

LPRHNet: Depth-Wise Convolution Enabled Feature Learning for License Plate Recognition through Hybrid Network Scheme

Deepesh Dewangan^{1*}, Dr. Ranu Pandey²

^{1,2} Shri Rawatpura Sarkar University, Raipur, Chhattisgarh, India

Abstract:

License plate recognition is challenging due to varied lighting, diverse plate designs, and occlusions. It is crucial for smart traffic management, enabling efficient monitoring, law enforcement, and traffic flow optimization, contributing to enhanced safety and streamlined urban mobility. In this direction, we introduce a hybrid Convolutional Neural Network (CNN) model for license plate detection, merging the strengths of DenseNet, InceptionNet, and XceptionNet through depth-wise convolution. Our model attains unprecedented accuracy, boasting remarkable results on diverse public datasets: 99.1% for AOLP, 96.8% for Chinese LP, 59.8% for RODOSOL-ALPR, and 78.3% for UFPR-ALPR. The model's collective prowess manifests in an impressive average accuracy of 83.5%, positioning it at the forefront of license plate detection methodologies. Notably, our approach surpasses several state-of-the-art methods and is validated through standard evaluation measures. This work signifies a substantial stride toward more accurate and robust systems crucial for real-world applications, such as smart traffic management and law enforcement.

Keywords: Convolutional Neural Networks, Feature Extraction and Learning, Hybrid Architecture, License Plate Recognition, Smart Traffic Systems

1. INTRODUCTION

The Smart traffic controller system and automobile industries, which are now equipped with the latest technologies, are growing with more advanced transportation demand from society. Various ground-breakers like Volvo, BMW, Google, and other research teams are also operative on their prototype of autonomous vehicles [1]. Similarly, Google AI-based traffic controller and other pioneers are also expanding their potential and are expected to be implemented in the near future to control the traffic flow.

However, even with these developments, full commercialization of these vehicles and automation of complete smart traffic control systems are still challenging due to the heterogeneous and unpredictable driving and road traffic environment. A report from the World Health Organization (WHO) reveals that deaths from road traffic accidents are high [2], and the reasons behind such mishaps are random [3] and may include casual driving, inability to observe the driving environment efficiently, varied road structures, illumination issues, drink-and-drive cases, etc.

Under various conditions, this system can be a better assistive tool to reduce such incidents, but recognizing each object in a dynamic environment is challenging and complex to implement. Moreover, these systems are customized with an ample number of sensors, which not only observe the traffic and driving scene but also behave intelligently in terms of controlled driving under standard conditions. Among these sensors, vision sensors are gaining popularity over LiDAR, radar, short-distance, and other key sensors [1].

Additionally, smart traffic systems mostly use vision sensors to recognize targets and primarily identify pedestrians and vehicles. Apart from recognition, they also execute necessary actions against the target object under certain sets of rules if any traffic violation happens. In this direction, most traffic violations are caused by vehicles driven by careless drivers. The recognition of such vehicles primarily focuses on the driver's identity and the license plate of the vehicle.

However, in most cases, accurately identifying the driver's identity inside a vehicle is a major challenge from a camera perspective, as the windshield or side glass reduces visibility. As an alternative solution, license plate recognition becomes a key objective. Through license plate recognition, a smart traffic system can immediately issue a ticket to the vehicle owner, and the traffic department can be notified about violations so that appropriate actions can be taken.

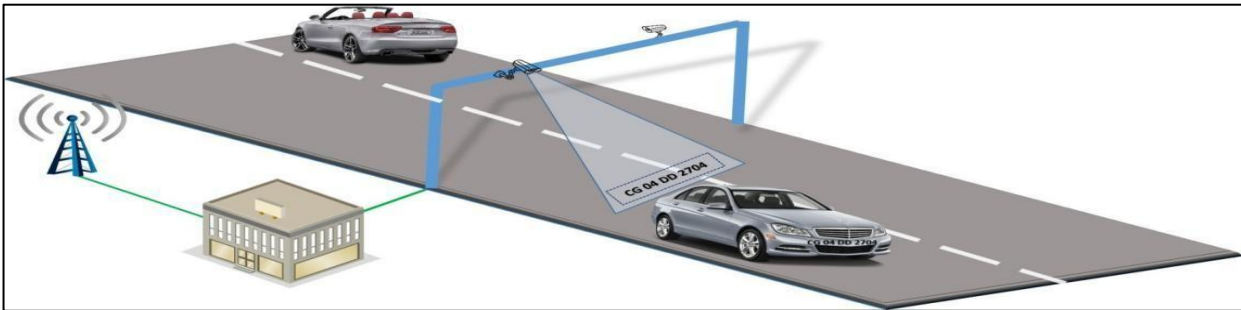
Most countries already follow standard license plate structures attached at both ends of a vehicle. However, different font types and styles used in license plates add complexity to recognition. A number of studies have been conducted to recognize number plates/license plates using edge information [4,5], color feature information [6,7], segmentation-based methods [8,9], and modern intelligent techniques [10,11].

Recent advancements in computer vision and deep learning-based approaches have significantly improved recognition performance in many applications. The key idea is to obtain significant features from the target object and process them to regulate vehicle detection through number plate recognition in a smart traffic environment using vision sensors. Vision sensors are gaining popularity in collaboration with multiple deep learning approaches [12–16,50,51].

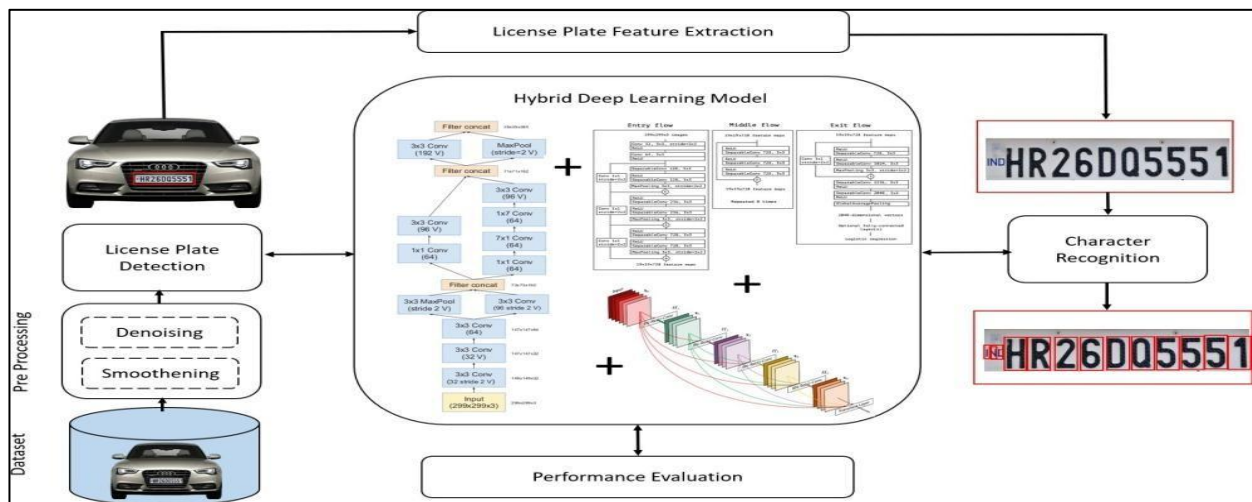
However, an important question remains: which features are significant? A target object represented through a large number of features may increase system complexity during the learning phase. Similarly, using too few features may result in poor recognition. Therefore, maintaining a balance between effective learning and a compact, significant feature pool is a challenging task.

Different state-of-the-art methods have shown fair performance in various domains. The ability to extract significant features and the behavior of these models largely depend on their architectural design, which focuses on feature mapping at lower, middle, and higher levels. Instead of using a single architecture, combining two or more models can be advantageous for license plate recognition in the proposed domain.

In this work, the potential of well-known deep learning models and their hybridization is investigated for license plate recognition, especially for smart traffic control systems using vision sensors. Moreover, in the adopted strategy, alphanumeric characters will also be obtained.



(a) General representation of a conventional smart traffic control system to recognize the license plate from the vehicle



(b) Process flow of the adopted strategy LPRHNet using hybrid network learning.

Figure 1: Schematic of the proposed approach

From the recognized license plates, the mechanism will not only identify the vehicle and its owner against careless driving in a busy road traffic environment, but it will also serve as an intelligent service for smart traffic and smart cities of the country. Additionally, the implemented hybrid scheme can be utilized in similar real-life domains.

The summarized schematic of the proposed approach is shown in Figure 1. Figure 1(a) describes the general representation of a typical smart traffic control system to recognize the license plate from the vehicle, and Figure 1(b) portrays the process flow of the adopted strategy.

The structure of the rest of the article is as follows: Section II highlights the major challenges of license plate recognition through standard and latest practices in this domain. Section III portrays the adopted scheme of implementation. The outcome and discussion of the implementation are presented in Section IV, and finally, the concluding remarks along with future scope and limitations are described in Section V.

2. RELATED WORK

The license plate detection step greatly influences the accuracy of the subsequent steps. The input is an image containing none, one, or multiple license plates, and the output should be the portion of the image containing only the license plates. Based on different schemes, the following studies are considered in this work to identify potential gaps.

2.1 Edge-based Schemes

Zheng et al. [17] utilize solely vertical edge detection in their approach, as horizontal edge detection was found to introduce higher errors due to potential confusion with the car bumper's horizontal edges [18]. Moreover, they exclusively consider rectangles with the same aspect ratio as license plates to enhance the detection process by excluding unrelated candidates. The reported detection rate achieved in their method is 99.7%. Alternatively, the Hough transform can also be employed to detect license plates in an image by detecting straight lines [19].

2.2 Color-based Schemes

A genetic algorithm is used in [20] to learn how to recognize color combinations. By training under different illumination conditions, it learns which colors to detect. Using the average brightness of each region, it differentiates license plate regions from others. The authors report an accuracy of 92.8%. The mean shift algorithm is used in [21] to segment regions based on color differences. Edge-based methods are then applied to locate the license plate within candidate regions.

2.3 Texture-based Schemes

License plates are identified as irregularities in image texture, relying on abrupt texture changes to detect potential plates. Kaushik et al. [22] introduce a technique combining sliding concentric windows and histograms, improving detection accuracy. Additionally, Zhang et al. [23] present a method utilizing both global statistical features and local Haar-like features. Global features help exclude background areas, while local features enhance robustness against variations in brightness, color, size, and position.

2.4 Character-based Schemes

Character-based approaches treat license plates as sequences of characters and analyze the image to identify regions containing character-like patterns. Matas et al. [24] adopt a region-based method where a neural network classifies regions based on character-like attributes. A license plate is identified when combinations of such regions are detected.

Li et al. [25] enhance this approach by detecting all characters using a 37-way output convolutional neural network (CNN), followed by another CNN to remove false positives. Although this method takes approximately 5 seconds per image, it eliminates the need for character segmentation.

In [26], an image saliency-based method is used. It first detects and segments characters using an intensity saliency map, achieving a high recall rate. A sliding window is then applied to detect license plates based on saliency features.

2.5 Hybrid Schemes

Combining multiple features proves to be effective for license plate detection. Porikli et al. [27] use a covariance matrix incorporating statistical and spatial information extracted from license plates. A neural network trained on these features and non-license plate regions achieves accurate detection.

In [28], two time-delay neural networks (TDNN) are used: one analyzes color and the other texture. Their outputs are combined to identify candidate regions, achieving a 100% detection rate on a dataset of 1000 toll-gate video sequences. CNN-based general object detection models also fall under this category.

2.6 Projection-based Schemes

Projection-based approaches utilize the contrast between character and background colors, producing opposing values after binarization. In [29], vertical projection identifies character boundaries, followed by horizontal projection to segment each character, achieving a segmentation rate of 97.5%. However, this method requires prior knowledge of character structure and is sensitive to font variations and noise.

2.7 Learning-based Schemes

Learning-based approaches use machine learning to identify characters based on features. Jiao et al. [30] use a neural network trained on image density features. CNN architectures are particularly effective, as they automatically learn relevant features without predefined rules.

Silva et al. developed a YOLO-based CNN architecture where the first phase detects license plates and the second phase extracts text [31]. Similarly, Shi et al. proposed an end-to-end trainable model for scene text recognition without character segmentation, resulting in a lightweight neural network [32]. Laroca et al. [33] introduced a new stage involving corner detection and character classification using two public datasets. Wang et al. [34] proposed GRCNN, focusing on contextual information. Feature learning approaches aim to extract significant features for CNN-based detection [35, 36] and text recognition tasks [37–42].

Table 1: Summary of various schemes for license plate detection

Schemes	Strength	Limitation
Color	Not affected by inclined or deformed plates	Limited results when used alone
Character	High recall rate and robust to rotation	Has trouble when image contains other text
Edge	Simple and fast	Returns too many candidates when image has many edges and has trouble when image is blurry
Texture	Not dependent on the license plates having regular shapes	Computationally complex when input image has many edges or different illumination.
Hybrid	Very robust and effective	Computationally complex
Projection	Simple and robust to rotation	Require prior knowledge, sensitive to font changes and noise affect the result.

Integrating two or more features in approaches is frequently more dependable and efficient than utilizing a single feature. Nevertheless, these methods tend to be more computationally intensive compared to single-feature approaches. These approaches exhibit robustness to distortions, and the recognition process is swift due to the utilization of fewer features compared to pixels. However, feature extraction can be time-consuming, and if non-robust features are chosen, it may lead to potential problems.

Key Contributions

1. The work introduces a novel hybrid neural network architecture specifically designed for license plate recognition that combines the strengths of standard convolutional neural networks (CNNs) for feature extraction and sequence modeling, significantly improving recognition accuracy.
2. Even though the adopted model uses a hybrid approach, with the integration of depth-wise convolution, it demonstrates robustness under challenging conditions, including varying lighting, weather, and license plate orientations.
3. Moreover, binarization schemes simplify the sample images, making them easier to interpret and analyze, and prove useful in optical character recognition with high recognition performance.

3. MATERIAL AND METHODS

3.1 Dataset

Any deep learning based model including the proposed approach requires a large amount of data to learn the important features. These models work fine on any dataset in the standard conditions. However, the diversity of the data challenges the computing of a model to obtain the significant feature pool. In addition, complexity in different font styling and designs of license plate are also required to provide some representational complexity to the model. Therefore, keeping these issues in the primary phase; four different public dataset is referred for the proposed work RODOSOLALPR [43], AOLP [44],

UFPR-ALPR [45] and Chinese LP [46]. The Chinese LP and AOLP dataset contains samples of various resolution whereas, the UFPR-ALPR and RODOSOL-ALPR mostly contains the sample images of 1920x1080 and 1280x780 resolutions respectively. For the proposed work; 5000 image samples, 2049 samples, 4500 samples and 20000 samples have been considered from the Chinese LP, AOLP, UFPR-ALPR and RODOSOL-ALPR respectively.

3.2 Pre-processing

Pre-processing in the proposed approach involves a series of critical schemes designed to prepare raw input data for effective feature learning analysis. Starting with the normalization schemes of Z-score normalization, pixel values across the dataset are standardized, that ensures the network is less sensitive to variations in illumination and pixel intensity. Data augmentation, employing techniques like random rotation, flipping, and cropping, diversifies the training dataset and combats over-fitting, enhancing the model's generalization capabilities. Further, the histogram equalization redistributes pixel values to achieve a more balanced histogram, making objects stand out against their backgrounds. Finally, the noise reduction, accomplished through Gaussian blur cleanses the images of unwanted noise and disturbances, enhancing the network's ability to focus on relevant object details. These meticulously applied pre-processing schemes collectively enable CNNs to detect objects with precision and robustness in diverse real-world scenarios, equipping them to extract meaningful features and make accurate object detection.

3.3 Architecture

The proposed approach considers the promising state-of-the-art architectures specialized in object proposal attention mechanism. Afterward, their hybridization is also deliberated and evaluated to identify the fair performance. The idea is to obtain the most significant feature representations through different learning capabilities of adopted models namely Inception-ResNet, DenseNet and Xception whose technical details are given below:

3.3.1 Phase-I

Most of the features pool is formed through convolving different kernels over the images. Larger kernels are favored to target the global features dispersed over an enormous range of the image. Likewise, kernels who are small in size are able to provide noble outcomes in perceiving area-explicit features distributed through the image. For the formation of fair variable-sized feature pool it requires kernels of variable sizes. In the context of the Inception model various kernels of variable sizes are employed in the configuration of the same layer. In this direction, the global feature representations are obtained through the 5x5 filters, while the 3x3 filters are employed to realize the disseminated features. The adopted version of the Inception (V4) has large number of layers as compared to its previous versions and functions without having the residual influences capability. Therefore, variations in filter sizes such as 1x1, 3x3, and 5x5 can be lodged to the convolution process. In the Inception-Resnet block, residual connection holds the convolutional filters with distinct sizes. The practice of having residual connections in the inception module avoids the degradation issues and hence minimizes the overall training time. Each inception block in the adopted strategy is operated through filter-expansion layer 1x1 convolution deprived of the activation function and applied for dimensionality scale-up factor before the input depth layer. This phase is required to reimburse the cost of dimensionality reduction brought by the inception module block (refer figure 2).

Though, the traditional scheme of convolution has the potential to determine the significant features from the object but it is the computation is expensive. For the resulting convolution outcome of $D \times K \times C$; where D is the subsequent dimension, $K \times K$ is the size of convolution filter and C describes the channel for single convolution operation. Therefore, for N depth of convolution the outcome will be $D \times K \times C \times N$ results in high computation. To minimize the computation complexity depth-wise separable convolutions is utilized in the present scenario through depth-wise and point-wise convolutions. It relates with the crosswise mapping transmission and spatial operation to completely decouple. The components of the Xception module is represented through its transitional wedges: entry, middle and exit flow (figure 3).

However, the problem of vanishing gradient still exists and the model may not perform with the highest accuracy. DenseNets are a type of neural network architecture that is characterized by the concatenation of all previous feature maps within a dense block as the input to each layer. This allows for the efficient flow of information and gradients throughout the network, leading to improved performance in many tasks. Therefore, with the capability of skip connection and considering the feature map concatenation to reduce the chances of vanishing gradient with optimal accuracy, DenseNet is deliberated. Towards

the non-linear transformation NT, consider a layered network with L layers. The final outcome of the network for the Lth layer is described by I_L , and the input is denoted by I_0 . To the traditional feed-forward network, there is a direct link between Lth layers to $(L + 1)$ th layer with the skip connection can be represented as:

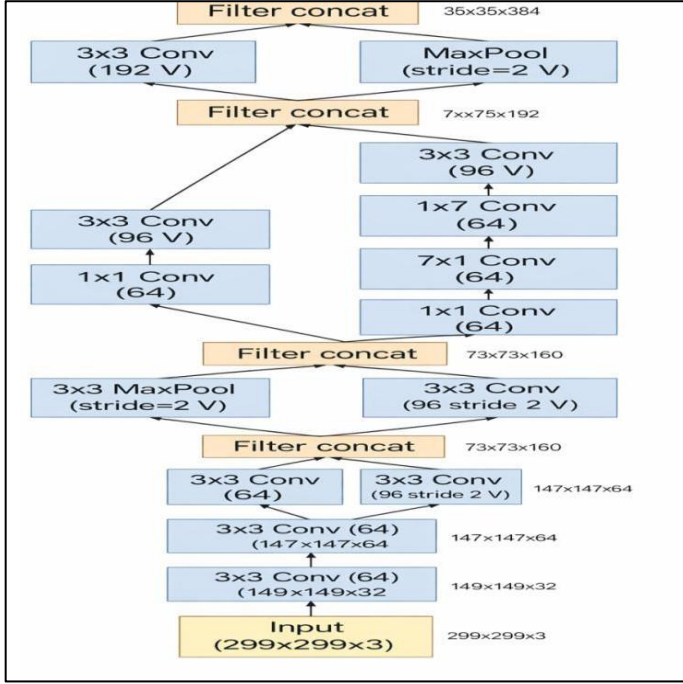


Figure 2: Inception-ResNet module [47]

In the adopted strategy, the convolution operation generates a pool of feature map through filter concatenation approach. The shapes of the different filters in the convolution operations are indicated as height X width X channels; where V indicates the valid padding [47].

$$I_L = N_T(I_{L-1}) + (I_{L-1})$$

For the same scenario, the DenseNet architecture can be represented as:

$$I_L = N_T(I_0, I_1, I_2 \dots I_{L-1}) + I_{L-1}$$

Where $[I_0, I_1, I_2]$ represents the feature map concatenation produced by the corresponding layers.

The first convolutional layer in a DenseNet typically uses 1x1 filters to reduce the number of channels, which helps to save computational resources. This layer is followed by a batch normalization (BN) and ReLU transformation, which is then followed by another convolutional layer that uses 3x3 filters to reduce the number of features to the final k output features. The use of 1x1 filters in the initial layer, followed by the 3x3 filters, helps to effectively extract and preserve important features in the input data.

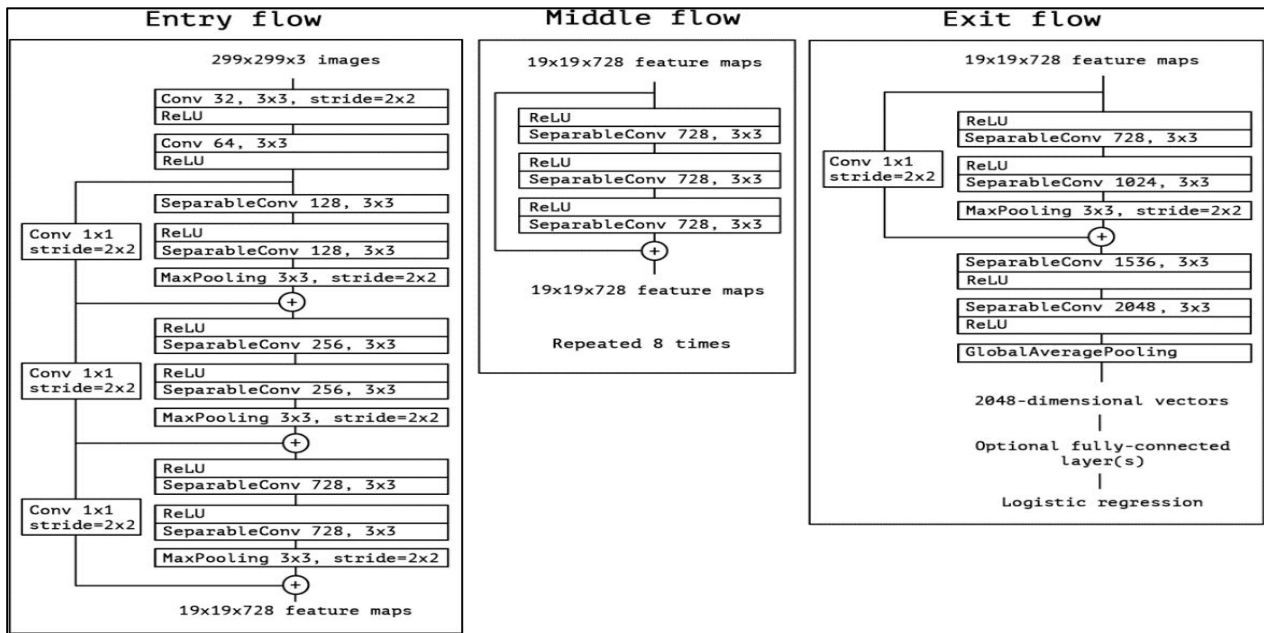


Figure 3: The Xception module through its internal configuration to obtain the significant features using skip connection [48].

The computation of the classifier in a DenseNet typically occurs in the first five convolutional layers, and takes time roughly proportional to the number of input pixels. As a preprocessing step, the input image is centered by subtracting the mean image created from a large data set. This helps to reduce the impact of lighting and other factors that might affect the input image. However, after this step, the computation performed in the convolutional layers is translationally invariant, meaning that the output of a node at the final convolution layer depends only on the value of the supporting image patch, not on the relative location of the node within the image plane. This means that for two overlapping region proposals of the same size and aspect ratio, the values of any nodes that share the same supporting image patch will be identical, and need not be recomputed, resulting in a speedup in computation.

A 5-layer dense block with a growth rate of considered $k = 4$ will be implemented as:

1. Each layer in the dense block takes all preceding feature maps as input.
2. The first layer in the dense block applies a set of convolutional filters to the input feature maps, resulting in k (4) new feature maps. These new feature maps are concatenated with the input feature maps.
3. The second layer in the dense block takes the concatenated feature maps from the first layer as input and applies a set of convolutional filters, resulting in another k (4) new feature maps. These new feature maps are concatenated with the input feature maps from the first and second layers.

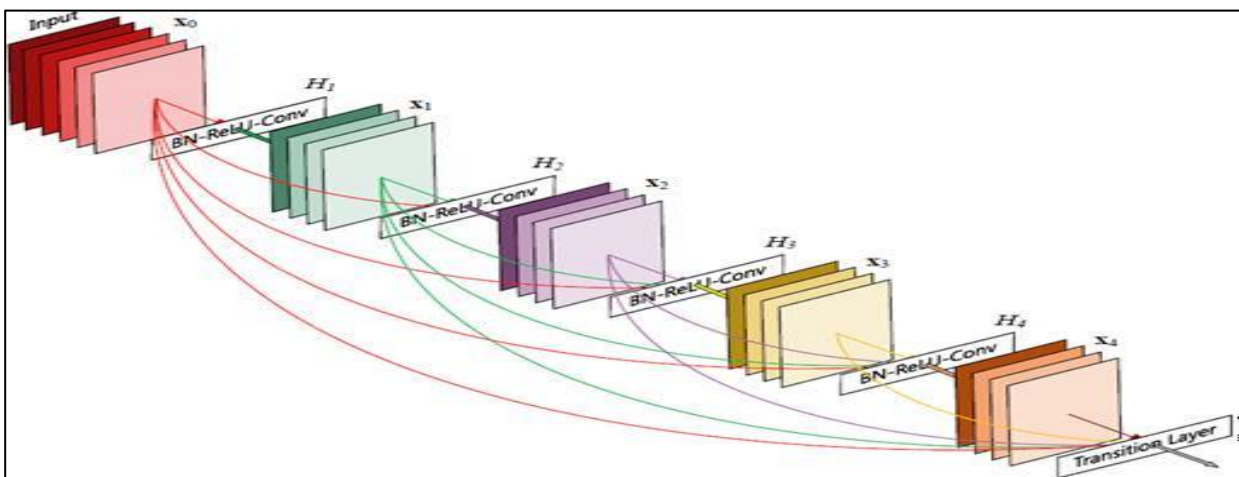


Figure 4: A Five layer dense blocks (growth rate of $k=4$) [49].

4. This process is repeated for the third, fourth, and fifth layers, resulting in a total of $k * 5 = 20$ new feature maps added to the input feature maps from each layer.
5. The output of the 5-layer dense block is a concatenation of all input feature maps and new feature maps from each layer.
6. The output of the dense block is then passed through a transition layer to reduce the number of feature maps and control overfitting.
7. This process can be repeated with different dense blocks to form a larger neural network architecture.

A depth-wise convolution applies a separate filter to each input channel. This means that if the input has n channels, n filters will be applied, one for each channel. This type of convolution is useful for preserving the spatial information in the input and is often used in mobile and embedded devices to reduce the number of parameters. Likewise, a point-wise convolution, also known as a 1×1 convolution, applies a single filter to the entire input. This means that the filter is applied to all channels of the input at once. Point-wise convolutions are often used to increase the number of channels in the output, which can increase the capacity of the network. When used together, depth-wise and point-wise convolutions can be used to increase the capacity of a network while maintaining a low number of parameters. For this depth wise separable convolution, where the spatial information is preserved by the depth wise convolution, and the pointwise convolution increases the number of channels. The concept discussed in this section and the operations of the depth-wise and point-wise convolution operations are illustrated in figure 5.

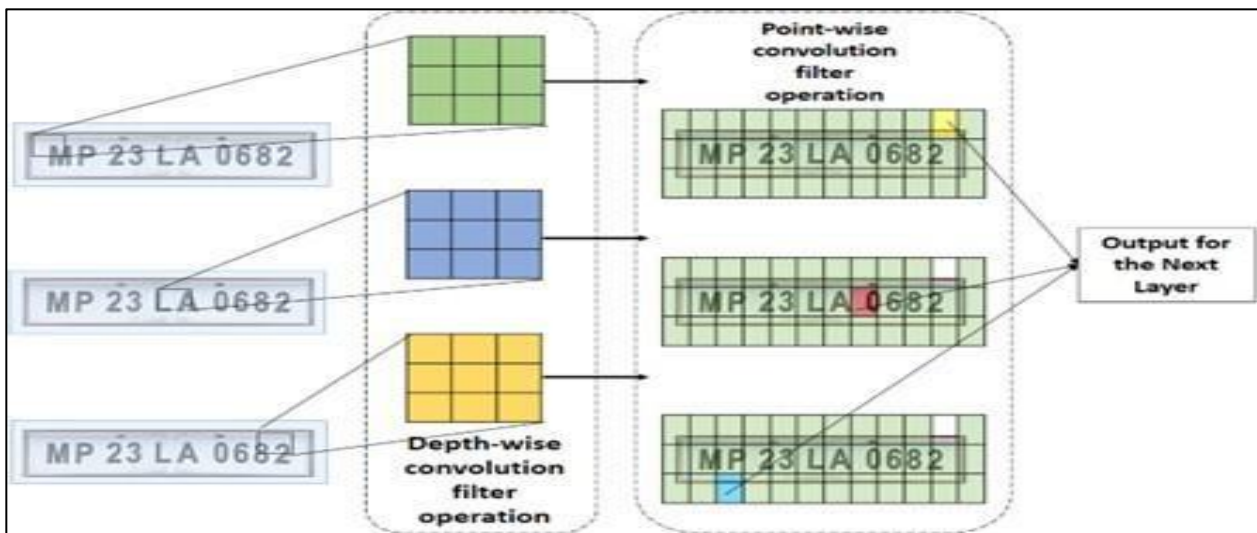


Figure 5: A 3×3 channel filter is applied as depth-wise convolution that is linked to 1×1 point-wise convolution to yield a linear arrangement

3.3.2 Phase-II

The basic idea of histogram equalization is to stretch the intensity levels of the pixels so that the intensity values are distributed more evenly across the entire range of possible values. The histogram equalization can be represented by a cumulative distribution function (CDF) which is used to map the original intensity levels of the pixels to new intensity levels. The CDF is defined as the probability that a given pixel has an intensity level less than or equal to a certain value. The CDF can be represented mathematically as:

$$F(r) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [f(i, j) \leq r]$$

Where $f(i, j)$ is the intensity level of the pixel at position (i, j) in the image, m and n are the dimensions of the image, and the sum is taken over all the pixels in the image. Once the CDF is calculated, it is used to map the original intensity levels of the pixels to new in-tensity levels. This is done by applying a function $g(r)$ to the CDF, where $g(r)$ is expressed as:

$$g(r) = (L - 1) \times F(r)$$

where L is the number of possible intensity levels in the image. The main advantage of using Gaussian filtering is its ability to preserve edges and details in the image while removing noise. The Gaussian function is a bell-shaped curve that gives more weight to pixels that are closer to the center of the window and less weight to pixels that are farther away. This allows the filter to preserve the edges and details of the image while smoothing out the noise. The mathematical representation of the Gaussian function is given by the following equation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where $G(x, y)$ is the Gaussian function, sigma is the standard deviation of the function, and x and y are the coordinates of the pixel in the window. The Gaussian filter is a convolution kernel. The convolution of an image with a Gaussian kernel is defined as:

$$G(i, j) = \frac{1}{N} \sum_{x=-n}^n \sum_{y=-n}^n f(i+x, j+y) \cdot G(x, y)$$

Where $G(i, j)$ is the filtered image, $f(i, j)$ is the original image, N is the normalizing constant and the sum is taken over all the pixels in the window. In practice, the convolution operation is performed by sliding the kernel over the image and performing the dot product between the kernel and the image at each position. The standard deviation of the Gaussian function (sigma) controls the amount of smoothing in the filtered image. Larger values of sigma result in more smoothing, while smaller values of sigma result in less smoothing.

Further, the Otsu algorithm is able to automatically determine the best threshold value for separating the foreground and background pixels in an image, which can save time and effort compared to manually selecting a threshold value. The stepwise representation of Otsu's method is based on the following steps:

1. Calculate the histogram of the image, which is a representation of the frequency of each intensity level in the image.
2. From the histogram, calculate the probability of each intensity level, $P(i)$, where I is the in-tensity level.
3. Calculate the mean intensity level of the entire image, μ .
4. For each possible threshold value t, calculate the probability of the pixels below the threshold, $P1(t)$, and the mean intensity level of the pixels below the threshold, $\mu1(t)$.
5. Similarly, calculate the probability of the pixels above the threshold, $P2(t)$ and the mean intensity level of the pixels above the threshold, $\mu2(t)$.
6. The goal of Otsu's method is to maximize the inter-class variance, which is defined as:

$$\sigma^2(t) = P1(t)[\mu1(t) - \mu]^2 + P2(t)[\mu2(t) - \mu]^2$$
7. The threshold value that maximizes the inter-class variance is chosen as the optimal threshold value.
8. Finally, the pixels that are below the optimal threshold value are set to black and the pixels that are above the optimal threshold value are set to white.

The Active Contour Model formulates the problem of character segmentation as the optimization of an energy functional:

$$E(C) = E_{data}(C) + E_{smooth}(C)$$

Where C is the contour that defines the boundary of the character and $E_{data}(C)$ and $E_{smooth}(C)$ are the data and smoothness terms of the energy functional, respectively.

The data term, $E_{data}(C)$, measures the fitness of the contour C to the image data, it can be defined as:

$$E_{data}(C) = \iint \Phi(I(x, y) - c(x, y)) dx dy$$

Where $I(x, y)$ is the image intensity, $c(x, y)$ is the average intensity of the pixels inside the contour C and $\Phi(I)$ is a function that assigns a high value to pixels that are likely to belong to the character, such as gradient magnitude or intensity gradient.

The smoothness term, $E_{\text{smooth}}(C)$, measures the smoothness of the contour C , it can be expressed as:

$$E_{\text{smooth}}(C) = \int \nabla C(s) ds$$

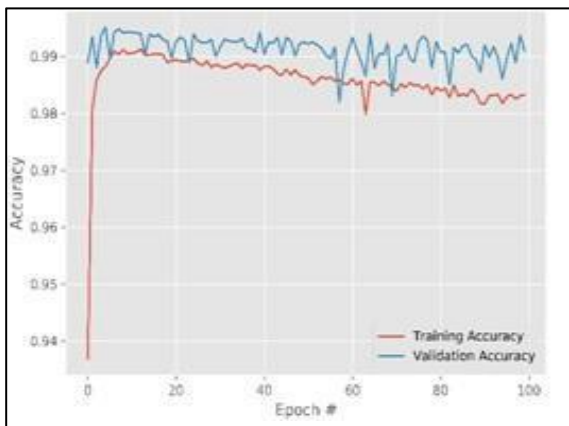
Where $\nabla C(s)$ is the gradient of the contour C and s is the arc-length parameter of the contour. By minimizing the energy functional $E(C)$, we can find the contour that best fits the character in the image. This is usually done using gradient descent optimization techniques, where the contour is iteratively updated to minimize the energy functional.

4. EXPERIMENTAL RESULT AND DISCUSSION

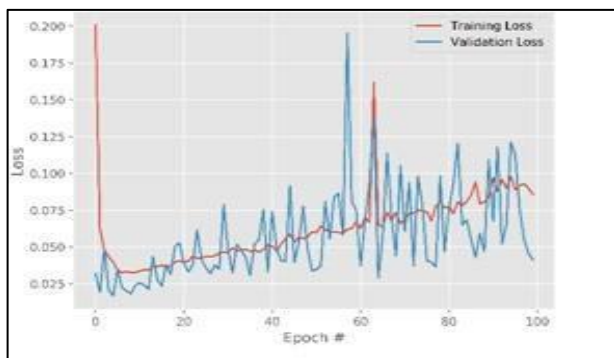
The adopted strategy for license plate recognition requires a high computational system. The configuration of the utilized platform is summarized as: NVIDIA-SMI 510.47.03, Driver Version: 510.47.03, CUDA Version: 11.6, GPU Name Tesla T4 Memory-Usage 15360MB. The pre/post processing of data does not include the computation on GPU (graphical processing unit).

4.1 Qualitative Assessment

The training graph of a deep learning model is a plot of the model's performance on the training data over time, usually as a function of the number of iterations or epochs. It provides a visual representation of how well the model is learning from the data and how it is improving over time. The training graph usually consists of two or more lines, representing the training loss and the validation loss. The training loss is a measure of how well the model is fitting the training data, while the validation loss is a measure of how well the model is generalizing to new examples. A well-trained deep learning model will show a decrease in both the training loss and validation loss over time, indicating that the model is learning the patterns in the data and generalizing well to new examples. A typical training graph for a well-trained deep learning model will show a smooth and consistent decrease in both the training loss and validation loss, with the validation loss remaining relatively close to the training loss.

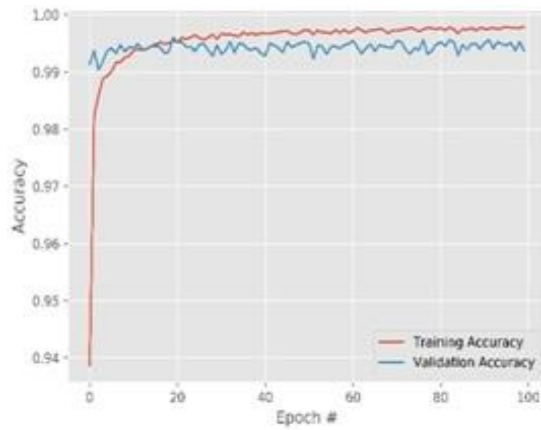


(a) Plot of accuracy parameter during training

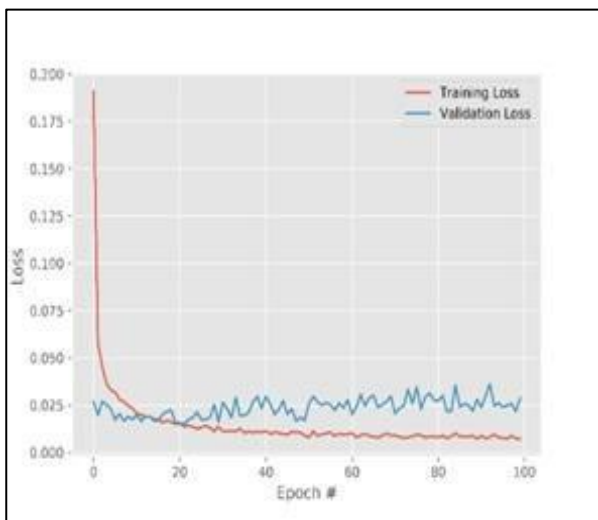


(b) Plot of loss parameter during training

Figure 6: Learning Curves of DenseNet Architecture

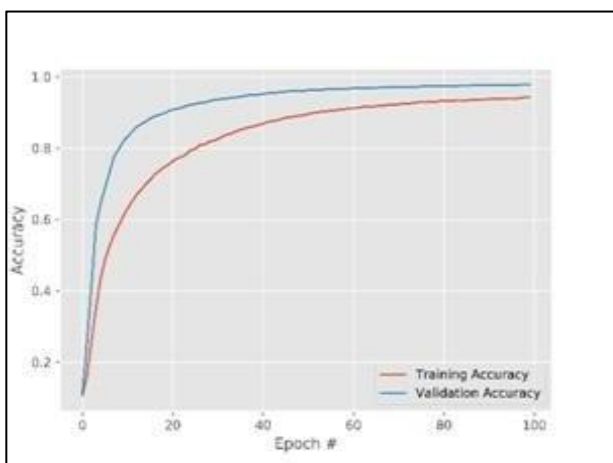


(a) Plot of accuracy parameter during training

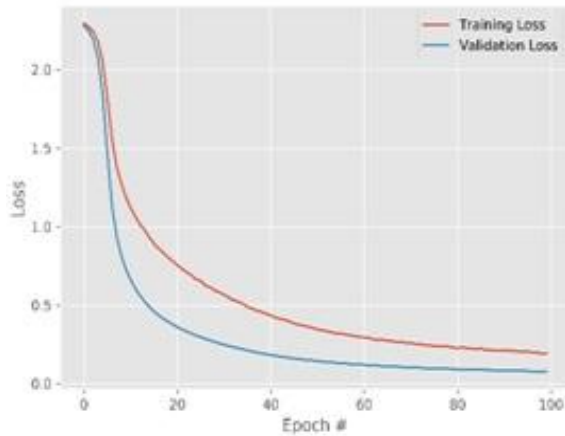


(b) Plot of loss parameter during training

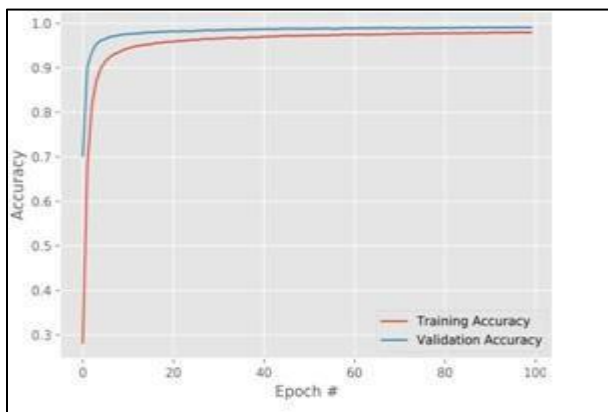
Figure 7: Learning Curves of Xception Architecture



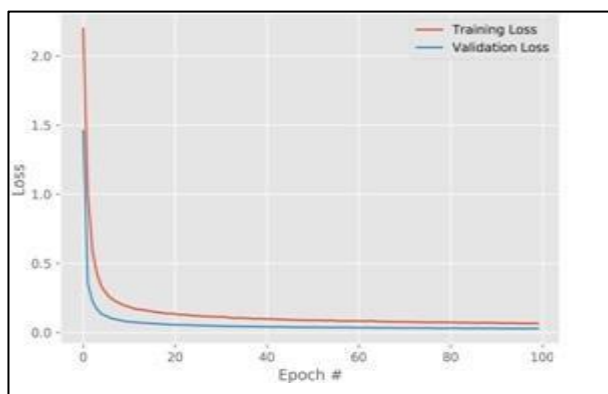
(a) Plot of accuracy parameter during training



(b) Plot of loss parameter during training

Figure 8: Learning Curves of Inception Architecture

(a) Plot of accuracy parameter during training



(b) Plot of loss parameter during training

Figure 9: Learning Curves of Hybrid Architecture

4.2 Quantitative Assessment

To confirm and validate the decent performance of the proposed work, the standard evaluation measures of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) have been considered for the recognition rate. The higher recognition rate indicates the algorithm in correctly identifying the target objects. Table 2 discusses the obtained scores from the proposed approach and its comparison with the state-of-the-art methods. The table also contains the scores attained by the individual models adopted for the hybrid approach and we show how their performance resemblance in with qualitative assessment. Later, this parameter will also be used to match how closely the performance is attained as compared with the qualitative analysis.

The performance of the model has been considered on the referred datasets namely AOLP [44], Chinese LP [46], RODOSOL-ALPR [43] and UFPR-ALPR [45]. The average score has also been considered to look in to the overall performance. The highest recognition rates obtained in each datasets is highlighted in bold text.

Table 2: Comparison of the proposed approach "LPRHnet" with the state-of-the- art methods on recognition rate score.

Study	AOLP	ChineseLP	RODOSOL-ALPR	UFPR-ALPR	Average Accuracy
R2AM[72]	83.1%	87.0%	48.6%	66.9%	71.4%
RARE[71]	93.4%	95.7%	51.6%	69.0%	77.4%
STAR-Net[69]	96.8%	96.9%	51.8%	70.9%	79.1%
CRNN[62]	87.6%	88.2%	48.2%	64.9%	72.2%
GRCNN[64]	85.9%	84.5%	48.4%	63.3%	70.5%
Holistic-CNN[66]	86.3%	90.7%	49.0%	70.3%	74.1%
Rosetta[70]	83.7%	90.1%	48.7%	63.9%	71.6%
Multi-task[65]	84.7%	86.3%	49.1%	65.3%	71.4%
TRBA[68]	99.0%	96.9%	59.6%	72.9%	82.1%
CR-NET[61]	97.7%	96.9%	55.8%	78.3%	82.2%
Fast-OCR[63]	95.8%	85.1%	49.7%	65.9%	74.1%
ViTSTR-Base[67]	95.6%	99.4%	49.3%	73.3%	79.4%
Inception-ResNet	99.0%	96.8%	55.4%	72.2%	80.8%
Xception	98.8%	96.2%	53.4%	70.2%	79.6%
DenseNet	98.1%	94.2%	48.4%	69.2%	77.4%
LPRHNet	99.1%	96.8%	59.8%	78.3%	83.5%

Lee et al. had applied CNNs for image encoding to obtain the important features and obtained 48.6% and 66.9% recognition rate for RODOSOL-ALPR and UFPRALPR dataset respectively [42]. With the image encoding CNN approach which is effective for capturing local patterns, textures, and features. but, this local processing is insufficient for understanding the global context and relationships between distant image region. Although, the individual performance of Dense-Net, InceptionNet and XceptionNet is not that decent, but comparing with the work [42], LPRHNet employed the feature fusion by combining the standard models and achieved 11.2% and 11.4% higher recognition rate. respectively. This to some extent alleviates the problem of insufficient information interaction during the fusion process.

Experimental results demonstrate that the proposed scheme can improve the performances. Considering the method to locate and identify characters at different positions in the sample image, Spanhel et al. had used CNN and obtains the lowest error rate but on a small number of sample images chances of over-fitting can not be ignored [36]. Comparing to this work, LPRHNet has demonstrated that with the hybrid mechanism of feature learning is not affected by the overfitting issue (figure 9) and therefore generalization is stable. However, the ViTSTR-Base [37] obtains the highest scores in the

Chinese LP datasets due to the attention mechanism. But, the overall performance of the LPRHnet in the context of average score is above all the scores obtained by the referred works in the mentioned datasets and thus outperforms the other studies.

Moreover, Hsu et al. had employed the Gaussian mixture model and edge clustering for the plate detection using Expectation? Maximization approach and obtains 93% score of accuracy parameter [44]. The authors have kept the constant configuration of the number of neurons in the implementation of their proposed probabilistic neural network and artificial neural networks. Comparing with the proposed work, the performance of the [44] is far behind the scores attained by the LPRHNet. Comparing the performance of LPRHNet with another feature based study; the authors have demonstrated the operations of principal visual word discovery and local feature matching in license plate detection and obtain the accuracy score of 93.2% [46]. But, the performance of [46] is not found suitable especially for the cases where the resolution is little low or observation angle is critical. As opposed to this, our approach detects approximately all the license plates even though the resolution is low. However, we have considered only the front angle of observation that is to be improved.

In another work, Laroca et al. had considered the YOLO detector to recognize the license plate and the overall accuracy obtained was 78.3% [45]. With the close score of accuracy parameters, the issues of the orientation of slanted license plate needs to be considered.

Silva et al. had utilized the hierarchical CNN and obtained the accuracy score of 65.62% for the UFPR-ALPR dataset; where as the LPRHNet had achieved 78.3% accuracy on the same dataset which is 12.68% higher than the Silva's approach [31]. It is shown by applying hybrid of the adopted models the results were significantly improved compared to the individual performance as discussed in the qualitative and quantitative assessment and the proposed model LPRHNet outperforms the other state-of-the-art methods in this domain.

5. CONCLUSION

The fusion of DenseNet, InceptionNet, and XceptionNet within our hybrid CNN model, leveraging the power of depth-wise convolutions, has proven to be a groundbreaking stride in the realm of license plate recognition. The depth-wise convolutional approach not only significantly reduced computational overhead but also enhanced the model's ability to capture intricate features at various scales. The fusion of these state-of-the-art architectures synergistically leverages the strengths of each component, resulting in a neural network that excels in discerning complex patterns and nuances within diverse datasets. Furthermore, the incorporation of depth-wise convolutions has proven to be a pivotal factor, contributing to the model's exceptional performance. This specialized convolutional technique has effectively minimized the number of parameters, fostering faster convergence during training and reducing the risk of over fitting. As a result, our hybrid CNN not only achieves superior accuracy but also demonstrates remarkable generalization across various image datasets. Our extensive experimentation and rigorous evaluation demonstrate that the hybrid model consistently outperforms its counterparts, showcasing unparalleled accuracy and efficiency with detection rate of 99.1% for AOLP, 96.8% for Chinese LP, 59.8% for RODOSOL-ALPR, 78.3% for UFPR-ALPR and an average score of 83.5% outperforming the other major state-of-the-art methods. Though our method achieves good results in license plate detection accuracy, the attention mechanism will require more computing resources than the traditional convolutional networks/hybrid approaches. License plate recognition is a very important practical application. Therefore, the follow-up work will also instantiate the algorithm of this paper to explore the real-time performance of the algorithm.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

REFERENCES

1. Dewangan, D.K., Sahu, S.P. Towards the design of vision-based intelligent vehicle system: methodologies and challenges. *Evol. Intel.* 16, 759–800 (2023). <https://doi.org/10.1007/s12065-022-00713-2>.
2. "Road traffic deaths, Global Health Observatory data repository by World Health Organization," 2019. <http://apps.who.int/gho/data/node.main.A997>.
3. D. K. Dewangan and S. P. Sahu, "Real time object tracking for intelligent vehicle," 2020 1st Int. Conf. Power, Control Comput. Technol. ICPC2T 2020, pp. 134–138, 2020, doi: 10.1109/ICPC2T48082.2020.9071478.
4. A. A. H. Farhat et al., "OCR-based hardware implementation for qatari number plate on the Zynq SoC," 2017 9th IEEE-GCC Conf. Exhib. GCCCE 2017, 2018, doi: 10.1109/IEEEGCC.2017.8448145.
5. J. Yepez and S. B. Ko, "Improved license plate localisation algorithm based on morphological operations," *IET Intell. Transp. Syst.*, ol. 12, no. 6, pp. 542–549, 2018, doi: 10.1049/iet-its.2017.0224.
6. F.Wang, L. Man, B.Wang, Y. Xiao, W. Pa, and X. Lu, "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 1007–1020, 2008, doi: 10.1016/j.patrec.2008.01.026.
7. J. fang Song, "Vehicle Detection Using Spatial Relationship GMM for Complex Urban Surveillance in Daytime and Nighttime," *Int. J. Parallel Program.*, vol. 46, no. 5, pp. 859–872, 2018, doi: 10.1007/s10766-017-0543-9.
8. J. M. Guo, Y. F. Liu, and J. Der Lee, "License plate localization and character segmentation with feedback self-learning and hybrid-binarization techniques," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 57, no. 3, pp. 1417–1424, 2007, doi: 10.1109/TENCON.2007.4429069.
9. C. Y. Lei and J. H. Liu, "Vehicle license plate character segmentation method based on Watershed algorithm," 2010 Int. Conf. Mach. Vis. Human-Machine Interface, MVHI 2010, pp. 447–452, 2010, doi: 10.1109/MVHI.2010.161.
10. J. F. Xu, S. F. Li, and M. S. Yu, "Car license plate extraction using color and edge information," *Proc. 2004 Int. Conf. Mach. Learn. Cybern.*, vol. 6, no. August, pp. 3904–3907, 2004, doi: 10.1109/icmlc.2004.1380528.
11. W. Weihong and T. Jiaoyang, "Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment," *IEEE Access*, vol. 8, pp. 91661–91675, 2020, doi: 10.1109/ACCESS.2020.2994287.
12. D. K. Dewangan and S. P. Sahu, "Lane detection in intelligent vehicle system using optimal 2- tier deep convolutional neural network," *Multimed. Tools Appl.*, vol. 82, no. 5, pp. 7293–7317, 2023, doi: 10.1007/s11042-022-13425-7.
13. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, no. January, p. 100057, 2021, doi: 10.1016/j.array.2021.100057.
14. X.Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 1, pp. 91–124, 2022, doi: 10.1109/MGRS.2021.3115137.20
15. Mhalla, T. Chateau, and N. E. Ben Amara, "Spatio-temporal object detection by deep learning: Video-interlacing to improve multi-object tracking," *Image Vis. Comput.*, vol. 88, pp. 120–131, 2019, doi: 10.1016/j.imavis.2019.03.002.
16. D. K. Dewangan and S. P. Sahu, "Optimized Convolutional Neural Network for Road Detection with Structured Contour and Spatial Information for Intelligent Vehicle System," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 36, no. 6, 2022, doi: 10.1142/S0218001422520024.
17. D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, 2005, doi: 10.1016/j.patrec.2005.04.014.
18. F. Faradji, A. H. Rezaie, and M. Ziaratban, "A morphological-based license plate location," *Proc. - Int. Conf. Image Process. ICIP*, vol. 1, pp. I-57–I-60, 2006, doi: 10.1109/ICIP.2007.4378890.
19. S. Rasheed, A. Naeem, and O. Ishaq, "Automated number plate recognition using hough lines and template matching," *Lect. Notes Eng. Comput. Sci.*, vol. 1, pp. 199–203, 2012.
20. S. K. Kim, D. W. Kim, and H. J. Kim, "Recognition of vehicle license plate using a genetic algorithm based segmentation," *IEEE Int. Conf. Image Process.*, vol. 2, pp. 661–664, 1996, doi: 10.1109/icip.1996.560964.

21. W. Jia, H. Zhang, and X. He, "Region-based license plate detection," *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1324–1333, 2007, doi: 10.1016/j.jnca.2006.09.010.
22. K. Deb, H. U. Chae, and K. H. Jo, "Vehicle license plate detection method based on sliding concentric windows and histogram," *J. Comput.*, vol. 4, no. 8, pp. 771–777, 2009, doi: 10.4304/jcp.4.8.771-777.
23. H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-based license plate detection using global and local features," *Proc. - Int. Conf. Pattern Recognit.*, vol. 2, pp. 1102–1105, 2006, doi: 10.1109/ICPR.2006.758.
24. J. Matas and K. Zimmermann, "Unconstrained licence plate and text localization and recognition," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2005, pp. 572–577, 2005, doi: 10.1109/ITSC.2005.1520111.
25. H. Li and C. Shen, "Reading Car License Plates Using Deep Convolutional Neural Networks and LSTMs," 2016, [Online]. Available: <http://arxiv.org/abs/1601.05610>.
26. K. H. Lin, H. Tang, and T. S. Huang, "Robust license plate detection using image saliency," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2, no. 1, pp. 3945–3948, 2010, doi: 10.1109/ICIP.2010.5649878.
27. F. Porikli and T. Kocak, "Robust license plate detection using covariance descriptor in a neural network framework," *Proc. - IEEE Int. Conf. Video Signal Based Surveill. 2006, AVSS 2006*, no. November 2006, pp. 107–112, 2006, doi: 10.1109/AVSS.2006.100.
28. K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, "Learning-based approach for license plate recognition," *Neural Networks Signal Process. - Proc. IEEE Work.*, vol. 2, no. C, pp. 614–623, 2000, doi: 10.1109/nnspp.2000.890140.
29. T. Nukano, M. Fukumi, and M. Khalid, "Vehicle license plate character recognition by neural networks," *Proc. 2004 Int. Symp. Intell. Signal Process. Commun. Syst. ISPACS 2004*, pp. 771–775, 2004, doi: 10.1109/ispacs.2004.1439164.
30. J. Jiao, Q. Ye, and Q. Huang, "A configurable method for multi-style license plate recognition," *Pattern Recognit.*, vol. 42, no. 3, pp. 358–369, 2009, doi: 10.1016/j.patcog.2008.08.016.
31. S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, p. 102773, 2020, doi: 10.1016/j.jvcir.2020.102773.
32. B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image- Based Sequence Recognition and Its Application to Scene Text Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2017, doi: 10.1109/tpami.2017.7885812.
33. R. Laroca, A. B. Araujo, L. A. Zanlorensi, E. C. De Almeida, and D. Menotti, "Towards Image-Based Automatic Meter Reading in Unconstrained Scenarios: A Robust and Efficient Approach," *IEEE Access*, vol. 9, pp. 67569–67584, 2021, doi: 10.1109/ACCESS.2021.3077415.
34. J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," *Adv. Neural Inf. Process. Syst.*, vol. 2017-December, no. Nips, pp. 335–344, 2017.
35. G. R. Goncalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Multi-task Learning for Low-Resolution License Plate Recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11896 LNCS, no. i, pp. 251–261, 2019, doi: 10.1007/978-3-030-33904-3_23.
36. J. Spanhel, J. Sochor, R. Juranek, A. Herout, L. Marsik, and P. Zemcik, "Holistic recognition of low quality license plates by CNN using track annotated data," 2017 14th IEEE Int. Conf. Adv. Video Signal Based Surveillance, AVSS 2017, no. August, 2017, doi: 10.1109/AVSS.2017.8078501.22
37. R. Atienza, "Vision Transformer for Fast and Efficient Scene Text Recognition," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12821 LNCS, pp. 319–334, 2021, doi: 10.1007/978-3-030-86549-8_21.
38. J. Baek et al., "What is wrong with scene text recognition model comparisons? dataset and model analysis," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019- October, pp. 4714–4722, 2019, doi: 10.1109/ICCV.2019.00481.
39. W. Liu, C. Chen, K. Y. K. Wong, Z. Su, and J. Han, "STAR-Net: A spatial attention residue network for scene text recognition," *Br. Mach. Vis. Conf. 2016, BMVC 2016*, vol. 2016-September, no. September, pp. 43.1–43.13, 2016, doi: 10.5244/C.30.43.
40. F. Borisjuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 71–79, 2018, doi: 10.1145/3219819.3219861.

41. B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust Scene Text Recognition with Automatic Rectification," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 4168–4176, 2016, doi: 10.1109/CVPR.2016.452.
42. C. Y. Lee and S. Osindero, "Recursive Recurrent Nets with Attention Modeling for OCR in the Wild," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, no. 3, pp. 2231–2239, 2016, doi: 10.1109/CVPR.2016.245.
43. R. Laroca, E. Cardoso, D. Lucio, V. Estevam, and D. Menotti, "On the Crossdataset Generalization in License Plate Recognition," *Int. Conf. Comput. Vis. Theory Appl.*, pp. 166–178, 2022, doi: 10.5220/0010846800003124.
44. G. Hsu, J. Chen, and Y. Chung, "Application-Oriented License Plate Recognition," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 552–561, 2013, doi: 10.1109/TVT.2012.2226218.
45. R. Laroca et al., "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018- July, 2018, doi: 10.1109/IJCNN.2018.8489629.
46. W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, 2012, doi: 10.1109/TIP.2012.2199506.
47. M. L'angkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, [Online]. Available: <http://arxiv.org/abs/1512.0056723>
48. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258. doi: 10.4271/2014-01-0975.
49. G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
50. Dewangan, D.K., Sahu, S.P. (2021). Lane Detection for Intelligent Vehicle System Using Image Processing Techniques. In: Verma, G.K., Soni, B., Bourennane, S., Ramos, A.C.B. (eds) *Data Science. Transactions on Computer Systems and Networks*. Springer, Singapore. https://doi.org/10.1007/978-981-16-1681-5_21.
51. Sahu, S., Sahu, S.P., Dewangan, D.K. (2023). Pedestrian Detection Using MobileNetV2 Based Mask R-CNN. In: Joby, P.P., Balas, V.E., Palanisamy, R. (eds) *IoT Based Control Networks and Intelligent Systems. Lecture Notes in Networks and Systems*, vol 528. Springer, Singapore. https://doi.org/10.1007/978-981-19-5845-8_22.