

# Enhancing license plate numeric character detection and extraction using Super Resolution

Nopphawan Numuansuwan  
School of Engineering & Technology  
Asian Institute of Technology  
Pathum Thani, Thailand  
st122410@ait.asia

Saratoon Khantasima  
School of Engineering & Technology  
Asian Institute of Technology  
Pathum Thani, Thailand  
st122149@ait.asia

Worachot Nakduk  
School of Engineering & Technology  
Asian Institute of Technology  
Pathum Thani, Thailand  
st122153@ait.asia

**Abstract**— The purpose of this paper is to answer 2 questions. 1.) How much does image super-resolution can improve the character extraction performance, 2.) Which deep-learning model is better suited for this operation, SRCNN or SRGAN? We first started by obtaining the low-resolution images by down-sampling the collected images, then using YOLOv5 to crop only the license plate. Then we used the cropped image for super-resolution. At that step, we then had 3 sets of images 1.) Low-res image 2.) High-res image from SRCNN 3.) High-res image from SRGAN. Then we performed Optical Character Recognition (OCR) on those sets of images and compare the results. We also used measuring tools such as PSNR and SSIM to observe the quality of the image produced by those two models. The result is that for the PSNR, the score of SRGAN is a little better than SRCNN. While for the SSIM the scores are about the same. But for the OCR, the character extraction accuracy of the images from SRCNN is higher than images from SRGAN.

**Keywords**—Super-resolution, SRCNN, SRGAN, License plate detection and extraction, OCR

## I. INTRODUCTION

Both image detection and image super-resolution can be applied to many real-world applications nowadays such as Face-recognition and character extraction and so on. But there is still room for improvement and many ways to do it. The reason we conducted this experiment is we want to know if the already good-performance image detection algorithms such as Pytesseract can be improved even further by using the image super-resolution techniques. This leads us to our first hypothesis: “the accuracy of character extraction should increase when we use the high-resolution image instead of the low-resolution ones”. As for the image super-resolution aspect of this experiment, we decided to use the SRCNN and SRGAN models. The reason we chose these two is that both CNN and GAN generally perform well with image datasets. But because the structure and working process of GAN is more complex. This leads to the second hypothesis: “SRGAN should perform better on image super-resolution than SRCNN”. Our experiment in the following section is revolved around these two hypotheses.

## II. BACKGROUND

### A. Super-resolution (SR)

a) *SRCNN*: The super-resolution convolutional neural networks (SRCNN) is consists of 3 layers. [1]

- Patch extraction and representation: this operation extracts (overlapping) patches from the low-resolution image and represents each patch as a high-dimensional vector. These vectors comprise a set of feature maps, of which the number equals the dimensionality of the vectors.

- Non-linear mapping: this operation nonlinearly maps each high-dimensional vector onto another high-dimensional vector. Each mapped vector is conceptually the representation of a high-resolution patch. These vectors comprise another set of feature maps.
- Reconstruction: this operation aggregates the above high-resolution patch-wise representations to generate the final high-resolution image. This image is expected to be similar to the input images. [1]

b) *SRGAN*: A super-resolution generative adversarial network (SRGAN) was proposed by C. Ledig et al. in 2017 [2] which the mean-opinion-score (MOS) scores of obtained images are closer to those of the original high-resolution images than images obtained with other methods. The structure of SRGAN was presented by replacing CNN algorithm in SRCNN structure with GAN algorithm. The GAN consists of a discriminator network D which adopts the VGG network [3] and a generator network G which uses a ResNet structure [4]. In SRGAN, the generator network G tries to generate super-resolution (SR) images from low-resolution (LR) images while the discriminator D tries to identify between SR images generated from generator network G and high-resolution (HR) images. According to the performance of SRGAN, there are many works developed which can be obtained with a satisfied result.

### B. Object detection

Object detection is among the classical computer vision problems to identify which objects are in the image and their corresponding locations. The object detection issue is more complex than the classification problem that consists of recognizing objects but without indicating their locations in the image. Moreover, images containing more than one object cannot be classified. [5]

YOLO (You Only Look Once) model structure is one of the most popular deep convolution neural models for object detection, due to its good performance and short time requirements. YOLOv5 model structure is described in Fig. 1.

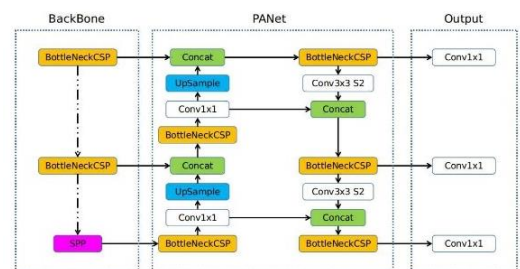


Fig. 1 Overview of YOLOv5 model structure

### C. Optical character recognition

Optical character recognition (OCR) is a method used in converting text images such as scanned images, handwriting images, photo-taken text images, or license plate images into editable text. OCR is a very useful and popular method used in various applications [6]. In license plate recognition study, OCR has been playing an important role in automatic license plate characters extraction [7] [8]. There are many techniques and tool based on OCR. Tesseract is an open-source optical character recognition engine which was developed between 1984 to 1994 [9].

### III. DATASET

In this study, the datasets were collected from 2 sources. According to the hypothesis that the super resolution technique can enhance the performance such as an accuracy of license plate numeric character detection and extraction, the first dataset has to be collected as the high resolution images for evaluating the test result compared with known license plate numbers. The second dataset was collected from the real world to show the result from super resolution and OCR techniques. Therefore, the license plate number in this dataset may be blurred and unreadable.

#### A. Train/Test dataset

The dataset for training and testing has to be collected with high resolution and have clear license plate numbers. The images were collected by using a smartphone camera when the cars were stationary in the daytime. The images were collected from a military base entrance in Thailand. An example of the images is shown in Fig. 2. The collected images were splitted to 16 images train dataset and 20 images test dataset.

#### B. Real world dataset

The real world dataset was collected by using a car camera attached to the front window inside a car. The images were taken while the car was both stationary and moving. Fig. 3 is an example of real world images.



Fig. 2 a train/test image



Fig. 3 a real-world image

### IV. METHODOLOGY

#### A. Downsampling

For image preparation, we downsized the image by 4 times its original size to compress the pixels, then we upsized it to be about half of the original size to obtain the low-resolution image. Then we input those images into a super-resolution deep learning model.

#### B. Object detection

In this paper, we used YOLOv5 model that was used to build our proposed learner model. To train this model, the SGD optimizer was used with an initial value of  $10^{-2}$  and a batch of size 16. All the other parameters are the standard parameters of the YOLOv5 code.

In the training phase, the YOLOv5 model learns to recognize all the clusters with 300 epochs. At the inference phase, once the model detects the bounding boxes, the center for each cluster is computed and this value will be the initial value for the initial cluster [5]. As YOLOv5 being very efficient and lightweight, it can detect quickly objects (clusters) and we used implemented on GPU.

#### C. Super-resolution (SR)

a) *SRCNN*: We used the pre-trained weight for the model [15], so we can bypass the training phase and start at the testing phase. The structure of the model can be seen in Fig. 4. The model has 3 layers. Where  $n$  is the number of filters and  $f$  is the filter size. The set of hyperparameters that we used are:  $n_1 = 128$ ,  $n_2 = 64$ ,  $n_3 = 1$  and  $f_1=9$ ,  $f_2 = 3$ ,  $f_3 = 5$  respectively. As for the activation function, we used Relu after each of the first two layers to add non-linearity and we used Adam as an optimizer, with a learning rate equal to 0.0003. [15] Before we input the image into the model, we have to preprocess it a little by cropping it so that it can have a divisible image size when passing through the kernel and convolutional layers, then we convert the image from BGR to YCrCb channel(3-channel-image) and normalize it as well as set the input channel into 1. because SRCNN works on 1-dimensional input or 3D inputs with depth 1. After this, we put the images into the model. Then, we have to post-process the image by unnormalizing it and crop the border size so that it has the same size as the input image. Lastly, we convert it back to the BGR channel and compare it with the input image to obtain PSNR and SSIM values.

b) *SRGAN*: In this study, Martin Krasser's pre-trained weight and model [10] was used for generate HR images from LR cropped images. The pre-trained model consists of generator and discriminator model. The generator of Martin Krasser's model was built from SR resnet algorithm which consists resnet blocks. A resnet block combine many layers such as Conv2D layers, BatchNormalization layers, and PReLU layers, complexly. The pre-trained generator in a model python file (srgan.py) was imported as a library. Next, the pre-trained weight (gan\_generator.h5) was loaded into the model. Then, super-resolution (SR) of the cropped LR images from test dataset was implemented in the SRGAN model.

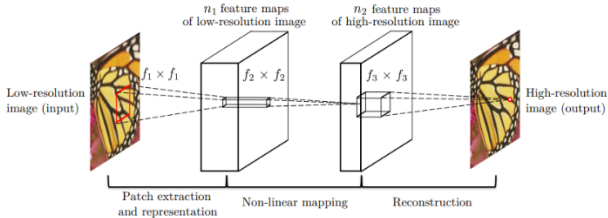


Fig. 4 The structure of the SRCNN model [1]

#### D. Optical character recognition using Tesseract

In this method, the numbers in cropped license plate image from 3 difference models (SRCNN, SRGAN, and LR) were extracted by using OCR technique. The image augmentation techniques were used for pre-processing the images before OCR implementation. The image augmentation steps following The AI Guy's GitHub [11] are image resizing, Gaussian blurring, median blurring, Otsu thresholding, element structuring, and dilating, respectively. After image augmentation, sorted contours was performed to find the rectangle boundary of each number or character. Next, numbers of license plate in pre-processed images were extracted by using Pytesseract, a Tesseract library for python.

### V. RESULT AND DISCUSSION

In this paper, we used two image quality metrics. The peak-signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) were compared the proposed method with previous Down sampling methods.

#### A. Peak Signal-to-Noise Ratio ( PSNR )

The PSNR is generally used to evaluate the quality between the original image and the resultant image in decibels, based on inversely proportional to the MSE (Mean Squared Error) [12]. this shows that a higher PSNR value provides a higher image quality. the PSNR is described by (1) and (2).

$$PSNR = 20 \log_{10} \left( \frac{MAX}{\sqrt{MSE}} \right)^2 \quad (1)$$

$$MSE = \frac{1}{m} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (2)$$

Here, MAX is the highest possible value of the image.

According to Fig. 5, blue line is the PSNR value of SRCNN which is less than orange line that is the PSNR value of SRGAN.

#### B. Structural Similarity Index Measure (SSIM)

The SSIM is a well-known quality metric used to measure the similarity between two images. It was developed by Wang et al. [13], and is correlated with the quality perception of the human visual system (HVS). The difference with other techniques such as MSE or PSNR is that these approaches estimate absolute errors [14]. The SSIM is defined as the Equations (3).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3)$$

With:

- $\mu_x$  is the average of  $x$ ,
- $\mu_y$  is the average of  $y$ ,
- $\sigma_x^2$  the variance of  $x$ ,
- $\sigma_y^2$  the variance of  $y$ ,
- $\sigma_{xy}$  the variance of  $x$  and  $y$ ,
- $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$  two variables to stabilize the division with weak denominator,
- $L$  the dynamic range of the pixel-values (typically this is  $2^{\#bits\_per\_pixel} - 1$ ,
- $k_1 = 0.01$  and  $k_2 = 0.03$  by default

According to Fig. 6, blue line is the SSIM value of SRCNN and orange line that is the SSIM value of SRGAN. Both is not much different.

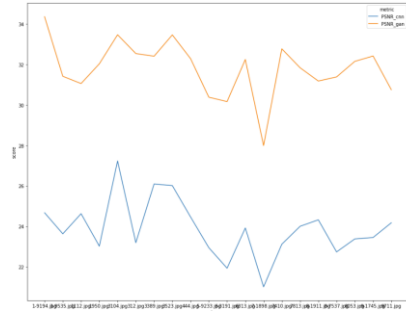


Fig. 5 PSNR Comparison of the SRCNN and the SRGAN models

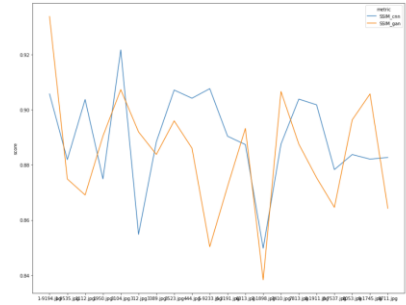


Fig. 6 Comparison of the SSIM of the SRCNN and the SRGAN

TABLE I. EXAMPLE OF LOW RESOLUTION AND SUPER RESOLUTION IMAGES

Low Resolution	Super Resolution	
	SRGAN	SRCNN



The result of super resolution images in Table I is not much different by human vision. So, different of image quality can be observed in graph (Fig. 5 and Fig. 6) by using PSNR and SSIM metric.

#### A. Accuracy

The experimental results verify that the two proposed algorithm (SRCNN & SRGAN), before images were used by Tesseract OCR algorithm for recognition that can improve the accuracy rate. In this paper, we evaluate only numeric (0-9) between super resolution and low resolution. If results are character or symbol. Results are wrong and show that in predicted results.

Table II presents the results were used by Tesseract OCR algorithm for recognition. The license plate accuracy rate of the low-resolution images is 81.4%. SRGAN accuracy rate is 84.88%. The best result is SRCNN which is 97.67%.

#### B. Confusion matrix

We used Confusion matrix for represent summarizing the performance of a numeric classification that can show what are often the wrong numeric by classification.

Table III represent the result of numeric classification from each processing. Some of result can be not detected. We replace them by “?” symbol. Besides, some of result are not numeric. We show them by themselves.

The best resolution processing is SRCNN that can see example result in the table IV. SRCNN can improve image quality for support license plate recognition by using Tesseract OCR.

TABLE II. ACCURACY COMPARISON OF LR, SRGAN AND SRCNN

Accuracy metric	LR	SRGAN	SRCNN
Correct (%)	81.4	84.88	<b>97.67</b>
Mis-detected (%)	3.49	11.63	<b>2.33</b>
Un-detected (%)	15.12	3.49	<b>0</b>

TABLE III. COMPARISON OF GROUND TRUTH RESULTS AND PREDICTED RESULTS

Ground truth	LR	SRGAN	SRCNN
312	3?2	3 2	312
444	444	444	444
1112	??22	1112	1112
1950	?950	1950	1950
3104	31Q4	31Q4	3104
3389	3389	338Q	3389
3523	3523	3523	3523
6313	63?3	6313	6313
7410	7410	7410	7410
7813	78?3	78?3	7813
8053	a053	?053	a053
9711	97??	971?	971q
19194	?9?Q4	1Q1QA	19194
19535	?9535	195a5	19535
59233	59233	59233	59233
63191	63191	631Q1	63191
71898	?7898	f1898	71898
81911	81911	81911	81911
87537	87537	87537	87537
91745	91745	91T45	91745

TABLE IV. EXAMPLE OF COMPARISON OF GROUND TRUTH RESULTS AND PREDICTED RESULTS

Ground truth	LR	SRGAN	SRCNN
	7410	7410	7410
	31Q4	31Q4	3104
	3389	338Q	3389
	?9?Q4	1Q1QA	19194

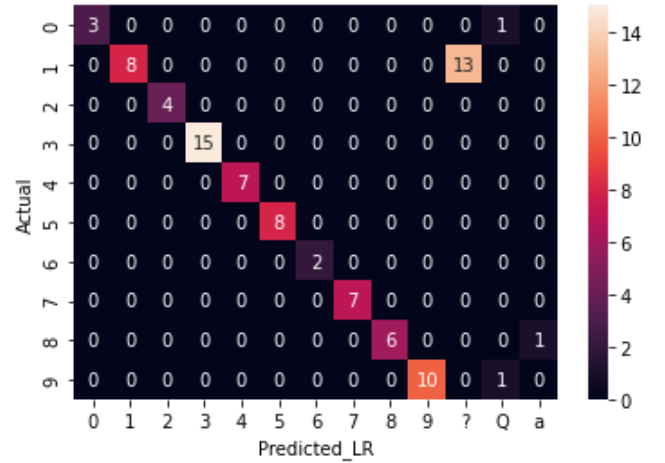


Fig. 7 Numeric-level confusion matrix of LR predicted

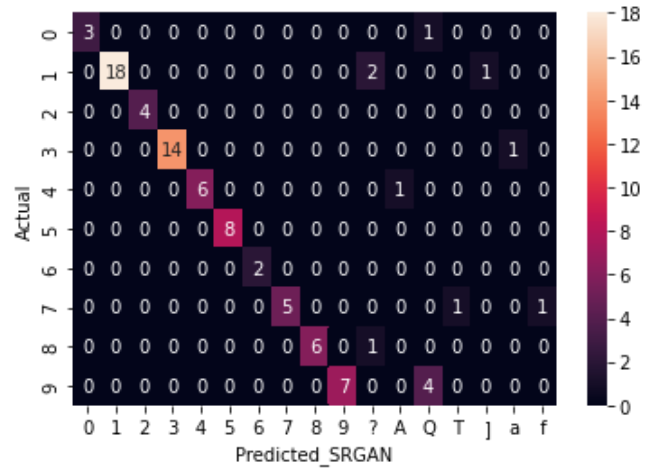


Fig. 8 Numeric-level confusion matrix of SRGAN predicted

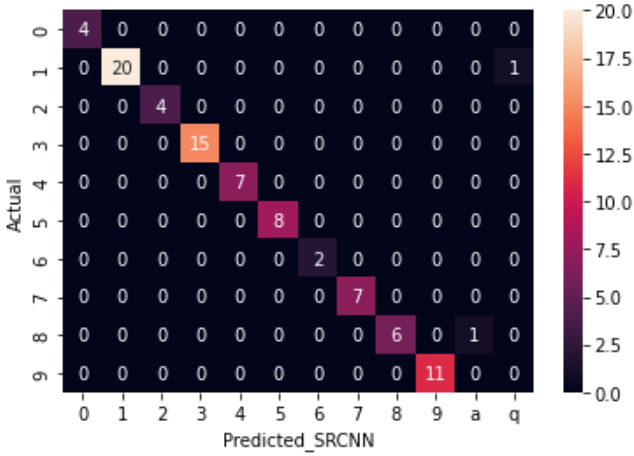


Fig. 9 Numeric-level confusion matrix of SRCNN predicted

Fig. 7 represent performance of LR that have 13 missing values which Tesseract OCR cannot detect the numeric in license plate. However, after we used SRGAN and SRCNN (Fig. 8 and Fig. 9) for improve image resolution that can decrease the number of missing value and improve the accuracy of classification. In SRCNN case, Fig. 9 that show the zero of missing values for license plate recognition.

### C. Real World Situation

After, we explore super resolution that can help to improve license plate recognition by using Tesseract OCR. So, we try to use these SRGAN and SRCNN in real world situation. We show the result in Table V.

## VI. CONCLUSION

In the experiment, it showed that our first hypothesis is correct, super-resolution did improve the performance of character extraction. But the conclusion for the second hypothesis is a little bit ambiguous. For this experiment, overall, SRGAN performs better than SRCNN. In terms of PSNR values, SRGAN is higher than SRCNN. Even though SRCNN accuracy is higher when performing OCR on the high-resolution train/test image. but when we used a real-world image for super-resolution, SRGAN perform much better than SRCNN. The reason for this may be the size of the real-world images input into the model is much smaller than the train/test images, which causes the SRCNN performance to drop significantly. because there are a lot of factors that influence the results of this experiment such as the weight of the model, the architecture of the model, the image preprocessing method, and the hyperparameters used. To clearly state which deep learning model is better required more experiments to be done.

## VII. FUTURE WORK

In this experiment, due to the number of real-time data that we collected being quite limited, we used the pre-trained weight for the image super-resolution technique. So, this aspect can be improved by collecting more data and training the model by ourselves in the future. There may also be some combinations of hyperparameters that can be adjusted to further improve the result than the ones we did, So, we could also improve this aspect in the next work by trying out more different sets of hyperparameters.

TABLE V. EXAMPLE OF LOW RESOLUTION AND SUPER RESOLUTION IMAGES

Low Resolution	Super Resolution	
	SRGAN	SRCNN

## REFERENCES

- [1] T. C. Dong, C. C. Loy, K. He and Xiaoou, "Image Super-Resolution Using Deep Convolutional Networks," 2015.
- [2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, 2017.
- [3] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum., "Image super-resolution using gradient profile prior.," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [4] K. He, X. Zhang, S. Ren, and J. Sun., "Deep residual learning for image recognition.," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [5] R. Couturier, H. N. Noura, O. Salman, A. Sider, "A Deep Learning Object Detection Method for an Efficient Clusters Initialization," 2021.
- [6] C. Patel, A. Pate, D. Patel, "Optical Character Recognition by Open Source OCR," *International Journal of Computer Applications*, vol. 55, no. 10, 2012.
- [7] B. V. Kakani, D. Gandhi, S. Jani, "Improved OCR based Automatic Vehicle Number," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, India, 2017.
- [8] S. Tanwar, A. Tiwari, and R. Chowdhry, "Indian Licence Plate Dataset in the wild," in *eprint arXiv:2111.06054*, 2021.
- [9] S. Ray, "An Overview of the Tesseract OCR Engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, Brazil, 2007.
- [10] K. Martin, "github," [Online]. Available: <https://github.com/krasserm/super-resolution>. [Accessed 4 December 2021].
- [11] The AI Guy, "github," [Online]. Available: <https://github.com/theAIGuysCode/yolov4-custom-functions>. [Accessed 4 December 2021].
- [12] F. A. Fardo, V. H. Conforto, F. C. de Oliveira, S. Paulo, "A Formal Evaluation of PSNR as Quality Measurement," *Centro Universitário da FEI, S-ao Paulo, Brazil*, 2016.
- [13] A. C. B. H. R. S. a. E. P. S. Z. Wang, "Image quality assessment: from error visibility to structural," *IEEE Transactions on Image Processing*, vol. 13, pp. 600-612, 2004.
- [14] A. Horé, D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *International Conference on Pattern Recognition*, Sherbrooke (Québec), Canada, 2010.
- [15] kunal-visoulia, "GITHUB," 25 February 2019. [Online]. Available: <https://github.com/kunal-visoulia/Image-Restoration-using-SRCNN>. [Accessed 13 November 2021].

