing multiple frames, accommodating oblique number plates, and tailoring it specifically for the Indian context. It consists of an integrated codebase which has four modules, *i.e.* vehicle detection, vehicle tracking, license plate detection and optical character recognition. The codebase is built using the state-of-art YOLOv5 framework vehicle detection and StrongSORT for vehicle tracking, while the license plate detection and optical character recognition is done using the module from ALPR unconstrained. Figure 1 gives the systemarchitecture of the proposed methodology.

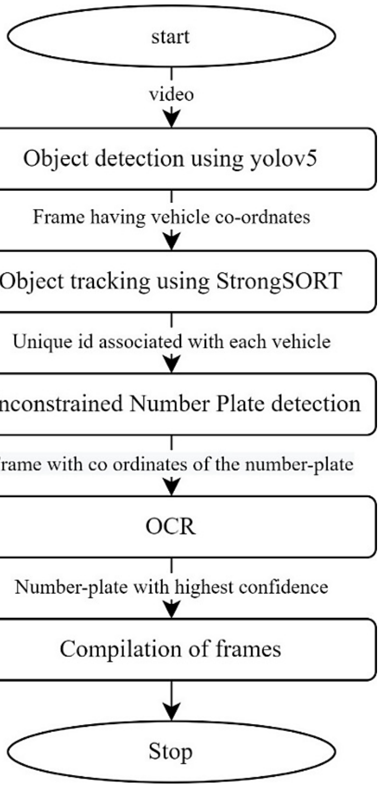

Figure 1. System architecture

2.1. Vehicle detection

It is the first module of the pipeline Figure 1 which uses the YOLOv5. YOLOv5 works by dividing an input image into a grid of cells and predicting bounding boxes around each object in the image. The algorithm predicts the class probabilities and location of the object within each bounding box [2]. This is achieved through a deep neural network architecture, which consists of a backbone network for feature extraction, and a head network for object detection and classification. The backbone network is typically a convolutional neural network (CNN) that extracts features from the input image at multiple scales. The head network then uses these features to make predictions about the location and class of objects in the image. YOLOv5 also incorporates anchor boxes, which help to improve the accuracy of bounding box predictions by providing a prior estimate of the shape and size of the objects in the image [4]. This neural network is used to detect all the vehicles present in a single frame, detect their positions and extract their coordinates and identify the category of vehicles that each vehicle belongs to. It takes in a single video frame as an input, along with the different hyper parameters specified by the user and returns the coordinates of each vehicle in the frame and which category it belongs to.

2.2. Vehicle tracking

StrongSORT is an advanced version of the simple online and real time tracking (SORT) algorithm, which is a widely used algorithm for tracking objects in video streams. StrongSORT combines the SORT algorithm with a deep learning-based object detector to improve tracking accuracy and robustness. The object detector is used to detect objects in each frame of the video stream, and the SORT algorithm is used to track the objects over time. StrongSORT can handle multiple objects with different motion patterns, sizes, and aspect ratios, and can track objects even when they are occluded or partially visible [10]. The usage of a deep learning-based object detector provides more accurate and robust object detections compared to traditional methods such as background subtraction or blob detection.

The module extracts feature from detections generated by the vehicle detection module, assigning a unique identifier (ID) to each entity. It processes a series of frames with vehicle detections, along with user-specified hyperparameters, to predict the object's location in subsequent frames, enabling object tracking. This tracking capability allows the module to assign a unique identifier to each vehicle sharing a common feature map.

2.3. License plate recognition

The ALPR unconstrained is a license plate detection technology that is able to perform well over a variety of scenarios and camera setups. It is capable of detecting the license plate in many different cameras poses and estimating its distortion, allowing a rectification process before OCR [15]. It was originally trained on a Brazilian number plate dataset with over 200 manually annotated images along with augmentations. This model was ineffective in the Indian scenario. Hence, the model was transfer-learned with a custom dataset of 2,000 images. The images included vehicles majorly from Indian roads. It mainly consisted of cars along which, there were trucks, buses, tempo travelers, vans, and auto rickshaws. Traditional datasets contain only 2 sets of coordinates representing the top left and bottom right corner, but this assumes that the license plate is in an upright position. Hence to take into consideration the oblique view of the license plates, 4 coordinates are required. These images were manually annotated with 4 sets of coordinates marking the corners of the license plate. For this purpose, an open-source tool “makesense.ai” was used to annotate the images [28]. These annotated images along with various hyper parameters were used to train the model and the results were analyzed. An optimal set of parameters that provided sufficient accuracy and limited false detections were taken for the final model. Figure 2 gives a set of images from the collected dataset.

This custom trained model is used for detecting the license plate and extracting their coordinates and the cropped license plate image for each vehicle detected in a video frame. The module also deals with performing a set of image augmentations to unwrap and rotate the cropped image to get an upright image. The module consists of multiple CNN models that are used to first detect the license plate, even in oblique angle scenarios. It generates a polygon-shaped bounding box to cover the license plate region. The cropped image is then rotated to an upright position. This can be further used by other modules for text detection and OCR.

Figure 2. Sample images of collected dataset

2.4. Optical character recognition

Optical character recognition uses a combination of image processing techniques and machine learning algorithms to identify and recognize the characters in an image or document. The OCR process typically involves several steps, including pre-processing, segmentation, feature extraction, and classification. In the preprocessing step, the image is enhanced and corrected for distortions, such as perspective distortion or uneven lighting. In the segmentation step, the individual characters are separated from the background and each other. In the feature extraction step, the unique features of each character are extracted, such as the shape, size, and orientation. In the classification step, the extracted features are used to classify the characters into their respective classes, using machine learning algorithms such as neural networks or support vector machines.

After obtaining the detections and the corrected license plate image, OCR is applied on the same. The ALPR Unconstrained pipeline provided an OCR network that was based on Darknet. This model was trained on an artificially created data consisting of pasting a string of characters onto a textured background and then performing random transformations, such as rotation, translation, noise, and blur [15]. This model was applied on the obtained upright license plates to perform optical recognition.

Each license plate detected produced an OCR result. Using the results obtained from the tracking module, the license plate recognitions related to each tracked vehicle over a set of frames were used to perform the majority pooling. The majority pooling consists of a two-layer filter to be able to accurately determine the license plate of the vehicle. The first level consists of having a lower bound for the number of OCR results needed to take the license plate into consideration. If a given vehicle has a minimum of 10 OCR results associated with it, then it would be considered for the next layer. Vehicles failing to get past this criterion are dropped from the character recognition as they would have very low accuracy. A set of viable candidates consisting of detections having nine or ten characters are selected from the set of results. The next layer consists of checking the string similarity of the viable candidates with the rest of the detections utilizing Levenshtein distance as a metric. The candidate having the highest similarity is selected as the final candidate. Once these conditions are satisfied, a number plate reading is associated with that vehicle and no more license plate detections are carried out for that vehicle until it exits the frame of the video. These results can then be updated in a database along with a timestamp associated with it for further use.

3. RESULT AND DISCUSSION

The overall model performs sufficiently well for Indian traffic scenarios when tested on traffic videos taken from Bangalore on mobile phones and traffic cameras. This was done to simulate the diverse set of conditions that the model could experience. The approach highlights the model's adaptability and effectiveness in real-world settings common to India.

3.1. Vehicle detection and tracking

The submodules for vehicle detection and tracking perform a frame-by-frame analysis of the video feed. In each frame, the detected vehicles are encapsulated with a bounding box. Additionally, each bounding box includes a label that displays the vehicle class, the unique vehicle ID and the confidence score of the model in predicting the vehicle class as illustrated in Figure 3.

Figure 3. Results from vehicle detection and tracking

3.2. License plate recognition and OCR

The submodules used for the license plate recognition and OCR process the cropped images of the vehicles to detect the license plate. Once a license plate is detected, the four two-point coordinates are extracted from the image and passed to the next module. This is used to draw a red quadrilateral, bounding box around the license plate region, and displayed for each vehicle as shown in Figure 4.

The license plate region of each vehicle is cropped and passed to CNN that rotates the license plate, if required, to a frontal view image. This ensures that the OCR module has the best possible chance of reading the characters accurately. The OCR module processes the rotated license plate images as shown in Figure 5 and predicts the characters of the license plate. The readings are then compiled along with the bounding boxes and the values which are displayed in the frame with black bold letters above the license plate region as shown in Figure 6.

Figure 4. Bounding boxes around the license plate regions

Figure 5. Rotated license plate cropped images

Figure 6. Vehicle image with outer bounding box, license plate, bounding box and OCR readings displayed

3.3. Overall performance of the system

Given the multiple distinct modules in the ALPR pipeline, each of which have different functions, a unified evaluation metric is infeasible. Hence, to obtain the efficiency and accuracy of the pipeline, each module is evaluated separately according to its own function. As there is not a metric that can correctly evaluate the performance, the evaluation is done manually on a test video of Mysuru road which is 37 seconds long. There are 54 vehicles that can be seen in the video, which belong to different categories such as trucks, cars, buses, bikes, autos and so on. The pipeline was able to detect 53 of them correctly. There are 48 vehicles that are present in the video for more than 3 frames that were successfully tracked by StrongSORT throughout the video and maintaining the same vehicle ID and category. Out of the 54 vehicles in the video, 50 of them have their license plates visible with either the front/rear license plate in view with some cases having extreme oblique angles. Despite the conditions, the module performs well by detecting 45 vehicles. From the 45 vehicles whose license plates were detected correctly, OCR readings were obtained for all 45. Manually annotating the license plate readings for each vehicle and then comparing it to the readings obtained from the pipeline was the metric used to evaluate the module. The comparison was done character wise as well as string length wise. The result for the OCR module after applying the post-OCR correction utilizing Levenshtein distance was 84%. The results are summarized in Table 1.

Table 1. Accuracy of the different modules of the system

Module	No. of instances	No. of correct detection	Accuracy
Vehicle Detection	54	53	98%
License Plate Detection	50	45	90%
OCR	45	38	84%

3.4. Model time analysis

A time analysis script was run on the entire model and its submodules on two different systems: i) a system with a NVIDIA GPU (Google Colab instance); and ii) a system with an Intel i3 Core 10th Gen CPU. The video sample in both cases was a daytime traffic video shot outside RV College of Engineering on a mobile phone in 4K resolution. The video clip is 30 seconds long and the input image/frame size in both cases is set at 1920×1080 pixels per frame. Table 2 shows the average of each individual frame's results over 630 frames of the 30 second clip. It can be inferred that with the help of a GPU, the computation speed is exponentially faster and close to a real-time analysis of the video.

Table 2. Processing metrics for various modules using GPU and CPU

Sub modules	GPU	CPU
Pre-process	0.9 ms	44.8 ms
Inference	34.2 ms	2296.9 ms
NMS	2.2 ms	76.3 ms
StrongSORT update	44.6 ms	481.1 ms
License plate detection	137.57 ms	754.75 ms
OCR	233.24 ms	450.36 ms
Generating output	75.28 ms	119.88 ms
Overall time taken by the pipeline	527.99 ms	2157.09 ms

3.5. Inferences

The results demonstrate the improvement of the proposed methodology over the current state-of-the-art systems in the field of automatic license plate recognition systems when used in the context of Indian vehicles and scenarios. While the differences are not significant when comparing the different vehicle tracking modules, there are considerable improvements in the remaining modules of license plate recognition and OCR due to the unique systems and schemes followed in Indian vehicles for which existing models are unable to detect and process accurately. By fine-tuning the models on Indian-specific datasets and employing post-processing techniques, the proposed system is a novel approach to creating an efficient and accurate model for automatic license plate recognition in the context of the Indian subcontinent that currently faces a dearth of viable solutions. The comparisons between the proposed system and the existing models have been consolidated in Table 3.

Table 3. Accuracy rates on the video shot of vehicles on Mysuru road

Model	Vehicle detection	License plate detection	OCR
ALPR [12]	98%	88%	79%
ANPR [23]	n/a	89%	82%
Proposed System	98%	90%	84%

4. CONCLUSION

The proposed technology can successfully detect vehicles, track vehicles, and obtain their number plates with sufficient accuracy in the Indian scenario. A dataset with more than 2,000 images containing various classes of vehicles including cars, buses, trucks, vans, and auto rickshaws along with annotations for their number plates in oblique scenarios was presented. A two-level majority pooling algorithm utilizing Levenshtein distance was also implemented to improve the accuracy of the OCR results. There is a dip in the accuracy of the model in situations where the quality of the input video is low or there are fast moving vehicles. These can be improved by further tuning the OCR model using Indian datasets.

ACKNOWLEDGEMENT

The authors thank the institutions involved in the support and backing of this paper along with the persons who have guided us throughout the way.

REFERENCES

[1] Q. Fan, L. Brown, and J. Smith, "A closer look at faster R-CNN for vehicle detection," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 124–129, doi: 10.1109/IVS.2016.7535375.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[3] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car detection using unmanned aerial vehicles: comparison between faster R-CNN and YOLOv3," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, Feb. 2019, pp. 1–6, doi: 10.1109/UVS.2019.8658300.

[4] J. A. Kim, J. Y. Sung, and S. H. Park, "Comparison of faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," *2020 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2020*, pp. 1–4, 2020, doi: 10.1109/ICCE-Asia49877.2020.9277040.

[5] X. Hu *et al.*, "SINet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010–1019, 2019, doi: 10.1109/TITS.2018.2838132.

[6] Z. Wang *et al.*, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 379–387, doi: 10.1109/ICCV.2017.49.

[7] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 941–951, 2019, doi: 10.1109/ICCV.2019.00103.

[8] Y. Zhang *et al.*, "ByteTrack: multi-object tracking by associating every detection box," in *Lecture Notes in Computer Science*, vol. 13682 LNCS, 2022, pp. 1–21, doi: 10.1007/978-3-031-20047-2_1.

[9] X. Hou, Y. Wang, and L.-P. Chau, "Vehicle tracking using deep sort with low confidence track filtering," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Sep. 2019, pp. 1–6, doi: 10.1109/AVSS.2019.8909903.

[10] Y. Du *et al.*, "StrongSORT: make DeepSORT great again," *IEEE Transactions on Multimedia*, vol. 25, pp. 8725–8737, 2023, doi: 10.1109/TMM.2023.3240881.

[11] Z. Selmi, M. Ben Halima, and A. M. Alimi, "Deep learning system for automatic license plate detection and recognition," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, pp. 1132–1138, doi: 10.1109/ICDAR.2017.187.

[12] R. Laroca *et al.*, "A robust real-time automatic license plate recognition based on the YOLO detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–10, doi: 10.1109/IJCNN.2018.8489629.

[13] M. A. Uthaib and M. S. Croock, "Multiclassification of license plate based on deep convolution neural networks," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 5266–5276, 2021, doi: 10.11591/ijece.v11i6.pp5266-5276.

[14] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A new CNN-based method for multi-directional car license plate detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, Feb. 2018, doi: 10.1109/TITS.2017.2784093.

[15] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Computer Vision - ECCV 2018*, vol. 11216 LNCS, Springer International Publishing, 2018, pp. 593–609, doi: 10.1007/978-3-030-01258-8_36.

[16] Q. Huang, Z. Cai, and T. Lan, "A single neural network for mixed style license plate detection and recognition," *IEEE Access*, vol. 9, pp. 21777–21785, 2021, doi: 10.1109/ACCESS.2021.3055243.

[17] M. Li *et al.*, "TrOCR: transformer-based optical character recognition with pre-trained models," in *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023*, 2023, vol. 37, pp. 13094–13102, doi: 10.1609/aaai.v37i11.26538.
[18] I. R. Khan, S. T. A. Ali, A. Siddiq, and S.-O. Shim, "Multi-string missing characters restoration for automatic license plate recognition system," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 835–843, 2023, doi: 10.14569/IJACSA.2023.0140395.
[19] S. Drobac and K. Lindén, "Optical character recognition with neural networks and post-correction with finite state methods," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 23, no. 4, pp. 279–295, Dec. 2020, doi: 10.1007/s10032-020-00359-9.
[20] S. Rijhwani, D. Rosenblum, A. Anastasopoulos, and G. Neubig, "Lexically aware semi-supervised learning for OCR post- correction," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1285–1302, 2021.
[21] P. Batra *et al.*, "A novel memory and time-efficient ALPR system based on YOLOv5," *Sensors*, vol. 22, no. 14, pp. 1–12, Jul. 2022, doi: 10.3390/s22145283.
[22] S. Ranjithkumar and S. Chenthur Pandian, "Automatic license plate recognition system for vehicles using a CNN," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 35–50, 2022, doi: 10.32604/cmc.2022.017681.
[23] J. Singh and B. Bhushan, "Real time indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Oct. 2019, pp. 347–352, doi: 10.1109/ICCCIS48478.2019.8974469.
[24] C. Dias, A. Jagetiya, and S. Chaurasia, "Anonymous vehicle detection for secure campuses: a framework for license plate recognition using deep learning," in *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*, Sep. 2019, pp. 79–82, doi: 10.1109/ICCT46177.2019.8969068.
[25] M. Bensouilah, M. Zennir, and M. Taffar, "An ALPR system-based deep networks for the detection and recognition," in *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods*, 2021, pp. 204–211, doi: 10.5220/0010229202040211.
[26] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, 2021, doi: 10.1049/itr2.12030.
[27] C. Tu and S. Du, "A hierarchical RCNN for vehicle and vehicle license plate detection and recognition," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 731–737, 2022, doi: 10.11591/ijece.v12i1.pp731-737.
[28] P. Skalski, "Make sense," *GitHub*, 2019. https://github.com/SkalskiP/make-sense (accessed Jan. 16, 2023).

BIOGRAPHIES OF AUTHORS

Abhinav Samaga is an undergraduate student in Vivesvaraya Technical University (VTU),pursuing a B.E. in computer science and engineering at Rashtreeya Vidyalaya College of Engineering (RVCE) Bangalore, India. His areas of studies include artificial intelligence and machine learning, computer vision, embedded software, and micro-controllers. He can be contacted at email:abhinavsamaga.cs20@rvce.edu.in.

Allen Joel Lobo is an undergraduate student in Vivesvaraya Technical University (VTU),pursuing a B.E. in computer science and engineering at Rashtreeya Vidyalaya College of Engineering (RVCE) Bangalore, India. His areas of studies include artificial intelligence and machine learning, computer vision, computer design and algorithms. He can be contacted at email: allenjoel-lobo.cs20@rvce.edu.in.

Azra Nasreen has been teaching in the Department of Computer Science and Engineering in RVCE, since 2007. She is currently an associate professor. She has authored or coauthored more than 30 referred journal and conference papers. Her research interests include video analytics and high-performance computing. She can be contacted at email: azranasreen@rvce.edu.in.

Ramakanth Kumar Pattar is a professor and the Head of the Department of Computer Science and Engineering at RV College of Engineering, Bangalore. He has been teaching for the past 29 years with over 14 years experience in research and development. He has authored or coauthored more than 120 journal and conference papers. His research interests include digital image processing, pattern recognition and natural language processing. He can be contacted at email: ramakanthkp@rvce.edu.in.

Neeta Trivedi is the CEO and Founder of Inferigence Quotient, a Deep Technology Startup that has been awarded prestigious grants from the Ministry of Defense as well as from Govtof Karnataka. She is a former Sc 'G', DRDO, a senior member of IEEE. She has over 30 years of experience in avionics for manned and unmanned aerial systems, UAV payload data processing. She also has experience in systems and software engineering, information fusion, computer vision, and artificial intelligent. She can be contacted at email: neeta@inferq.com.

Peehu Raj is director of research development at Inferigence Quotient. He has expertise in signal and image processing, computer vision, deep learning, embedded computing. He can be contacted at email: peehu@inferq.com.

Koratagere Sreelakshmi is a professor and Head of Department of the Telecommunication Engineering at RV College of Engineering, Bangalore. She has received her M.E. and Ph.D. and hasbeen teaching for the past 25 years. She has authored or coauthored over 20 journal and conference papers. Her research areas include microwave communication and nano electric devices. She can be contacted at email: sreelakshmik@rvce.edu.in.