License Plate Image Resolution Enhancement Using Super-Resolution Generative Adversarial Network

Yuzheng Mei
Dept. of Electrical and Computer Eng.
Georgia Southern University
Statesboro, Georgia, USA
ym01245@georgiasouthern.edu

Mark Moelter
Dept. of Electrical and Computer Eng.
Georgia Southern University
Statesboro, Georgia, USA
mm15116@georgiasouthern.edu

Rami J. Haddad
Dept. of Electrical and Computer Eng.
Georgia Southern University
Statesboro, Georgia, USA
rhaddad@georgiasouthern.edu

***Abstract*—Car license plates play a crucial role in vehicle identification. However, obtaining high-quality images of license plates, particularly in real-world scenarios, poses a significant challenge due to limitations in camera resolution. To address this issue and enhance image quality for license plate identification, this research proposes a novel approach employing two Convolutional Neural Networks: a Generator and a Discriminator. These networks are trained simultaneously within a Generative Adversarial Network framework. The primary objective is to improve the visual quality of low-resolution license plate images to facilitate accurate tag recognition. The training process involves generating high-resolution (HR) license plate images with random tag numbers, followed by downsampling and inputting them into the Generator to produce an upscaled super-resolution (SR) image. The Discriminator then discerns whether the image is SR or HR, and this feedback refines the Generator's performance. Low-resolution images spanning from 16x16 to 128x128 are up-scaled to super-resolution, ranging from 32x32 to 256x256. Evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR) and Optical Character Recognition (OCR), are employed to compare the resultant SR images with interpolated images of the same size combinations. Notably, the PSNR of the SR images is nearly double that of the interpolated low-resolution (LR) images. Additionally, the OCR accuracy of LR images, such as 16x16 and 32x32, improved to 66.59% and 71.28% from 0.02% and 5.74%, respectively. This proposed image enhancement method was utilized for license plate image enhancement; which will offer law enforcement agencies a powerful tool to extract actionable intelligence from surveillance footage.**

***Index Terms*—Deep learning, computer vision, generative adversarial networks, license plate recognition, synthetic dataset.**

I. Introduction

According to reports from the National Highway Traffic Safety Administration (NHTSA), 2015 witnessed a staggering 737,100 hit-and-run crashes across the United States [1]. This alarming figure equates to a hit-and-run incident occurring approximately every 43 seconds. These incidents range from causing mere property damage to tragic fatalities, often resulting in protracted investigations that frustratingly remain unsolved. The disheartening reality is that the rates of successfully resolving hit-and-run cases are extremely low. A 2021 report revealed that only a mere 10 percent of such cases are solved nationwide [2].

However, with the rapid proliferation of surveillance cameras across the United States in recent years, they have emerged as invaluable tools to act as eyewitnesses in hit-and-run incidents and subsequently aid in resolving these cases. Prior to the widespread deployment of surveillance cameras, law enforcement's primary method of identifying a suspect's vehicle relied heavily on eyewitness accounts. Unfortunately, these witnesses often could only recall basic details such as the make and color of the vehicle, resulting in a relatively large pool of potential suspects. However, the advent of surveillance cameras has revolutionized this process. Law enforcement now has the ability to access video footage captured by these cameras, which can provide a more accurate means of identifying vehicles through their registration tags. Yet, even with the advantages offered by surveillance cameras, there remain challenges. Frequently, the images of license plates captured by these cameras are characterized by low resolution, awkward angles, and considerable noise. Even the use of a 4K camera mounted on a vantage point fails to capture a high-resolution image of a license plate when zoomed in, rendering the registration tag unreadable.

Addressing this issue, this paper introduces a solution harnessing the power of machine learning and deep learning techniques. Specifically, we propose the utilization of the Super Resolution Generative Adversarial Network (SR-GAN) architecture, which trains two convolutional neural networks simultaneously: a generator and a discriminator. Once these neural networks are trained, the generator can take low-resolution images and enhance their quality, resulting in more readable and detectable license plate images while retaining the crucial original information contained in the images.

II. Background

A. GAN Structure

A Generative Adversarial Network (GAN) is a fundamental neural network architecture comprised of two essential components: a generator and a discriminator. These two networks operate in tandem, learning simultaneously. The generator's primary function is to create synthetic images that closely resemble real training data, with the generated images serving as negative examples for the discriminator. The discriminator,

SoutheastCon 2024 | 979-8-3503-1710-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/SOUTHEASTCON52093.2024.10500226

on the other hand, is tasked with distinguishing between the fake images generated by the generator, labeled as '0,' and the authentic training data, labeled as '1.'

During the initial stages of training, the generator produces obviously fake data, allowing the discriminator to easily discern the disparity between fake and real data. However, as the training progresses, the generator becomes more adept at generating fake data that closely mimics the real dataset. The increased fidelity of these synthetic images confounds the discriminator, leading to a decline in its accuracy. Upon completion of the training, the discriminator is typically discarded, and the generator is retained for future use [3].

B. SR-GAN

The Super Resolution Generative Adversarial Network (SR-GAN) shares its foundational structure with the GAN but introduces notable modifications in the generator's layer map. Unlike traditional GANs, where random noise serves as input for the generator, SR-GAN takes a low-resolution image as its input. The generator's role in this context is to upscale the input image by a factor of two, resulting in an image with higher resolution and enhanced details [4]. The SR-GAN operation is illustrated in Figure 1.

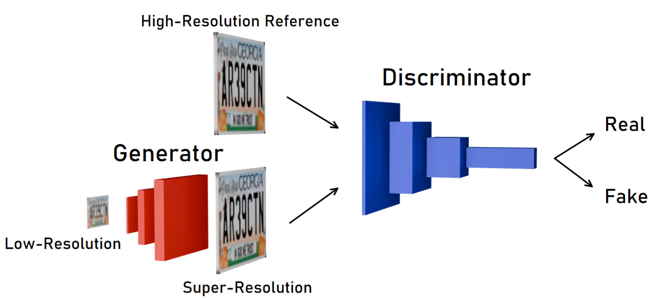

Fig. 1. SRGAN Flowchart

C. Loss Function

Loss functions are pivotal in deep learning, serving to quantify the disparity between actual data and predicted values. Various types of loss functions exist, including mean square error, mean squared logarithmic error loss, and binary cross-entropy. The output value of the loss function plays a crucial role in updating the neural network's weights.

In the GAN framework, the discriminator's classifications are integrated into the loss function to measure the dissimilarity between the training data and the generator's predictions. The resulting output is then used to adjust the weights of both the discriminator and the generator, refining their capabilities over the course of training. This iterative process drives the network to produce more realistic, high-quality synthetic images [3].

III. Methods

In this proposed solution, the two neural networks were constructed and trained using MATLAB. Their primary objectives were to upscale input pixel sizes and deblur images of car license plates. Employing the SRGAN model, four distinct input low-resolution sizes were up-sampled and deblurred. A critical addition to this model was the generation of datasets on which both networks were trained. This approach significantly reduced the need for extensive preprocessing of real-world data, considering that images captured by cameras may not be perfectly rectangular. Leveraging generated data expedited the data collection process and mitigated bias during training.

A. Datasets Generation

The dataset played a pivotal role as the ground truth for the model. High-resolution training data underwent down-sampling before being fed into the SRGAN. All training images shared a consistent background – the "Georgia State 2012 Peach State" template, as depicted in Figure 2. This template featured a peach tree, seven identically spaced black characters, peaches in the bottom corner, "Peach State" in the top left corner, and "Georgia" in the top right corner.

Fig. 2. 2012 Peach State License Plate Template

Standard Georgia license plates encompassed seven possible character positions, including characters ranging from A to Z and 0 to 9. To eliminate bias during training, a character generation pattern was introduced. Six characters were randomly selected, while one character, occupying a fixed position, iterated through all 36 possible characters. This process was replicated for all seven potential positions, yielding a set of 36 * 7 unique license plate images.

This system facilitated the effortless generation of multiple datasets, each containing 252 images, thus creating a substantial and diverse dataset. Subsequently, the datasets were partitioned into a 70:30 ratio for training and validation purposes.

B. Network Architecture

The architectural configuration of both the generator and the discriminator networks was depicted in Figure 3. The labels on top of the convolution layer indicated essential parameters: kernel size (k), output channel size (n), and stride (s). All Leaky Rectified Linear Unit (LReLU) functions adhered to a consistent hyperparameter value of 0.2 for the negative slope [5].

1) Generator Layers: The generator consisted of a total of nine B-residual blocks. Each B-residual block comprised the following layers:

- Convolution layer - k3n64s1
- LReLU activation layer
- Convolution layer - k2n64s1
- Element-wise sum layer

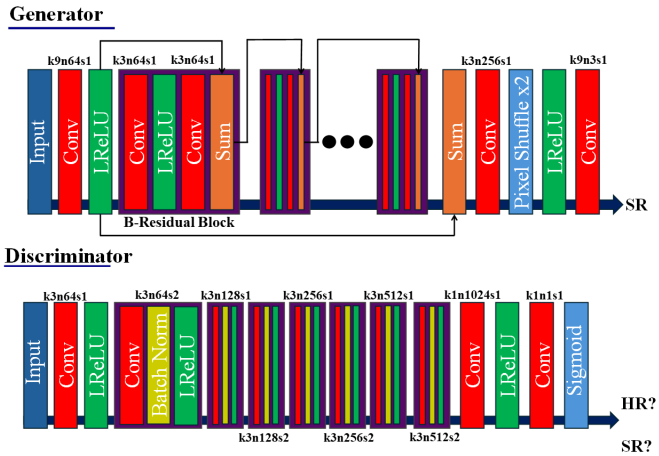

Fig. 3. Layer Map for Generator and Discriminator

Each block featured an element-wise sum layer establishing a connection to the previous block.

2) Discriminator Layers: The discriminator network was responsible for feature extraction from the input image and subsequent flattening, aiming to classify the input image as either high-resolution (1) or super-resolution (0). The convolution blocks within the discriminator comprised the following layers:

- Convolution layer
- Batch Normalization layer
- LReLU activation layer

The kernel size remained constant at 3 for all convolution blocks. Nevertheless, the output channel size began at 64 and doubled every two blocks, reaching a final value of 512. Additionally, the stride alternated between 1 and 2 throughout the convolution blocks.

C. Datasets Preprocessing

While image preprocessing to identify the region of interest was unnecessary, specific steps were essential before the training and testing phases. First, the original images were down-sampled to achieve the desired high-resolution reference images. After obtaining the high-resolution reference images, Gaussian blur was applied to them. Then, the blurred high-resolution images were further down-sampled by a factor of two to obtain the low-resolution images. After generating the low-resolution images, they were interpolated back to the same size as the high-resolution images. During the training process, only the high-resolution reference and low-resolution images were utilized. However, the testing process incorporated all three types of preprocessed images.

D. Training Process

The generator's capability to upscale images was limited to a factor of 2 in each iteration. Achieving higher upscaling factors of 4 or 8 times necessitated the training of multiple networks, collaborating sequentially to achieve the desired upscaling.

In the initial step, low-resolution images were input into the generator to generate super-resolution images. Subsequently, both the high-resolution reference and super-resolution images underwent classification by the discriminator. The discriminator's classification outputs were integrated into the loss function to optimize the weights of both networks. The training process followed the subsequent steps:

1) Down-sample the original image to obtain the desired high-resolution image.
2) Apply Gaussian blur to the high-resolution image.
3) Down-sample the blurred high-resolution image to produce a low-resolution image. the low-resolution image into the generator to obtain the super-resolution image.
4) Present both the high-resolution and super-resolution images to the discriminator.
5) Obtain the discriminator and generator losses.
6) Adjust the weights of both the discriminator and generator.
7) Preserve the trained generator for future utilization.

Upon the successful training of the initial two-times upscale network, it was employed to generate training images for subsequent networks. The key distinction in this process was the additional preprocessing steps, involving down-sampling the high-resolution images by half of the desired upscale factor before applying Gaussian blur. Following this, the blurred images were down-sampled by a factor of 2. The earlier steps were then applied iteratively to obtain the low-resolution input. The subsequent stages of the training process mirrored those of the initial step.

E. Validation

To validate the trained generator, the validation set of the dataset is first blurred and down-sampled to match the low-resolution size that the generator was trained on. Subsequently, the low-resolution images are passed through the generator to produce super-resolution images. The low-resolution images are also interpolated to reach the high-resolution pixel size. Then, the super-resolution and interpolated images are compared to their respective high-resolution counterparts using Peak Signal-to-Noise Ratio (PSNR) to assess their similarity.

The super-resolution images are subjected to Google PyTesseract Optical Character Recognition (OCR) to attempt character recognition and compare the results with the actual tag characters [6]. Before being processed by the OCR, the images are cropped to isolate only the character portion. They are then converted to grayscale and binarized to remove the background.

The OCR output consists of a string of detected characters. To determine the accuracy of the OCR result, a bitwise comparison is made between the OCR output and the actual tag characters. The correct characters are counted, and the count is divided by the total number of characters to calculate the percentage accuracy. It should be noted that the pre-trained OCR may not always perfectly convert images to strings, and in cases where the OCR output has fewer than 7 characters, all 7 positions are considered incorrect.

IV. Results

The result of the Super Resolution Generative Adversarial Network (SR-GAN) significantly improved the quality of the license plate. In this section, the performance of the SR-GAN is evaluated in three categories: Visual Presentation, Peak Signal to Noise Ratio (PSNR), and PyTesseract Optical Character Recognition (OCR) results.

A. Visual Presentation

While visual presentation may not provide as objective data as systematically calculated values like PSNR, it remains an important aspect to consider. After successfully training the two networks in MATLAB, the validation dataset was subjected to blurring and downscaling to match the dimensions in which the networks were trained, as illustrated in Figure 4.

Fig. 4. Low-Resolution input in different sizes

The blurred low-resolution images are input into the trained generator, as depicted in Figures 5, 6, 7, and 8. In each set of images, the left image represents the super-resolution output generated by the generator, the middle image depicts the interpolated result of the low-resolution input, and the right image serves as the high-resolution reference. A significant improvement in image quality is evident. The super-resolution image closely resembles the high-resolution reference, albeit with some residual noise in the background. Despite these minor imperfections, the registration tag remains readable. With larger input image sizes, the interpolated image gradually incorporates more information from the low-resolution input, yet it remains markedly inferior in quality compared to the super-resolution counterpart.

Fig. 5. 16X16 input enhanced to 32X32, 64X64, and 128X128

Fig. 6. 32X32 input enhanced to 64X64, 128X128, and 256X256

Fig. 7. 64X64 input enhanced to 128X128 and 256X256

Fig. 8. 128X128 input enhanced to 256X256

B. Peak Signal-to-Noise Ratio

The Peak Signal-to-Noise Ratio (PSNR) quantifies the ratio of noise relative to the signal with respect to the reference. A higher PSNR value signifies a closer resemblance between the signal and the reference.

In the context of this model, the PSNR value assesses the noise between the input image, whether super-resolution or interpolated, and the high-resolution reference image. The PSNR values for both super-resolution and interpolated images are presented in Table I. The left column denotes the input pixel size of the pre-upscaled low-resolution images, while the second row indicates the output size of the super-resolution and interpolated images. As depicted in the table, the PSNR value for super-resolution images is nearly double that of interpolated images. This discrepancy demonstrates that super-resolution images exhibit significantly less noise compared to their interpolated counterparts.

TABLE I
PSNR OF SR AND INTERPOLATED IMAGES

PSNR(dB)		Output Size							
		32		64		128		256	
		SR	INT	SR	INT	SR	INT	SR	INT
Input Size	16	24.3	9.9	19.5	10.3	20.5	10.2	-	-
	32	-	-	27.0	13.3	23.6	13.1	25.1	13.3
	64	-	-	-	-	27.8	16.7	30.1	17.4
	128	-	-	-	-	-	-	31.3	23.0

C. Optical Character Recognition

Optical Character Recognition (OCR) was employed to verify if the network maintained character integrity while deblurring and increasing image size. However, it is worth noting that the pre-trained OCR does not perfectly distinguish certain characters, such as ”0” and ”O,” ”1” and ”I,” or ”8” and ”B”. Therefore, to benchmark the performance of the OCR algorithm, the highest-resolution reference images were fed to the OCR to obtain the best accuracy that can be achieved. As demonstrated in Table II, the OCR accuracy of the highest resolution reference image is 74.01%. All three sizes of the reference images exhibit similar accuracy and were used as benchmarks for evaluating the super-resolution images’ accuracy.

TABLE II
OCR ACCURACY OF HR REFERENCE

Size	Accuracy
256	74.01%
128	75.64%
64	74.26%

For the low-resolution input, as shown in Table III, the OCR accuracy is notably low, especially for the 16 by 16 and 32 by 32 images, which are almost unrecognizable by the OCR. The 64 by 64 images show relatively higher accuracy compared to the lower-resolution counterparts but still fall short of the reference images.

TABLE III
OCR ACCURACY OF LR INPUT

Size	Accuracy
64	67.66%
32	5.74%
16	0.02%

Table IV illustrates the OCR accuracy of super-resolution images. The first column indicates the size of the input low-resolution image, while the second row represents the output super-resolution image size. The generator successfully enhances the readability of the license plates. After upscaling the low-resolution images to the highest pixel size using the generator, OCR accuracy significantly improves, particularly for the 16 by 16 and 32 by 32 images, where accuracy increased from 0.02% and 5.74% to 66.59% and 71.28%, respectively. For larger input sizes, the super-resolution image’s OCR accuracy approaches that of the reference images, indicating successful restoration by the generator.

TABLE IV
OCR ACCURACY OF SR IMAGES

SR Accuracy	Output Size			
Input Size	32	64	128	256
16	29.40%	58.45%	66.59%	-
32	-	64.29%	68.08%	71.28%
64	-	-	68.70%	74.06%
128	-	-	-	74.26%

V. CONCLUSION AND FUTURE WORK

In this paper, a novel approach has been introduced to enhance the image quality of vehicle license plates through upsampling and deblurring of low-resolution images using the SRGAN model. The proposed method has significantly improved the legibility of low-resolution license plate images. Multiple networks were trained to produce super-resolution images of varying sizes using input images of different sizes. A synthetic training dataset was created to mitigate bias related to the initial quality of the license plates. After model training, validation datasets were subjected to PSNR and OCR testing.

The PSNR measurement quantifies the noise in the input image relative to the reference image, and as demonstrated in Table I, the PSNR values of the super-resolution images are nearly double those of the interpolated images. Additionally, the OCR results have shown significant improvements in image readability. The OCR testing evaluates the networks’ ability to preserve character integrity while deblurring and increasing image size. The results of this evaluation demonstrated that characters in a low-resolution image of 16 by 16 pixels were identified with a mere 0.02% accuracy. In contrast, characters in the enhanced super-resolution image of 128 by 128 pixels achieved a remarkable 66.59% accuracy rate, indicating a substantial enhancement in image quality and character legibility. Higher-resolution input images, like 64 by 64 and 128 by 128, reached the OCR accuracy benchmark set by reference images. It is worth noting that the utilization of a preexisting OCR does not yield the most accurate validation results. Future work will involve training an OCR specifically tailored to license plate fonts to enhance accuracy.

The results clearly demonstrate the model’s effectiveness in enhancing the image quality of low-resolution license plate images. However, the training and validation datasets were synthetically generated to form perfect rectangles. In reality, cameras are unlikely to capture images with perfectly aligned boundaries. Future plans include integrating an algorithm to locate and isolate the region of interest in raw footage and transforming unprocessed images into a state similar to the one the models were trained on. Additionally, incorporating more variations of license plate backgrounds will enhance the model's applicability beyond the state of Georgia.

REFERENCES

[1] Benson, A., Arnold, L., Tefft, B., Horrey, W. (2017) Hit-and-run crashes: Prevalence, contributing factors, and countermeasures (research brief). AAA Foundation for Traffic Safety. [Online]. Available: https://aaafoundation.org/wp-content/uploads/2018/04/18-0058_Hit-and-Run-Brief_FINALv2.pdf

[2] Patricio G. Balona. (2023, Dec.) They took somebody from our family; Hit-and-run drivers are often never caught. Daytona Beach News-Journal. [Online]. Available: https://www.news-journalonline.com/story/news/2023/12/28/why-are-hit-and-run-killings-so-difficult-to-solve/71711253007/

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *arXiv:1406.2661*, 2014.

[4] C. Ledig, L. Theis, F. Huszár, 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114.

[5] Manoreken. (2022, May) Srgan-vgg54 single image super resolution matlab port. MATLAB Central File Exchange. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/95228-srgan-vgg54-single-image-super-resolution-matlab-port

[6] R. Smith, "An overview of the tesseract ocr engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.