



Computational Intelligence in Electrical Engineering  
Vol. 14, No. 4, 2024  
pp. 47-60  
Research Paper

## Automatic License Plate Recognition (ALPR) Using Improved Convolutional Neural Network

Fariba Soleimanzadeh Rasteh<sup>1</sup>, Sara Motamed<sup>2\*</sup>

<sup>1</sup> Department of Computer Engineering, Fouman and Shaft Branch, Islamic Azad University, Fouman, Iran

frybaslymanzadh@gmail.com

<sup>2</sup> Department of Computer Engineering, Fouman and Shaft Branch, Islamic Azad University, Fouman, Iran

motamed.sarah@gmail.com

### Abstract :

According to statistics, over half a billion vehicles are moving on the roads worldwide. All vehicles have a license plate as their primary identifier, which is one of the most suitable tools for vehicle identification. Automatic License Plate Recognition (ALPR) can be effective in improving road security, reducing traffic congestion, enhancing transportation efficiency, preventing car theft, and more. Traditional methods proposed for license plate detection mainly relied on manual feature extraction and lacked could not be generalized to variable image components in different conditions. With recent advancements in deep learning, algorithms have emerged that can automatically extract high-level representations of images in addition to learning complex image structures. Therefore, in this paper, the high capacity of deep neural networks is utilized for learning license plate identifiers. The proposed model consists of two stages: license plate localization and plate identification. For localization, a combination of Convolutional Neural Network (CNN) and Generative Adversarial Network (GAN) is used in an encoder-decoder network. The proposed model is evaluated on two datasets, FZU Cars and Stanford Cars, and based on the experimental results, it outperforms baseline methods in terms of accuracy on both datasets.

**Keywords:** Automatic License Plate Recognition, Deep Learning Networks, Convolutional Neural Networks, Generative Adversarial Networks, Encoder-Decoder Architecture.

### 1. Introduction

Due to the increase of cars, the control and

management of traffic resources and vehicles in streets and parking lots, office places, and intercity roads have become a critical issue. The excessive growth of cars has caused problems such as traffic control, highway tolls, parking lot management, etc. The control of this huge flood of cars is beyond the power of today's humans alone and without the use of computer systems [1, 2].

---

<sup>1</sup> Submission date: 09, 11, 2022

Acceptance date: 07, 06, 2023

Corresponding author: Department of Computer Engineering, Fouman and Shaft Branch, Islamic Azad University, Fouman, Iran



This is an open access article under the CC BY-NC-ND/4.0/ License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).



<https://doi.org/10.22108/ISEE.2023.135661.1595>

All vehicles are assigned a vehicle identification number, which serves as their primary identifier. This identification number, also known as a license plate, is a legal requirement for cars, and all vehicles must have this ID to drive. It can be said that the license plate is currently one of the most reliable means of authenticating a car [3]. Therefore, there is a need for systems that can identify license plates by capturing images through cameras placed strategically across the city, at intersections, and on highways. The vehicle license plate recognition system is a mechanized and computerized system that utilizes image processing techniques to identify and read the license plate characters, including letters and numbers, from images taken from cars using surveillance cameras. By implementing this system, many of these issues can be effectively addressed [4, 5].

In general, the accurate recognition of license plates can be enhanced through the effective use of image processing methods. However, various factors such as lighting conditions at different times of the day, weather conditions, and license plate degradation can introduce unwanted effects on the appearance of the license plate in the captured images. These effects may distort the image in a way that makes it difficult to correctly identify the license plate or its ID. Additionally, the camera angle about the horizon may result in a perspective error when viewing license plates from an angled perspective [6].

The aforementioned factors represent environmental challenges in the license plate recognition process. Additionally, variables like the license plate's nationality directly influence its color, design, and the symbols displayed on it. Manipulation and accidents leading to changes in license plate identifiers further complicate the automatic recognition of car license plates. The approach to addressing these issues depends on the chosen method, but the license plate recognition process typically involves three stages:

1. License Plate Localization: The initial objective is to locate the license plate within the image captured by the camera. Various techniques, such as analyzing edge density in potential plate regions, are employed. Additionally, corrective measures can be applied to rectify angular distortions in the license plate.

2. Segmentation and Extraction: In this stage, the image is typically enhanced and binarized. Different image processing methods, such as morphological operators, are used to separate the license plate image into two regions: the identifier area and the background.

3. Character Categorization and Retrieval: Each digit and letter of the license plate are identified individually in this step. Various methods and algorithms are utilized to accurately classify and extract the characters.

It is crucial to note that the accuracy of each step significantly impacts the outcome of the license plate recognition process. Errors resulting from incorrect positioning of the license plate can lead to wasted time and inaccurate calculations in subsequent stages. The zoning method employed should have the capability to effectively separate the license plate's background from the identifiers, irrespective of the license plate's color. It should also perform well under various lighting and environmental conditions. Different lighting conditions can result in certain parts of the image being opaque or excessively dark. Additionally, some methods may be sensitive to rotation angles and require correction for plate deviations before zoning. Given that imaging is often performed on moving vehicles, there is an increased likelihood of image blurring, particularly in the license plate area. Consequently, the zoning methods employed should be efficient in handling the challenge of image blurring [8, 9].

Over time, various research studies have been conducted to read IDs and recognize license plates. One of these methods is perceptron neural networks. However, perceptron networks have limitations in accurately learning complex structures in images. In contrast, deep learning, which encompasses more complex structures, has gained popularity. Convolutional neural networks (CNNs) are one type of deep learning network that has been successfully applied in various applications, including letter and object recognition.

Given the specific challenges present in each camera-captured image, using a single enhancement method is not effective for all images. Different images may suffer from issues such as strong reflections, intense shadows, or blurred IDs due to spots. To address these problems, specific image enhancement methods can be employed to improve the quality of the

image. However, it is important to note that image enhancement methods can often be complex and time-consuming [5, 9, 10]. To mitigate these challenges, utilizing the high capacity of deep neural networks for learning license plates is recommended. By training deep neural networks with license plate images, the task of extracting license plate IDs can be accomplished directly and, in less time, without relying on complex enhancement methods. The objective of this article is to leverage deep learning networks for license plate recognition.

The proposed model in this article consists of two stages: license plate highlighting and ID reading. To achieve this, the model utilizes a combination of a convolutional neural network and a competitive generative network, which is divided into three parts: encoder, feature transformation, and decoder. The encoder layer takes the binarized license plate image and enhances the license plate identifiers in the resulting images. The goal is to create an image of the license plate where the identifiers are black, while the other components of the license plate appear closer to white, representing the background of the image.

The input of the proposed model is the license plate image, and the target images are the binary license plate images that have been labeled by users. The model was evaluated on two datasets, namely FZU Cars and Stanford Cars. The test results demonstrate that the proposed model achieves higher accuracy on both datasets, indicating that the combination of two convolutional neural networks and a competitive generator leads to improved accuracy. The model is categorized accordingly. The remaining sections of the article are organized as follows: The second part provides an overview of previous works. The third part explains the proposed method in detail. The fourth section describes the experiments conducted and presents the results. The fifth section includes conclusions drawn from the research and suggestions for future work.

## 2. Literature Review

Intelligent transportation systems have a long history in image processing research [11-13]. As mentioned, a license plate recognition system typically consists of three stages. The first step aims to identify the regions in the received image from the imaging system that contains the license plate. Various features, such as edge density in the plaque area, are used to locate the presence of the

license plate. Additionally, issues like the angularity of the license plate can be addressed.

In the second step, image brightness problems are usually corrected, and the extracted plaque image is enhanced using methods like morphological operators. The enhanced image can be further processed using binarization techniques to separate it into two regions: The IDs and the background.

The third step involves the recognition of numbers and letters of the alphabet separately. In the following sections, each step of the license plate recognition system will be described in more detail. In the process of locating the license plate, one common method used in many studies is to identify areas with high edge density. Other approaches include using a moving window, leveraging color space information, and extracting texture information using wavelet transformations [14-16]. Some research has also explored learning-based methods. For instance, training cascade classifiers using Haar-like features [17] and employing convolutional neural networks to locate the license plate have been utilized in more recent studies [12, 18].

### 2.1. Upgrading and Segmenting License Plate Identifiers

The license plate, which is separated from the positioning stage, faces various issues such as variable lighting conditions, shadows, contamination, and plate misalignment, resulting in poor quality for identifier recognition. Therefore, it requires upgrading. In reference [19], instead of a global threshold, the license plate image is converted into a binary image using the thresholding method introduced in [20]. Connected regions are then extracted, and excessively large areas are removed at this stage. Finally, the remaining areas are horizontally aligned to correct plate misalignment. In [21], thresholding is performed using both histogram information of illumination intensity and a neural network. The neural network takes a vector as input, which consists of the number of pixels with brightness falling within a specific interval. The global threshold value is calculated as the output. In reference [22], a separate threshold is calculated for each pixel. The threshold value for each pixel at the center of a window is set as a fixed number lower than the average brightness of the window. This helps eliminate the edges connecting two regions.

In [23], adaptive thresholding is utilized. The

method involves calculating a local threshold for each pixel based on the mean and variance in its neighborhood. After thresholding, connected component analysis is performed to extract the region containing the identifiers as a mask. Using this mask, the binary image of the license plate is filtered to remove extra regions caused by binarization errors. In this step, candidate regions for identifiers are extracted using connected components, and the identifiers themselves are obtained by comparing the size of these regions. Due to the significant color difference between the identifiers and the license plate background, the projection profile of the license plate can provide crucial information about the identifiers. In [24], two lines on the plate are separated using the horizontal projection profile. The valley appearing in the projection profile represents the distance between the two lines, allowing the determination of the line separating them.

In [25], the identifiers are separated from each other using the vertical projection profile. Subsequently, the border of each identifier is extracted with the aid of vertical profile information. Binary image projection is employed in the aforementioned studies, but in [26], the gray image of the license plate is used instead.

Research [27], employs a hybrid method to separate identifiers from the background. In this study, an adaptive thresholding technique is applied to binarize the license plate image. The excess areas remaining in the image are then removed using the joinery algorithm. Finally, the identifiers are separated by analyzing and evaluating the vertical projection profile of the binary license plate image. Deep learning has also demonstrated its capability to enhance license plate images. Reference [28] presents a method for binarizing the license plate using a convolutional neural network with an encoder-decoder structure.

## 2.2. Reading Vehicle License Plate Identifiers

Once the license plate identifiers are segmented, conventional optical character recognition methods can be employed for their recognition. Due to variations in the distance between the imaging system and the license plate, the license plate image may suffer from distortions, resulting in perspective effects. As a result, the separated identifiers may appear in different sizes.

Also, when separating identifiers, there is a

possibility that they break or connect with other background components. The methods used for reading license plate IDs must be capable of recognizing the identifiers despite these challenges. Since identifiers within license plates of the same nationality often have similar shapes, categorizing each identifier by matching its pattern with predetermined patterns is possible. However, the rotation and deviation of the license plate can alter the shape of the identifiers. In [29], this issue is addressed by considering different patterns of the same identifier with varying rotation angles. To match isolated identifiers with pattern images, a similarity measure needs to be applied. The Mahalanobis distance and cross-correlation are among the criteria utilized in the studies [30] and [31].

Different classification methods such as the support vector machine, artificial neural networks, and hidden Markov models can be utilized for reading the identifiers separated from the license plate. Generally, classification and machine learning methods require feature extraction from license plate identifiers for training and classification purposes. In reference [21], the skeleton of the identifiers obtained using morphological operators is used for feature extraction. The window containing the license plate ID skeleton is divided into nine areas, and the angle of the skeleton parts is extracted as a feature from each area. Finally, an artificial neural network recognizes the ID using these features.

Perceptron neural networks have been employed as a method to read isolated identifiers from license plate images in various studies, such as [27]. This classification method also necessitates the extraction of appropriate features from license plate identifiers. In [32], the contour curve of the identifiers is used as a feature independent of their shape and size. Additionally, the Gabor filter is employed as a feature extraction method for identifier categorization. In research [33], the brightness intensity vector of normalized license plate identifiers is used as a feature in two categories: nearest neighbor and support vector machine.

In [34], identifiers are recognized using the geometric characteristics of the identifiers and the distribution of brightness intensity. In [35], the gray image of the identifiers is used as input to train a neural network. Reading license plate IDs, in addition to being considered a final goal in

research, can also serve as a confirmation method for locating license plate detection. In [36], the number of IDs detected by a neural network is used as feedback for the license plate location method. In [37], the YOLO network is employed for the automatic recognition of car license plates. In another study [38], character segmentation is performed to extract the license plate region from an image using the R-CNN method. In this study, optical character recognition is used to accurately identify the characters, and the collected data is compared with relevant reference databases to retrieve specific information such as the car owner, registration location, address, etc.

### 2.3. Learning Methods Based on Neural Network

The perceptron neural network is one of the earliest machine learning architectures, consisting of multiple layers, with each layer comprising several learning units or neurons. Structurally, this network can be used as a comprehensive model to fit data, but as the complexity of the data increases, a more sophisticated model is required. Issues in training these networks persisted until 2006 when they gradually began to be resolved [7, 12]. However, deep learning had been explored in some research before that. These methods were rooted in the concept of the visual structure found in living organisms. The Neocognitron network was the first neural network created based on this concept. Due to its shared-weight region, which functions similarly to a joint mathematical operator, it is also referred to as a convolutional neural network [18].

The convolutional neural network can be considered one of the earliest deep learning methods that achieved significant results in machine learning even before 2006. The Neocognitron convolutional neural network bears a strong resemblance to modern neural networks used in image processing research today [39]. In general, convolutional neural networks consist of three layers with different functions. The initial layer in these networks is the convergence layer, comprising multiple central cores or filters. Each filter possesses shared weights, enabling it to detect specific features in various locations within the input image. During the network's training process, the weights of each filter are adjusted until the learned combination can extract important and informative details from the input image. Consequently, there is no need to explicitly extract image features in these

networks. These filters can be employed in each layer, alongside pooling, similar to the merging operator. In data such as images, which may contain uniform and identical areas, one point can serve as a representative for other points. This sampling layer is referred to as the pooling layer in convolutional neural networks. Moreover, in some studies, the pooling layer has been utilized to increase the dimensionality of the data. Both of these layers play vital roles in feature extraction from the input data. The output of each layer is referred to as a feature map. Finally, to carry out the final processes of the neural network, such as classification, fully connected layers, which are similar to perceptron neural networks, are employed [40].

One of the domains where deep learning has achieved success is encoder-decoder structures, or more broadly, encoders themselves. A self-encoding network initially maps the input data to the feature space and then reconstructs this feature space back to the original space in two steps. The training requirement for this network is the reversibility of the data in the output of the decoder, ensuring that the input information is not lost in the encoder's output. Depending on the training type and approach, the encoder can extract information and features that are also suitable for other machine-learning tasks [41].

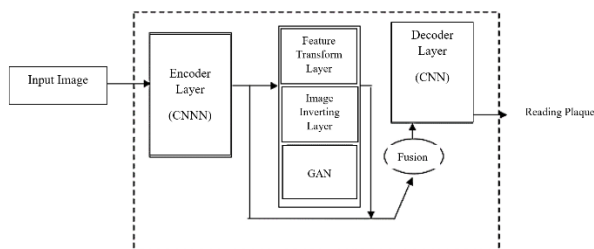
Another widely used type of neural networks is a recurrent network, which can process various types of sequences. In this network, unlike most neural networks where each learning unit is solely connected to neurons in the next layer, each neuron can also be connected to units within the same layer. One popular variant of this network is the long short-term memory network [42].

### 3. Proposed Model

The proposed model in this article consists of two stages: license plate ID highlighting and ID reading. The highlighting of license plate identifiers is achieved through a combination of Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) with an encoder-decoder structure called CNGA. Additionally, a recurrent neural network (RNN) is employed for reading the license plate. Given the diversity and variations in the database images, this research utilizes deep learning methods, which are more effective, compared to conventional plaque zoning approaches. Simple machine learning architectures like perceptron neural networks cannot learn and segment license

plate IDs for the following reasons: If a simple perceptron neural network is used as a self-encoder, the entire license plate image needs to be fed into its input. This would require a large number of learning units in the input layer, resulting in suboptimal computational efficiency. The perceptron neural network structure only considers a sequence of light intensities, disregarding the semantic relationship between adjacent pixels and the visual appearance and structures of identifiers. Given that license plate images are typically extracted from a license plate recognition system and can exhibit slight angular deviations, displacements within the extracted window, and appearance issues, a simple perceptron neural network lacks the necessary capability to segment the license plate accurately. It is worth noting that the perceptron network structure for plaque zoning should also be deep and incorporate an encoder and a decoder to effectively segment the plaque. Consequently, the proposed model employs a convolutional neural network within an encoder-decoder structure. Figure 1 illustrates the architecture of the proposed model, and its details are described below.

The proposed model is capable of learning binary representations of license plates under various conditions and subsequently highlighting the license plate identifiers in new images. The objective of highlighting the vehicle IDs is to generate an image of the license plate where the license plate IDs appear black, while the other components of the license plate fade towards white. The input images for the proposed system are license plate images, while the target images for training are binary license plate images that have been preprocessed by the user.



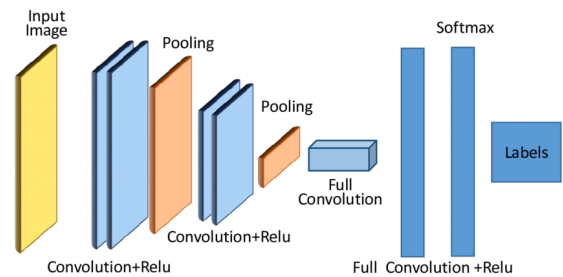
**Fig. 1. General schematic of the proposed model**

Since different lighting conditions can significantly alter the color of the license plate, color-based features cannot be reliably utilized.

Moreover, extracting features from color plates necessitates a network with a larger number of learning units. However, increasing the input volume of the neural network reduces computational speed and increases network inference time. Therefore, to optimize resource usage, the input image is initially converted to grayscale and then rescaled. The rescaling amounts for the length and width of the images are set to 398 and 80, respectively.

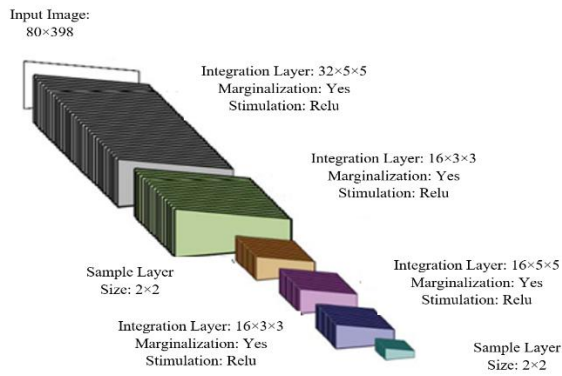
**3.1. Encoder Layer**

License plate identifiers generally come in two colors: white and black. However, there are more license plates with black identifiers than those with white identifiers, which can pose a challenge for the network to learn license plates with white identifiers effectively. To address this issue, the plates with white identifiers are inverted before training. In this process, a convolutional neural network is employed for encoding. Its purpose is to convert the initial grayscale license plate image into a binary license plate image. Convolutional networks consist of one or more convolutional layers, where each layer functions as a local filter. Figure 2 provides an example of a convolutional network.



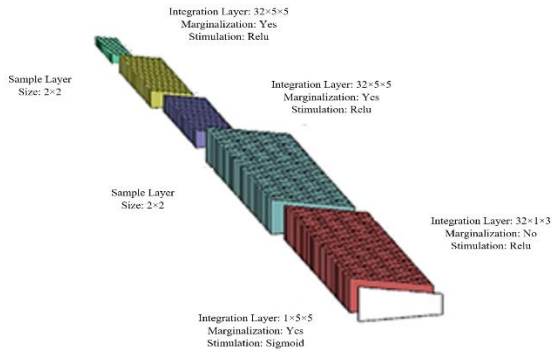
**Fig. 2. Convolutional neural network structure**

In the provided figure, the convolutional layer is shown to function as a local filter on three adjacent inputs. Typically, a convolutional network is accompanied by an integration layer. The role of this layer is to extract crucial information from the output obtained by the convolutional layer. The integration layer is designed to be capable of capturing a specific feature regardless of its exact location within the input. Furthermore, the details of the proposed encryption network are outlined in Figure 3.



**Fig. 3. Schematic of the proposed encoder**

To train the network, the training decoder structure illustrated in Figure 4 is utilized. This structure comprises two interconnected networks, with the gray-scale image of the license plate being fed into the encoder input. The desired output of the training decoder is the binary image of the license plate.



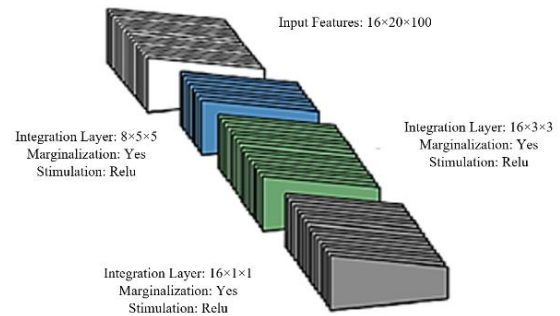
**Fig. 4. The proposed decoder network structure for encoder training**

However, it should be noted that the proposed encoder network, when trained using the training decoder structure, cannot effectively highlight license plate identifiers in white. One common solution to address this issue is to invert the license plate image and zone the license plate, which introduces its own set of image-related challenges. Therefore, several different solutions were explored for the zoning of license plate identifiers. Among these solutions, the most effective one in terms of time and computational cost involves adding an intermediate network after the encoder. This intermediate network estimates the features related to the inversion of the original image using the information available in the encoder. Consequently, both categories of features, related to the original image and its inversion, are available in the input of the

decoder. As a result, the decoder can generate the zoned image in the output without the need to detect the color or invert the image in the encoder's input. This intermediate network, which transforms the features related to the input image (obtained at the output of the encoder) into the features related to the image itself, is referred to as a feature converter.

### 3.2. Feature Transform Layer

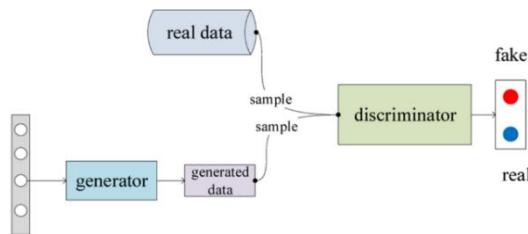
The structure of the proposed decoder cannot generate a suitable output image of the license plate with white identifiers. Therefore, the network structure can be modified to incorporate characteristics of both the input image and its inverse in the decoder input. However, it is important to note that this modification requires repeating the feature extraction process in the encoder, which can be time-consuming. Hence, a method that incorporates features related to the inverted image in the encoder output is necessary. This paper proposes a solution by introducing a small network that transforms the encoder output features into features related to the inversion of the original image. This network is referred to as the "feature transform network" in the paper. Since the proposed network is designed to be small for feature transformation purposes, it offers an optimal solution in terms of time. The structure of this layer is depicted in Figure 5.



**Fig. 5. Feature transform layer**

Since the feature transform network is applied to all input images, there is no need to classify or separate the license plates before the encoder or feature transform input. This means that the same system can be used to identify all types of license plates. The feature converter is responsible for transforming the features extracted from the input image into features obtained from the inverse of the image. To train this network, features extracted from images with white identifiers are

used as the input training data. The target data for training are the features extracted by the encoder for the inverse of those images. In the feature transformation layer, a competitive generative neural network is employed to generate new data to increase the amount of training data. This network consists of two parts: the generating part and the discriminating part. The generating part of the competitive generative network adds noise to the existing images and creates new images for training purposes. Subsequently, the discriminator compares the generated images with the original training images. If the generated images do not match the original images, the discriminator rejects them; otherwise, artificial images are generated. The structure of the competitive generator network is illustrated in Figure 6.



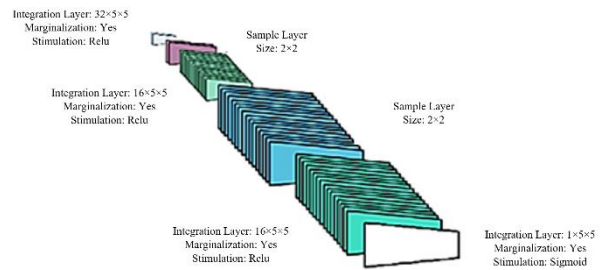
**Fig. 6. Structure of Competitive Generative Network (GAN)**

In this method, the network learns to generate new data from the training data, aiming to create statistically similar output data. The responsibility of generating the output lies with the producer, while the discriminator's task is to assess the similarity between the generated data and the training data. The process can be likened to a game, where the discriminator's goal is to identify any differences between the main input of the network and the output produced by the producer. If the discriminator successfully detects dissimilarities, it wins the game, and the producer must improve the output to the point where the generating part can deceive the discriminating part, resulting in the game ending. Trained generative networks can generate new images that are visually correct and incorporate many features of the training data. For example, imagine the task of generating high-resolution images from low-resolution images, where the generated images not only have larger dimensions and higher quality but also precisely match the input image. In such a scenario, the competitive generative network proves to be beneficial, as it can compensate for

the lack of training data and generate high-quality images.

**3.3. Decoder Layer**

The purpose of employing a feature converter is to ensure that both the features of the original images and their inverses are available as inputs to the decoder. Consequently, the architecture of the proposed decoder needs to be designed in a way that it can accommodate both sets of features as inputs. The final structure of the decoder, as depicted in Figure 7, reflects this design. In the proposed decoder, the input features consist of the summation of the features extracted from an image and its inverse. This approach enables the decoder to operate independently of the color of the license plate IDs while effectively highlighting them. By utilizing this method, the decoder can successfully emphasize the license plate IDs regardless of the input color.



**Fig. 7. Proposed structure for creating a plate with a prominent ID**

The decoder in this proposed model has twice the number of input features compared to the training decoder. Additionally, the input features are merged within the decoder's second layer. This not only helps enhance the process of highlighting the license plate IDs but also contributes to its speed improvement.

It is important to address the issue of overfitting in deep neural networks. In the proposed model, several measures have been implemented to mitigate overfitting:

1- Sufficient training examples: Given the large number of learning connections in a deep neural network, an ample number of sample images have been used in the training process. Moreover, the number of examples has been increased by introducing noise and reducing the image quality.

2- Soft weight adjustment clause: In addition

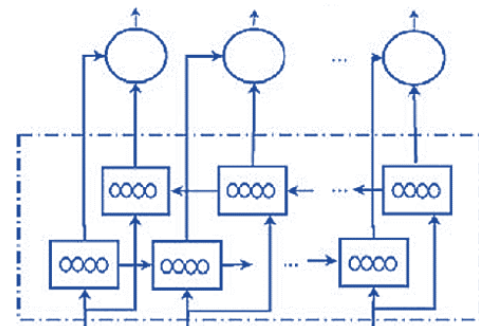


to the loss function used for training the network, a soft weight adjustment clause has been employed. This mechanism helps regulate the weights in the network, preventing overfitting.

**Dropout technique:** The dropout technique has been utilized to reduce overfitting. This technique randomly deactivates a portion of learning units in a layer during training, ensuring that only other units participate in that particular stage of training. This helps prevent the network from relying too heavily on specific units and encourages more robust learning.

### 3.4. Reading the Embossed License Plate IDs

In the article, a recurrent neural network (RNN) is employed to read the license plate IDs from an image where the IDs have been highlighted. The advantage of using an RNN is that it eliminates the need to separate the identifiers from each other during both the training and reading phases. The RNN can interpret the output image from the decoder as a sequence and produce the license plate number as the output. Figure 8 illustrates a simple structure of the recurrent neural network used for reading license plate IDs. Recurrent neural networks are commonly applied in tasks such as optical character recognition. However, it is important to ensure that the input image contains only a sequence of letters and numbers for the network to accurately read the text. Therefore, in the proposed model, before using the recurrent neural network to read the image identifiers, any prominent background components and the size of the background are removed. This ensures that the input to the recurrent neural network consists of a sequence of identifiers, with dashes indicating regions that are empty of identifiers. The working mechanism of the recurrent neural network involves feeding the input image to the network as a sequence. The information from each location on the input image is stored in the memory of the network's learning units. Finally, the input image is labeled as a sequence, which represents a combination of identifiers, with dashes denoting empty regions. The recurrent neural network then processes this sequence to produce the license plate number as the output.



**Fig. 8. Recurrent neural network for reading license plate IDs**

In the proposed approach, the last layer of the recurrent neural network is utilized, and the output is combined and connected to generate a single sequence of license plate identifiers. This method differs from common techniques where the license plate image is first binarized and then the identifiers are segmented for reading. By using a recurrent neural network for reading license plate IDs, the approach becomes more effective in handling certain challenges related to license plate segmentation. For instance, during the zoning process, some background components might remain on the license plate image, or the IDs themselves may be fragmented into multiple parts. The advantage of employing a recurrent neural network is that it treats the license plate ID image as a sequence. This allows the network to better handle these issues and perform more robustly when faced with such problems. Compared to traditional methods that rely on binarization and zoning, the recurrent neural network-based approach offers enhanced adaptability and resilience in dealing with common challenges encountered during license plate reading.

## 4. Experimental Results

In this section, the data set, evaluation criteria, hardware and software requirements are introduced first, and then the tests will be described.

### 4.1. Dataset

In the article, two standard datasets commonly used in the field of license plate recognition are employed for testing purposes. The first dataset is the FZU Cars dataset, which comprises 297 car models and a total of 43,615 images. The second dataset is the Stanford Cars dataset, consisting of 196 car models and a total of 16,185 images<sup>1</sup>. Since these datasets do not come with pre-defined training and test sets, the article adopts a mutual validation method for evaluation. This method

involves performing cross-validation, which assesses the generalizability and independence of the statistical analysis results concerning the training data. To implement this method, the entire dataset is randomly divided into five parts. Three parts are utilized as the training set, while the remaining two parts are used as the validation and test sets. It's important to note that the reported results in the article are based on the average performance across all five divisions of the dataset.

#### 4.2. Evaluation Criteria

To evaluate the proposed model, the evaluation criteria of accuracy, correctness, and recall are used and they are calculated according to the (1), (2), (3) and (4):

$$Accuracy = (TP + TN) / N \quad (1)$$

$$precision = TP / (TP + FP) \quad (2)$$

$$recall = TP / (TP + FN) \quad (3)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

In the above relationships, TP and TN are respectively positive and negative examples that are correctly classified. FP and FN are the misclassified positive and negative samples, respectively, and N is the total number of samples.

#### 4.3. System Requirement and Training Parameters

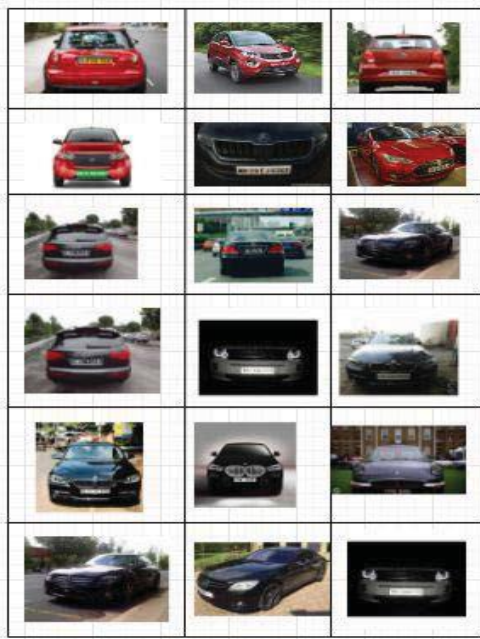
The implementation of the program in this article relies on deep learning, which involves processing and calculating information from millions of data points. Consequently, a regular processor is insufficient for this task, necessitating the use of powerful hardware with higher speed capabilities. In this regard, the implementation requires hardware that meets the demanding requirements of the process.

For the implementation of the proposed method, the Python programming language has been employed. Python provides a convenient environment for designing and implementing machine learning and deep learning algorithms. To utilize the Python programming environment effectively, the article utilizes Anaconda, a free and open-source distribution of Python and R programming languages specifically designed for scientific computing, including applications in data science, machine learning, large-scale data processing, and predictive analytics. Anaconda

simplifies package management and deployment, making it easier for researchers and developers to work with Python in these domains. It is worth noting that Anaconda is widely used, with over 15 million users, and includes more than 1,500 popular data science packages suitable for Windows, Linux, and MacOS platforms.

In summary, all the implementations described in the article have been carried out using Python 3 and the TensorFlow 0,1, 2 library. The hardware utilized for the experiments consisted of a system equipped with an Intel Xeon 2 E5-2620 2.0 GHz processor and 8 GB of RAM, operating in a Linux environment. This hardware configuration was chosen to meet the computational demands of the deep learning tasks involved in the proposed method. In the proposed model, a competitive generative neural network is utilized to generate new images. Figure 9 in the article showcases some examples of the images produced in this step.

To implement the CNN, in the proposed model, filters with sizes of 16, 32, and 64 are employed, and the number of filters is set to 150. The non-linear Rectified Linear Unit (ReLU) function is used as the activation function in this network. Additionally, a max-pooling function with a size of (2x2) is applied. For training the model, the ADADELTA weight update rule is used with a learning rate of 0,01. The ADADELTA algorithm is an adaptive learning rate optimization method that aims to mitigate the need for manual tuning of the learning rate. A dropout rate of 0.05 is also implemented, which helps prevent overfitting by randomly deactivating a fraction of the learning units during training. It's important to note that the proposed model is trained using 100 iterations per epoch (IPECs).



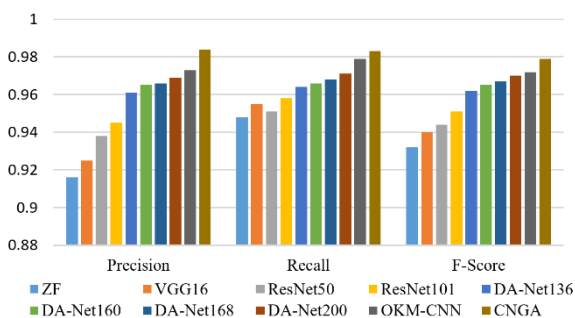
**Fig. 9. Results produced by competitive generative neural network**

#### 4.4. Implementation Results

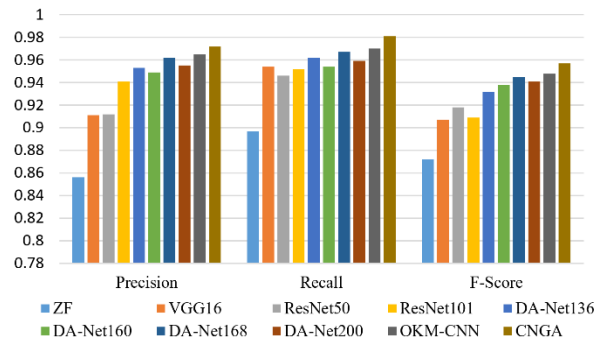
In this section, the article focuses on evaluating the efficiency of the proposed model compared to other traditional models. The process involves preprocessing, feature extraction, and modeling of the data. The goal of the experiments conducted in this section is to answer two fundamental questions:

Question 1: Does the proposed method exhibit higher prediction accuracy than existing methods?  
 Question 2: Can the combination of a competitive generative neural network and a convolutional neural network enhance the accuracy of car license plate classification?

The process of license plate recognition using the proposed model is illustrated in Figure 1. To answer these questions, the proposed model is compared with a set of previous models, and the results are presented in Table 1, as well as Figures 10 and 11.



**Fig. 10. The results of implementing the proposed model on the FZU Cars dataset**



**Fig. 11. The results of implementing the proposed model on the Stanford Cars dataset**

The implementation results of the proposed model demonstrate that it achieves higher accuracy on both datasets. Evaluating the FZU Cars dataset, the proposed model achieves an accuracy, recall, and F-score of 0,984, 0,983, and 0,979, respectively, outperforming previous models in terms of accuracy. On the Stanford Cars dataset, the proposed model achieves an accuracy, recall, and F-score of 0,972, 0,981, and 0,957, respectively. Comparing these results with previous models, the proposed model demonstrates higher accuracy on the Stanford Cars dataset as well.

#### 5. Conclusion

Automatic license plate recognition (ALPR) systems play a crucial role in car identification. These systems utilize image processing techniques to extract license plate information from passing vehicles, eliminating the need for additional devices like GPS or radio tags. Special cameras capture images of vehicles, which are then processed by ALPR software on a computer. Implementing ALPR systems can provide several benefits for municipalities, traffic police, and relevant authorities. They can make informed decisions to improve traffic flow and enhance road safety without relying on manual interventions. Additionally, these systems enable quick and automatic identification of suspicious vehicles, aiding in the prevention of driving crimes and enhancing overall road security. Recognizing the significance of ALPR systems, the article introduces an improved neural network-based model for automatic license plate recognition. The proposed model, called improved convolutional neural network (CNGA), consists of two steps: license plate ID highlighting and ID reading. The model utilizes a convolutional neural network with an encoder-decoder structure to independently highlight license plate identifiers,

regardless of their color (white or black). However, the decoder component lacks the ability to create a suitable image of the license plate with white identifiers. To address this, the input of the encoder is modified to incorporate both modes of the input image. A feature transform layer, implemented using a small network, converts the encoder's output features to those related to the inversion of the original image. It is expected that the combination of the encoder-decoder layer and the feature transform layer will significantly enhance the accuracy of license plate recognition

systems. The proposed model is evaluated using two datasets, namely FZU Cars and Stanford Car. The experimental results demonstrate that the proposed model outperforms previous methods in terms of accuracy on both datasets. In future research, the article suggests exploring the application of non-linear functions with random characteristics, unsupervised learning approaches, and random integration to improve the generalizability and accuracy of deep learning models in license plate recognition systems.

**Table 1. The results of the tests on the tested data set**

Models	FZU Cars Dataset			Stanford Cars Dataset		
	Precision	Recall	F-Score	Precision	Recall	F-Score
ZF	0,916	0,948	0,932	0,856	0,897	0,872
VGG16	0,925	0,955	0,940	0,911	0,954	0,907
ResNet50	0,938	0,951	0,944	0,912	0,946	0,918
ResNet101	0,945	0,958	0,951	0,941	0,952	0,909
DA-Net136	0,961	0,964	0,962	0,953	0,962	0,932
DA-Net160	0,965	0,966	0,965	0,949	0,954	0,938
DA-Net168	0,966	0,968	0,967	0,962	0,967	0,945
DA-Net200	0,969	0,971	0,970	0,955	0,959	0,941
OKM-CNN	0,973	0,979	0,972	0,965	0,970	0,948
<b>CNGA</b>	<b>0,984</b>	<b>0,983</b>	<b>0,979</b>	<b>0,972</b>	<b>0,981</b>	<b>0,957</b>

## References

- [1] Huang C H, Sun Y, and Fuh C S. Vehicle License Plate Recognition With Deep Learning. *Technologies to Advance Automation in Forensic Science and Criminal Investigation 2022*, IGI Global: 161-219.
- [2] Abdellatif M M et al. A low cost IoT-based Arabic license plate recognition model for smart parking systems. *Ain Shams Engineering Journal 2023*; 14(6): 102-178.
- [3] Antar R, et al. Automatic number plate recognition of Saudi license car plates. *Engineering. Technology & Applied Science Research 2022*; 12(2): 8266-8272.
- [4] Pirgazi J, Pourhashem Kallehbasti M M, Ghanbari Sorkhi A. An End-to-End Deep Learning Approach for Plate Recognition in Intelligent Transportation Systems. *Wireless Communications and Mobile Computing 2022*.
- [5] Weihong W, Jiaoyang T. Research on license plate recognition algorithms based on deep learning in complex environment. *IEEE Access 2020*; 8: 91661-91675.
- [6] Zhang Y, Wang Z, Zhuang J. Efficient license plate recognition via holistic position attention. in *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021.
- [7] Mufti N, Shah SAA. Automatic number plate Recognition: A detailed survey of relevant algorithms. *Sensors 2021*; 21(9).
- [8] Balia R, et al. A Deep Learning Solution for Integrated Traffic Control Through Automatic License Plate Recognition. *International Conference on Computational Science and Its Applications 2021*. Springer.
- [9] Vaiyapuri T, et al. Automatic vehicle license plate recognition using optimal deep learning model. *Computers, Materials and Continua 2021*; 67(2): 1881-1897.
- [10] Gnanaprakash V, Kanthimathi N, Saranya N. Automatic number plate recognition using deep learning. in *IOP Conference Series: Materials Science and Engineering 2021*. IOP Publishing.
- [11] Mohamed A M, Hazeem B. Automatic License Plate Recognition Based on

- Surveillance Videos. *Journal of Optoelectronics Laser* 2022; 12: 403-406.
- [12] Kashyap A, et al. Automatic number plate recognition. *International conference on advances in computing, communication control and networking (ICACCCN)* 2018. IEEE.
- [13] Deb K, et al. Optical Recognition of Vehicle license plates. *Proceedings of 2011 6th International Forum on Strategic Technology* 2011. IEEE.
- [14] Wang F, et al. Fuzzy-based algorithm for color recognition of license plates. *Pattern Recognition Letters* 2008; 29(7): 1007-1020.
- [15] Yu S, et al. A novel license plate location method based on wavelet transform and EMD analysis. *Pattern Recognition* 2015; 48(1): 114-125.
- [16] Rashedi E, Nezamabadi-Pour H. A hierarchical algorithm for vehicle license plate localization. *Multimedia Tools and Applications* 2018; 77(2): 2771-2790.
- [17] Ibitoye O, et al. Convolutional neural network-based license plate recognition techniques: a short overview. *International Conference on Computational Science and Computational Intelligence (CSCI)* 2020. IEEE.
- [18] Chang S L, et al. Automatic license plate recognition. *IEEE transactions on intelligent transportation systems* 2004; 5(1): 42-53.
- [19] Nakagawa Y, Rosenfeld A. Some experiments on variable thresholding. *Pattern recognition* 1979; 11(3): 191-204.
- [20] Nukano T, Fukumi M, Khalid M. Vehicle license plate character recognition by neural networks. in *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems* 2004. IEEE.
- [21] Llorens D, et al. Car license plates extraction and recognition based on connected components analysis and HMM decoding. in *Iberian conference on pattern recognition and image analysis* 2005. Springer.
- [22] Giannoukos I, et al. Operator context scanning to support high segmentation rates for real time license plate recognition. *Pattern Recognition* 2010; 43(11): 3866-3878.
- [23] Duan T D, et al. Building an automatic vehicle license plate recognition system. *International Conference of Computing Science* 2005.
- [24] Shan B. Vehicle License Plate Recognition Based on Text-line Construction and Multilevel RBF Neural Network. *Journal Computing* 2011; 6(2): 246-253.
- [25] Rahman C A, Badawy W, Radmanesh A. A real time vehicle's license plate recognition system. *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance* 2003. IEEE.
- [26] Nomura S, et al. A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition* 2005; 38(11): 1961-1975.
- [27] Rakhshani S, Rashidi A, Nezamabadipour H. Binarization method of car license plate using deep learning with encoder-decoder neural network structure. *10th Iran Machine Vision and Image Processing Conference* 2016.
- [28] Naito T, et al. Robust license-plate recognition method for passing vehicles under outside environment. *IEEE transactions on vehicular technology* 2000; 49(6): 2309-2319.
- [29] Xiaobo L, Xiaojing L, Wei H. Vehicle license plate character recognition. in *International Conference on Neural Networks and Signal Processing* 2003. IEEE.
- [30] Miyamoto K, et al. Vehicle license-plate recognition by image analysis. *Proceedings IECON'91: 1991 International Conference on Industrial Electronics, Control and Instrumentation* 1991. IEEE.
- [31] Kim M K, Kwon Y B. Multi-font and multi-size character recognition based on the sampling and quantization of an unwrapped contour. in *Proceedings of 13th International Conference on Pattern Recognition* 1996. IEEE.
- [32] Hu P, et al. Recognition of gray character using gabor filters. in *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002*.(IEEE Cat. No. 02EX5997).
- [33] Ko M A, Kim Y M. A simple OCR method from strong perspective view. in *33rd Applied Imagery Pattern Recognition Workshop (AIPR'04)* 2004. IEEE.
- [34] Jiao J, Ye Q, Huang Q. A configurable method for multi-style license plate recognition. *Pattern Recognition* 2009; 42(3): 358-369.
- [35] Khosravi H. A sliding and classifying approach towards real time Persian license plate recognition. *International Journal of Engineering, Transactions A: Basics* 2015; 28(1): 74-80.
- [36] Ke X, Zeng G, Guo W. An Ultra-Fast

- Automatic License Plate Recognition Approach for Unconstrained Scenarios. IEEE Transactions on Intelligent Transportation Systems 2023.
- [37] Pujar A, Kulkarni P B. Automatic Number Plate Recognition Using Machine Learning. in International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022) 2023. Atlantis Press.
- [38] LeCun Y, et al. Learning algorithms for classification: A comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective 1995; 261(276).
- [39] Kendall A, Badrinarayanan V, Cipolla R. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv 2015.
- [40] Vincent P, et al. Extracting and composing robust features with denoising autoencoders. in Proceedings of the 25th international conference on Machine learning 2008.
- [41] Zou Y, et al. A robust license plate recognition model based on bi-LSTM. IEEE Access 2020; 8: 211630-211641.
- [42] Wani M A, Bhat F A, Afzal S, Khan A I. Advances in deep learning. Springer 2020.

---

<sup>i</sup> <https://www.kaggle.com/jessicali9530/stanford-cars-dataset>