License plate location and recognition

TCW group: Fengyin Cen, Honghui Weng, Zian Tang

Abstract

We present a streamlined adaptation of the Multi-task Cascaded Convolutional Neural Network (MTCNN) for efficient license plate detection, eliminating the complexity of feature point identification. Incorporating the Spatial Transformer Network (STN), the system corrects image distortions to facilitate accurate plate recognition. Utilizing LPRNet for its lightweight and robust character recognition, our method forgoes the need for segmentation. This end-to-end framework ensures real-time processing suitable for embedded devices. The combined approach promises enhancements in automated license plate recognition, with broad implications for vehicular security and traffic systems.

1 Background

There are plenty of situations where the license plate location and recognition technique could be used. For example, it could help with social security. License plate data can be valuable in criminal investigations, providing a record of vehicle movements and aiding law enforcement agencies in solving crimes. What's more, it could also help with parking system. Automated recognition facilitates efficient parking management, including identifying violators, managing parking spaces, and streamlining payment systems.

2 Algorithm

The algorithm for license plate location and recognition integrates modified versions of MTCNN and STN frameworks, along with LPRNet, to create a seamless and efficient process for identifying and interpreting license plates.

2.1 License plate location base on MTCNN

MTCNN[1] was proposed by Zhang et al. in 2016, which is a deep cascaded multi-task framework that detects faces and facial landmarks on images. It consists of 3 neural networks connected in a cascade:

1. P-Net(Proposal): Scan images and perform first detection. It has a low threshold for detection and therefore detects many false positives.

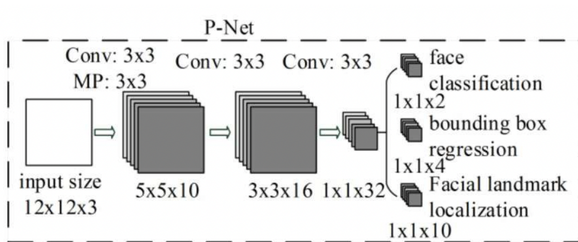

Figure 1: P-Net Architecture

2. R-Net(Refine): Filter detections to obtain quite precise bounding boxes.

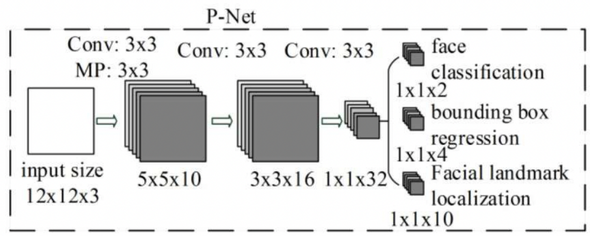

Figure 2: R-Net Architecture

3. O-Net(Output): Perform the final refinement of the bounding boxes. This way not only faces are detected, but bounding boxes are very right and precise.

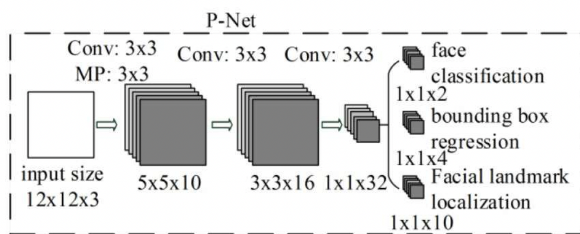

Figure 3: O-Net Architecture

In order to use it to detect the location of the license plate, the original algorithm need to be modified. We abandon the R-Net because it's much easier to detect the license plate than faces, and there is no need to locate feature points. We also modify the parameters of the convolutional and pooling layers within P-Net and O-Net. And the input and output dimensions of the networks are adjusted to meet the requirements of license plate detection.

2.2 License plate calibration base on STN

The STN (Spatial Transformer Network)[2] is a learnable module in a neural network that allows the network to perform spatial transformations on an input feature map. It consists of three main components: a localization network, a grid generator, and a sampler. The localization network predicts the parameters of the spatial transformation, which are then used by the grid generator to create a sampling grid. The sampler uses this grid to produce a transformed output, which is then used for further processing by the network.

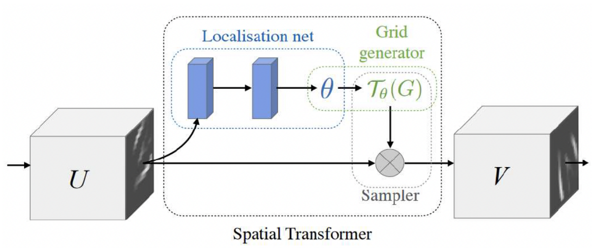

Figure 4: STN model Architecture

It is a novel approach in computer vision that allows a neural network to learn how to perform complex spatial transformations on images, such as translation, scaling, rotation, cropping, as well as non-rigid deformations. Since the image we get from MTCNN may be skewed or off center images, we could use STN to calibrate the image.

2.3 License plate Recognition base on MTCNN

LPRNet(License Plate Recognition Net) [3] is a lightweight neural network which takes the plate image as a whole as input and then outputs the plate number. It has several advantages:

1. The network is lightweight enough so that it can be integrated to run in embedded devices.
2. The character segmentation link is eliminated, and character segmentation and character recognition are integrated and realized in a single network.
3. Strong robustness to various types of interference, such as uneven illumination, viewing angle distortion, etc.

It utilizes a streamlined CNN architecture with depth-wise separable convolutions and omits fully connected layers in favor of LSTM modules. This architecture allows for direct end-to-end training without the need for segmenting images, using Connectionist Temporal Classification (CTC) to handle sequences where input-to-target alignment is not predefined. This approach streamlines the learning process and reduces the computational load, leading to more efficient model training and inference.

3 Implementation

We use MTCNN to detect and localize the specific location of the license plate in the input image. Since the network framework of MTCNN was originally designed for face detection and recognition, it needs to be modified accordingly when it is used for license plate detection and localization, which includes two parts. (1) Discard the R-Net network part and retain only the P-Net and O-Net network parts. Since the task of license plate detection is relatively simple, and only need to complete the license plate detection and bounding box regression task, and do not need to carry out the task of feature point localization, so through the experiments found that the removal of the R-Net part of the team after the final detection of the accuracy does not have a direct impact. (2) The parameters of the convolutional layer and pooling layer inside P-Net and O-Net are modified, and the input and output dimensions of the network are adjusted to meet the requirements of license plate detection. The modified MTCNN network structure is shown in Figure 5 and 6.

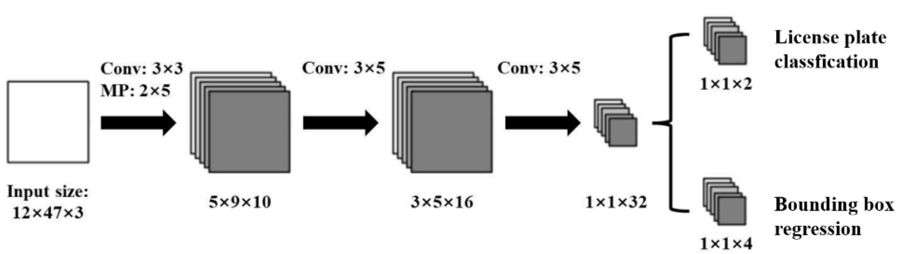

Figure 5: Modified P-Net Network Architecture

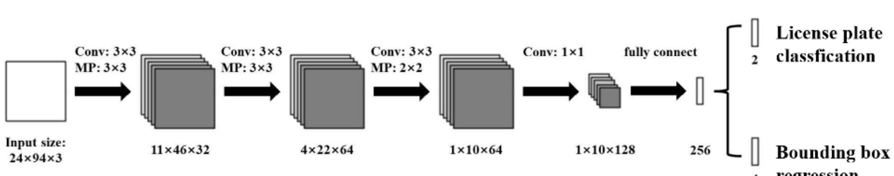

Figure 6: Modified O-Net Network Architecture

In this paper, MTCNN is trained using CCPD[4] license plate dataset. The main steps are as follows:(1) Generate positive samples, negative samples with intermediate samples for training P-Net. (2) Train the P-Net network model. (3) Generate positive samples, negative samples and intermediate samples for training O-Net. (4) Train the O-Net network model When the original input image is processed by MTCNN, its predicted license plate area coordinates will be obtained, and the license plate image can be obtained by cropping the original input image according to the coordinates, and the license plate image can be obtained by inputting the license plate image to LRPNet for inference.

4 Result

To verify the feasibility of the license plate localization and recognition algorithm in this paper, 100 images are randomly selected from the CCPD dataset as the test set. In order to evaluate the robustness of the algorithm, the selected test images include both standard vehicle license plate images and vehicle license plate images with various types of interference such as blurring, tilting and rotation. The results show that the accuracy of the algorithm in this paper reaches 98%. Figure 7 shows some concrete examples.

Figure 7: The testing examples

References

[1] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

[2] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[3] Sergey Zherzdev and Alexey Gruzdev. Lprnet: License plate recognition via deep neural networks. *arXiv preprint arXiv:1806.10447*, 2018.

[4] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European conference on computer vision (ECCV)*, pages 255–271, 2018.