Journal of Trends and Challenges in Artificial Intelligence

ASPUR PUBLISHING

Vol. 02, No. 01 (2025) 19-26, DOI: 10.61552/JAI.2025.01.003 - http://jai.aspur.rs

ENHANCING LICENSE PLATE RECOGNITION USING YOLO-NAS, YOLOV8, AND SORT ALGORITHMS

Charvi Khanna [1]
Bhavini Bish
Dhruv Kamshetty
Harshit Bhardwaj
Monica Bhutani

Received 29.06.2024.
Revised 15.07.2024.
Accepted 25.08.2024.

Keywords:

License Plate Recognition (LPR) Technology, Computer Vision, YOLO (You Only Look Once) Algorithm, Deep Learning Neural Network Models, Convolutional Neural Networks (CNN), YOLO-NAS (You Only Look Once - Neural Architecture Search), Quantization Support, Deformable Convolution, Multi-scale Detection Layers, YOLOv5, Recurrent Neural Networks (RNNs), Model Optimization, Character Segmentation.

Original research

A B S T R A C T

Targeting the challenge posed by the traditional license plate recognition approach's deficiencies in precision and speed, a novel end-to-end deep learning model has been introduced. This model employs YOLO-NAS for the accurate detection and recognition of license plates in real-world scenarios. Employing the YOLO-NAS model, our approach to license plate identification involves comprehensive training on diverse datasets, spanning small, medium, and large scales to achieve optimal accuracy. YOLO-NAS introduces an innovative quantization-friendly basic block, mitigating a key limitation in earlier YOLO models. Performance is further heightened through the incorporation of advanced training methodologies and post-training quantization techniques. In conjunction, YOLOv8 serves to categorize vehicles into specific types, such as cars or bikes. The SORT algorithm assigns distinct identity numbers to vehicles, facilitating seamless linkage with their corresponding detected license plates. This associational data is systematically stored in a CSV file for reference. For visualization, EasyOCR is deployed to recognize alphanumeric characters on license plates. This recognition output is visually represented as a box above the identified vehicles. Leveraging YOLO-NAS for license plate detection not only ensures superior accuracy but also optimizes performance through quantization support and strategic accuracy-latency trade-offs, contributing to a more refined and efficient recognition system. The accuracy that we obtained for our YOLO-NAS (small) model was 90.2%. Using YOLO-NAS for license plate detection we are able to develop a model which combines high speed with accuracy.

1. INTRODUCTION

In the realm of scientific research, the automated number plate recognition (ANPR) system holds paramount importance for its multifaceted applications. ANPR's role in enhancing traffic management, bolstering law enforcement, and fortifying security infrastructure is pivotal. Automation of this system streamlines data processing, ensuring real-time analysis and response. Research efforts focus on refining accuracy, optimizing algorithms, and exploring integration with intelligent transportation systems. Such advancements contribute significantly to crime prevention, efficient surveillance, and overall societal safety. The scholarly exploration of ANPR automation enriches the field, fostering innovations that transcend traditional surveillance

[1] Corresponding author: Charvi Khanna
Email: khanna.charvi@gmail.com

paradigms and bolster the foundations of a secure and seamlessly connected urban environment.

Despite the widespread application of license plate recognition (LPR) technology, its predominant use in fixed settings limits its precision and robustness for complex and real-time scenarios. The evolving landscape of computer hardware has propelled deep learning neural network models as optimal tools for intricate computer vision challenges (Arshaghi et al., 2021; Bezsonov et al., 2021; Mohi Alden et al., 2022). Among these, Convolutional Neural Networks (CNN), particularly the YOLO (You only look once) algorithm introduced by (Redmon et al., 2016), have gained prominence for their efficiency in target detection and recognition tasks.

The YOLO approach ingeniously integrates the candidate area and target recognition phases into a unified forward operation, leading to a substantial reduction in image processing time and a notable improvement in the overall efficiency of the model. Numerous advanced Automatic License Plate Recognition (ALPR) systems have been successfully created by leveraging the cutting-edge YOLO object detector (Laroca et al., 2018).

On one front, collaborative effort involve employing YOLOv3 to extract and categorize underwater objects, coupled with a deep learning approach incorporating Long Short-Term Memory (LSTM) to ascertain their precise locations (Kshirsagar et al., 2023). Additionally, a novel feature fusion module is redesigned to heighten the accuracy of detecting small objects, incorporating multiple scale detection layers.

The iteration, YOLOv5, stands out for its precision, speed, and generation of lightweight detection files, facilitating seamless deployment on embedded devices. Innovative applications, such as S-DAYOLO for autonomous driving assistance, demonstrate YOLOv5's adaptability (Murthy et al. 2022). Concurrently, recurrent neural networks (RNNs) have emerged as pivotal in end-to-end license plate recognition, eliminating the need for character segmentation.

To enhance the efficiency of identifying vehicles in CCTV footage swiftly, a deep learning tool called the YOLO v8 algorithm is employed for real-time object recognition within the system (Talaat et al., 2023). Furthermore, TensorFlow, an open-source machine learning platform, is utilized for recognizing vehicle number plates and conducting image processing.

In the realm of model optimization, attention mechanisms, like Squeeze and Excitation (SE), are incredibly important in enhancing the performance of neural networks by establishing connections between different channels. While SE is commonly used, it fails to adequately consider the distinct features related to specific locations, which are crucial for identifying target structures in tasks involving detection and recognition (Hu et al., 2018). The ongoing efforts to enhance these mechanisms highlight their significance in improving the precision and usability of Licence Plate Recognition (LPR) models.

The collaborative research efforts have led to the development of a remarkable innovation by Deci AI, called YOLO-NAS (*You Only Look Once - Neural Architecture Search*). This revolutionary creation utilizes advanced technology in Neural Architecture Search, which was meticulously crafted to overcome the limitations of previous YOLO models. YOLO-NAS represents a substantial progress in the field of object detection, showcasing notable enhancements in quantization support and skilfully achieving a delicate equilibrium between speed and precision.

YOLO-NAS's salient features include the incorporation of blocks that are cognizant of quantization as well as selective quantization for maximum efficiency. Interestingly, the model shows very little decrease in precision when it is transformed into its INT8 quantized iteration, which is a significant improvement over other model. Together, these crucial developments result in an improved architectural design, giving YOLO-NAS state-of-the-art object detection capabilities.

This paper makes significant contributions in the following key areas:

- We use YOLO-NAS model to identify the license plate. We are able to obtain the model with the best accuracy by training YOLO-NAS for small, medium and large datasets. YOLO-NAS introduces a new basic block that is friendly to quantization, addressing one of the significant limitations of previous YOLO models. It also leverages advanced training schemes and post-training quantization to enhance performance.
- YOLOv8 is used to categorize vehicles as a specific vehicle type like car, bike etc.
- SORT algorithm is used to give specific identity numbers to vehicles in order to link the vehicle with the respective detected license plate and this data is stored in a csv file.
- For visualisation EasyOCR is used to recognise the alpha-numeric characters on the license plate. This output is displayed as a box above the respective detected vehicles.
- Using YOLO-NAS for license plate detection provides greater accuracy through quantization support and accuracy-latency trade-offs.
- In order to obtain the best possible results, we trained our dataset on three different models of YOLO-NAS namely small, medium and large. The model with the best accuracy was used for detection of the license plate.

2. LITERATURE REVIEW

Conventional methods for license plate localization, which rely on predefined data, are commonly classified into categories such as color texture analysis, shape regression, and edge detection in the process of locating license plates.

To construct the GrabCut method for automatic license plate localization, (Salau et al., 2021) utilized geometric information regarding the aspect ratio of the license plate as a threshold for foreground extraction. However, this approach has limitations due to the diversity of license plate aspect ratios in different locations. Traditional localization methods, which depend on manually designed feature extraction, are unsuitable given the diverse range of images. Consequently, conventional techniques for license plate detection are both imprecise and inefficient.

Target identification techniques have come a long way in the last several years thanks to deep learning breakthroughs. These algorithms can be divided into two major categories. One method is having the algorithm start the process by creating a candidate region, then classifying it and then fine-tuning its exact location. For example, (Bulan et al., 2017) presented a unique approach for Automatic License Plate Recognition (ALPR) that does not require segmentation or annotation thanks to new methodologies. Moreover, automated failure identification and improved plate location are integrated into this method.

Another approach (Saini & Saini, 2017; Saini & Saini, 2017a; Yu et al. 2015) developed a powerful technique based on wavelet transform and EMD (Empirical Mode Decomposition) analysis to solve real-world problems such as illumination fluctuations, complex backgrounds, and viewing angles when identifying license plates in photos. Variety: the method consists of projecting the obtained details onto the vehicle image, performing a wavelet transform, and creating crests representing the license plate. EMD analysis is used to identify the desired peaks in nonlinear and nonstationary projection data sets. The reconstructed projection data and the Hilbert transform of the intrinsic mode function components are then used to determine the position of the license plate (Saini & Saini, 2017).

A recently employed approach is the end-to-end detection method, which directly acquires the target's location coordinates and class probability. (Li et al., 2018) introduced a unified deep neural network that performs simultaneous localization of license plates and recognition of letters in a single forward pass. This not only circumvents the accumulation of intermediate errors but also significantly accelerates the processing speed.

However, challenges arise in scenarios where the system's accuracy is affected by image blurring due to motion blur, low resolution, low luminosity, low contrast, and noise. To address these concerns, (Duan et al., 2019) proposed an end-to-end fast solution for number-plate recognition. The widely used Single Shot MultiBox Detector algorithm was applied for the automatic detection of license plate locations. Additionally, for recognizing consecutive characters within the detected license plate area, an end-to-end Convolutional Neural Network classification model was devised.

(Al Batat et al., 2022) incorporated YOLO v2 in the initial phase of the pipeline, and the subsequent stages were built upon the advanced YOLO v4 detector. Various data augmentation and generation techniques were employed to achieve license plate recognition accuracy comparable to the currently proposed methods. The final phase of an Automatic Number Plate Recognition System (ANPRs) involves Optical Character Recognition (OCR), where the characters on the number plate image are transformed into encoded texts. (Zhai et al., 2012) introduced an OCR algorithm for ANPR applications based on Artificial Neural Network (ANN). The performance of the proposed algorithm was evaluated using a database comprising 3700 binary character images from the UK.

After the license plate characters have been fully segmented using the honeybee algorithm (Pereira et al. 2017; Shatnawi et al., 2018) employs a support vector machine (SVM) to identify the characters. The approach has a decent effect on recognizing license plates, but its recognition efficiency is low, according to experimental results.

A comprehensive model incorporating GRU (Gated Recurrent Unit) and CTC (Connectionist Temporal Classification) was proposed by (Shi & Zhao, 2023) utilizing YOLO-V5. Character segmentation-free license plate identification is made possible by this creative method that incorporates GRU and CTC into the recognition network. Specifically, this technique can speed up convergence, drastically cut down on training time, and enhance the detection network's accuracy overall.

Conventional license plate recognition techniques usually depend on discrete modules for both location and recognition, which means that intricate algorithms must be created to handle each difficulty separately. By comparison, our method effectively unifies the location and recognition operations by utilizing a deep learning-based neural network model. This study presents a new end-to-end approach for both character recognition using EasyOCR and license plate detection using YOLO-NAS. Our approach greatly improves recognition accuracy and efficiency by utilizing deep learning capabilities. The workflow is made simpler by the integrated design, which results in a more unified and effective solution for license plate recognition and detection.

3. METHODOLOGY

Although these approaches are pretty accurate, fast and rely on state-of-art computer vision and machine-learning algorithms, they often need improvements in many aspects to obtain the desired accuracy, time efficiency and customization when used for real-world scenarios. With that motivation in mind, we have developed an Automatic License Plate identification (ALPR) system merging the state-of-art computer vision and machine-learning techniques into a synergistic human/machine combination workflow. It includes the OpenCV-based basic image preprocessing, which nicely promotes the robustness of the YOLO-NAS license plate detection with state-of-the-art performance, the SORT based object-tracking totally decoupled from object detection, and the character recognition at an absolute Olympic level by leveraging the power of conditional random fields or other competitive algorithms from the EasyOCR package. This algorithm can work on different license plate formats.

The network has a bunch of neurons corresponding to each of the 28*28 pixels of the input image has 784 neurons in total each one of these holds a number that represents the grayscale value of the corresponding pixel ranging from 0 for black pixels up to 1 for white pixels. This number inside the neuron is called its activation (Figure 1).

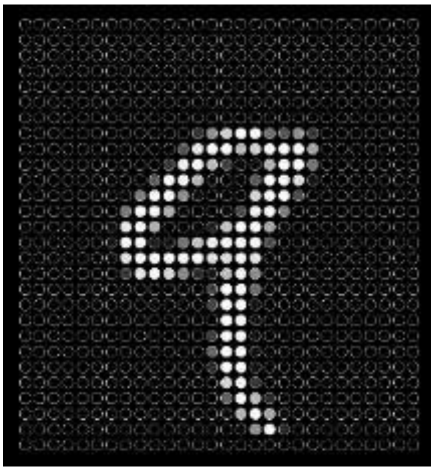


Figure 1. OCR based on weighted average

A weight assigned to each one of the connections between the input pixel neuron and the neurons from the first layer. These weights, quantified as numerical values, modulate the importance of information flowing through the network. Then all those activations from the first layer are taken and their weighted sum is computed.

$$a_0^{(1)} = w_{0,0}a_0^{(0)} + w_{0,1}a_1^{(0)} + \cdots + w_{0,n}a_n^{(0)} \quad \text{Eq. (1)}$$

Weighted sum can be any number, but for this network what we want is for activations to be some value between 0 & 1 so a common thing to do is to pump this weighted sum into some function that squishes the real number line into the range between 0 & 1. Therefore we use the sigmoid function also known as a logistic curve.

Incorporating biases further refines this process. Biases introduce an additional term to the weighted sum before applying a sigmoid function. So the weights tell you what pixel pattern this neuron in the second layer is picking up on and the bias tells you how high the weighted sum needs to be before the neuron starts getting meaningfully active.

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad \text{Eq. (2)}$$

An interpretation of neuron activations that is binary or probabilistic is made easier by this transformation. In essence, biases set the activation thresholds and weights define the patterns or features that a neuron is sensitive to, allowing for subtle sensitivity to changing input circumstances.

$$a_0^{(1)} = \sigma(w_{0,0}a_0^{(0)} + w_{0,1}a_1^{(0)} + \cdots + w_{0,n}a_n^{(0)} + b_0) \quad \text{Eq. (3)}$$

In summary, every neuron in the OCR-focused deep learning architecture is closely related to the layer that came before it, with each neuron being controlled by unique weights and biases. These weights and biases work together to create a complex network of information processing that makes it possible to recognize the characters effectively. The full transition from one layer to the next can be shown as matrix multiplication as shown below:

$$a^1 = \sigma\left(\begin{bmatrix} w_{0,0} & \cdots & w_{0,n} \\ \vdots & \ddots & \vdots \\ w_{k,0} & \cdots & w_{k,n} \end{bmatrix}\begin{bmatrix} a_0^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix}\right) \quad \text{Eq. (4)}$$

The equation 4 of transition from one layer to another can be tightly written as:

$$a^1 = \sigma(Wa^{(0)} + b) \quad \text{Eq. (5)}$$

The main steps involved are data collection, data preprocessing, detection, tracking and character recognition. Criteria for performance evaluation and experimental analysis for establishing results are shown online to prove the effectiveness of the system. Applications include the development of intelligent traffic control systems, surveillance, and law enforcement, where the whole system is designed as a comprehensive ‘end-to-end’ ALPR solution.

3.1 Dataset

This task is based on the Automatic License Plate Detection dataset, which has been provided by Roboflow, a platform that specialises in annotating, preprocessing and managing data for computer vision applications. This dataset have totally 712 photos, which including lots of occurrences in real world such as different scenes and situation, which are split to several categories, including few with license plate and without license plate.

The most interesting thing about the data set is that the box surrounding the plate has been demarcated for training the detection model. But there’s also huge variability within the fonts, colors and styles of the licensed plates in that file, and that kind of variability can be useful to make the model sturdier, and to accept the conditions we want to apply to the real world.

3.2 YOLO-NAS for Object Detection

YOLO takes an input image and divides it into a grid. Each grid cell is responsible for predicting bounding boxes and class probabilities. Prior to training, anchor boxes were defined based on the shapes and sizes of objects in the dataset. These anchor boxes helped the model make accurate predictions for License Plate bounding boxes.

3.3 SORT Algorithm

The SORT algorithm is an acronym for "Simple Online and Realtime Tracking" which is a tracking algorithm designed to track multiple objects in video sequences. It focuses on efficient association of object detections across frames in real-time scenarios. SORT is particularly useful for applications such as video surveillance, autonomous vehicles, and human-computer interaction. In the first step of SORT, object detections are obtained in each frame of a video sequence. These detections were from the pre-trained object detection model (YOLO-NAS) and provided bounding boxes around the License Plate.

SORT uses a simple and effective approach for data association. It employs the Hungarian algorithm (or linear assignment problem solver) to associate detections in the current frame with the existing tracked objects from the previous frame. The Hungarian algorithm optimally assigns detections to existing tracks based on a cost matrix that considers both spatial and appearance information. Each object being tracked is represented by a state vector that typically includes the object's position (x, y coordinates), size, and a unique identifier. The state vector is updated based on the association between the detections and the existing tracks.

SORT utilizes Kalman filter to estimate state of each object being tracked. The Kalman filter is a recursive algorithm that predicts the next state of an object based on its previous state and the system dynamics. It is quite handy in cases involving noisy measurements, since it gives a smoothed estimate for the state of the object.

3.4 EasyOCR for Optical Character Recognition

OCR is a technology used to convert various types of documents, such as scanned paper documents, PDFs, or images taken by a digital camera, into searchable and editable data. It was specifically applied on License Plate Detection to identify the characters on license plates. Automatic License Plate Detection pipeline integrated it as one of its components for license plates text recognition within the recognized registration regions. Consequently, this allowed for the extraction of license plate number from images.

3.5 OpenCV for Image Processing

The function provided by OpenCV for adjusting image contrast, brightness, and sharpness helps to make vehicle license plates visible even in poor lighting.

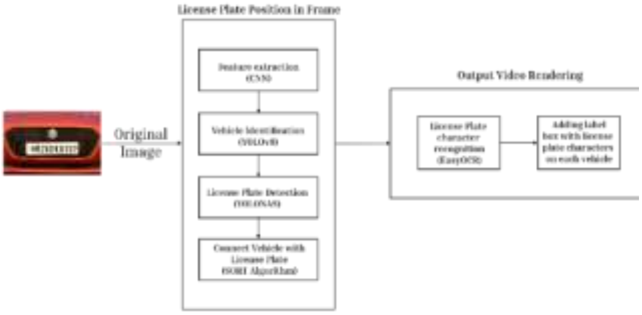


Figure 2. LPR Architecture

Gaussian blur or median blur are examples of filters and techniques available in OpenCV which can be used to reduce noise in an image. Reducing noise improves the accuracy of subsequent image processing steps. Developers can segment images using color with OpenCV. An example of it is when Automatic License Plate Detection was used for isolating "Regions of Interest" that are more likely to have license plates (Figure 2). Usually, number plates have distinct colors compared to their surroundings.

4. RESULTS

If we run the code successfully then it will only detect the vehicle's license plate from both given input image as well as from the real time feed, as shown in below images:

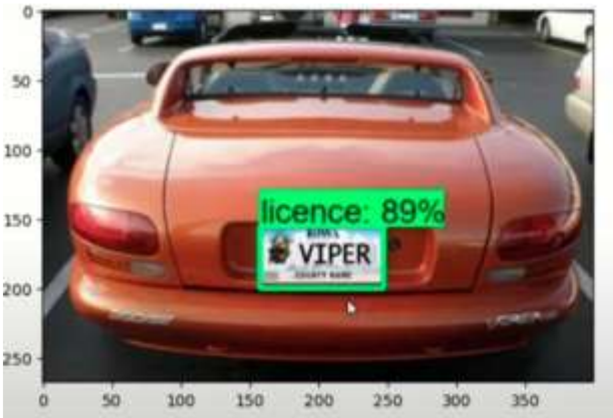
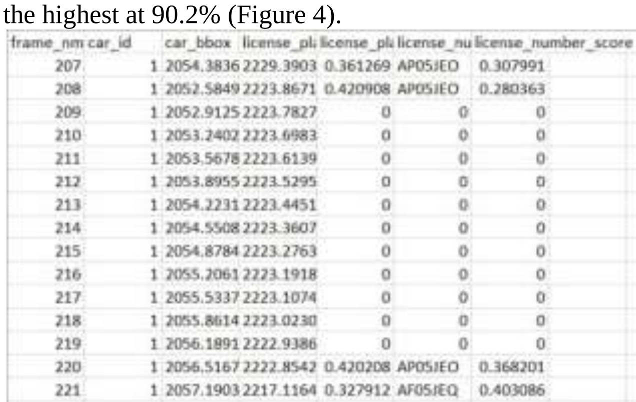


Figure 3. License Plate detected from car

The above output depicts the license plate recognition model which was implemented using YOLO-NAS. Here we can observe real time license plate detection where we feed a real time video in our model and receive identified license plate on a vehicle (Figure 3).

Efficiency of different models of YOLO were studied as a part of comparative study and the above result was obtained depicting efficiency of different models using the parameter "mean average precision" (mAP) at an IoU of 0.50 and efficiency of YOLO-NAS was found to be the highest at 90.2% (Figure 4).



frame_nm	car_id	car_bbox	license_pl	license_pl	license_nu	license_number_score
207	1	2054.3836	2229.3903	0.361269	AP05JEO	0.307991
208	1	2052.5849	2223.8671	0.420908	AP05JEO	0.280363
209	1	2052.9125	2223.7827	0	0	0
210	1	2053.2402	2223.6983	0	0	0
211	1	2053.5678	2223.6139	0	0	0
212	1	2053.8955	2223.5295	0	0	0
213	1	2054.2231	2223.4451	0	0	0
214	1	2054.5508	2223.3607	0	0	0
215	1	2054.8784	2223.2763	0	0	0
216	1	2055.2061	2223.1918	0	0	0
217	1	2055.5337	2223.1074	0	0	0
218	1	2055.8614	2223.0230	0	0	0
219	1	2056.1891	2222.9386	0	0	0
220	1	2056.5167	2222.8542	0.420208	AP05JEO	0.368201
221	1	2057.1903	2217.1164	0.327912	AF05JEQ	0.403086

Figure 4. License Plate detected from car output as CSV file

This result is obtained by downsizing the real time video into static frames post which the concerned vehicle is recognized then our license plate detector is applied on the license plate location, further the details of the obtained license plates and vehicle are stored and mirrored with each other using SORT algorithm.

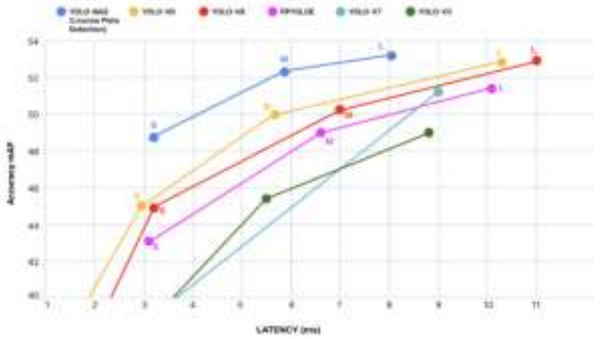


Figure 5. Latency vs. Accuracy graph for various YOLO models

The below Latency vs Accuracy graph was plotted for different models for YOLO on small, medium and large models where small, medium and large dictates the depth and breadth of the network of the model (Figure 5). The average of "mean average precision" (mAP) of these models in the IoU range of 0.50 – 0.95 is considered.

5. CONCLUSION

This paper addresses the limitations of traditional license plate recognition methods, emphasizing their shortcomings in accuracy and speed. In response, we propose an innovative end-to-end deep learning model designed for license plate localization and recognition in natural settings. Our experimental focus involves enhancing the YOLO-NAS target detection algorithm through the integration of a quantization-friendly basic block, effectively overcoming a significant constraint observed in prior YOLO models. To further elevate performance, we incorporate advanced training techniques and post-training quantization methods. Notably, our YOLO-NAS (small) model achieves an accuracy of 90.2%. Leveraging YOLO-NAS for license plate detection enables the development of a model that seamlessly integrates high speed and precision.

References:

Al-Batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., & Kapetanios, E. (2022). An end-to-end automated license plate recognition system using YOLO based vehicle and license plate detection with vehicle classification. *Sensors*, *22*(23), 9477. DOI:10.3390/s22239477.

Arshaghi, A., Ashourin, M., & Ghabeli, L. (2021). Detection and Classification of Potato Diseases Potato Using a New Convolution Neural Network Architecture. *Traitement du Signal*, *38*(6). DOI:10.18280/ts.380622.

Bezsonov, O., Lebediev, O., Lebediev, V., Megel, Y., Prochukhan, D., & Rudenko, O. (2021). Breed recognition and estimation of live weight of cattle based on methods of machine learning and computer vision. *Eastern-European Journal of Enterprise Technologies*, *6*(9), 114.

Bulan, O., Kozitsky, V., Ramesh, P., & Shreve, M. (2017). Segmentation-and annotation-free license plate recognition with deep localization and failure identification. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2351-2363. DOI:10.1109/TITS.2016.2639020.

Duan, N., Cui, J., Liu, L., & Zheng, L. (2019). An end to end recognition for license plates using convolutional neural networks. *IEEE Intelligent Transportation Systems Magazine*, *13*(2), 177-188.DOI:10.1109/MITS.2019.2898967.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).

Kshirsagar, A. P., & Azath, H. (2023). YOLOv3-based human detection and heuristically modified-LSTM for abnormal human activities detection in ATM machine. *Journal of Visual Communication and Image Representation*, *95*, 103901. DOI:10.1016/j.jvcir.2023.103901.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018, July). A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 international joint conference on neural networks (IJCNN)* (pp. 1-10). IEEE.

Li, H., Wang, P., & Shen, C. (2018). Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, *20*(3), 1126-1136. DOI:10.1109/TITS.2018.2847291.

Mohi-Alden, K., Omid, M., Soltani Firouz, M., & Nasiri, A. (2022). Design and evaluation of an intelligent sorting system for bell pepper using deep convolutional neural networks. *Journal of food science*, *87*(1), 289-301. DOI:10.1111/1750-3841.15995.

Murthy, J. S., Siddesh, G. M., Lai, W. C., Parameshachari, B. D., Patil, S. N., & Hemalatha, K. L. (2022). Objectdetect: A real-time object detection framework for advanced driver assistant systems using yolov5. *Wireless Communications and Mobile Computing*, *2022*. DOI:10.1155/2022/9444360.

Pereira, N. A., Rao, P., Kallianpur, A. K., & Srinivasa, K. G. (2017, December). Discrete artificial bee colony algorithm based optical character recognition. In *2017 14th IEEE India Council International Conference (INDICON)* (pp. 1-6). IEEE. DOI:10.1109/INDICON.2017.8487826.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). DOI:10.1109/CVPR.2016.91.

Saini, M. K., & Saini, S. (2017). Multiwavelet transform based license plate detection. *Journal of Visual Communication and Image Representation*, *44*, 128-138.DOI:10.1016/j.jvcir.2017.01.003.

Saini, M. K., & Saini, S. (2017a, March). Analysis of license plate using MWT. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-6). IEEE. DOI:10.1109/ICIIECS.2017.8276069.

Salau, A. O., Yesufu, T. K., & Ogundare, B. S. (2021). Vehicle plate number localization using a modified GrabCut algorithm. *Journal of King Saud University-Computer and Information Sciences*, *33*(4), 399-407. DOI:10.1016/j.jksuci.2019.01.011.

Shatnawi, N. M. (2018). Bees algorithm and support vector machine for Malaysian license plate recognition. *International Journal of Business Information Systems*, *28*(3), 284-298.

Shi, H., & Zhao, D. (2023). License plate recognition system based on improved YOLOv5 and GRU. *IEEE Access*, *11*, 10429-10439. DOI:10.1109/ACCESS.2023.3240439.

Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Computing and Applications*, *35*(28), 20939-20954. DOI:10.1007/s00521-023-08809-1.

Yu, S., Li, B., Zhang, Q., Liu, C., & Meng, M. Q. H. (2015). A novel license plate location method based on wavelet transform and EMD analysis. *Pattern Recognition*, *48*(1), 114-125. DOI:10.1016/j.patcog.2014.07.027.

Zhai, X., Bensaali, F., & Sotudeh, R. (2012, July). OCR-based neural network for ANPR. In *2012 IEEE International Conference on Imaging Systems and Techniques Proceedings* (pp. 393-397). IEEE. DOI:10.1109/IST.2012.6295581.

Charvi Khanna
ECE Department
Bharati Vidyapeeth's College of Engineering, New Delhi, India (Guru Gobind Singh Indraprastha University)
khanna.charvi@gmail.com
ORCID 0009-0009-9960-8406

Bhavini Bisht
ECE Department
Bharati Vidyapeeth's College of Engineering, New Delhi, India (Guru Gobind Singh Indraprastha University)
bhavini20.20@gmail.com
ORCID 0009-0004-0221-6611

Dhruv Kamshetty
ECE Department
Bharati Vidyapeeth's College of Engineering New Delhi, India (Guru Gobind Singh Indraprastha University)
dhruvkamshetty123@gmail.com
ORCID 0009-0009-9029-1916

Harshit Bhardwaj
ECE Department
Bharati Vidyapeeth's College of Engineering, New Delhi, India (Guru Gobind Singh Indraprastha University)
harshitbhardwaj001@gmail.com

Monica Bhutani
ECE Department
Bharati Vidyapeeth's College of Engineering, New Delhi, India
Monica.bhutani@bharatividyapeeth.edu
ORCID 0000-0003-1924-3056