Inteligencia Artificial, 27(73) (2024), 129-141
doi: 10.4114/intartif.vol27iss73pp129-141

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

# A Robust Approach for Licence Plate Detection Using Deep Learning

Shefali Arora*[1], Ruchi Mittal[2], Dhruv Arora[3], Avinash K Shrivastava[4]

[1]Department of Computer Science and Engineering
National Institute of Technology Jalandhar, India
chouhansa@nitj.ac.in

[2]DataTech Department,Iconic Data, Tokyo, Japan
ruchi.mittal138@gmail.com

[3]School of Industrial Engineering
Purdue University, West Lafayette, Indiana, United States of America
aroradhruv10@gmail.com

[4]Department of Operations Management Quantitative Techniques
International Management Institute Kolkata, India
a.shrivastava@imi-k.edu.in

**A**bstract Intelligent transport systems must be developed due to the rising use of vehicles, particularly cars. In the field of computer vision, the identification of a vehicle's license plate (LP) has been crucial. Various methods and algorithms have been used for the detection process. It becomes challenging to find similar images, nevertheless, because the features of these plates change depending on the characters' color, font, and language. The research proposes a robust deep-learning framework based on feature extraction using convolutional neural networks and localization using an improved approach integrating bilateral filtering and Canny edge detection. Further, a CNN architecture is used to extract features from images and classify the presence of license plates in unseen vehicles. If present, the stage is followed by recognition of numbers written on the plates. An extensive experimental investigation is performed on Stanford Cars, Car Licence Plate Detection dataset, and Indian Licence Plates Dataset. The attained simulation outcome ensures a superior performance over existing techniques in a significant way.

**Keywords**: Vehicle plates, OpenCV,Deep learning, CNN, Recognition

## 1 Introduction

The surge in road vehicle traffic has dramatically boosted the demand for traffic and management systems. In this situation, it is nearly impossible to follow cars traveling swiftly down the road manually. Time and labor resources are wasted in this case. Even if done manually, this will result in significant problems and errors. There is now research being conducted with the use of machine learning algorithms to track vehicles and license plates[1]. An automated system must be created to track cars by tracing their license plates. Parking management systems, toll payment processing systems, and other systems that monitor vehicles rely on Automatic Number Plate Recognition (ANPR). Security staff can save a significant amount of time by automating the process. Computer vision technology has evolved significantly in many real-world issues in recent decades. Plate numbers' height, breadth, contour, and so on were previously found using template matching techniques. Numerous deeper models developed from massive data are now frequently employed in plate recognition.[2]. Even though the ANPR uses

**ISSN: 1137-3601 (print), 1988-3064 (on-line)**

multiple algorithms, it still lacks real-time accuracy. Applying profound learning strategies may help to get past it. Deep learning is a massive field of artificial intelligence (AI) that uses neural networks to learn from a lot of data[9]. Today, deep learning is used in almost all real-time applications as it gathers high-level features from raw input to extract features. Compared to other algorithms, it exhibits good precision and fewer errors. The CNN network is used to identify vehicles and license plates in various ways. The system's primary goal is to develop a deep learning model that can use a surveillance camera to scan a number plate on a roadside. Various researchers are working on hybrid deep learning frameworks to automate the process of license plate detection[31]. In this study, we use deep CNN architecture to extract features from moving vehicle license plates. We propose a framework that uses an improved localization process involving bilateral filters and Canny Edge detection. To apply the findings to previously unknown car plate photos, feature extraction, and classification are performed after the annotation of license plates using the proposed approach. Our main contribution is as follows:

- The study proposes an improved segmentation technique that combines bilateral filters with Canny Edge detectors to locate license plates on a car. The approach improves the detection rate as well as the Performance of feature extraction in the following stage.
- Furthermore, images of localized license plates are labelled separately from images of unlocalized vehicles, allowing CNN to extract features and classify unseen cars based on the presence of license plates. This categorization is carried out using the Softmax classifier. As a result, the framework can organize unseen images for the presence or absence of license plates and detect license plates followed by character recognition via Optical Character Recognition (OCR).
- We also compare the results with approaches successfully utilized to identify license plates. The proposed solution has been discovered to provide higher classification accuracy on par with existing techniques.

# 2 Related Work

A lot of research has been done for license plate recognition in the past. In [4], authors use morphology to identify license plates in crowded images. There are three main parts to the proposed system. First, a morphology-based approach is suggested to extract significant contrast elements that serve as a road map for finding the appropriate license plates. The contrast characteristic can withstand changes in lighting and is invariant to several adjustments, including scaling, translation, and skewing. The authors of [5] provide an algorithm to extract the area of a car license plate from an image using grey-scale morphological techniques. This algorithm stands out from other solutions to the issue since the input photos are not subject to any particular limitations. In [6], the authors suggest a real-time multiple license plate (LP) detection method for dense traffic scenarios. While a straightforward yet successful color detection method is utilised to find yellow LP sections, the chromatic component of the YDbDr color space is proposed to detect the blue parts. The low-intensity pixel values are deleted as a pre-processing step to improve the LP regions. The Otsu method is then applied to obtain the binary image. To prevent missing plate position, especially in low-quality photos, the study's goal in [8] is to maximize the contrast of plate-like regions. To enhance the performance of the license plate detection problem, a new collection of features known as structured Histogram of Oriented Gradients (sHOG) is presented in [9]. The shallow ANN was trained using the sHOG features, which gives the candidate regions a level of confidence score and directs the GSA search to a less-than-ideal solution in the search space of a particular input picture. Results show that the proposed technique may archive an IOU detection rate of up to 98.74% when its Performance was assessed using private and public license plate datasets.
Several deep learning algorithms are being used to classify license plates of vehicles automatically. A survey conducted by authors in [1][2] gives an overview of various deep learning algorithms which can be utilized in this domain. In [10], an integrated deep neural network is proposed to find the license plate and identify characters in a single forward pass. The approach uses just one network to simultaneously accomplish both objectives, in contrast to earlier systems that treated license plate recognition and identification as separate issues that needed to be resolved individually. Processing is sped up, in addition, intermediate errors are removed. The video is converted into a picture to identify the car in each frame. Then, the indicated vehicle's license plate is read.
The work presented in [11] detects the front/rear view of the car via the upgraded ANPR pipeline that is being offered, and the number plate area is then localized using the YOLOv4 (You Only Look Once) object detection models. The final phase in the pipeline involves recognizing the number plate label using a deep learning architecture (e.g., AlexNet or R-CNNL3) after an algorithm recognizes the distinctive plate layout, which can be either a single or double row layout depending on the country. According to the results, the number plate localization model achieves a 99.71% mean Average Precision (mAP) score on a 0.50 threshold, and our trained YOLOv4 model for vehicle front/rear view identification earns a 98.42% mAP score. In [14], authors employ an end-to-end generic ALPR pipeline based on YOLO for the detection and recognition of license plates (LP), vehicles (VD), and objects (O) without the use of prior knowledge or additional inference stages. They evaluate the entire

ALPR pipeline, including the vehicle classifier for big trucks and emergency vehicles, from vehicle detection to the LP recognition stage. YOLO v2 is used in the pipeline's initial stage, and the state-of-the-art YOLO v4 detector in the latter stages, together with a variety of data augmentation and creation strategies, to achieve LP recognition accuracy on par with recently proposed methods. A survey was done in [19] on various ANPR (automatic number plate recognition) installation approaches. Nearly 78 reference documents were gathered by the researchers, and the findings were evaluated. The primary functions of the ANPR are the gathering of vehicle pictures, recognition of number plates, segmentation, and character recognition. Using a camera module, sensor, control device, GSM, and an integrated server, the Automatic Number Plate Recognition (ANPR) evaluation was carried out in [20]. By matching the previously saved vehicle information, it tried to stop illicit autos. A histogram adjustment was used to convert the camera images into grey and enhance them. The edge detection method of Sobel was used to locate the rims. The photos were then subjected to morphological processing. The characters were identified using machine learning.

In [20], a deep learning plate identification system was created. The dataset was created artificially by collecting pictures from the internet and adding noise and background. SUN and Standford's databases were utilized for background purposes. An object detection framework was utilized for number plate detection YOLO(You only look once). Convolutional Neural Network (CNN) was used for character recognition.

# 3 Methodology

The overall working principle of the proposed framework is shown in Figure 1. In the first stage, license plates are localized using an improved algorithm involving bilateral filters and Canny edge detection. This is followed by contour detection for the annotation of license plates. Further, labelling of localized and unlocalized images is done, and it serves as input to a CNN architecture in order to detect features and classify licence plates in unseen vehicles (i.e. whether they have license plates or not). In the third stage, character recognition is done for vehicles that contain a license plate so that that number can be recognized.

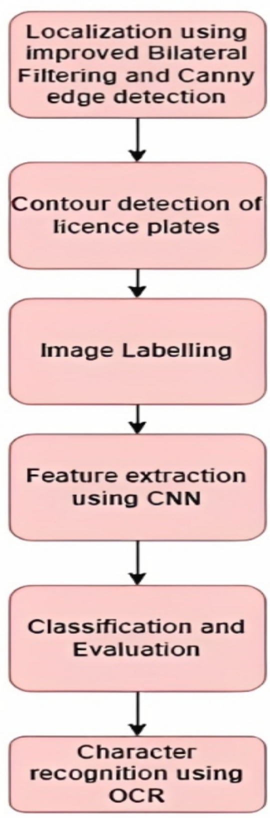


Figure 1: Proposed Methodology for LP detection

## 3.1 LP Localization

### 3.1.1 An Improved Bilateral Filtering Based Canny Edge Detection

In this stage, two local methodologies are combined to enhance the localization process of license plates. To detect edges, image noise must be removed. Gaussian blur can be formulated as follows:

$$GB[I]_p = \sum_{q \epsilon S} G_\sigma I_q \tag{1}$$

where $GB[I]_p$ is the result at pixel p, and the right-hand side shows the sum of all pixels q weighted by a Gaussian function.

In the case of bilateral filtering, noise reduction is done with the help of a normalization factor and range weight factor. This gives an improvement over Gaussian blurring commonly used in Canny edge detection. Bilateral filtering is a method for blurring pictures while keeping their edges sharp. Several characteristics of the bilateral filter contribute to its effectiveness:

- Its formulation is straightforward: a weighted average of each pixel's neighbours is used to replace each one. This feature is crucial since it makes it simple to construct, adjust to application-specific requirements, and develop an understanding of its behaviour.
- It may be utilized in a non-iterative way;
- It simply depends on two parameters defining the characteristics' size and contrasts to be preserved. Their impact does not compound across several repetitions, making the parameters simple to set.
- It can be computed at real-time speed, even on big pictures, and even in real-time if graphics hardware is available.

Let $\sigma s$ denote the kernel's spatial extent, and $\sigma r$ denote the edge of the amplitude. With the help of bilateral filtering, pixels with intensity values similar to central pixels are considered for blurring. The sharp changes in intensity are maintained in bilateral filtering.

$$GB[I]_p = 1/W_p \sum_{q \epsilon S} G_{\sigma s}(||p-q||) G_{\sigma r}(||I_p - I_q||) I_q \tag{2}$$

The value of $\sigma s$ in color space and $\sigma r$ in the coordinate space are chosen to optimize the mixing of pixels.
In bilateral filtering, the kernel coefficients of the filter are determined by the combined closeness and similarity function. It combines the functions of range filtering and domain filtering, thus enhancing the performance of the Canny edge detection algorithm compared to its usage with Gaussian filters. Let $f : R- > R\hat{a}$ be the original brightness function of an image which maps the coordinates of a pixel (x, y) to a value in light intensity. Let $a_0$ be the reference pixel. The coefficient assigned to intensity value f(a) for range filter r(a) is computed as:

$$r(a) = s(f(a), f(a_0) = e^{(f(a)-f(a_0)^2)/(2*\sigma_1^2)} \tag{3}$$

The coefficient assigned to intensity value f(a) at for domain filter g(a) is computed as:

$$g(a) = c(a, a_0) = e^{(a-a_0)^2)/(2*\sigma_2^2)} \tag{4}$$

Therefore, for the reference pixel 0 a , its new intensity value, denoted by 0 h a( ), is

$$h(a_0) = k^{-1} \sum_{i=0}^{n-1} (f(a_i) * r(a_i) * g(a_i) \tag{5}$$

where k is a normalization constant. The results from the filtering can be calculated as follows:

$$G = \sqrt{G_x^2 + G_y^2} \tag{6}$$

$$\theta = tan^{-1}(G_x/G_y) \tag{7}$$

The algorithm suppresses false edges using a technique called non-maximum suppression of advantages. This filters out unwanted pixels. Each pixel is compared with its neighbouring pixels in a positive and negative direction.

#### 3.1.2 Canny Edge detection and refining

With the Canny edge detection technology, the quantity of data that must be processed may be drastically reduced while still extracting meaningful structural information from various visual objects. It is frequently used in many computer vision systems. According to Canny, the prerequisites for applying edge detection to different vision systems are the same. Thus, a solution for edge detection that meets these needs can be used in multiple contexts. Edge detection is performed after the stage of bilateral filtering:

- Low error rate edge detection indicates that the detection should precisely capture all the edges visible in the picture.
- The operator's edge point detection should correctly locate the edge's center.
- If feasible, picture noise should not produce false edges, and an edge in the image should only be marked once.

To meet these conditions, Canny utilizes the calculus of variations to locate the function that best optimizes a given process. The Canny edge detection algorithm is one of the most precisely specified edge detection techniques created to date. It offers accurate and dependable detection. It became one of the most widely used edge detection algorithms because of its superior ability to satisfy the three edge detection requirements and ease of implementation [12,13]. The five phases that make up the Canny edge detection and refining method are shown in Figure 2:

After the edge detection step, all pixels have been marked as either edge points or non-edge points. Further, a threshold value is calculated to obtain a final edge map. The threshold is calculated based on a percentage of possible edge points, manually set as strong edge points by the user. Like the original Canny's detection algorithm, we first mark the pixels with gradient magnitudes more significant than the higher threshold as the edge pixels. Then, the process is performed on every neighbouring pixel value of the edge marked. All other magnitudes in between the thresholds are marked as weak edges.

#### 3.1.3 Contour Detection and Labelling

After Canny Edge detection, the process of contour detection is done to detect the boundaries of license plates. For this purpose, only the number of contours equal to 4 is considered for detection. All contours with sides not equal to 4 are rejected in the final image. Thus, localized license plates are obtained after this process. All the images with unlocalized license plates are labelled in one category, and those with localized license plates are labelled in the second category. Figure 3 shows the localization of a license plate after following the stages above:

#### 3.1.4 Character Segmentation

This stage uses Tesseract's OCR machine to detect the characters printed on a license plate.

#### 3.1.5 Character recognition from plates

The OCR and Tesseract library in Python is used to transform two-dimensional images of text into handwritten text. The OCR makes use of pre-processing and character segmentation to capture unique words from written text. This is depicted in Figure 4.

The retrieved data is finally saved in a spreadsheet.

### 3.2 Feature extraction using CNN

A Convolution Neural Network(CNN) is a specific instance of a neural network with several layers that are completely coupled. In terms of complexity and memory needs, using CNNs offers higher Performance, combining the weights of convolution layers during features extraction with fully connected layers utilized for classification processes. Each CNN neuron creates output by computing dot products from inputs to neuron weight. The following are shown:

$$z = g(w.x + b) \tag{8}$$

Where w is the neuron's weight, x is the entry, and b is the neuron's bias unit. g is a function of activation such as sigmoid, ReLu etc. In the convolution layer, backpropagating filter weights are utilized to convert with the input image matrix to acquire the features. Say m filters for layer l. At every position of the input, each filter recognizes a certain characteristic. The output of the $yi^l$ function map in a particular convolution layer is calculated as:

$$y_i^l = b_i^l + \sum_{j=1}^{m} f_{i,j}^l * y_j^{l-1} \tag{9}$$

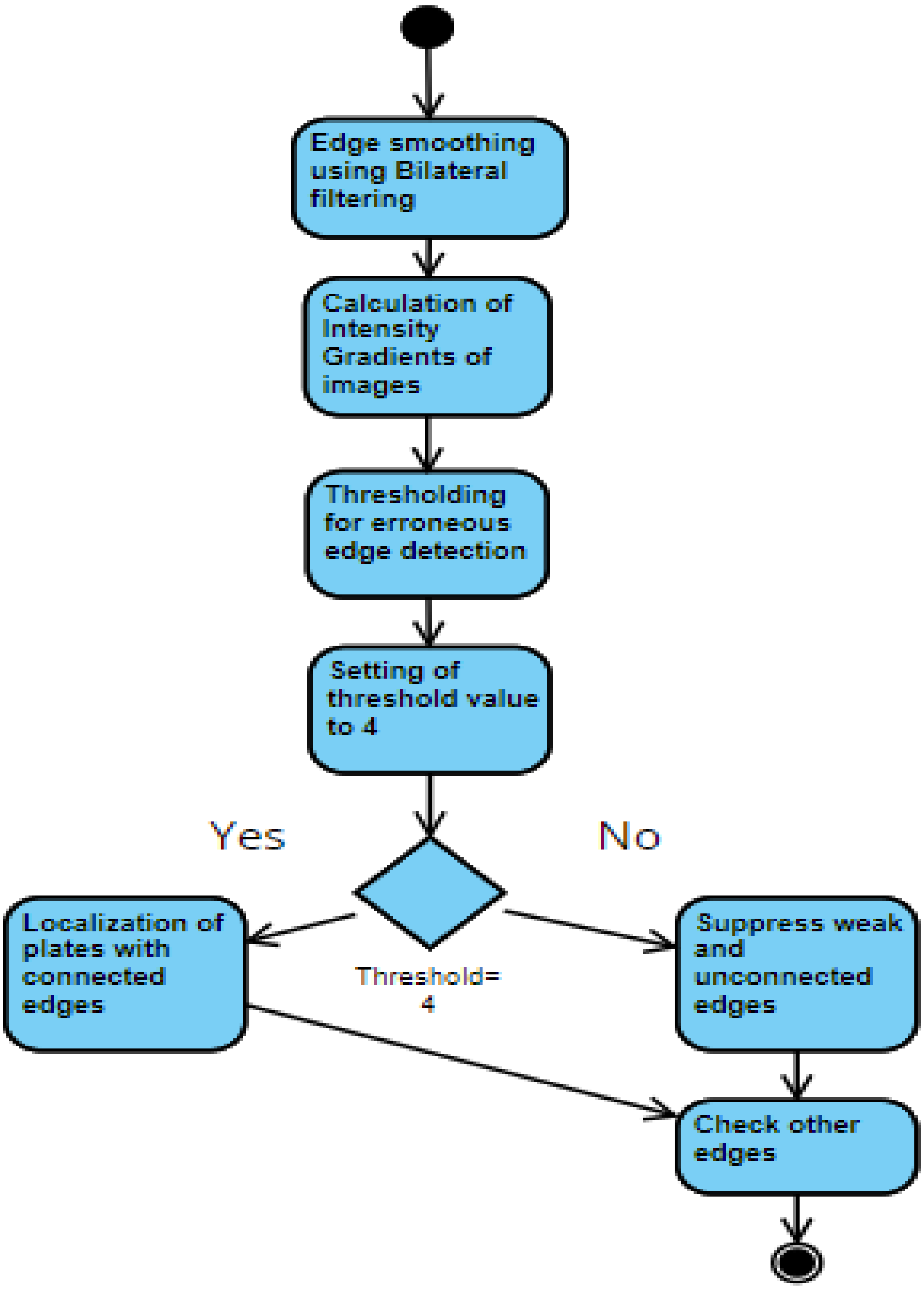


Figure 2: Proposed framework for localization of licence plates

The convolution operator depicts convolution, $yj/l-1$ and convolutional filter $fi, jl$, between the input of the preceding layer l-1. $bi^l$ is a bias matrix. The max pooling layer follows this layer. With a n * n window that is utilized for max pooling. The spatial size of each representation is gradually reduced. Thus, max pooling applies a function of u(n, n) to each patch for each patch represented by s*s on the images matrix and calculates the


Figure 3: Localization of licence plate

maximum in the vicinity. At every location of the window, the decrease is repeated.

$$y_j = max_{n*n}(y_i^{n*n} u(s*s)) \tag{10}$$

The next stage is to extract features from the labelled images. The presented network comprises overlapping, pooling, and completely linked layers. The Max pooling process is utilised to sample pictures. Table 1 shows the architecture of CNN used in the framework. In the proposed framework, the following layers have been used in the CNN architecture used for feature extraction:

- The architecture includes two convolutional layers, each consisting of 32 filters with a size of 3x3. ReLU activation functions are utilized to introduce non-linearity.
- After the convolutional layers, a max pooling layer with a 2x2 filter size is applied. This operation aids in reducing the dimensionality of the feature matrix obtained from the convolutional process.
- Following the max pooling layer, two more convolutional layers are introduced. These layers consist of 64 filters, each with a size of 3x3, and again employ ReLU activation functions. Additionally, max pooling with a 2x2 filter size is performed on the resultant matrix.
- This is succeeded by two additional convolutional layers, each employing 128 filters of size 3x3 and ReLU activation functions. Similar to previous layers, max pooling is applied with a 2x2 filter size to downscale the obtained matrix.
- The final stage involves convolutions on the output matrix, utilizing a layer comprising 256 filters of size 3x3.
- At the end, the output matrix is given as an input to a fully connected layer of 1024 neurons after passing through a Flatten layer.
- The final layer consists of 2 neurons that use a softmax layer to classify whether the input image has a license plate.

Table 1 shows the architecture of CNN used in the framework.

The next stage is a forward through a CNN network using the layers indicated. The network trains pre-processed labelled images based on the localization of license plates. The images received are classified using a softmax classifier, which predicts whether the image has a license plate. The layers of CNN are entirely linked. This is shown in Table 2.

The network handles a randomly tested mini-batch for every forward pass. Our framework is 100 epochs trained. The advantages of this training framework include the network's multi-frame training and the extraction of features effectively, i.e., better discriminatory functionality. The learning rate has been set to 0.01 for the selected number of epochs.

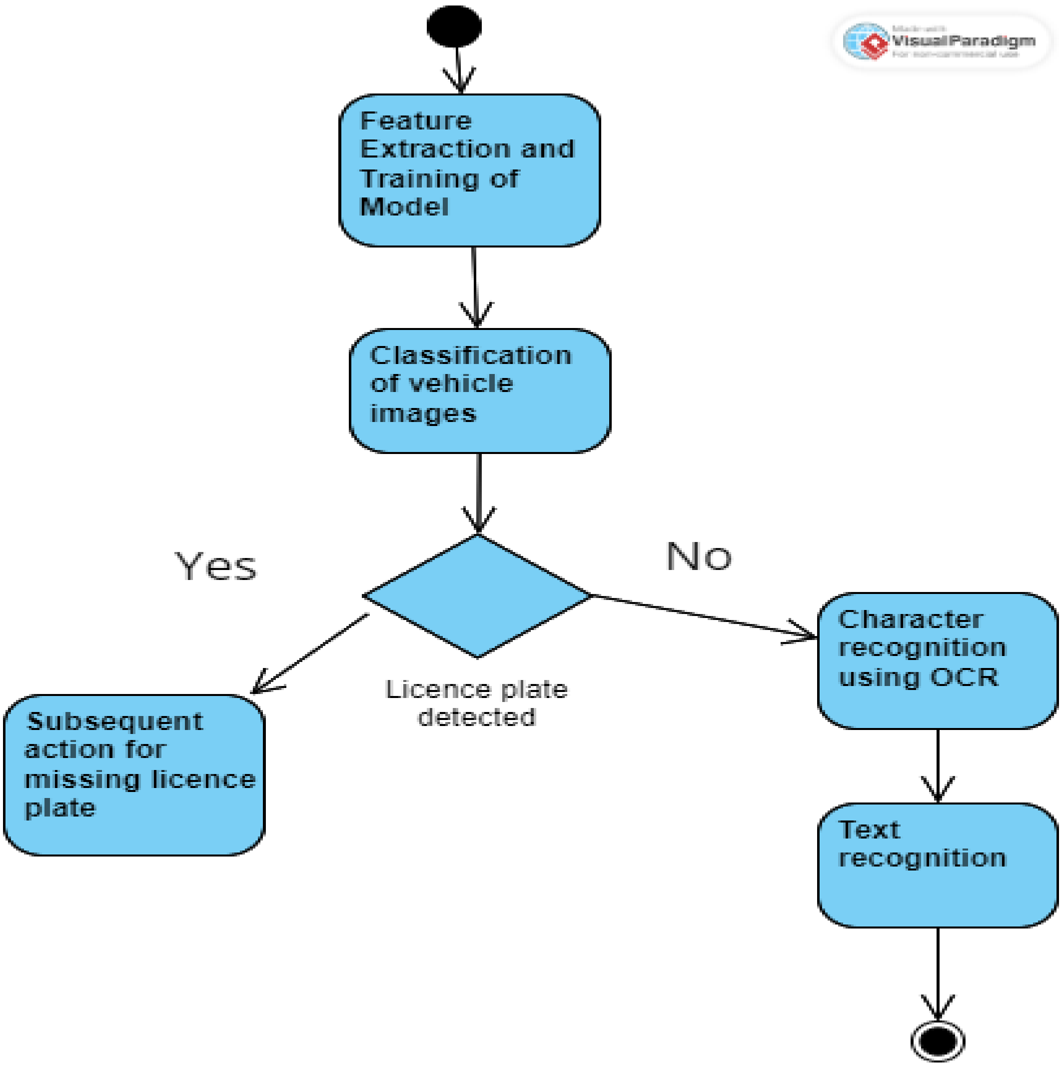


Figure 4: Character recognition post classification

#### 3.2.1 Pre-trained models

The use of pre-trained networks is referred to as transfer learning. In transfer learning, knowledge is transferred from one field to another to generate better results. Thus knowledge from previous tasks is used to learn new tasks. The various pre-trained CNN networks used to compare the results achieved with the proposed framework are VGG-Net and ResNet-50.

# 4 Experimental Setup

## 4.1 Implementation details

The suggested framework's implementation is accelerated on the Google Cloud Platform via a GPU. The approach is carried out using Python and TensorFlow, Pillow, OpenCV, OCR, and PyTesseract. A total of three datasets were used for the experimental study.

## 4.2 Datasets used

We evaluate the proposed architecture on three popular benchmarks: the Stanford Cars dataset, Indian Licence Plates Dataset, and Car Licence Plate Detection Dataset. The datasets are described along with the quality of

Table 1: Convolutional and Max Pooling Layers used in CNN architecture

| Layer | Number of filters | Size | Activation |
|---|---|---|---|
| Convolution | 32 | 3 | Relu |
| Convolution | 32 | 3 | Relu |
| Max Pooling | - | 2 | - |
| Convolution | 64 | 3 | Relu |
| Convolution | 64 | 3 | Relu |
| Max Pooling | - | 2 | - |
| Convolution | 128 | 3 | Relu |
| Convolution | 128 | 3 | Relu |
| Max Pooling | - | 2 | - |
| Convolution | 256 | 3 | Relu |
| Convolution | 256 | 3 | Relu |

Table 2: Fully connected layers used in CNN architecture

| Layer | Number | Activation |
|---|---|---|
| Flatten | - | - |
| Fully connected layer | 1024 | Relu |
| Fully connected layer | 2 | Softmax |

images as follows:

- Stanford Cars dataset[28]: It comprises a set of 297 model cars with 43615 images. The second Stanford Cars dataset encloses a set of 196 model cars with 16,185 images. The dataset 16,185 good quality images captured from the rear of each of the 196 car classes studied. With 8,144 training images and 8,041 scoring images, the images are approximately evenly split between training and scoring.
- Indian Licence Plates Dataset[29]: This database consists of over 10000 images of vehicle licence plates on Indian cars. The dataset contains over 20,000 photos collected from 4000+ crowdsource contributors utilising mobile phones. All photographs have a minimum quality of HD (1920x1080) and include 700+ cities and villages in India, displaying a diversity of lighting conditions, including day and night, as well as different distances and angles. The dataset is useful in a variety of fields, including number plate detection, Automatic Number Plate identification (ANPR), number plate identification, self-driving systems, and others.
- Car Licence Plate Detection Dataset[30]: This dataset contains 433 images of various car license plates. The images of licence plates have good quality.

# 5 Results

On the three benchmarks, this section compares the results obtained using pre-trained models with those acquired using the proposed framework. Table 3 shows the findings obtained utilizing performance metrics like Precision, Recall, F1-score, and Accuracy based on the Stanford Cars Dataset. Table 4 shows the performance metrics acquired from the Car Licence Plates Database regarding the specified performance measures. Table 5 displays the results from the Indian Licence Plates Dataset. Compared to pre-trained models in Tables 3, 4, and 5, the proposed framework outperforms them on all three datasets: Stanford Cars Dataset, Car Licence Plates Dataset, and Indian Licence Plates Dataset. The use of our proposed architecture increases the framework's overall Performance. Table 4 presents a comprehensive analysis of performance metrics for the Stanford Cars Dataset, including Precision, Recall, F1 score, and accuracy. The results reveal the remarkable superiority of the proposed framework over pre-trained models like ResNet-50 and VGG-16. The proposed framework excels in license plate detection for the Cars License Plates Dataset and the Indian License Plates Dataset, achieving accuracy rates of 99.1% and an impressive 100%, respectively, which is better than the 98.2% accuracy achieved on the Stanford Cars dataset. Additionally, VGG-16 demonstrates superior Performance compared to ResNet-50, showcasing higher Precision, Recall, F1 score, and accuracy. Specifically, VGG-Net achieves 99% accuracy on the Indian License Plates Dataset and 96% accuracy on the Car License Plates Dataset. These findings depict the clear advantage of the proposed model over pre-trained alternatives in terms of accuracy.

The character recognition process following feature extraction and classification on localized images is visually depicted in Figure 4. During this process, various test samples are input into the model, yielding an impressive character recognition accuracy of up to 96%

Figure 5 graphically illustrates the achieved accuracy across the three benchmarks, showcasing a direct comparison between the Performance of pre-trained models and the proposed framework.

Table 3: Performance metrics obtained on Stanford Cars dataset

| Models | Precision | Recall | F1 score | Accuracy |
|---|---|---|---|---|
| ResNet50 | 0.929 | 0.932 | 0.917 | 0.931 |
| VGG-Net | 0.952 | 0.96 | 0.949 | 0.956 |
| Proposed Model | 0.975 | 0.98 | 0.977 | 0.982 |

Table 4: Performance metrics obtained on Car Licence Plates Dataset

| Models | Precision | Recall | F1 score | Accuracy |
|---|---|---|---|---|
| ResNet50 | 0.945 | 0.957 | 0.951 | 0.95 |
| VGG-Net | 0.965 | 0.972 | 0.968 | 0.96 |
| Proposed Model | 0.992 | 0.991 | 0.992 | 0.991 |

Table 5: Performance metrics obtained on Indian Licence Plates Dataset

| Models | Precision | Recall | F1 score | Accuracy |
|---|---|---|---|---|
| ResNet 50 | 0.95 | 0.966 | 0.958 | 0.96 |
| VGGNet 16 | 0.99 | 0.99 | 0.98 | 0.99 |
| Proposed Model | 1.00 | 1.00 | 1.00 | 100 |

The model was trained for 100 epochs using the Adam optimizer with a batch size of 32, an initial learning rate of 0.01, and a gradient noise scale of 0.001. On the datasets, procedures such as rotation, scaling, and shift were used to augment the data. The images were shuffled and separated into two equal subsets for training and testing in a 60:40 ratio. Further, the promptness of the recognition system is as important as its accuracy. For instance, the accuracy and Time required to recognize the characters from test samples of the Stanford Cars Dataset are shown in Table 6. This shows that the Time taken for the recognition is very less, and accuracy is high.

### 5.1 Comparison with existing literature

Table 6 depicts the time and accuracy metrics for two test samples for character identification on the Stanford Cars Dataset. According to the results, Test Sample 1 was recognised in 1.9 seconds with a 92% accuracy, while Test Sample 2 took 2.3 seconds with a 96% accuracy. Table 7 compares the work done in the existing literature on Stanford Cars Dataset and the Indian Licence Plates Dataset. Various results include [21] achieving an accuracy of 98.1% on Stanford Cars and [31] achieving an accuracy of 98.45% on Indian Licence Plates. The table provides a comprehensive assessment of the performance of various strategies implemented in this domain.

The technique used in [22] employs an Improved Bernsen Algorithm (IBA) and Connected component analysis. (CCA) models to localize and detect licence plates. The techniques used in [23] trains end-to-end two EfficientNet-b0 models on disjoint subsets of data followed by efficient adaptive ensemble technique. [24] makes use of attention-aware data augmentation techniques with features derived from deep learning model InceptionV3, pre-trained on large scale datasets. Table 7 shows a comparison with existing literature in terms of accuracy as compared with the benchmarks used.

## 6 Conclusion and Future Work

The field of licence plate detection has been the subject of numerous investigations. Various researchers have developed various implementation methods and techniques for this technology. Each strategy has its pros and

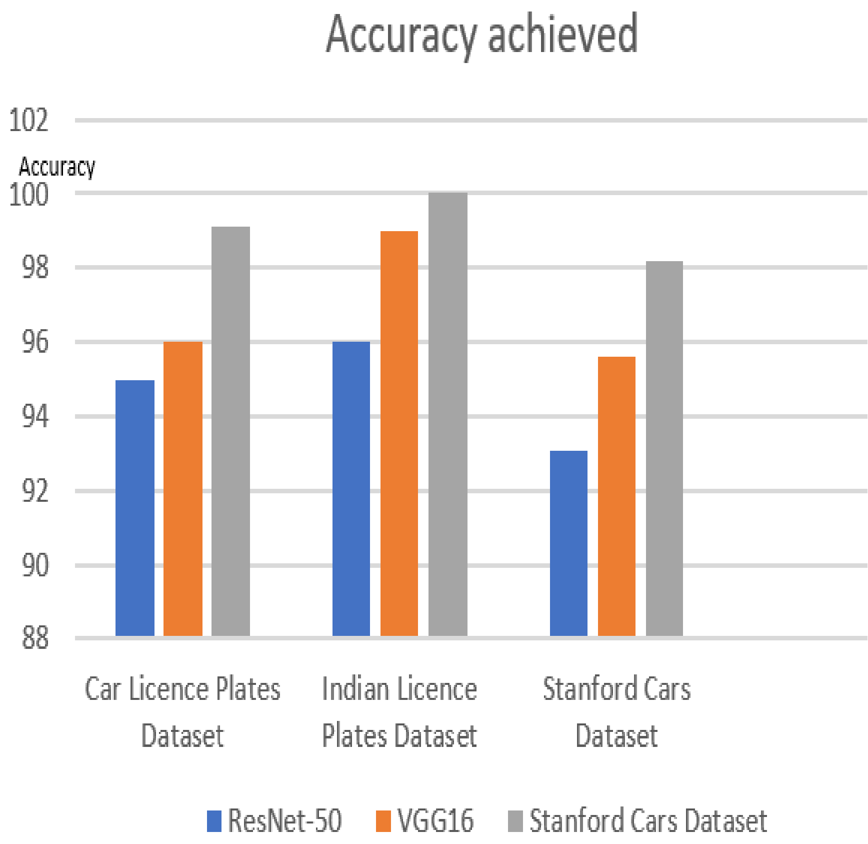


Figure 5: Accuracy achieved on datasets

Table 6: Time and accuracy achieved for character recognition in Stanford Cars Dataset

| | Time(sec) | Accuracy |
|---|---|---|
| Test Sample 1 | 1.9 | 92% |
| Test Sample 2 | 2.3 | 96% |

cons with respect to the detection of licence plates in vehicles. Even though LP detection and identification have been the subject of numerous studies, our approach uses a pre-processing framework with enhanced localization and feature extraction using improved Canny edge detection and bilateral filters. These localization procedures are implemented before the proposed model is employed to detect license plates. This is followed by feature extraction and classification using a CNN architecture. OCR character recognition aids in character comprehension when a licence plate is present. The results obtained are compared using pre-trained CNN models on three popular benchmarks. Our proposed approach attains an accuracy of more than 98% for the three datasets. It is observed that the results obtained are superior as compared to previous approaches on the benchmarks. Future studies will focus on the accuracy of detection and recognition plates under different restrictions and constrained environments. We will also attempt to use modern technologies (smartphones, tablets, etc.)Â  and video surveillance to build a real-time system in the future.

# 7 Compliance with Ethical Standards

## 7.1 Funding

The authors have not received any funding for this research

Table 7: Comparison with existing literature on Stanford Cars Dataset and Indian Licence Plates Dataset

| Method | Dataset | Accuracy(in %) |
|---|---|---|
| [21] | Stanford Cars | 98.1 |
| [22] | Stanford Cars | 96.41 |
| [23] | Stanford Cars | 96.13 |
| [24] | Stanford Cars | 95.96 |
| [25] | Stanford Cars | 96.32 |
| [26] | Indian Licence Plate | 82.68 |
| [26] | Indian Licence Plate | 87.2 |
| [31] | Indian Licence Plate | 97.8 |
| [31] | Indian Licence Plate | 98.45 |

### 7.2 Conflict of interest

The authors Ruchi Mittal, Shefali Arora, Dhruv Arora, Avinash Shrivastava declare no conflict of interest.

# References

[1]T. Saidani, Y.E Touati, A vehicle plate recognition system based on deep learning algorithms, Multimedia Tools and Applications, 80(30), 36237-36248,2021.

[2]N. Mufti, S.A.A Shah,Automatic number plate Recognition: A detailed survey of relevant algorithms, Sensors, 21(9), 3028,2021.

[3]N. Szabo and R. Tanaka, Residue Arithmetic and Its Applications to Computer Technology, New York: McGraw Hill, 1967.

[4]Hsieh, J. W., Yu, S. H., Chen, Y. S. (2002, August). Morphology-based license plate detection from complex scenes. In 2002 International Conference on Pattern Recognition (Vol. 3, pp. 176-179). IEEE.

[5]Poon, J. C., Ghadiali, M., Mao, G. M. T., Sheung, L. M. (1995, July). A robust vision system for vehicle licence plate recognition using grey-scale morphology. In 1995 Proceedings of the IEEE International Symposium on Industrial Electronics (Vol. 1, pp. 394-399). IEEE.

[6]Asif, M. R., Chun, Q., Hussain, S., Fareed, M. S. (2016). Multiple licence plate detection for Chinese vehicles in dense traffic scenarios. IET Intelligent Transport Systems, 10(8), 535-544.

[7]Deb, K., Jo, K. H. (2009). A vehicle license plate detection method for intelligent transportation system applications. Cybernetics and Systems: An International Journal, 40(8), 689-705.

[8]Abolghasemi, V., Ahmadyfard, A. (2009). An edge-based color-aided method for license plate detection. Image and Vision Computing, 27(8), 1134-1142.

[9]Muhammad, J., Altun, H. (2023). Guiding genetic search algorithm with ANN based fitness function: a case study using structured HOG descriptors for license plate detection. Multimedia Tools and Applications, 82(12), 17979-17997.

[10]Tom, R. J., Kumar, A., Shaik, S. B., Isaac, L. D., Tripathi, V., Pareek, P. (2022, April). Car License Plate Detection and Recognition Using Modified U-Net Deep Learning Model. In 2022 8th International Conference on Smart Structures and Systems (ICSSS) (pp. 01-06). IEEE.

[11]Khan, M. G., Saeed, M., Zulfiqar, A., Ghadi, Y. Y., Adnan, M. (2022). A Novel Deep Learning Based ANPR Pipeline for Vehicle Access Control. IEEE Access, 10, 64172-64184.

[12]Srikanth, P., Kumar, A. (2022). Automatic vehicle number plate detection and recognition systems: Survey and implementation. In Autonomous and Connected Heavy Vehicle Technology (pp. 125-139). Academic Press.

[13]Kaur, P., Kumar, Y., Ahmed, S., Alhumam, A., Singla, R., Ijaz, M. F. (2022). Automatic license plate recognition system for vehicles using a cnn. Computers, Materials Continua, 71(1), 35-50.

[14]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.

[15]Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J., Gupta, D., Khanna, A., Shankar, K., ... Joshi, G. P. (2020). Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. Ieee Access, 8, 92907-92917.

[16]Ganta, S., Svsrk, P. (2020). A novel method for Indian vehicle registration number plate detection and recognition using image processing techniques. Procedia Computer Science, 167, 2623-2633.

[17]Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164.

[18]Yu, W., Yang, K., Bai, Y., Xiao, T., Yao, H., Rui, Y. (2016, June). Visualizing and comparing AlexNet and VGG using deconvolutional layers. In Proceedings of the 33 rd International Conference on Machine Learning.

[19]Patel, C., Shah, D., Patel, A. (2013). Automatic number plate recognition system (anpr): A survey. International Journal of Computer Applications, 69(9).

[20]Arsenovic, M., Sladojevic, S., Anderla, A., Stefanovic, D. (2017). Deep learning driven plates recognition system. In Proc. 17th Int. Sci. Conf. Ind. Syst.(IS) (pp. 1-4).

[21]I. V. Pustokhina et al., "Automatic Vehicle License Plate Recognition Using Optimal K-Means With Convolutional Neural Network for Intelligent Transportation Systems," in IEEE Access, vol. 8, pp. 92907-92917, 2020, doi: 10.1109/ACCESS.2020.2993008

[22]Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E. and Noy, A., 2021. Ml-decoder: Scalable and versatile classification head. arXiv preprint arXiv:2111.12933.

[23]Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z. and Duerig, T., 2021, July. Scaling up visual and vision-language representation learning with noisy text supervision. In International Conference on Machine Learning (pp. 4904-4916). PMLR

[24]Foret, P., Kleiner, A., Mobahi, H. and Neyshabur, B., 2020. Sharpness-aware minimization for efficiently improving generalization. arXiv preprint arXiv:2010.01412.

[25]Behera, A., Wharton, Z., Hewage, P.R. and Bera, A., 2021, May. Context-aware attentional pooling (cap) for fine-grained visual classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 2, pp. 929-937).

[26]Batra, P., Hussain, I., Ahad, M.A., Casalino, G., Alam, M.A., Khalique, A. and Hassan, S.I., 2022. A Novel Memory and Time-Efficient ALPR System Based on YOLOv5. Sensors, 22(14), p.5283.

[27]Damak, T., Kriaa, O., Baccar, A., Ayed, M. B., Masmoudi, N. (2020). Automatic number plate recognition system based on deep learning. International Journal of Computer and Information Engineering, 14(3), 86-90.

[28] Krause, J., Stark, M., Deng, J., Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In Proceedings of the IEEE international conference on computer vision workshops (pp. 554-561).

[29] Data Cluster Labs. https://www.kaggle.com/datasets/dataclusterlabs/indian-number-plates-dataset

[30] Car Licence Plates Dataset.https://makeml.app/datasets/cars-license-plates

[31] Sharma, N., Haq, M. A., Dahiya, P. K., Marwah, B. R., Lalit, R., Mittal, N., Keshta, I. (2023). Deep Learning and SVM-Based Approach for Indian Licence Plate Character Recognition. Computers, Materials Continua, 74(1).