

Unleashing the power of Convolutional Neural Networks in license plate recognition and beyond

Maolin Wang

Logistics Engineering College, Shanghai Maritime University, Shanghai, 201306,
China

202110216054@stu.shmtu.edu.cn

Abstract. This article explores the application of Convolutional Neural Networks (CNN) in the field of license plate recognition. It begins by introducing the architecture of CNN, which consists of three key layers: Convolutional Layers, Pooling Layers, and Fully Connected Layers. The article then references three relevant papers that demonstrate how CNNs are applied in license plate recognition. The first paper utilizes TensorFlow to construct a CNN model and integrates it with an STM32MP157 embedded chip for license plate recognition. The second paper presents a real-time car license plate detection and recognition method called Multi-Task Light CNN, emphasizing robustness. The third paper employs the ResNet+FPN feature extraction network of the Mask R-CNN model and annotates a license plate dataset. The article highlights the promising future of CNNs in various fields beyond license plate recognition, emphasizing their potential for further development and industrial applications. CNNs have proven to be versatile and powerful tools in computer vision, offering solutions to a wide range of problems. Their adaptability and effectiveness make them a key player in the ongoing advancement of artificial intelligence and automation technologies.

Keywords: CNN, Deep Learning, License Plate Recognition.

1. Introduction

With the development of society and technology, various machines have been applied in various aspects of our lives, and people are constantly exploring the potential of machines to better serve humanity. Over the past decade, deep learning artificial intelligence has seen rapid growth and widespread applications in fields such as healthcare, autonomous driving, natural language processing, and more. Among the most classic and commonly used algorithms in artificial intelligence is the CNN [1].

This article delves into its utilization within the realm of license plate recognition. In the second chapter, we predominantly delve into CNN's architecture, encompassing three distinct layers. The Convolutional Layer assumes the crucial role of feature extraction, employing a set of trainable filters (kernels) to convolve across input data. This process captures various attributes such as edges, textures, and patterns. Subsequently, the Pooling Layers step in to reduce the spatial dimensions of the feature maps produced by the convolutional layer. Commonly utilized pooling techniques encompass max pooling and average pooling. Lastly, we encounter the Fully Connected Layer, which typically comprises interconnected neurons. This layer leverages the high-level features gleaned from preceding layers and amalgamates them to facilitate predictions or classifications. In image classification tasks,

the final layer furnishes probabilities pertaining to different classes. Chapter three makes reference to a trio of articles. The initial article employs TensorFlow for constructing a CNN model and integrates an STM32MP157 embedded chip for creating peripheral circuits, ultimately achieving license plate recognition. The second paper introduces a robust real-time method for car license plate detection and recognition, denoted as Multi-Task Light CNN for License Plate Recognition. The third paper scrutinizes the feature extraction network, ResNet+FPN, within the context of the Mask R-CNN model, and meticulously annotates a license plate dataset via the VIA software.

This research will contribute to the further development of CNNs in the field of license plate recognition, facilitating their mechanization and industrial applications.

2. Theoretical Foundation-CNN Algorithm

2.1. Definition and Structure

A Convolutional Neural Network, commonly referred to as CNN, is a deep learning model designed for processing data that possesses a grid-like structure. CNNs excel at handling data like images and videos and draw inspiration from the biological visual cortex, seeking to replicate the workings of the human visual system. A CNN is primarily composed of three distinct layer types: Convolutional Layers, Pooling Layers, and Fully Connected Layers.

Convolutional Layers serve as the cornerstone of CNNs, tasked with extracting features from input data through convolution operations. This process entails using small matrices known as convolutional kernels (or filters) to perform computations at various positions within the input data, effectively capturing local features [2]. By systematically sliding these convolutional kernels across the entire input and conducting the convolution operation, a series of two-dimensional feature maps emerge.

Pooling Layers are instrumental in reducing the spatial dimensions of the feature maps, thereby trimming network parameters while preserving crucial features. A frequently employed pooling technique is Max Pooling, which selects the maximum value within each region to generate the pooled outcome.

Fully Connected Layers facilitate the propagation of information by establishing connections between all neurons from the preceding layer to each neuron in the current layer. The primary objective of these fully connected layers is to transform the feature maps extracted by the convolutional and pooling layers into a format that is conducive to classification or regression tasks [3].

CNNs have attained remarkable success in various computer vision tasks, including but not limited to image classification, object detection, semantic segmentation, and face recognition. Through multi-level feature extraction and abstraction, CNNs adeptly capture both local and global information within images, resulting in more precise predictions and improved comprehension. For reference, Figure 1 depicts the structure of a CNN.

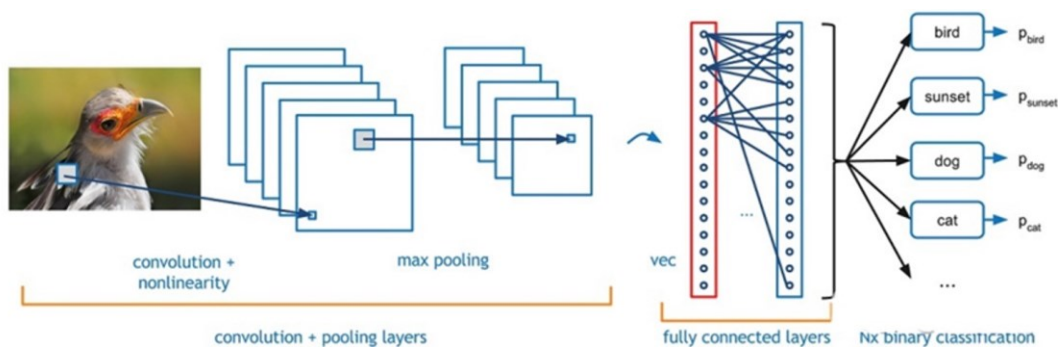


Figure 1. CNN structure.

2.2. Convolution

Certainly, convolution stands as a fundamental element within CNNs. It entails the process of convolving a filter, also referred to as a kernel, with the input image to execute convolutional operations and extract essential features. The convolution operation unfolds by sliding a filter window across the image and subsequently computing the dot product between the filter and the corresponding portion of the input data. The outcome of this computation at each position contributes to the formation of a feature map. Importantly, the convolution operation operates within local receptive fields, effectively capturing local features embedded within the image. This localized approach allows CNNs to identify and highlight distinctive features in different regions of the input, enhancing their ability to discern intricate patterns and structures.

The convolutional layer in CNN is a fundamental component responsible for convolving the input data and extracting features. In a convolutional layer, the input data is typically a multi-channel 2D image or a multi-dimensional feature map. The layer consists of multiple convolutional kernels (or filters), each of which can be seen as a small weight matrix [4]. These filters slide over the input data using a windowing approach.

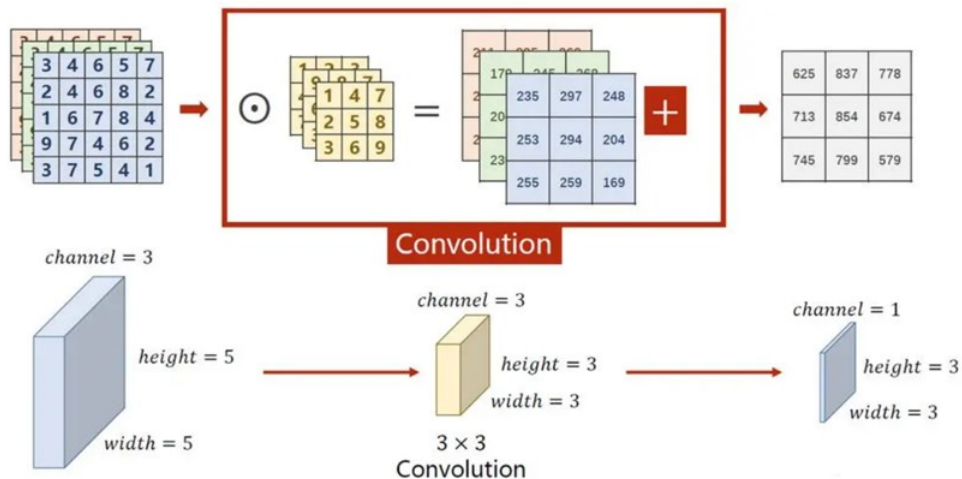


Figure 2. Convolution layer [4].

2.3. Pooling

Pooling layers play a crucial role in diminishing the spatial dimensions of feature maps, lowering parameter count, and retaining the stability of the features extracted. Typically, two widely-used pooling techniques are employed: max pooling, which chooses the highest value in a window as the output, and average pooling, which calculates the average value within the window. These pooling operations are instrumental in reducing the feature map size, which, in turn, enhances computational efficiency. Overall, pooling layers represent another vital element within CNNs, effectively reducing feature map dimensions and consequently simplifying the computational demands of the model [5]. In a pooling layer, the input is typically the feature maps from convolutional layers. The pooling operation involves using a sliding window to perform calculations on the input feature map and outputting a down sampled pooled feature map with reduced spatial dimensions.

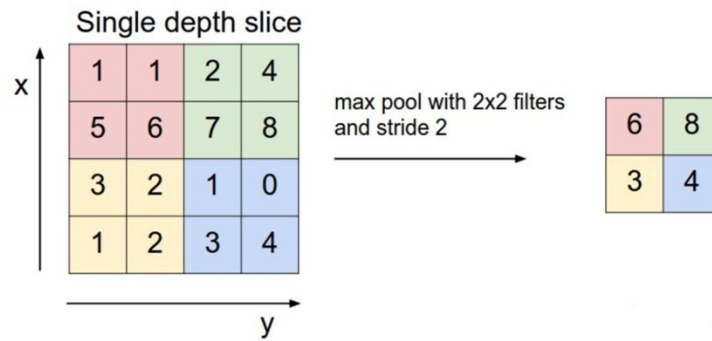


Figure 3. Pooling layer [5].

2.4. Fully Connected

Fully connected layers serve the purpose of classifying or performing regression on the features extracted through convolutional and pooling operations. These layers flatten the feature maps into a vector and execute a sequence of fully connected operations to make predictions. Activation functions, such as softmax or sigmoid, are applied to convert the outputs into either probability distributions or real-valued predictions. Situated typically towards the latter part of a Convolutional Neural Network (CNN), the fully connected layer is responsible for transforming the outputs from convolutional and pooling layers into a format conducive to further processing [6]. Each neuron within the fully connected layer establishes connections with all neurons in the preceding layer, with the weights between them facilitating the learning of feature combinations and abstract representations.

Within a CNN, convolutional and pooling layers primarily extract local features and spatial structures from the input data. In contrast, the fully connected layers focus on acquiring global feature representations and performing classification or regression tasks. The output of the fully connected layer can be directed towards a softmax or regression layer for classification or regression predictions. The input to the fully connected layer consists of a flattened feature vector, which entails converting the multi-dimensional feature maps from convolutional and pooling layers into a one-dimensional vector [7]. This flattening process involves transforming each feature map into an elongated vector and subsequently concatenating these vectors to construct the input.

Each neuron within the fully connected layer possesses a distinct set of weights and biases, establishing connections with all neurons in the preceding layer. These weight and bias parameters undergo training via the backpropagation algorithm, adapting to the specific task at hand. Throughout the training process, the fully connected layer learns the nonlinear mapping that links input features to output labels, facilitating the accomplishment of classification or regression objectives.

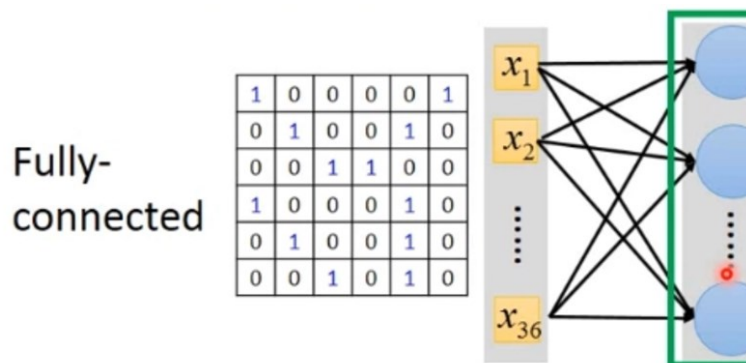


Figure 4. Fully Connected layer [7].

3. Analysis of CNN-Based Applications

3.1. Implementation of Recognition

To perform CNN-based license plate recognition using MATLAB, you can follow the steps below:

Data Preparation: First, collect a dataset containing license plate images and annotate them by marking the position of each license plate and its corresponding character label. Make sure the dataset includes samples with different types of license plates, fonts, and layouts, and covers various angles, lighting conditions, and occlusions.

Data Preprocessing: Preprocess the data to prepare it for input to the CNN model. This includes resizing the images, converting color images to grayscale, normalizing the images, and so on. Ensure that all images have the same size and map the pixel values to the range of 0 to 1.

Constructing the CNN Model: Employ the Deep Learning Toolbox within MATLAB to create the CNN model. You have the option to utilize the network design tool or tailor your own code for network construction. CNN models typically comprise convolutional layers, pooling layers, and fully connected layers. When determining the network structure, take into account the specific requirements of the license plate recognition task to select an appropriate configuration.

Model Training: Train the CNN model using the prepared dataset. During training, pass the input images to the CNN model and compare the predicted license plate labels with the expected labels. Update the weights and parameters of the model through backpropagation and optimization algorithms to gradually learn the association between input images and the correct license plate labels. MATLAB provides functions and tools for training deep learning models.

Model Evaluation: Evaluate the trained CNN model using an independent test set separate from the training data. Evaluate the accuracy and performance of the model by inputting test images and comparing the predicted license plate labels with the actual labels. Evaluation metrics such as classification accuracy, confusion matrix, etc., can be computed.

License Plate Recognition: Employ the pre-trained CNN model to perform license plate recognition. Forward the input images through the CNN model to acquire the model's output, representing the predicted license plate label. As needed, consider the utilization of character segmentation algorithms and character recognition algorithms to conduct additional processing on the license plate image, culminating in the final recognition outcome.

3.2. Analysis of Application Effectiveness

Currently, Convolutional Neural Networks (CNNs) are widely applied in various aspects of life, such as handwritten recognition, license plate recognition, and many scholars have conducted in-depth research in this field. The author Peng Peng first uses TensorFlow to build a CNN model and trains it using collected license plate data. Then, an embedded system peripheral circuit for vehicle recognition is built using the STM32MP157 embedded chip. Finally, through experiments, license plate recognition is achieved. The process is divided into four stages: license plate localization, character segmentation, character recognition, and non-segmented license plate recognition. The recognition speed and accuracy both achieved expected performance. Moreover, the overall hardware cost of the solution was significantly reduced [8].

The second author, Wanwei Wang, proposed the following operations. Firstly, the images or videos of vehicles captured by cameras are processed and analyzed. Then, techniques such as digital image processing and pattern recognition are employed to extract license plate numbers and color information. In real-world scenarios, images are captured under highly complex environments with varying lighting and weather conditions, making detection, segmentation, and recognition challenging tasks. In this paper, an innovative approach is introduced, namely the Multi-Task Light CNN for License Plate Recognition (MTLPR), which aims to provide a robust and real-time solution for car license plate detection and recognition. This method employs an end-to-end algorithm to efficiently recognize license plate characters. The proposed framework is built on cascaded networks, offering several benefits, including simplicity, high accuracy, and minimal computational demands. Notably, the Chinese license plate

dataset has been expanded and diversified, resulting in a substantial enhancement of the License Plate Recognition (LPR) system's performance [9].

The third author Jianxing Zeng started from traditional detection algorithms, this paper proposes a location method that combines edge detection and color information. This method achieves good localization results only for blue license plates in scenarios such as residential areas and parking lots, with certain limitations. To further enhance the applicability and robustness of the license plate localization algorithm, a license plate region detection algorithm based on Mask R-CNN is proposed. The paper selects and analyzes the ResNet+FPN feature extraction network of the Mask R-CNN model, annotates a license plate dataset using VIA software, exports the annotated dataset as a JSON file, and uses it for training and testing the Mask R-CNN model, ultimately obtaining experimental results [10].

4. Challenges and Future Prospects

Although Convolutional Neural Networks (CNN) have achieved great success in license plate recognition tasks, there are still some limitations and challenges. Here are some of the main shortcomings of CNN for license plate recognition:

Restrictions on dataset size and diversity: CNN requires large and diverse training data to learn effective feature representations. However, for certain regions or specific types of license plates, the available training data may be relatively limited. This can result in decreased performance of CNN models when dealing with these specific cases of license plates.

Illumination and angle variations: License plate recognition systems need to be robust to changes in illumination and viewing angles to adapt to different scenarios. However, the sensitivity of CNN to lighting and angle variations is still not fully resolved. In cases of strong lighting, shadows, or plate tilting, CNN may struggle to accurately recognize the license plates.

Occlusion issues: Occlusion is a common challenge in license plate recognition. When the license plate is partially obstructed, for example, by mudguards, stickers, or other objects, CNN may struggle to extract valid features, leading to recognition errors.

Issues with rare license plate types: Different regions and countries may have different license plate types and formats, including different character sets, fonts, and layouts. When CNN encounters rare types of license plates, it may face difficulties in recognition.

Real-time requirements: In some real-time applications, license plate recognition systems need to process images quickly and accurately within a short time frame, such as at highway toll booths or parking lot entrances. Traditional CNN models, due to their high computational complexity, may not meet the real-time requirements.

5. Conclusion

In summary, Convolutional Neural Networks (CNNs) have firmly established themselves as formidable image recognition algorithms with extensive applications spanning a multitude of domains. Their versatility and adaptability have made them indispensable tools in the realm of computer vision and beyond. As it continues to refine and optimize CNN models, their performance and effectiveness are poised to reach even greater heights.

The challenges and opportunities presented in this article serve as guiding beacons for the future of CNN technology in the realm of computer vision. Addressing challenges such as data diversity and scalability will be pivotal in ensuring CNNs can tackle an ever-expanding array of recognition tasks. Moreover, the promising prospects outlined here, particularly in license plate recognition, illustrate the boundless potential for CNNs to revolutionize automation, surveillance, and security systems.

Furthermore, CNNs' influence transcends their immediate applications, as they serve as pivotal components in the broader landscape of artificial intelligence and automation technologies. Their ability to process and interpret complex visual data is not only transforming industries but also shaping the way it interacts with machines and the world around.

References

- [1] Zherzdev S, Gruzdev A. Lprnet: License plate recognition via deep neural networks. arXiv preprint arXiv:1806.10447, 2018.
- [2] Li H, Wang P, Shen C. Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 2018, 20(3): 1126-1136.
- [3] Montazzolli S, Jung C. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. 2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI). IEEE, 2017: 55-62.
- [4] Masood S Z, Shu G, Dehghan A, et al. License plate detection and recognition using deeply learned convolutional neural networks. arXiv preprint arXiv:1703.07330, 2017.
- [5] Silva S M, Jung C R. Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*, 2020, 71: 102773.
- [6] Lin C H, Lin Y S, Liu W C. An efficient license plate recognition system using convolution neural networks. 2018 IEEE International Conference on Applied System Invention (ICASI). IEEE, 2018: 224-227.
- [7] Kurpiel F D, Minetto R, Nassu B T. Convolutional neural networks for license plate detection in images. 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017: 3395-3399.
- [8] Peng P. Research on license plate recognition based on CNN Convolutional neural network. Shandong University, 2020.
- [9] Wang W, Yang J, Chen M, et al. A light CNN for end-to-end car license plates detection and recognition. *IEEE Access*, 2019, 7: 173875-173883.
- [10] Zeng Jianxing. Research on license plate recognition based on convolutional neural network. Shandong University of Science and Technology, 2020.