Design and Evaluation of a Vehicle Detection System in Low Light Conditions

Pranav Bidare
Dept. of ECE
PES University
Bengaluru, India
pranavrbidare@gmail.com

Siri S
Dept. of ECE
PES University
Bengaluru, India
sirisrinivas@pesu.pes.edu

Param M
Dept. of ECE
PES University
Bengaluru, India
parammilindagrawal@pesu.pes.edu

Raghavendra M.J
Dept. of ECE
PES University
Bengaluru, India
raghavendramj@pes.edu

2023 10th International Conference on Signal Processing and Integrated Networks (SPIN) | 978-1-6654-9099-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/SPIN57001.2023.10116614

***Abstract*—The advent of autonomous vehicles has motivated researchers to focus their research on developing improved advanced driver assistance systems (ADAS). A key area of study in ADAS is vehicle detection. Many have proposed vehicle detection systems in daylight conditions but very few have proposed vehicle detection systems in night time. We have proposed a vehicle detection system in low light conditions in this paper. Cycle Generative Adversarial Networks (CycleGAN) is used to translate images from night to day. This makes use of unpaired image-to-image translation. Super Resolution Convolutional Neural Networks (SRCNN) is used to improve the resolution of the image. Finally, several object detector models like YOLOv4, Faster-RCNN, SSD, ResNet are used to detect vehicles. The results are compared. We have achieved a maximum mAP of 61.9% using the proposed model.**

***Index Terms*—Generative Adversarial Network, Super Resolution, Convolutional Neural Networks, Cycle GAN, Generator, Discriminator.**

I. Introduction

Advanced Driver Assistance Systems (ADAS) is a group of electronic technologies that help the driver in driving tasks. ADAS is one of the fastest growing technologies in the automotive industry. There are a lot of advantages of ADAS. It improves the safety of both pedestrians and passengers. Detection of vehicles is of utmost importance to ensure the safety of everyone. With the advent of various ADAS features, the human factor in making decisions is reduced. A vehicle detection system in low light conditions is proposed in this paper. Generative Adversarial Networks (GANs) is a machine learning algorithm. GAN consists of two neural networks namely Generator and Discriminator which contest each other in a game. Image Super Resolution (SR) is a technique where the resolution of the image is improved using various algorithms. Convolutional Neural Networks is a type of artificial neural networks which is used to perform tasks related to images. In recent times, deep learning and CNNs have been used to detect and classify objects. These machine learning (ML) algorithms have the ability to detect objects even in a live video stream, which makes them a good fit for self-driving vehicles. Several of the techniques use deep learning for extracting features and detecting objects. Some examples are You Only Look Once (YOLO), Faster R-CNN and Single Shot Detector (SSD).

II. Literature Survey

In [1], the authors present a method to translate an image from one domain to another in the absence of paired samples. They introduce two types of losses namely, Adversarial loss and Cycle consistency loss. The network includes 2 generators and 2 discriminators, one of each for each domain of images. Instance normalization is implemented. Discriminator networks use 70x70 PatchGANs. Generator networks consist of 6 Residual Blocks for 128x128 training images and 9 Residual Blocks for 256x256 for higher resolution training images. In [2], the authors proposed a 3 CNN layer architecture. The 3 layers performed patch extraction, non-linear mapping, and reconstruction. The authors obtained best results for the 3 CNN layer model. The authors achieved higher PSNR values when compared to bicubic interpolation and sparse coding based methods. The PSNR values improved with an increase in filter sizes. But the training time was longer. In [3], the authors have proposed a new object detection algorithm based on the Darknet with optimized feature concatenation, hard negative mining, multi-scale training, model pretraining, and proper calibration of key parameters. Bag of Freebies and Bag of Specials were used in the backbone and the detector to optimize the computing resources and to provide robust training. The minor drawbacks to YOLOv4 was its incapability to uniquely identify clusters of small objects hence significant localization loss which led to the presence of high false negatives in special conditions. The authors in [4] proposed a CycleGAN transformation for the images before the YOLOv4 detector model. The authors also performed gamma correction on images and compared the results. They found that the YOLOv4 worked best on the original night time images

978-1-6654-9099-3/23/$31.00 ©2023 IEEE

compared to the transformed images. By removing the bicubic interpolation and replacing it with a deconvolutional layer, the authors in [5] proposed a model with lesser computational complexity compared to the original SRCNN model proposed in [2]. The authors in [6] proposed a multi-scale super-resolution convolutional neural network (MSRCNN) to improve the resolution of the images of license plates in video footages. The proposed MSRCNN was inspired by the Inception architecture of the GoogLeNet. In [7], the authors proposed a self super resolution algorithm using deep learning as there are no external training atlases to help the model learn a mapping between low resolution and high resolution. The authors in [8] have used various CNNs models to compare the object detection in unmanned aerial vehicles.

III. Proposed Model

The proposed model consists of three parts. First block is the Cycle Generative Adversarial Networks (CycleGAN). This is used to translate a nighttime image to a daytime image. Image-to-image translation is a task that requires synthesizing a new image by a moderated modification of the given image. Usually, training such a translation model needs dataset to comprise of paired examples. For example if we wish to convert scenic pictures into Monet paintings, we will require images of scenic pictures and the corresponding Monet paintings. This however is very difficult to collect. Any two sets of images can be used. The model then extracts the general characteristics from each collection which is used in the image translation process. This is called unpaired image-to-image translation. CycleGAN uses this approach. Second block is the Super Resolution Convolutional Neural Network (SRCNN). This is used to enhance the resolution of the image. A low resolution image is converted to a high resolution image using CNN. SR is based on the idea that group of low resolution pixels can be used to create a high resolution image. Images with higher resolution offer more details of a scene. In the third block, we have used various detector models to detect vehicles in the images and compared the results.

We have used separate datasets for training CycleGAN, SRCNN and the various detector models. For training the CycleGAN model, we have used a day-night dataset from Kaggle. For training the SRCNN model we have used the T91 dataset which contains 91 high resolution images and for training the YOLOv4 model we have used images from the DETRAC dataset.

The Fig. 1 shows the block diagram of the proposed model. A night image is passed through the CycleGAN model which translates the night image into a day image. This translated image is passed through the SRCNN model which will improve the resolution of the image. Finally, this image is passed though the detectors models which identify vehicles in the image.

A. CycleGAN

CycleGAN is a machine learning algorithm which performs image-to-image translation in the absence of paired examples.

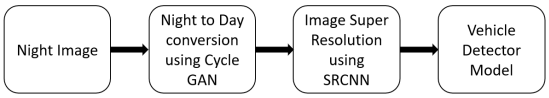


Fig. 1: Block Diagram of Proposed Model

The CycleGAN has two generators and two discriminators. One generator converts the day images into a night image and the other generator converts the night images into a day image. The two discriminators are used to train and update the generators. The CycleGAN model is also designed to maintain cycle consistency. The generator models are trained to reproduce the original source image. We then compare the images and calculate the loss. This loss is the cycle consistency loss.

Adversarial loss: This loss is applied to both mapping functions, for example, for G: X $\rightarrow$ Y and its discriminator DY, the loss objective is expressed as follows [1]:

$$Loss_{\mathrm{GAN}}(G, DY, X, Y) = \mathbb{E}_{y \sim p_{\mathrm{data}}(y)} [logDY(y)] + \mathbb{E}_{x \sim p_{\mathrm{data}}(x)} [log(1 - DY(G(x)] \quad (1)$$

This loss helps to identify between the real images and the fake images in their corresponding domains.

Cycle consistency loss: To ensure forward and backward cycle consistencies, this loss function is introduced in the model. The equation below summarizes the objective for one domain. Inversely, the same objective can be described for the other domain.

$$\mathcal{L}_{\mathrm{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\mathrm{data}}(x)} [F(G(x)) - x1] + \mathbb{E}_{y \sim p_{\mathrm{data}}(y)} [G(F(y)) - y1] \quad (2)$$

Hence the full objective of the CycleGAN model can be expressed as the below function:

$$Loss(G, F, DX, DY) = Loss_{\mathrm{GAN}}(G, DY, X, Y) + Loss_{\mathrm{GAN}}(G, DX, Y, X) + \lambda \mathcal{L}_{\mathrm{cyc}}(G, F) \quad (3)$$

The CycleGAN model is trained on 522 day images and 227 night images taken from the day-night dataset from Kaggle. Training of all models is done using Google Colab Pro. The details of the architectures of the generator and the discriminator of the proposed CycleGAN is tabulated in Table I and Table II.

B. SRCNN

Our proposed SRCNN model consists of 3 components: patch extraction, non-linear mapping and reconstruction. The low-resolution image is bicubic interpolated into X, with the same size as the low-resolution image, Y. The model aims to learn a mapping F: Y $\rightarrow$X. Our model consists of 4 convolutional layers. The first layer has filters f1 = 256, kernel size of (7,7), activation function is leakyrelu with alpha = 0.1.

TABLE I: Generator details

Layer	Activation Size
Input	3x256x256
64x7x7 conv, stride 1, pad same	64x256x256
128x3x3 conv, stride 2, pad same	128x128x128
256x3x3 conv, stride 2, pad same	256x64x64
9 Consecutive Residual Blocks	256x64x64
128x3x3 ConvTranspose, stride 2, pad same	128x128x128
64x3x3 ConvTranspose, stride 2, pad same	64x256x256
3x7x7 conv. stride 1, pad same	3x256x256

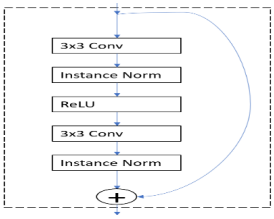


Fig. 2: Architecture of Residual Block

The second layer has filters f2 = 128, kernel size of (5,5), activation function is leakyrelu with alpha = 0.1. The third layer has filters f3 = 64, kernel size of (3,3), activation function is leakyrelu with alpha = 0.1. Finally, the fourth layer has f4 = 1, kernel size of (1,1), activation function used is linear. We have used an upscale factor of 4.

Our first layer is denoted by the operation defined by

$$A1(X) = max(0.01X, F1 * X + B1) \tag{4}$$

with LeakyRelu activation function, where F1 denotes the filters, B1 denotes the biases, * denotes the convolution operation.

This is our second layer. Using A1(X) we will forward propagate to n2 feature maps of this layer which gives us

$$A2(X) = max(0.01X, F2 * A1(X) + B2) \tag{5}$$

with LeakyRelu Activation function, where F2 denotes the filters, B2 denotes the biases, * denotes the convolution operation.

Our third layer. Using A2(X) we will forward propagate to n3 feature maps of this layer which gives us

$$A3(X) = max(0.01X, F3 * A2(X) + B3) \tag{6}$$

TABLE II: Discriminator Details

Layer	Activation Size
Input	3x256x256
64x4x4 conv, stride 2, pad same	64x128x128
128x4x4 conv, stride 2, pad same	128x64x64
256x4x4 conv, stride 2, pad same	256x32x32
512x4x4 conv, stride 2, pad same	512x16x16
512x4x4 conv, stride 1, pad same	512x16x16

with LeakyRelu Activation function, where F3 denotes the filters, B3 denotes the biases, * denotes the convolution operation.

Finally, A3(X) is used over a linear layer to get the final reconstruction of the high-resolution image.

$$A(X) = F4 * A3(X) + B4 \tag{7}$$

This layer uses a linear activation function, B4 denotes the biases, * denotes the convolution operation.

Peak signal-to-noise ratio (PSNR) represents the ratio of the maximum possible power of the signal to the noise affecting its accuracy. It is usually used to measure the quality of the reconstructed image [5]. The pixel-wise Mean Squared Error (MSE) between the reconstructed image A(X) and the original (ground truth) image Y is used. This will ensure that the PSNR is maximized during training. The MSE is calculated using Equation (8) and the PSNR is calculated using the Equation (9).

$$MSE = \frac{1}{n}\sum_{i=1}^{N} |F(Y_i;\theta) - X_i|^2 \tag{8}$$

Where Y and X denote the corresponding LR images and HR images, respectively, i denotes the different color channels, Θ=(Wi,Bi) denotes the convolution parameters of the network. F(Yi;Θ) denotes the input result of the network, and n denotes the number of training samples.

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{9}$$

The SRCNN model is trained on 91 high resolution images from the T91 dataset. The images are converted into sub images with a patch size of 32 and stride 14 as shown in Fig. 3. To avoid border effect, the padding in all layers is kept as same hence the final size of the image is reduced. For training we fixed the size of the input images but the model can be used on images of any size. The architecture details of the SRCNN model is given in Table III.

(a) Full training image (b) Patches of training image

Fig. 3: Image showing the training image and the patches

C. Detector Models

Seven different models are used for detection. YOLOv4 is a single shot detector implemented on a single CNN. Its novel structure is simple and small making it robust in training and implementation. Faster Region based CNN, latest version of RCNN which uses a sliding window approach. Inception

TABLE III: SRCNN Model Details

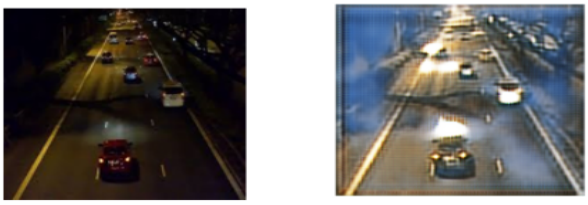
Layer	Activation Size
Input	3x256x256
256x7x7 Conv, pad valid	256x250x250
128x5x5 Conv, pad valid	128x246x246
64x3x3 Conv, pad valid	64x244x244
1x1x1 Conv, pad same	1x244x244

Resnet uses parallel architecture of multiple differently sized convolution kernels. It provides real-time capability but has low accuracy. Single Shot detector removes the regional proposal network in RCNN variants. This provides high accuracy but lacks small object detection. Mobilenet stacks convolution layers depth wise and provides interconnections. It provides lightweight computing while maintaining high accuracy at high FPS.

The YOLOv4 model is trained on images from the DETRAC dataset which is split into training and testing sets. The training dataset contains 1000 images. Night time images were also present in the training dataset. Faster RCNN Resnet50 v1, Faster RCNN Resnet101 v1, Faster RCNN Inception Resnet v2, SSD Resnet50 v1 FPN, SSD Resnet101 v1 FPN, SSD MobileNet v1 FPN, these six models are generated by transfer learning. The pre-trained model architecture and weights are copied from TensorFlow 2.0 official release. These are then trained on a custom dataset which updates all the weights. This approach is used due to the lack of computing resources.

IV. Results

The CycleGAN model has been trained on 522 day images and 227 night images using Google Colab Pro. The dataset is obtained from Kaggle. The learning rate is set to 0.0002 and the model is trained for 10 epochs. A batch size of 1 is used. The model is tested at the end of 5 and 10 epochs to ensure proper training. During training, we have also accounted for the cycle consistency of the CycleGAN model. Cycle Consistency loss, Discriminator loss and GAN loss are monitored throughout the training. The below images show the working of our CycleGAN model.



(a) Input to CycleGAN (b) Output from CYcleGAN

Fig. 4: Image showing the translation the CycleGAN performs on the night time image

Fig. 4 shows the translation of the cycleGAN. The Night time image is fed as input to the CycleGAN and a translated image is received as the output. Fig. 5 shows the cycle consistency property of the CycleGAN. Fig. 6 shows the output of the CycleGAN model after 5 and 10 epochs. From Fig. 4, we can identify that the CycleGAN is introducing some noise into the image during translation.

(a) Input (b) Generated (c) Reconstructed

Fig. 5: Image showing the cycle consistency of the CycleGAN

(a) Input to CycleGAN (b) Output after 5 epochs (c) Output after 10 epochs

Fig. 6: Image showing the input to the CycleGAN and the output of the CycleGAN after 5 and 10 epochs

The proposed SRCNN model is trained for 500 epochs on 91 high resolution images using Google Colab Pro. Validation loss of 0.00402 was achieved on the set14 + set5 images. We achieved an average PSNR value of 34.44dB for 19 test images in the set14 + set5 dataset and average PSNR Value of 32.01dB was achieved for 250 test images translated from the proposed CycleGAN model.

The SRCNN model is evaluated using the PSNR values. We calculate the PSNR values between the image before and after super resolution. The Table IV shows the average PSNR values obtained by the proposed model. We have compared the PSNR values for two image sizes for set14 + set5 images as well as the images from the CycleGAN model.

TABLE IV: Results of SRCNN Model

Image Size	Average PSNR Value (dB)	
	Cycle GAN Images	*Set 14 + Set 5 Images*
256x256	32.01	33.64
512x512	33.85	34.45

From Table IV, it can be seen that as the size of the image is increased, the average PSNR value increases.

Each of the seven models in table 5 are fed the same two sets of images for testing. Set A contains 250 night time images. Set B contains 250 daytime high-resolution images. The images in set B are a result of the night time images in set A undergoing translation and super resolution.

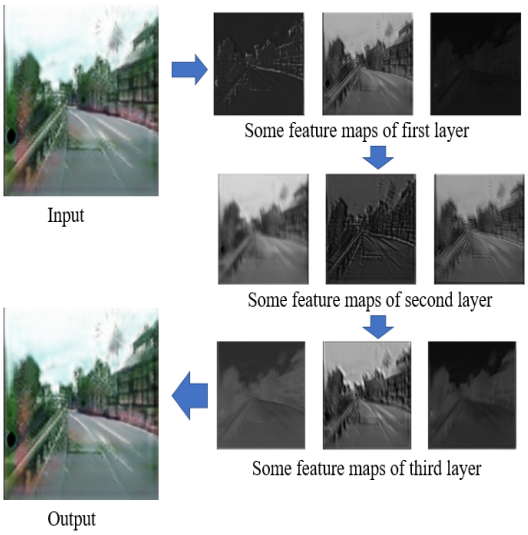
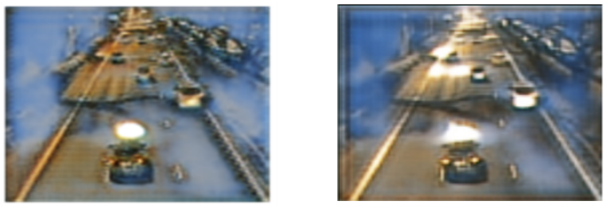


Fig. 7: Feature maps in the SRCNN model



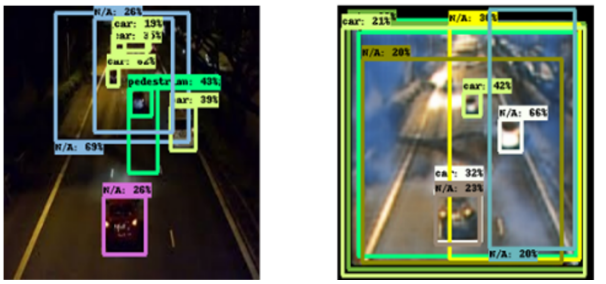
(a) Input to SRCNN (b) Output from SRCNN

Fig. 8: Input and output of the SRCNN model for translated image

The detector models then detect the vehicles in the two sets of images. Both the image outputs are then compared to evaluate the effectiveness of the translation and image super resolution tasks. We have compared the results of the night time image and the translated image. We have used various object detection models like YOLOv4, Faster RCNN Resnet, SSD Resnet and SSD Mobile. For the YOLOv4 model, the IoU threshold is kept at 0.5 to calculate the mAP and precision values.

The Fig. 9 illustrates the working of the detector model. The leftmost image is the night time image. The YOLOv4 model detects the vehicles in the image. Rightmost image is the translated and high-resolution image of the night image. The YOLOv4 model detects the vehicles present in the image. There is still some confusion between objects of the same shape (like van-car-bus) but overall, the performance acceptable. The table 5 contains the mAP values of the various detector models detecting vehicles on the two sets of images.

From the Table V, we can see that the YOLO v4 model tested on night time images shows mAP of 67.1%. The YOLOv4 model tested on the translated images has a mAP



(a) YOLOv4 detector working on night image (b) YOLOv4 detector working on translated image

Fig. 9: Image showing the detection of vehicles in the images using YOLOv4 model

TABLE V: Results of the various detector models

Model	Normal Image mAP	Translated + SR Image mAP
YOLO v4	0.671	0.619
FRCNN Resnet50 v1	0.623	0.429
FRCNN Resnet101 v1	0.393	0.425
FRCNN Inception v2	0.534	0.277
SSD Resnet50 v1	0.351	0.281
SSD Resnet101 v1	0.351	0.281
SSD Mobilenet v1 FPN	0.280	0.263

of 61.9%. This is mainly due to the noise introduced by the CycleGAN model. Due to this noise, there is some misclassification of cars as other type of vehicles. When FRCNN Resnet101 v1 was used as the detector model, the proposed system performed better on images in set B compared to the night images in set A and achieved a higher mAP.

V. Conclusion and Future Scope

In this paper, a vehicle detection system in low light conditions is proposed. The proposed model with the YOLOv4 detector model achieved a highest mAP of 61.9% using translated and super resolution images (night image is passed through CycleGAN and translated image is passed through SRCNN). The night time image without translation and super resolution achieved an accuracy of 67.1% using the YOLO v4 model. This is with hardware limitations. The training of the CycleGAN requires a large amount of RAM and a good GPU. The RAM and GPU available on Google Colab Pro was not sufficient to efficiently train the CycleGAN model. Given good hardware resources, like higher amounts of RAM and a good GPU, the proposed model may achieve better results. Our next steps will be to ensure that the proposed detector works in real time.

References

[1] J. -Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2242-2251.

[2] Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang. "Image super-resolution using deep convolutional networks." IEEE transactions on pattern analysis and machine intelligence 38, no. 2 (2015): 295-307.
[3] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, " YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020, https://arxiv.org/abs/2004.10934.
[4] N. Ho, M. Pham, N. D. Vo and K. Nguyen, "Vehicle Detection at Night Time," 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), 2020, pp. 250-255
[5] T. Wu, X. Song, T. Gan, B. Zeng and J. Chen, "Super-resolution Reconstruction of Night-light Images Based on Improved SRCNN," 2022 4th International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), 2022, pp. 1-5.
[6] Y. Yang, P. Bi and Y. Liu, "License Plate Image Super-Resolution Based on Convolutional Neural Network," 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), 2018, pp. 723-727.
[7] C. Zhao, A. Carass, B. E. Dewey and J. L. Prince, "Self super-resolution for magnetic resonance images using deep networks," 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), 2018, pp. 365-368.
[8] I. V. Saetchnikov, E. A. Tcherniavskaia and V. V. Skakun, "Object Detection for Unmanned Aerial Vehicle Camera via Convolutional Neural Networks," in IEEE Journal on Miniaturization for Air and Space Systems, vol. 2, no. 2, pp. 98-103, June 2021.
[9] G. ÖZTÜRK, R. KÖKER, O. ELDOğAN and D. KARAYEL, "Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2020, pp. 1-8.
[10] M. Schutera, M. Hussein, J. Abhau, R. Mikut and M. Reischl, "Night-to-Day: Online Image-to-Image Translation for Object Detection Within Autonomous Driving by Night," in IEEE Transactions on Intelligent Vehicles, vol. 6, no. 3, pp. 480-489, Sept. 2021.
[11] C. -T. Lin, S. -W. Huang, Y. -Y. Wu and S. -H. Lai, "GAN-Based Day-to-Night Image Style Transfer for Nighttime Vehicle Detection," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 2, pp. 951-963, Feb. 2021.
[12] C. Voreiter, J. -C. Burnel, P. Lassalle, M. Spigai, R. Hugues and N. Courty, "A Cycle Gan Approach for Heterogeneous Domain Adaptation in Land Use Classification," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 1961-1964.
[13] J. Kang, C. Sun and C. Zhu, "Cycle-GAN based Face Aging Method," 2022 International Conference on Big Data, Information and Computer Network (BDICN), 2022, pp. 616-625.
[14] J. Kim et al., "Performance Comparison of SRCNN, VDSR, and SRDenseNet Deep Learning Models in Embedded Autonomous Driving Platforms," 2021 International Conference on Information Networking (ICOIN), 2021, pp. 56-58.
[15] H. Ji, Z. Gao, T. Mei and B. Ramesh, "Vehicle Detection in Remote Sensing Images Leveraging on Simultaneous Super-Resolution," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 4, pp. 676-680, April 2020.