# Automated License Plate Recognition using Single Image Super-Resolution

Linfeng DU, Yile ZHENG, Afzal AHMAD
The Hong Kong University of Science and Technology
{linfeng.du, leonyile.zheng, afzal.ahmad}@connect.ust.hk

## Abstract

*The rapid increase in the number of vehicles on roads has given rise to novel surveillance challenges. With the proliferation of AI, governments around the world are increasing spending on AI-based systems capable of automated license plate recognition for monitoring traffic infractions and surveillance. In this project, we explore the impact of several enhancements and super-resolution (SR) to license plate images on their recognition accuracy. We propose enhancements allowing brightness and rotation/tilt correction, followed by a super-resolution network, enabling significant improvements to the fidelity of license plate image. Experiments show our methods improve the accuracy of license plate recognition by up to 31.88% on a base subset, while also showcasing significant improvements on several challenging subsets of the Chinese City Parking Dataset (CCPD), the largest known dataset for this problem.*

## 1. Introduction

Estimates suggest there were over 1.4 Billion vehicles on the world's roads by 2018 [1]. This rapid rise in road vehicles has introduced novel challenges for traffic monitoring and surveillance. The use of AI is increasingly becoming crucial in designing robust systems for automated license plate recognition (ALPR). The core challenge in designing AI-based ALPR systems is that of generality; ALPR systems need to provide high-accuracy over a range of working conditions including day/night times, weather conditions, and on blurry images. Furthermore, their accuracy has to be consistent over different camera positions and views of the vehicles.

Towards this end, we design a robust ALPR pipeline that is able to apply several enhancements to the license plate region before applying recognition. These enhancements improve the image fidelity of the detected license plate region, which is then passed to a recognition network. Fig. 1 shows the overall pipeline of our system. It consists of a detection module to detect and crop the license plate area, followed by a non-learning based enhancement module to apply rotation/tilt and brightness corrections to the detected license plate. The low-resolution, enhanced license plate crop is passed to a modified ESRGAN [17] for $4\times$ super-resolution. Using the ESRGAN, we explore the impact of different upsampling techniques on alleviating the checkerboard artifact problem in the license plate images. The $4\times$ high-resolution, enhanced images are passed to a recognition module based on CRNNs [14] for character recognition.

Defining an accurate recognition as correctness on all characters in a license plate, experiments show that our methods improve the accuracy of license plate recognition by up to 31.88% on the base subset of Chinese City Parking Dataset (CCPD) [19]. On blur, challenge, dark/bright, far/near subsets, the improvement ranges from 5.57% to 20.12%. The greatest effect appears on rotated, and tilted license plate subset, where results on low resolution plate images are only 4.16% and 2.29%. Through our pipeline, the accuracy increases to 55.24% and 35.27%, respectively. To the best of our knowledge, this is the first work that studies the impact of super-resolution on license plate recognition.

## 2. Related Work

The ALPR problem has been well studied within the past two decades, using both non-learning based techniques—such as connected-component analysis (CCA), edge detection, template matching and geometric analysis [11, 13, 2]—and learning based techniques using conventional machine learning and also state-of-the-art DNN backbones [18, 18, 15, 22]. Recent works are predominantly utilizing DNNs owing to their adaptability; their ability to generalize to a variety of different environmental conditions and camera views is unparalleled.

[19] proposed a huge dataset of over 340k images of vehicles with license plates, called Chinese City Parking Dataset (CCPD), while also establishing a baseline recognition network. The recognition network, called roadside parking network (RPNet), despite being a simple 10-layer convolutional network, was able to perform well on the

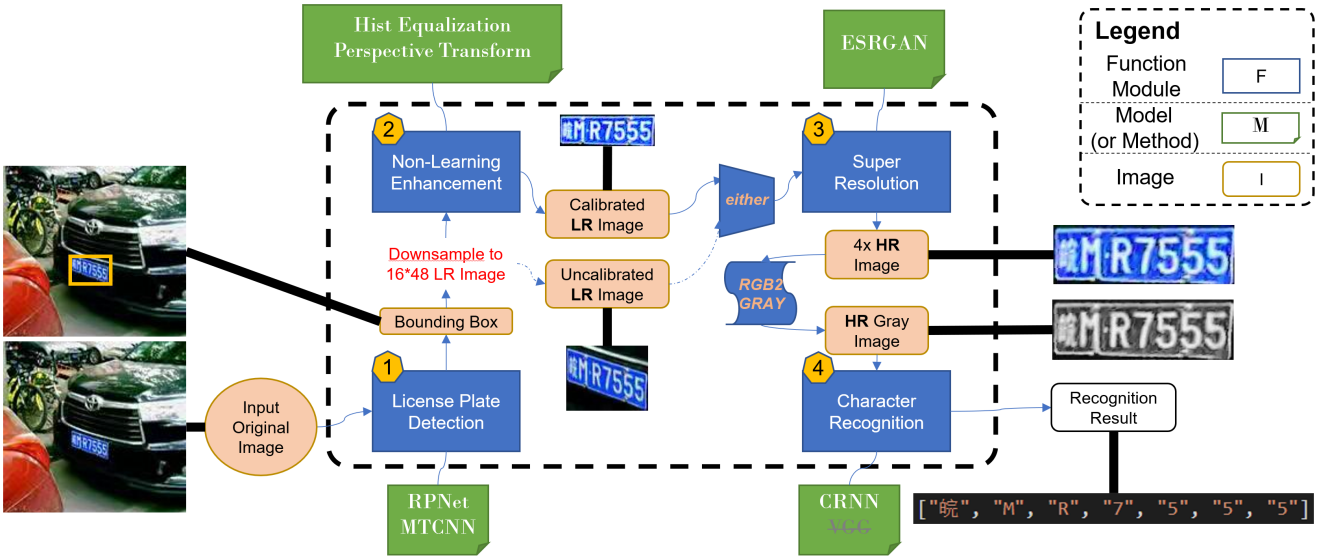



Figure 1: Overall pipeline of our license plate recognition system.

ALPR task on the CCPD dataset. Given that RPNet is able to perform recognition with very high accuracy on the CCPD dataset (over 99% on base split), we felt the need to make the problem more challenging. Specifically, our work studies the impact of super-resolution on recognition accuracy using *low-resolution* input images. Hence, we do not use the CCPD directly in our experiments but instead *downsample* the detected license plate area (Fig. 1) to $16 \times 48$ LR image in order to make the ALPR problem more challenging and to demonstrate the role that super-resolution plays in boosting our recognition accuracy.

To increase the resolution of an image, one of the earliest effective end-to-end super-resolution work was SRCNN [3, 4]. Then, diverse architectures are proposed including residual block [10], dense block [7], residual dense block [21], memory block [16], etc. ESRGAN [17] is improved based on SRGAN [10], benefiting from Residual-in-Residual Dense Block (RRDB) in generator, relative realness in discriminator, and features before activation in perceptual loss.

## 3. CCPD Dataset

We utilize the Chinese City Parking Dataset [19] (CCPD) to demonstrate the effectiveness of our methods. CCPD is the largest known dataset for this problem, consisting of over 340k images. It is annotated with bounding boxes, vertice coordinates of license plates (LPs), and license plate numbers. Each image in the dataset consists of only one licence plate. The dataset consists of a base subset utilized for training and validation (each using 100k images), and several other subsets having different properties. These properties include different illumination conditions, excessively far/near distance between camera and vehicle, and rotated/titled camera angles. Table. 1 shows the number of images in different subsets of the dataset. Each image in the dataset is of resolution $720 \times 1160$. As shown in Figure. 1, we downsample the *detected* bbox to a LR image of size $16 \times 48$ in order to make the recognition problem more challenging.

Table 1: CCPD Subset Description and Sizes

| Subset[1] | Description | #Images |
|---|---|---|
| base | Base Split | 200k |
| db | Dark/Bright Illumination | 10k |
| fn | Far/Near View | 21k |
| rotate | Vertical Tilt $(-10°, +10°)$ | 10k |
| tilt | Vertical Tilt $(+15°, +45°)$ | 30k |
| blur | Blurred due to hand jitter | 21k |
| cha | Challenging split | 50k |

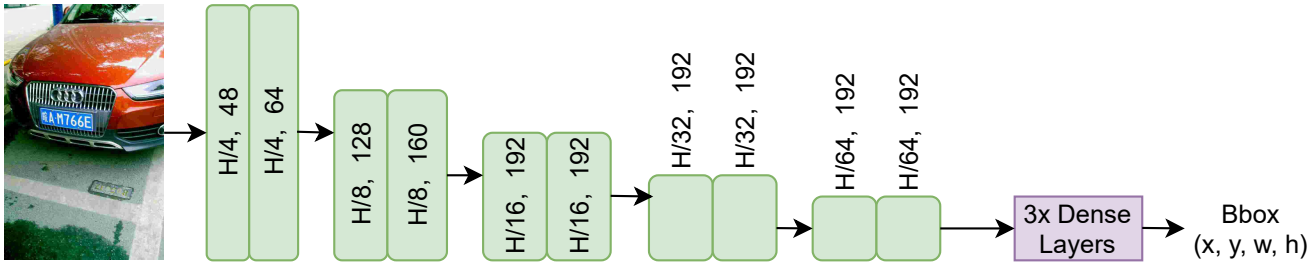
## 4. Proposed Pipeline

Fig. 1 shows the overall pipeline of our system. Our pipeline consists of four main modules:

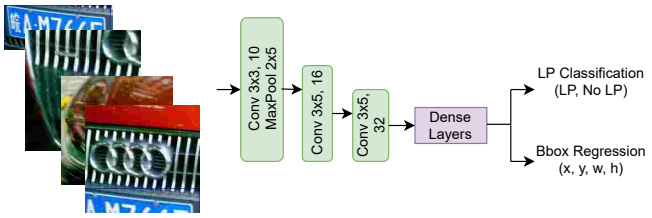
1. A detection module to detect and crop the license plate region in the images.

2. A non-learning enhancements module where we perform perspective transform and histogram equalization to correct the rotation/tilt, brightness of the input license plate. This module is utilized mainly to rectify the abnormalities in some of the non-base subsets.

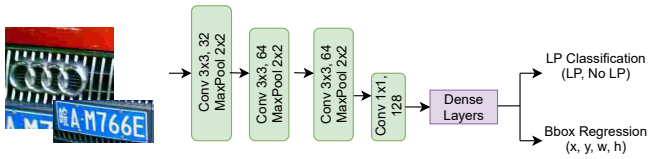
[1]Note that the dataset has been updated, hence the numbers in this table do not conform with those in Xu et al. [19]




(a) RPNet architecture.




(b) Proposal Network (PNet) Architecture.




(c) Output Network (ONet) Architecture

Figure 2: Detection Networks RPNet [19] (a), and MTCNN [20] cascaded CNNs, PNet (b) and ONet (c).

3. A modified ESRGAN for super-resolution. We study the impact of different upsampling techniques in ESRGAN on perceptual quality, checkerboard artifacts, and improvements in overall recognition accuracy.

4. A recognition module for character recognition.

In the following subsections, we describe each of these modules in greater detail.

## 4.1. License Plate Detection

Given the fact that each image in the dataset only contains a single license plate, detection is a regression problem with only one bounding box (bbox) coordinates. We experimented with two detection modules for this purpose; Roadside Parking Network (RPNet) [19] and Multi-Task CNN (MTCNN) [20].

### 4.1.1 Roadside Parking Network (RPNet)

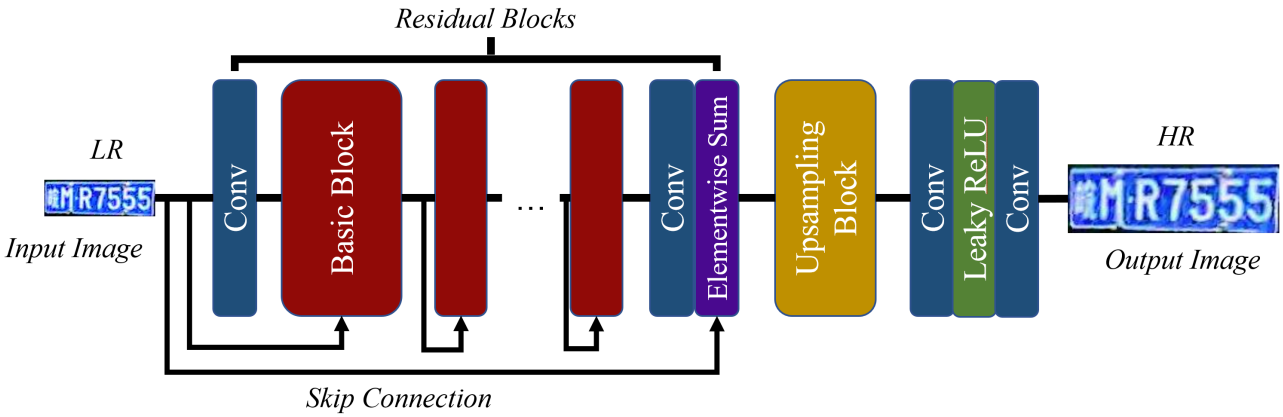
RPNet [19] is a rather unsophisticated CNN architecture designed for the task of ALPR on the CCPD dataset. We removed the recognition part of the network while extending the detection to allow evaluation using the detected bounding box IoUs instead of the character recognition accuracy. Figure. 2a shows the architecture of simplified RPNet that we utilize. RPNet consists of a total of 10 convolutional layers, each followed by batch normalization, relu activation, and maxpool operation. Bbox is regressed by passing the output of the last conv layer through a cascade of 3 dense layer, where the last layer outputs four values corresponding to center x/y and width/height of the bbox. One noticeable property of RPNet cost function is the fact that it penalizes the detected bounding boxes significantly more on the center coordinates and much less so on the height/width of the boxes[1].

### 4.1.2 Multi-Task CNN (MTCNN)

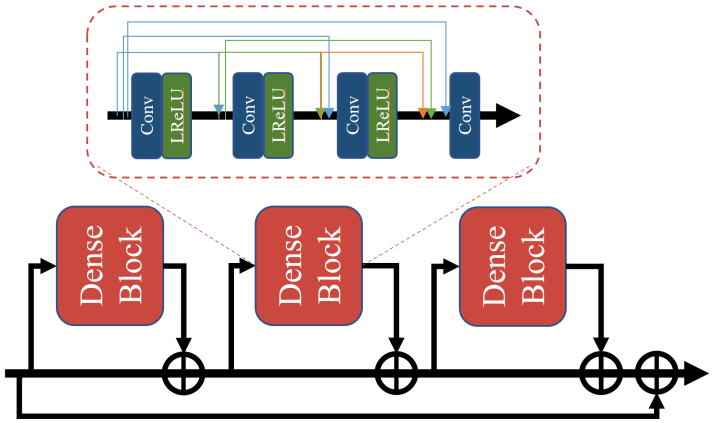
MTCNN [20] is a seminal work on face detection and alignment using cascaded CNNs. MTCNN consists of three cascaded CNNs, where each CNN refines the outputs of the previous CNN. The first CNN is responsible for generating region proposals using a shallow CNN. The two subsequent CNNs refine the predictions using more complex network architectures, eventually generating bounding boxes and their classification.

Given that our problem does not involve landmark detection, a simplified network architecture of MTCNN is utilized which only consists of two cascaded CNNs; Proposal Network (PNet) and Output Network (ONet). The middle CNN, Refinement Network (RNet) is excluded since the simplified network can achieve sufficiently high accuracy. Fig. 2b and 2c show the network architectures of the two CNNs in simplified MTCNN. MTCNN also utilizes multi-source training using which the network is explicitly trained with non-licence plate region proposals and partially aligned license plate region proposals. Online hard sample mining is also utilized which allows gradient computation using only the input sample images that generate large loss values, hence ignoring the 'easy' samples which are unhelpful during the training process.

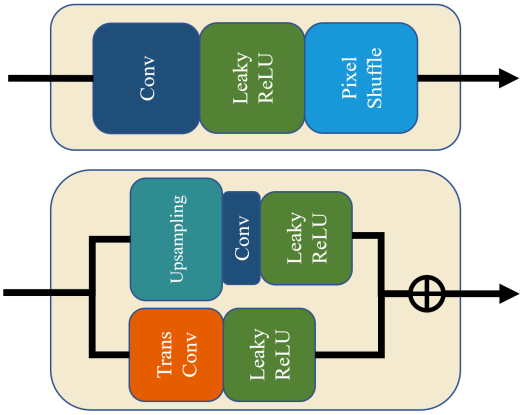
[1]Our experiments showed that equal weights to each loss term generates worse results, hence this weighted loss is purely based on heuristics.




(a) SRResNet [10] architecture.




(b) Residual-in-Residual Dense Block (RRDB).




(c) Original ERSGAN upsampling (upper) and out improvement for artifact removal (lower).

Figure 3: Super Resolution network architecture.

### 4.2. Non-Learning Enhancements

CCPD test splits contain several challenging subsets for machine recognition, e.g., rotated or tilted LPs different from upright base subset used for training, and too dark/bright plate even unclear to human eyes. In order to generalize our solution, two non-learning image processing techniques are applied here.

To fix the rotated or tilted numbers, we apply perspective transformation to license plate area using the four plate corner vertices. By multiplying the original image coordinate with a $3 \times 3$ projection matrix, mapping to a rectangle region (known as viewing plane) to calibrate the angle of characters. However, since the quality of original image is non-uniform, perspective mapping towards a uniform size could cause information loss. When it's applied to rotated or tilted numbers, great improvement is anticipated. However, when applied to blurred, or dark/bright plates, there's potential degradation in the recognition results. More details are discussed later in Section. 5.2.

To enhance the too dark or too bright images, plate-area contrast is effectively adjusted through histogram equalization, which allows the area with lower local contrast to gain a higher contrast–making plate characters stand out against the background. Equalization is done by normalizing cumulative distribution function of original images to get a target distribution, then pixel-wise mapping from original distribution to target space.

### 4.3. Super Resolution

The low-resolution output of our non-learning enhancement module is passed to the super-resolution module, where we utilize Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) [17] to generate $4\times$ high-resolution plate images.

The overall generator network architecture of ESRGAN is inspired by SRResNet [10], as Fig. 3a shows, where there

are multiple choices for "basic blocks". In ESRGAN, instead of residual block [6] or dense block [7], Residual-in-Residual Dense Block (RRDB) without batch normalization is used as the basic building block in generator to facilitate a deeper network. Fig. 3b shows the architecture of an RRDB. Experiments by [17] showed that RRDB demonstrates greater improvement in quality of generated images. ESRGAN also utilzied an improved discriminator using Relativistic average GAN (RaGAN) [8] which learns sharper edges and detailed textures effectively.

One of our major observation with super-resolution process is that checkerboard artifact and other sources of noise dramatically affects the quality of generated images, and subsequently the recognition accuracy. To remove the artifacts and noise, we used [12] for reference, which proposed that upsampling strategy strongly mediates the perceptual quality—original ESRGAN utilizes pixel shuffle algorithm in their upsampling block, while a better way is to separate upsampling with feature computation—replacing transposed convolution with resizing and convolution. The authors claimed that nearest-neighbor interpolation achieved the best result, and bilinear interpolation is not as effective. We compare the SR results obtained using these different upsampling techniques in Section 5.3. Our final implementation refers to the super-resolution model in cycle-in-cycle GAN [9], as Fig. 3c shows, we replaced original pixel shuffle with a combination of interpolation + convolution and transposed convolution, by assigning less weight to the latter path.

### 4.4. Character Recognition

Owing to its high accuracy, we adopt CRNN (Convolutional Recurrent Neural Network) as our recognition network [14]. As shown in Fig. 4, CRNN consists of a convolution network followed by a recurrent network. Convolutional layers help extract features from horizontal aspect-ratio input images which contain character combinations while recurrent layers segment and recognize each character. The CRNN convolutional layers perform symmetric down sampling in their max pooling layers, which scales down the image size in both dimensions and can be seen as enlarging the perceptive field on the input image. Afterward, the network starts performing some asymmetric down-sampling, which only scales down the height of feature maps while the width of the feature maps remains unchanged until the height equals to 1. This means we extend the perceptive field in only one dimension. Finally, we obtain a feature map of dimensions $1 \times W \times C$ with each position in the width dimension representing a vertical area in the original image. After getting features in this shape, the bidirectional RNN takes in the feature as a sequence with the width size as time steps. Here we use bidirectional LSTM (Long Short-Term Memory) as the recurrent

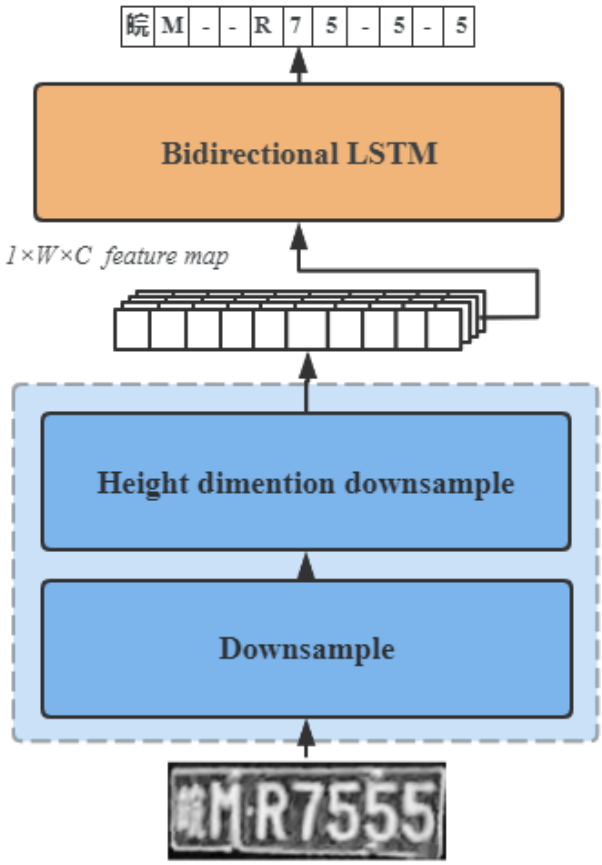



Figure 4: The adopted recognition network.

portion [5]. The final output obtained has the same length as the width of input feature map. Whereas the prediction can be any length shorter than this size. Because the vertical areas projecting to the input image could have some overlay, it predicts the same character as others. Or the vertical area covers a region where there is no character. In this case, it predicts "blank". This also means we need to have one more 'blank' class for invalid predictions. The total number of classes in CRNN is $68 + 1 = 69$, where 68 corresponds to chinese province characters, digits, and alphabets.

The original CRNN architecture is designed for input image size of $100 \times 32$ while our input image size is $64 \times 192$ ($4\times$ upsampled of LR size). So we alter the network to adapt to our input size (Please see Appendix Fig. 8 for the modified network architecture). We add one more down-sampling layer at the first part of the convolutional network to make sure the perceptive field is large enough before doing vertical down sampling.

Before deciding to use CRNN, we also tried directly utilizing VGG-19 for recognition. We implemented the convolution layer part of the VGG-19, while dividing the fully connected layers into 7 classifier heads corresponding to the 7 characters in each license plate. The accuracy obtained us-

ing VGG-19 is only around $1\%$ on the base subset even after long training. The details of this network were elaborated in the millstone report. This abysmal accuracy of VGG-19 drove us to find methods that can perform better on recognition, eventually stumbling across CRNN. Please refer to the milestone report for experiments with VGG-19.

# 5. Experiments

Given the modularity of our system, we are able to experiment with each stage of the pipeline without needing the results of the previous stages. Hence, in this section, we demonstrate qualitative and quantitative results of both independent modules and the overall recognition results.

For training, a predefined 100k image split from the base subset of CCPD is utilized. The remaining 100k split is used for validation. Testing is done on non-base subsets of the dataset. This makes achieving high-accuracy on the test subset much more challenging since there is a huge disparity between the training and test images. Specifically, training images consist only of frontal views of the license plates in normal light/weather conditions, no rotation/tilt and roughly constant distance from the camera. The test images, however, are significantly different in brightness, weather, rotation, tilt, etc compared to the base training subset.

## 5.1. License Plate Detection

The detection accuracy evaluation is based on the intersection-over-union (IoU) metric between the predicted bounding box and the ground truth bounding box. Specifically, if the IoU between these two bboxes is greater than 0.7, the prediction is classified as correct.

### 5.1.1 RPNet

We adopted the training hyperparameters from [19]; mean absolute error loss with step learning rate, momentum of 0.9, and SGD optimizer. The network is trained for 106 epochs using batch size of 6 on a Tesla V100S GPU, taking over 1.5 days. The training accuracy saturates at roughly 88.5% after the first 50 epochs[2].

### 5.1.2 MTCNN

Due to a lack of compute resources, we were only able to train MTCNN[3] using a smaller subset of 40k images out of the full 100k training split. Training parameters are adopted from [20]; Cross-Entropy loss for classification and MSE loss for regression, batch size of 64. We trained both PNet and ONet in MTCNN for 16 epochs for several hours on a Tesla V100S GPU. Although training is done on a small subset of 40k images, validation and testing are done on the full val and test splits. Despite being trained using only 2/5 of the training data used by RPNet, MTCNN quickly saturates to high training accuracy of over 99.9%. For MTCNN, the training accuracy is measured as the classification accuracy of the prediction label (LP, no LP, partially aligned LP). The network is trained for a fixed 16 epochs, after which we measure the validation and test accuracy using our normal, IoU based evaluation metric.

Table 2: Detection Accuracy Results - RPNet and MTCNN. 100k held-out split from base subset is used for validation. The remaining subsets are used only for testing. The accuracy metric is based on IoU $\geq 0.7$ metric described in Section. 5.1.

| Subset | RPNet | MTCNN [4] |
|---|---|---|
| Base (Validation) | 80.90% | 99.16% |
| Blur | 42.33% | 53.85% |
| Weather | 83.91% | 98.49% |
| Dark/Bright | 36.78% | 51.87% |
| Far/Near | 13.91% | 75.11% |
| Tilt | 12.25% | 47.65% |
| Challenging | 38.64% | 77.17% |
| Rotate | 0.04% | 59.92% |

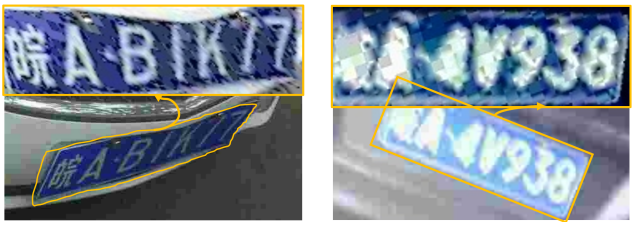
### 5.1.3 Comparison

Table. 2 shows the detection results on the base (validation) and test subsets of CCPD using both RPNet and MTCNN. Despite being trained for fewer epochs using only a 2/5 fraction of the training data, MTCNN performs significantly better than RPNet across the board. RPNet completely fails on the rotated subset while MTCNN achieves detection accuracy of 59.92% (This comparison is qualitatively shown in Appendix Fig. 7c and 7d). This could be attributed to a more exhaustive approach that MTCNN adopts using the region proposals. MTCNN utilizes a large number of region proposals which are refined using the two cascaded CNNs. RPNet utilizes a naive approach to detection, regressing a bounding box relative to the dimensions of the input image. MTCNN, on the other hand, only has to regress relative to the proposed regions and fine-tune the proposed regions. Multi-source training and online hard sample mining also help MTCNN since its training objective is tri-fold; license plate, no license plate, and partially aligned license plate, and training is done on 'hard' input samples that generate large loss.

[2]Full training loss and accuracy plots can be found here: https://wandb.ai/afzal/alpr/runs/11u40qn8?workspace=user-afzal

[3]We adopted the open-source simplified MTCNN from https://github.com/xuexingyu24/License_Plate_Detection_Pytorch, adding our own IoU-based evaluation metric

[4]Trained on only 40k samples from training data instead of the full 100k samples due to lack of compute.



(a) From "tilt" subset.



(b) From "rotate" subset.

(c) From "dark" subset.



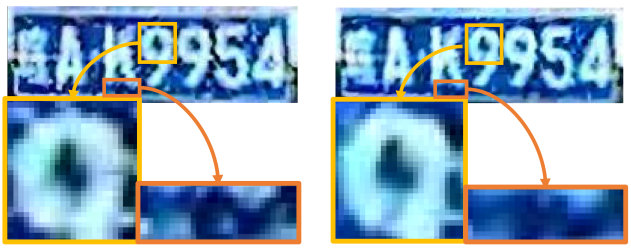
(d) From "blur" subset.

Figure 5: Non-learning enhancement on CCPD dataset.

(a) LR.



(b) SR by Pixel Shuffle.



(c) SR by Bilinear-Upsampling.



(d) SR by Nearest-Upsampling.

Figure 6: Comparison between low-resolution and $4\times$ Super Resolution results with different upsampling strategies.

### 5.2. Non-Learning Enhancements

We applied perspective transformation and histogram equalization on different subsets in CCPD dataset, as Fig. 5 shows. For tilted (e.g. Fig. 5a) or rotated (e.g. Fig. 5b) license plates, perspective transformation calibrates them into upright shape. For too dark (e.g. Fig. 5c) or too bright cases, histogram equalization balances local contrast, and demonstrates good perceptual result. However, we are concerned about two possible negative effect from these enhancements. First, for blurry figures (e.g. Fig. 5d), neither of our enhancements is effective since interpolation during perspective mapping could distort the little remaining information. Second, the label quality of some plate vertices are poor, which matters especially for those images in 'base' subset, where original plate numbers are already upright, while getting distorted after transformation.

### 5.3. Super Resolution

The super-resolution module takes a 16 (height) $\times$ 48 (width) low-resolution license plate as input, as Fig. 6a shows. The original ESRGAN generally applies pixel shuffle algorithm in its upsampling block. Fig. 6b shows the generated 64 (height) $\times$ 192 (width) high-resolution license plate from ESRGAN with pixel shuffle. To remove artifacts, we alternatively used a combination of 1.0 (resizing–convolution) + 0.1 transposed convolution. For interpolation method of resizing part, we tried bilinear-upsampling, as Fig. 6c shows, and nearest upsampling, as Fig. 6d shows.

We have the following qualitative (perceptual) observation:

1) All super-resolution results well recover the characters' shape. In LR plate, the hole of number "9" is only one dark pixel, and by contrast, the high-resolution "9" has a smoother shape.

2) Pixel shuffle method as analyzed before, causes serious checkerboard artifact. As the yellow square in Fig. 6b shows, the outline of the "circle" in number "9" is almost in squared shape. However, when using upsampling + convolution, both bilinear and nearest interpolation smoothen the circle shape.

3) Except for checkerboard artifacts, super-resolution could exaggerate noise in the image. As the orange rectangles show, nearest-upsampling method achieves the best result in filtering noise pixels. However, bilinear-upsampling even generates more noise (isolated white pixels) than original pixel shuffle.

The influence of super-resolution module on recognition is demonstrated in Section 5.5.

### 5.4. Character Recognition

For the recognition network, we train the model with both 1) images that are cropped directly from the *bounding boxes* $(x, y), (h, w)$ and 2) that adopt aforementioned perspective transformation method using *plate vertices coordinates* $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$. The training and validation datasets are split from the base subset of the CCPD2019 containing 100k images each. And the test dataset consists of other non-base subsets.

Table 3: Overall Recognition Accuracy

| | Validation | Test | | | | | |
|---|---|---|---|---|---|---|---|
| | Base | Blur | Challenge | Dark/Bright | Far/Near | Rotate | Tilted |
| LR | 56.90% | 12.37% | 20.10% | 9.08% | 19.44% | 4.16% | 2.29% |
| HR-Box-PixelShuffle | 88.48% | 17.94% | 31.46% | 26.61% | 33.56% | 22.42% | 12.54% |
| HR-Box-Bilinear | 88.02% | 16.91% | 31.09% | 29.20% | 33.17% | 22.39% | 11.49% |
| HR-Box-Nearest | 88.78% | 13.88% | 29.14% | 24.21% | 31.87% | 21.70% | 11.36% |
| HR-Cor-PixelShuffle | 77.92% | 13.95% | 30.95% | 11.86% | 33.47% | 54.94% | 34.43% |
| HR-Cor-Bilinear | 68.78% | 12.49% | 25.14% | 8.79% | 27.31% | 46.97% | 27.31% |
| HR-Cor-Nearest | 78.23% | 16.93% | 32.85% | 13.76% | 34.56% | 55.24% | 35.27% |

After several epochs of training[5], the accuracy starts to saturate. Then we perform the inference on the validation and test subsets. The results are shown in Table 4.

Table 4: Character Recognition Model Accuracy on Ground Truth Cropped LP using BBoxes/Corners

| Subset | BBox[8] | Corner/Vertices[9] |
|---|---|---|
| Base (Validation) | 93.86% | 94.06% |
| Blur | 41.42% | 42.18% |
| Challenge | 57.22% | 58.32% |
| Dark/Bright | 51.46% | 44.96% |
| Far/Near | 70.39% | 64.27% |
| Rotate | 84.33% | 79.19% |
| Tilted | 77.49% | 67.45% |

The results show that the difference is recognition accuracy between bbox-based and vertices-based pipeline is negligible on base, blur and challenging subsets. However, looking at the results of the other remaining subsets, we observe that the 'Box' based pipeline performs better on these subsets. That is because the 'Box' pipeline is trained on images cropped from their bounding boxes, which are more diversified and more irregular compared to 'Corner/vertices'-based pipeline which applies perspective transform.

### 5.5. Overall Recognition Results

In this section, we describe the recognition results from three modules cascaded together; non-learning enhancements, SR, and recognition. We cascade these three modules and perform inference on validation and test datasets like in Section 5.4. Our objective is to quantify the recognition accuracy gain from utilizing the enhancements and SR, compared to directly passing the LR image to the recognition network.

Table 3 shows the recognition accuracy using different configurations of our pipeline. For each subset, we measure the accuracy with 7 different configurations, namely, low-resolution (LR), high-resolution (HR) with "Cor" corners used for perspective transoform, or using bboxes, i.e. "Box". We utilize different upsampling techniques including original pixel shuffle, bilinear, and nearest neighbor upsampling. A recognition is classified as correct if all seven characters in the LP are recognized correctly.

The results show that our enhancements and super-resolution improve the recognition accuracy by up to 31.88% on the validation subset. On the test subsets i.e. "Blur", "Challenge", "Dark/Bright", "Far/Near", the improvement ranges from 5.57% to 20.12%. Our methods produce the most improvement on "Rotate" and "Tilted" subsets, where results on low resolution plate images are only 4.16% and 2.29%. Through our pipeline, the accuracy increases to 55.24% and 35.27%, respectively.

It is worth noting that best results on "Challenge", "Far/Near", "Rotate", "Tilted" subsets are all achieved by using non-learning enhancements ('Cor'), and nearest upsampling strategy. This is because 1) convolution is sensitive to rotation or tilt and our enhancements help mitigate this sensitivity, and 2) nearest upsampling has demonstrated to be the best in removing checkerboard artifacts and noise. However, for other subsets, since some plate vertices are not well labeled, perspective mapping could cause degradation for those characters originally upright but tilted after transformation, hence non-learning methods are not as effective.

## 6. Conclusion

In this paper, in considering the challenge in traffic license plate recognition, we proposed several methods for enhancing the recognition accuracy using some non-learning image enhancements as well as super resolution. We utilized ESRGAN and tuned its upsampling layers, which had a significant impact on perceptual quality of the generated HR images. Non-learning methods, such as perspective transformation and histogram equalization, helped mitigate the effects of rotational dependencies of convnets. And we also explored two different license plate area detection networks; RPNet and MTCNN, as well as license plate character recognition network models including CRNN and VGG-19. Results showed that applying enhancements and super-resolution can improve the ALPR accuracy significantly; up to 31.88% on a base subset. Furthermore, varying degrees of improvement on test subsets in different environmental and camera view conditions shows that our methods can help obtain better generalization in ALPR systems.

[5]Training and inference logs and plots can be found on $wandb.ai/leleleooonnn/plateRecog\_crnn$

[8]Model trained on images cropped by bounding box

[9]Model trained on images with perspective transformation using LP vertices coordinates

## References


[1] A. Chesterton. How many cars are there in the world?, 2018.

[2] J. Chong, C. Tianhua, and J. Linhao. License plate recognition based on edge detection algorithm. In *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 395–398, 2013.

[3] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[4] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[5] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. 1999.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] 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*, pages 4700–4708, 2017.

[8] A. Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

[9] G. Kim, J. Park, K. Lee, J. Lee, J. Min, B. Lee, D. K. Han, and H. Ko. Unsupervised real-world super resolution with cycle generative adversarial network and domain discriminator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 456–457, 2020.

[10] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[11] Y. Liao. Research on edge detection in license plate recognition. pages 1139–1142. Atlantis Press, 2012/08.

[12] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[13] C.-M. Pun and W.-Y. Ho. An edge-based macao license plate recognition system. *International Journal of Computational Intelligence Systems*, 4(2):244–254, 2011.

[14] 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 transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[15] S. M. Silva and C. R. Jung. License plate detection and recognition in unconstrained scenarios. In *Proceedings of the European conference on computer vision (ECCV)*, pages 580–596, 2018.

[16] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[17] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[18] Y. Wang, Z.-P. Bian, Y. Zhou, and L.-P. Chau. Rethinking and designing a high-performing automatic license plate recognition approach. *arXiv preprint arXiv:2011.14936*, 2020.

[19] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. 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.

[20] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[21] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[22] S. Zherzdev and A. Gruzdev. Lprnet: License plate recognition via deep neural networks. *arXiv preprint arXiv:1806.10447*, 2018.

# Appendix

(a) Tilt - RPNet. (b) Tilt - MTCNN.

(c) Rotate - RPNet. (d) Rotate - MTCNN.

Figure 7: Qualitative comparison between detection results of RPNet and MTCNN. RPNet fails to localize the bbox precisely in the tilted sample in (a) (IoU $< 0.7$), while MTCNN performs well. RPNet completely fails on rotated subset while MTCNN can regress the bbox with a fairly good precision.

| Layer | Configurations | Size |
|---|---|---|
| Input | | 1x64x196 |
| Conv1, ReLU | k: 3x3x64, s: 1, p: 1 | 64x64x192 |
| MaxPooling | k: 2x2, s: 2 | 64x32x96 |
| Conv2, ReLU | k: 3x3x128, s: 1, p: 1 | 128x32x96 |
| MaxPooling | k: 2x2, s: 2 | 128x16x48 |
| Conv3, ReLU | k: 3x3x128, s: 1, p: 1 | 128x16x48 |
| MaxPooling | k: 2x2, s: 2 | 128x8x24 |
| Conv4, BatchNorm, ReLU | k: 3x3x256, s: 1, p: 1 | 256x8x24 |
| Conv5, ReLU | k: 3x3x256, s: 1, p: 1 | 256x8x24 |
| MaxPooling | k: 2x2, s: (2,1), p: (0,1) | 256x4x25 |
| Conv6, BatchNorm, ReLU | k: 3x3x512, s: 1, p: 1 | 512x4x25 |
| Conv7, ReLU | k: 3x3x512, s: 1, p: 1 | 512x4x25 |
| MaxPooling | k: 2x2, s: (2,1), p: (0,1) | 512x2x26 |
| Conv8, BatchNorm, ReLU | k: 2x2x512, s: 1, p: 0 | 512x1x25 |
| to-sequence | - | 25 x batch_size x 512 |
| Bidirectional-LSTM | hidden feature: 256 | 25 x batch_size x 256 |
| Bidirectional-LSTM | hidden feature: 256 | 25 x batch_size x 68 (#class) |

Added layer Asymmetric downsampling

Figure 8: Configuration details of the recognition network.