ASEAN Engineering Journal

Full Paper

ENHANCEMENT OF TRAFFIC IMAGES UNDER DIFFERENT WEATHER CONDITIONS USING PYNET

Miza Fatini Shamsul Azmi, Fadhlan Hafizhelmi Kamaru Zaman*, Husna Zainol Abidin

Vehicle Intelligence and Telematics Lab, School of Electrical Engineering, College of Engineering, 40450 Shah Alam, Selangor, Malaysia

Article history
Received
03 July 2023
Received in revised form
10 October 2023
Accepted
26 November 2023
Published online
31 May 2024

*Corresponding author
fadhlan@uitm.edu.my
r

Graphical abstract

Abstract

Convolutional neural networks have real practical application potentials, such as for autonomous driving and semantic segmentation, but they are known to be sensitive to image degradations and corruption. Over the years, image enhancement in deep learning has shown drastic improvement. However, the clarity and quality of images still badly affect the robustness of semantic segmentation, especially for traffic images under different weather conditions. This paper proposes image enhancement performance using PyNET deep learning methods for more robust semantic segmentation of traffic images under different conditions. This work also proposes a new metric to objectively estimate the performance known as Image Similarity Metrics (ISM). Modification to PyNET is made in this work to allow this deep learning model to be used to enhance traffic images under various weather conditions such as fog, night, and rain. We compared PyNET performance against the Deep Convolutional Networks (DPED) and the Cycle Generative Adversarial Networks (CycleGAN) to evaluate the improvement gained by these image enhancement methods. Based on our experiment, PyNET gives the best image enhancement performance among those three methods in all weather conditions according to our proposed ISM. To support the validity of the ISM result, we performed tests by semantic segmentation of traffic images using ResNet-18 on the PyNET-enhanced images. Based on semantic segmentation results, PyNET improves the semantic segmentation of traffic images under different weather conditions by as much as 17% accuracy and delivers performance that directly validates the ISM scores, by showing that PyNET delivers the best semantic segmentation improvement in fog and night images.

Keywords: Image enhancement, deep learning, semantic segmentation, traffic images

1.0 INTRODUCTION

Machine learning (ML) application has been getting a lot of attention, especially in computer vision, prediction, and semantics segmentation [1], [2]. Besides that, several industries such as healthcare [3] are using machine learning to predict disease and cancer, transportation uses ML for mapping and self-driving [3] and so does cybersecurity which prevents cyber-attacks [4]. Whereas early-day ML is used to predict the stock market and finance [5]. In retail [6], ML is utilized to improve the efficiency and time required for weighing food for labels and it is also used to analyze the social media user content [7]. Conventionally, traditional ML is a set of algorithms that will learn and apply the learning program to make decisions. A newer approach called deep learning is a sub-field of machine learning where the backbone is the neural network. Deep learning neural networks can be used to analyze massive data with human rationality to make a decision [8]. Feed-forward Neural Network, Convolution Neural Network (CNN), and Recurrent Neural Network (RNN) are basic examples of different types of neural networks in deep learning.

Deep learning neural networks will use a massive amount of data, resulting in a high computation time. Therefore, CNN is usually trained with GPU accelerators such as Caffe, Pytorch, or Tensorflow to reduce the computation time [9]. CNN is popular and has a significant role in image processing and computer vision tasks. The image processing task focuses on processing raw images to prepare them for the following tasks. Some of the image processing techniques are image enhancement [10], image restoration [11], image segmentation [12], image compression [13], image manipulation [14], image generation [15], object detection [16], and recognition [17]. Meanwhile, computer vision focuses on extracting input images or videos to create a prediction like a human brain. Here are some examples of computer vision tasks, image segmentation, pose estimation, and image classification [18].

In this work, we specifically focus on the task of enhancing noisy traffic images under various conditions. This can improve the quality of images for information extraction. There are two types of image enhancement techniques: the spatial domain and the frequency domain. The spatial domain style will perform on pixels to achieve the intended enhancement, while the frequency domain style is applied at Fourier Transform, where all pixel operates in groups [19]–[21]. Several methods used in image enhancement have created Histogram equalization (HE) [20], Wiener Filter, Deep neural network [22], and unsharp mask filtering. In this paper, the image enhancement method is used to remove blurring or noise and reveal the detail of an image under adverse weather conditions. Examples of these weather conditions are foggy weather, night, or low light, and rainy weather. We also focus only on the deep neural network method for image enhancement as it will remove noise and construct a high resolution from images that contain various types of noise while taking advantage of CNN.

Choosing the right deep learning neural network for all our conditions is non-trivial because a certain method only works well on a certain noise based on [23]. Based on [24], there are several factors affecting the deep learning neural network performance, which are the model used and setup, data structure, and learning hyperparameters. Additionally, to measure the image enhancement performance, most previous works used Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Root Mean Square Error (RMSE), Structural Similarity Index (SSIM), Universal Image Quality Index (UQI), Visual Information Fidelity (VIF), and Spectral Similarity Measurements (SAM).

Thus, the purpose of this paper is to explore the performance or effectiveness of image enhancement using several deep-learning models and to calculate the overall image quality assessment in different weather conditions for traffic images. We propose the use of a Pyramidal CNN (PyNET) architecture [25] for image enhancement, and the performance is compared against DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks (DPED) [26], and the Cycle Generative Adversarial Network (CycleGAN) [27]. Then, we introduce a new method called the Image Similarity Method (ISM) to quantify the overall performance image assessment better. Finally, we evaluate the improvement gained from image enhancement by semantic segmentation task.

1.1 Previous work on Foggy Weather Images

Bad weather, particularly fog, commonly limits drivers from observing and reacting to road conditions and when driving on a clear day. Haze removal has two removal algorithms which are a single image and a multiple image algorithm. A simple image enhancement based on haze removal was proposed in [28]. To maintain or balance the underexposure image the model proposed used the Koschmider, image processing such as detail enhancement, gamma correction, image fusion, and tone mapping to change under exposure image to be clear or its details to be restored. In [29], the authors mentioned that the main problem faced by a driver is faded visual scenes which lead to road accidents while the road is covered with fog. Therefore, this reference has proposed a new method by using deep neural networks in real-time and does not need additional information. The deep neural network for visibility enhancement will accept the foggy images as input and then defogged them as output. This method was made to function as a pair. The image used in this experiment was from the Foggy Road Image Database (FRIDA) database where the images provided are synthetic images with fog and without fog. This method is good at restoring the details of the image, but this method is carried out on greyscale images.

Previously, [30] focused on automatic license plate recognition in robust weather. The problem is that when weather such as fog and low light or at night-time the recognition system is low quality. This happened because during foggy weather the scattering effect of the atmosphere leads to the low resolution of an image. While at night the brightness and contrast of images are very low compared to daylight conditions. The data used is mixed or generated by the Python script, data from KarPlate, and the real image captured outside. They also made some changes to the generated license plate to make it more diverse and realistic. The method consists of image processing, license plate detection (LPD), and license plate recognition (LPR). The image processing consists of defogging, dehazing, and low-light enhancement. For defogging and dehazing they applied a dark channel prior algorithm and for low light enhancement, they were using Contrast Limited Adaptive Histogram Equalization (CLAHE). Then to improve the resolution image they applied Super-Resolution using a deep learning-based image super-resolution reconstruction algorithm since the hardware is high cost. The method is then tested on several conditions such as foggy image, low light image, blurred image, and basement parking. The result overall does show a great improvement.

1.2 Previous work on Poorly Lighted Images

Images captured in a poorly lit environment are often of low visibility and affect many high-level computer vision tasks such as detection and recognition. The general cause of low light can be caused by insufficient light sources, dark or shaded regions, and underground areas such as tunnels or mines. Several image enhancement methods are proposed for using the traditional algorithm such as Histogram Equalization, gamma correction, and retinex theory [31]. DriveRetinex-Net was used to enhance the low-light image as discussed in [32]. DriveRetinex-Net is a method that consists of a deep neural network and Retinex theory. In this case, it was divided into two subnetworks: Decom-Net and Enhanced-Net. The Decom-Net will decompose a color image into a reflectance and illumination map. Enhanced-Net will enhance light intensity on the map. About 10,000 images were captured on the driving scene in different conditions such as morning, afternoon, dusk, night, and rain.

In [33], the authors proposed using a stacked Sparse Denoising Autoencoder (SSDA), which consists of a Low-light Net (LL-Net) to enhance low-light images. They used both synthetic and natural images as the dataset. The images then will be transformed into patches by MATLAB and will be shuffled before it is divided into training sample or validation sample. To compare the method proposed performance, they used Histogram Equalization, CLAHE, Gamma Adjustment (GA), and Histogram Equalization with 3D Block Matching (HE+BM3D). The result shows that their method works well on natural lowlight. In [34], a new Self-Calibrated Illumination (SCI) was developed for a faster, more flexible, and robust image enhancement for real-life low-light images. The self-calibrates module will gradually correct the illumination with weight sharing between each stage. A trainable hybrid network for image enhancement called a spatially variant Recurrent Neural Network (RNN) is a novelty method proposed in [35]. RNN consists of two parts which are the content stream and the edge stream. For benchmarking, they used multiple different methods, datasets from DPED, and image assessment metrics.

1.3 Previous Work On Rainy Weather Images

Bad weather conditions do affect the accuracy of visual perception to safely navigate in an autonomous vehicle. To solve this problem, Mukhtarjee et al. [36] have proposed to apply image enhancement which consists of two end-to-end deep learning CNN for removing haze and rain from the image and the method called NVDeHazenet and NVDeRainNet. The paper considers two types of weather rain and haze or fog (rain at a far distance where it seems like fog). However, it is not applicable to a large volume of rain streaks and thick mist. For comparison, they used Deep Detail Net and DehazeNet and then calculated its PSNR and SSIM. In [22], Shi et. al. are using one method to solve night-time rain weather. The method that the paper is using is decom-net, enhance-net, and derain-net. As a result, they managed to enhance the image, remove the rain streaks, and improve the image details. The method proposed has increased the PSNR score from 22% to 29.88% and 0.97% SSIM on flower images. In the rabbit image, the PSNR is about 28% and the SSIM is 0.97%.

Meanwhile, in [37], the authors proposed a universal multiple bad weather method that is robust while working on high-level vision tasks and to develop a suitable low-memory network that will be hardware-friendly on the embedded system. This method consists of a Harmonic Dense Block (HBlock) and a Feature Identity Extraction Module (FIE) which was inspired by DenseNet. RESCAN, DeRainDrop, and PReNet were used as a benchmark for those high-level tasks in bad weather 3 tasks. It also proves that the method proposed outperforms the existing methods and can be effective in bad weather. Furthermore, a deep neural network called ResDerainNet was proposed in [38] to remove streaks of rain from low to high streaks. This method is proposed because rain streaks cause a blurring effect and degrade the image quality. Therefore, this paper's focus is the rain streaks. To de-rained rained images, they were using residual learning of CNN. The result of the output image and image assessment metrics shows that this method does remove the rain streaks very well compared to other existing methods. They adopt the dataset with two rain streaks, two raindrops, and two haze simultaneously on each image to create a robust weather effect. Besides that, they also test the effectiveness of their method based on three different tasks, which are lane detection, monocular depth estimation, and object detection.

1.4 DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks (DPED) and Cycle Generative Adversarial Networks (CycleGAN)

According to [26], DPED consists of two networks in the photo enhancer, first the image enhancement network and the discriminator network. The image enhancer network starts with a 9x9 layer followed by a 4 same block which consists of two 3x3 layers and a batch normalizer which alternates with each other. Then they used another 3x3 layer and a 9x9 layer after the residual blocks. ReLU activation is also in all layers except the output where tanh is applied. The discriminator CNN consists of five convolution layers and is followed by a LeakyReLU, and at the output, a sigmoidal activation function is applied. For benchmarking this reference uses Apple photo Enhancer (APE), Deep Convolution Network for Image Super-Resolution, and feed-forward networks. To measure the quality of the image, the result of average PSNR and SSIM on different enhancement methods was recorded.

The Cycle Generative Adversarial Network, or CycleGAN [27], is an approach to training a deep convolutional neural network for image-to-image translation tasks. The network learns a mapping between input and output images using an unpaired dataset. The model architecture consists of two generator models: one generator (Generator-A) for generating images for the first domain (Domain-A) and the second generator (Generator-B) for generating images for the second domain (Domain-B). Each generator has a corresponding discriminator model (Discriminator-A and Discriminator-B). The discriminator model takes real images from the Domain and generates images from the Generator to predict whether they are real or fake.

1.5 Image Quality Metrics

Several popular image quality metrics are Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Structural Similarity Index (SSIM), Universal Quality Image Index (UQI), Spectral Angle Mapper (SAM), and Visual Information Fidelity (VIF) [39]–[41] to measure the image quality. To calculate image enhancement performance in [20], they used PSNR, Normalized cross-correlation (NCC), Execution Time (ET), and Discrete Entropy (DE). In their cases, they used PSNR to calculate the visual quality of the image tested. In [22], the authors suggested a joint deep neural network method for nighttime rainy conditions. The objective is to remove the low visual effect that was caused by low illumination. The object in the image only contains flowers and rabbits. Their enhancement successfully removed the synthetic rain streaks and, at the same time, maintained the image details when it was zoomed in. Then to calculate the visual difference, they used PSNR and SSIM. Similar to [33], this work used PSNR and SSIM to calculate the image assessment. Besides, in [32], PSNR and SSIM are used to calculate the comparison of the proposed method, which is DriveRetinex-Net, and the existing method Retinex-Net. Based on the result obtained, PSNR and SSIM do increase by about 10 -20 percent compared to the existing method of PSNR and SSIM. While in [42], they use Root Mean Square (RMSE) to measure the changes per pixel in processing data. Table 1 summarizes several popular metrics to calculate image quality or image similarity.

1.6 Residual Networks

Residual Networks, or ResNet, is a classic neural network that is the backbone of computer vision which introduced the concept of skip connection. Based on [43], ResNet is one of the well-known and used CNNs where it consists of 3 layers, which are convolutional layers, nonlinear layers, and pooling layers.

Table 1 Summary of the image assessment metrics

Metrics	Purpose	Notes
Peak Signal-to-Noise Ratio (PSNR)	Quality measurement between the original and a compressed image	The higher the PSNR, the better the quality
Root Mean Squared Error (RMSE)	Measures the amount of error in statistical models	The lower the RMSE, the better the model
Structural Similarity Index (SSIM)	Measuring the similarity/shape overlap between two images.	Ideally, it should be zero.
Universal Quality Image Index (UQI)	Calculate the quality of an image using a loss	The higher the value the better.
Spectral Angle Mapper (SAM)	An automated method for directly comparing image spectra	The lower the value the better.
Visual Information Fidelity (VIF)	Measure the accuracy of the reconstructed image	If the VIF value is greater, it improves. If VIF is less, it loses visual quality

In [44], Res-Net is used to predict accidents in Citywide. Based on the paper reviewed, the existing method does not consider the density of the population. Therefore, the proposed method considers the road network structure, meteorological data, calendar data, and human mobility data. By using ResNet, the prediction has outperformed other existing methods and has 88.9% accuracy. The method proposed consists of a CNN feature extractor, feature fusion module, and accident prediction based on feature sequence. In [45], ResNet is used to detect and predict accidents, where three types of extraction are used, which include ResNet and Bag of Visual Words Construction using K-means as a cluster. Besides, Res-Net is also used in driving trajectory prediction [46]. Due to the residual connection, the network can learn deep layers while maintaining the performance rates.

2.0 METHODOLOGY

This section will elaborate on the framework which starts with the dataset preparation for PyNET, DPED, and CycleGAN. Then we elaborate on the proposed PyNET architecture and setup including DPED and CycleGAN are used for benchmarking. Then we discuss how to compute the overall image quality assessment using a new method called Image Similarity Metrics (ISM). Then lastly, to prove the improvement delivered by PyNET and to validate ISM, we conduct another experiment using ResNet-18 semantic segmentation of the traffic images.

2.1 CamVid Dataset Preparations

The dataset used in the experiment is from the CamVid dataset [47] which consists of 707 traffic scene images. The CamVid also provides ground truth with 32 semantic classes. The image was captured from a driving automobile perspective. The dataset was divided into three groups which are 367 for training, 233 for the test, and 101 for validation. We chose the CamVid dataset because it is the most compatible with our experimental setup and environment. We applied synthetic fog, night, and rain to imitate real-life weather, and the example is shown in Figure 1.

Figure 1 Sample of CamVid dataset images (original) along with the synthesized images: fog weather, night condition, and rainy weather.

For fog and rain, we used a photo editor named Photo Kako to generate fog and rain. The app was created by Hirohisa Fujita from Japan. As for the night, we used an application called pho.to which used artificial intelligence and facial landmark detection face tracking to create a neural art style transfer technique, automatic face retouch. After that, all training, test, and validation images will be transformed into patch sizes of 448x448 and 100x100, as shown in Figure 2. These 448x448 and 100x100 image patches will be used by PyNET and DPED respectively for training and validation, while for testing, we will use full-resolution images. For PyNET, about 10110 patches are used for training and testing the model. Patches of the image are often used in deep learning because most of the time, deep learning has no prior knowledge of the input and label. Therefore, to fit in GPU support images, researchers tend to patch the images into smaller pieces. The bigger the patches, the more patch prediction happened during the process.

(a)

(b)

(c)

Figure 2 Sample of patch images of normal, fog, night, and rain conditions for (a) training, (b) test, and (c) validation images.

2.2 PyNET Modelling Workflow

The original PyNET architecture shown in Figure 3 is modified to suit the requirement of images used in this work and to suit the objective of enhancing noisy images. Based on the original work in [25], the model has an inverted pyramidal shape and is processing the images at a different level. Therefore, each level has its scope of work. Levels 4-5 will mainly focus on downscaling the image and factorizing. Level 2-3 goal is to process the color and shape of the object on the image, and level 1 is to correct the image texture, noise, and local color. The original model is specifically designed to use RAW images as input and a Bayer filter to encode image color information. The image that was encoded by the Bayer filter has the same format as the grayscale image and carries less information. Overall, the Bayer conversion will transform the image from a Bayer pattern that was kept in grayscale into a true color image [48]. The overall workflow for modeling PyNET is given in Figure 4.

In this work, we modified the original PyNET architecture proposed in [25] by not using RAW images and the Bayer channel, which contains RGGB image space. We had to combine it into RGB while loading the data for training and testing. Further, as shown in Figure 4, training consists of 5 levels of multi-stage training. The model is trained, starting from the lowest layer. This allows for achieving good results on smaller scales. After the bottom layer is pre-trained, the same procedure is applied to the next level till the training is done on the original resolution. It starts by collecting the training dataset, test dataset, and previous model data (for levels 4 – 0), then continues training the new model for the current level. After that, the image will perform the image transformation for the specific level and save the current model and data. This will be repeated until it reaches level 0. For train size, we start with 10,000 for levels 5 and 4, then 8000 for levels 3 and 2, and for levels 1 and 0, we use 5000. As for batch size, we start with 100 for level 5, 100 for level 4, 50 for level 3, 20 for level 2, level 1 is 12 and for level 0 is 10. Once model level 0 is trained, it will be used for testing whereby full-resolution images will be used instead of patches. The trained model will enhance the full-resolution image and the enhanced image will be compared against the target image. To calculate the performance of image enhancement, we used 6 performance metrics such as PSNR, SSIM, RMSE, UQI, VIF, and SAM. These metrics are obtained by comparing the similarity between the enhanced images against the original images. The result was tabulated and discussed in section results.

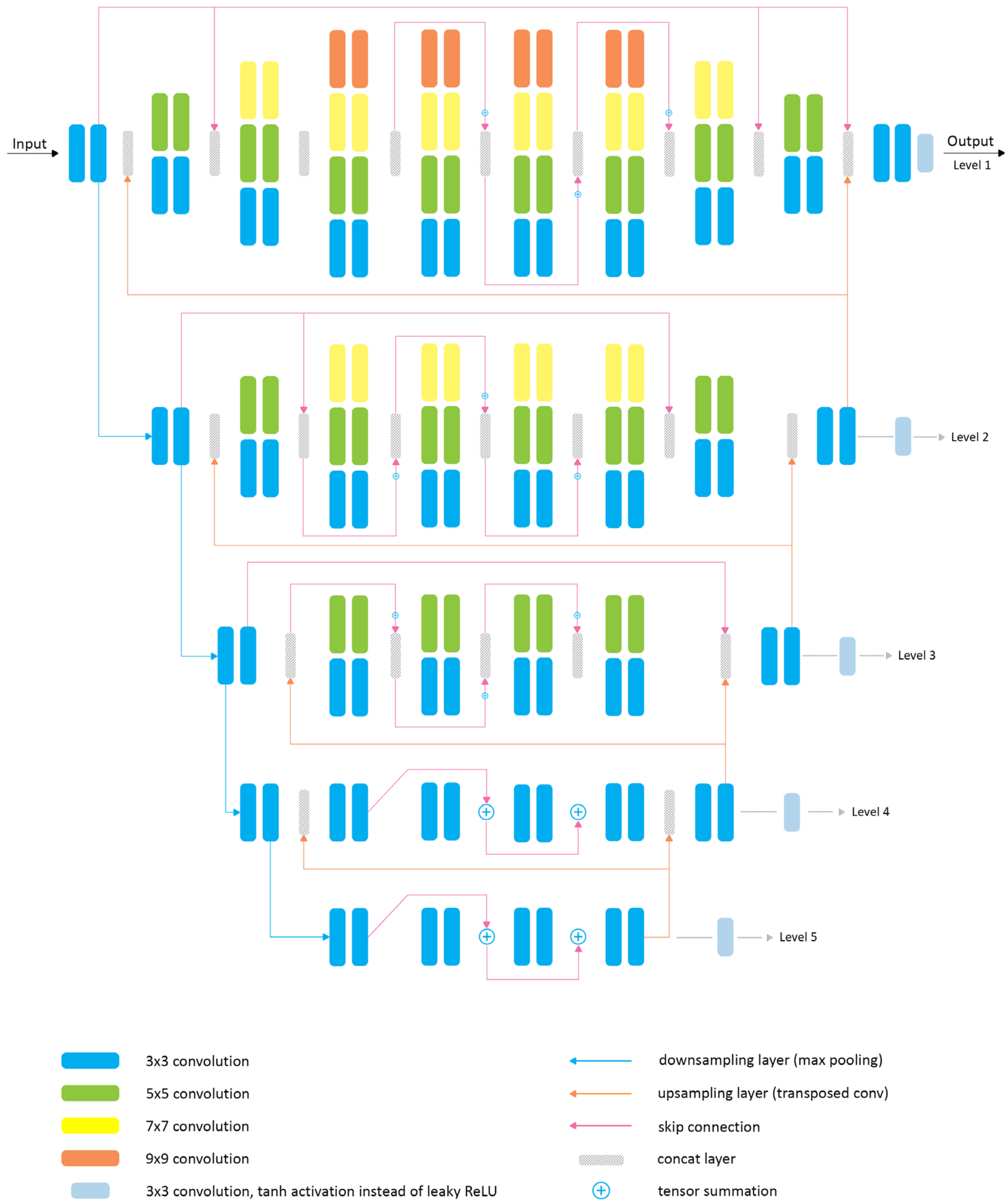


Figure 3 The PyNET architecture adopted in [25].

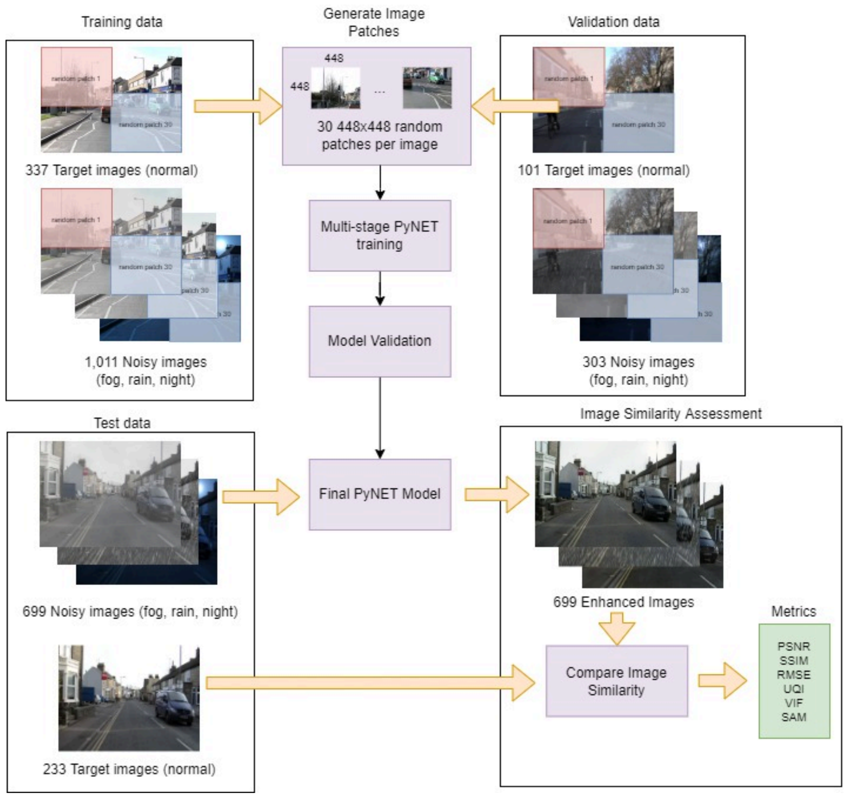


Figure 4 Overall workflow of PyNET Modelling

As for benchmarking, we used DPED and CycleGAN, and these two methods are designed to follow the same workflow as PyNET where applicable as described in Figure 4, and here we explain the parameters and dataset used and setup for them. We choose DPED and CycleGAN as the benchmark because both are among state of the art in deep learning. About 3030 patches are used for DPED training and testing the model with the size of 100x100. To train the DPED model, it required python, pillow, spicy, NumPy, imageio, and TensorFlow 2.0 libraries with suitable CUDA and CuDNN for GPU accelerator. As for the parameters, we used 19000 iterations with 50 batch sizes and a 5e-4 learning rate. DPED is also trained using patch images and tested using full-resolution images. On the other hand, CycleGAN will be using full resolution 960x720 for training, testing, and validation. For CycleGAN training, it will use 337 images, 233 images for testing, and 101 images for validation.

2.3 Image Similarity Metrics (ISM)

To obtain better quantification of the performance of the image enhancement from the 6 metrics used, we propose to combine the metrics into a single metric called Image Similarity Metrics (ISM). The quantification process uses Geometric Mean (GM) or sample geometric mean which is a whole dataset value that measures the average value or means of a product. It multiplies all the data values and takes the *n*-th root of the numbers as shown in (1) [49]. There are different types of Arithmetic Mean (AM), GM, and Harmonic mean (HM). The arithmetic mean is an average of value divided by its count, and it is more beneficial to calculate the average where the number does not depend on each other. At the same time, geometry is calculated for a series of numbers by taking into account many aspects [50].

$$G.M = \sqrt[n]{x^1 x^2 x^3 \ldots \ldots x_n} \tag{1}$$

According to Figure 4, we used multiple image assessments to calculate the performance of the image enhancement method. However, each metric we used has its specific role, so the value represents a different aspect of the findings depending on the characteristics of the metrics. Therefore, if we rely on each of the individual metrics, we cannot conclusively determine which single method is the best for all weather. Thus, we will be using ISM to select the best enhancement method. The formula for ISM is therefore given in (2), derived from GM. ISM is the root of multiple scores of PSNR, SSIM, UQI, VIF, and the inverse of RMSE and SAM.

$$ISM = \sqrt[6]{PSNR \times \frac{1}{RMSE} \times SSIM \times UQI \times VIF \times \frac{1}{SAM}} \tag{2}$$

2.4 Semantic Segmentation using ResNet-18

We further performed tests using semantic segmentation of traffic images using ResNet-18 to validate the result we obtained from ISM. There are two parts to the experiments carried out here. First, we train a ResNet-18 semantic segmentation model using normal CamVid images and then test the model against the noisy test images. The second part is where we used the model against the enhanced images that were obtained from PyNET, DPED, and CycleGAN.

The overall workflow of semantic segmentation using ResNet-18 is given in Figure 5. In this test, there will be 11 segmentation classes used namely bicyclist, building, pole, road, pavement, tree, sign symbols, fence, car, and pedestrian, as shown in Figure 5. For this experiment, the number of classes in CamVid is reduced from 32 segmentation classes to 11 segmentation classes. For example, a truck-bus or other big moving objects will be labeled as a car. After defining the classes, it then sets the color label for each class. Then we start our training and testing in this experiment. Once testing is completed, Intersection-Over Union (IoU) and the accuracy for each class will be calculated as the Jaccard index given in (3). The purpose of calculating IoU is to confirm the visual results and to calculate network performance.

$$J(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} \tag{3}$$

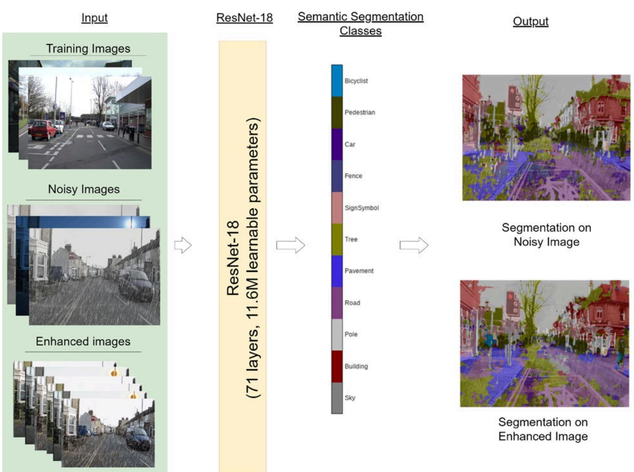


Figure 5 Overall workflow of semantic segmentation using ResNet-18

3.0 RESULTS AND DISCUSSION

In this section, we discuss the results obtained from the image enhancement experiment according to the weather or condition, starting with fog, night (low light), and rain. We also present a scatter chart of each type of test image to show the VIF vs. SAM, and VIF vs. PSNR correlation and a summary of performance for each image enhancement method.

3.1 Image Enhancement on Fog Images

Figure 6 shows an example of the results of the images before and after the enhancement experiment using PyNET, DPED, and CycleGAN. From Figure 6 (a), we can observe that the fog has been significantly reduced, and the image has better clarity compared to the image before enhancement. To observe all test image performance in each image enhancement method, we plot the results as shown in Figure 7. In this figure, the higher the scores of VIF and PSNR will indicate better image enhancement, and the lower the SAM score, the better the performance of the enhancement. Thus, for Figure 7(a), the best performance is achieved in the northwest direction, whereas for Figure 7(b), the best performance is achieved in the northeast direction. VIF-SAM and VIF-PSNR plots highlight image information extracted by the human visual system plot. Table 2 is given to summarize the image quality in terms of PSNR, SSIM, RMSE, UQI, VIF, SAM, and ISM.

Table 2 Image quality assessment for fog image enhancement

Image Quality Assessments	PyNET	DPED	CycleGAN
PSNR	24.01	**24.87**	24.13
SSIM	**83.79**	81.28	65.72
RMSE	0.92	0.90	**0.87**
UQI	0.93	**0.95**	0.94
VIF	**0.60**	0.49	0.47
SAM	0.12	**0.10**	0.11
Proposed ISM (BE)	1.05	1.05	1.05
Proposed ISM (AE)	**1.52**	1.50	1.50

*Bold values indicate the best image enhancement method based on image assessment metrics.

Table 2 tabulates the average of all metrics for each method, indicating that each method does perform well in a certain aspect. For example, DPED has the best performance if we look at its PSNR, UQI, and SAM scores. While PyNET is the best if we consider the SSIM and VIF scores. Meanwhile, if we use RMSE, CycleGAN will be the best method. This situation highlights that certain methods will excel at certain performance metrics, and to select the best method, it is better to use a single metric that will take into consideration the geometric average of all scores, which in this case we will use ISM. The table shows ISM (BE), which refers to the score of ISM Before Enhancement, and ISM (AE), which refers to the score of ISM After Enhancement. From ISM, it shows that PyNET is the best method for the enhancement of foggy images. Table 2 also indicates that there is a difference in ISM before and after enhancement such that PyNET, which obtains the highest score of ISM has improved from ISM = 1.05 to ISM = 1.52 after the enhancement. The ISM for DPED and CycleGAN are the same at ISM = 1.50, which is consistent with the plot in Figure 7. We illustrate several examples of foggy images, before and after enhancement with their respective ISM scores in Figure 8.

3.2 Image Enhancement on Night Images

We can observe from Figure 6 (b) that the low illumination component has been removed, so the enhanced image is more detailed and clearer compared to the image before the enhancement. The sky and sign symbols are clearer than the images before enhancement. To observe all test image performance in each image enhancement method, the VIF-SAM and VIF-PSNR plot is given in Figure 9.

According to Figure 9, we can observe that PyNET has better performance in both VIF-SAM and VIF-PNSR plots. We can also see a clear improvement in the image enhancement methods compared to the noisy image scores. In both cases, PyNET has the best scores, whereby DPED performs better than CycleGAN. A summary of the average scores is tabulated in Table 3.

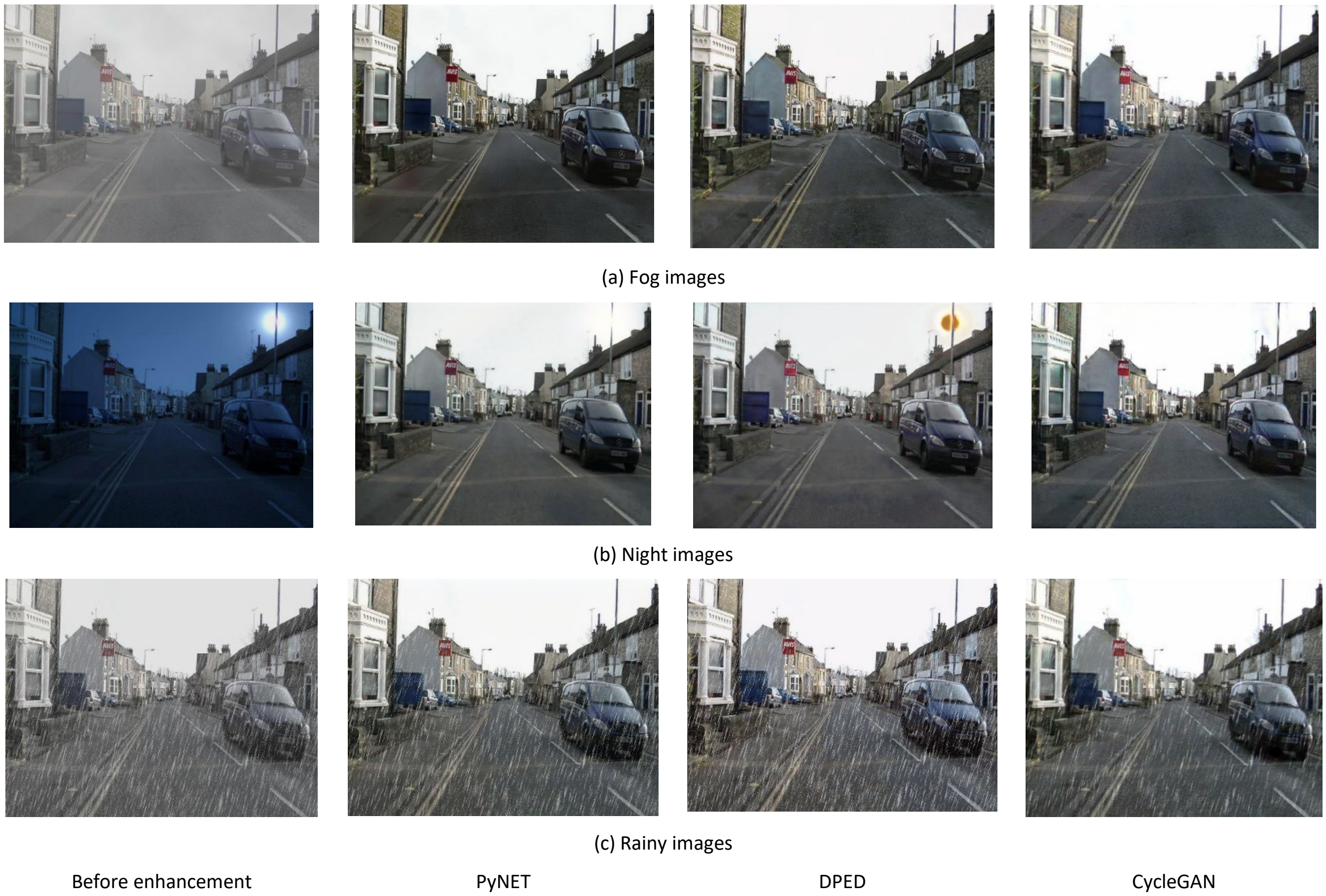


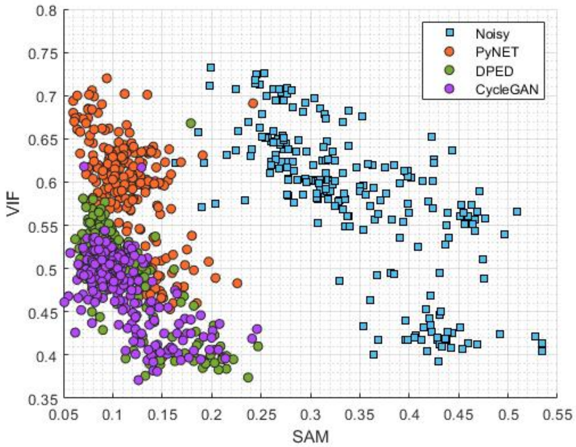
Figure 6 Example of results for image enhancement of (a) fog images, (b) night images, and (c) rainy images before and after enhancement using PyNET, DPED, and CycleGAN.

In Table 3, we highlight the best score in terms of each metric from the image assessment metric. Again, this table shows that CycleGAN and PyNET perform better in maintaining the quality of images for the night or low light image enhancement if looking at metrics such as UQI, VIF, SAM, PSNR, and RMSE. On the other hand, considering SSIM, DPED delivers the best performance. In terms of PSNR, VIF, and SAM, PyNET has the best UQI at 0.98, VIF = 0.49, and SAM = 0.11. Based on the ISM, PyNET is the best method with ISM = 1.53, followed by DPED and CycleGAN at 1.47 and 1.43 respectively. Overall, PyNET has significantly improved the noisy images from ISM = 0.85 to ISM = 1.53. We illustrate several examples of foggy images, before and after enhancement with their respective ISM scores in Figure 10.

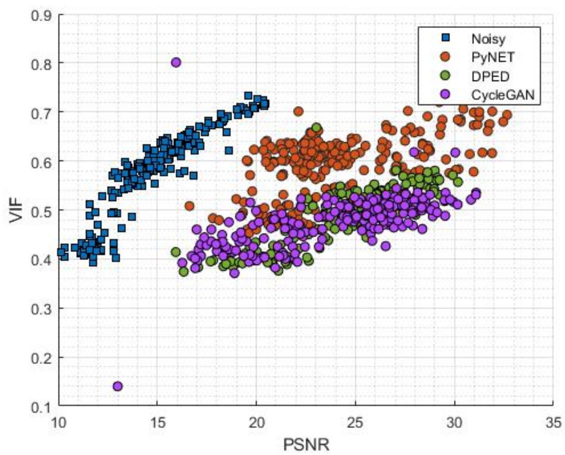
3.3 Image Enhancement On Rainy Images

Figure 6 (c) shows an example of rainy weather images enhanced by PyNET, DPED, and CycleGAN. Compared to other conditions, the weather rain is the most difficult condition to enhance and remove its noise. From the figure, we could see that the rain streaks were still present in the enhanced image, but it was not as heavy as before, and its background was much more detailed and precise. To illustrate all test image performance in each image enhancement method, we did a test and collected the scores of the metric for each image and plotted them into scatter charts, as shown in Figure 11.

According to Figure 11, for rainy weather images, the performance improvement in all tested methods is not too obvious. In terms of VIF and SAM, there is not much difference between PyNET, DPED, and CycleGAN. Based on Table 4, PyNET is the best method if we consider PSNR, UQI, VIF, and SAM, while DPED performs best in SSIM and RMSE. CycleGAN, however, did not perform best in any of the metrics. Based on ISM, PyNET has the best ISM score and manages to improve the ISM from 1.29 to 1.65. We illustrate several examples of foggy images, before and after enhancement with their respective ISM scores in Figure 12.



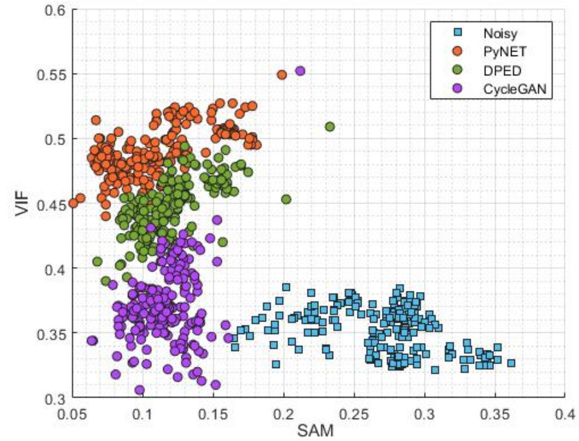
(a)



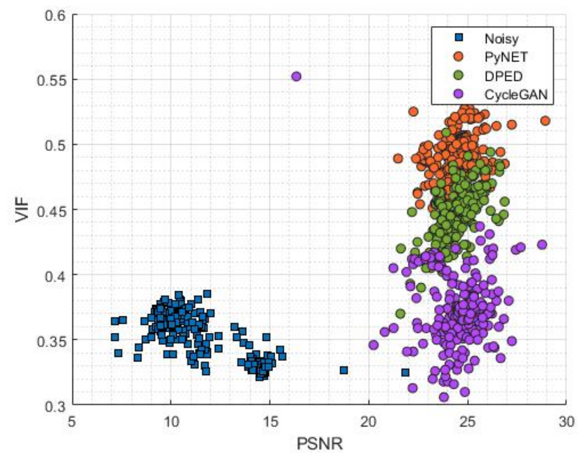
(b)

Figure 7 Comparison between (a) VIF-SAM and (b) PSNR-VIF scatter plots showing the distribution of image quality before and after image enhancements for fog images.

Figure 8 ISM scores for several sample foggy weather images using PyNET, DPED, and CycleGAN image enhancement.



(a)



(b)

Figure 9 Comparison between (a)VIF-SAM and (b)PSNR-VIF scatter plots showing the distribution of image quality before and after image enhancements for night images.

Table 3 Image quality assessment for night image enhancement

Image Quality Assessments	PyNET	DPED	CycleGAN
PSNR	24.32	24.39	**24.57**
SSIM	60.67	**65.93**	60.44
RMSE	0.91	0.89	**0.84**
UQI	**0.98**	0.97	0.96
VIF	**0.49**	0.44	0.37
SAM	**0.11**	0.12	0.12
Proposed ISM (BE)	0.85	0.85	0.85
Proposed ISM (AE)	**1.53**	1.47	1.43

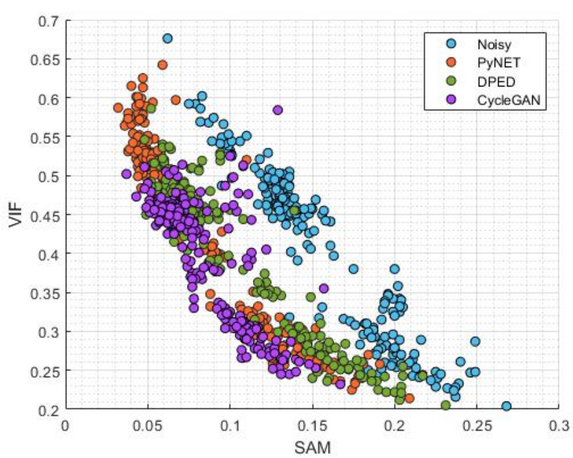
*Bold values indicate the best image enhancement method based on image assessment metrics.

3.4 Semantic Segmentation using ResNet-18

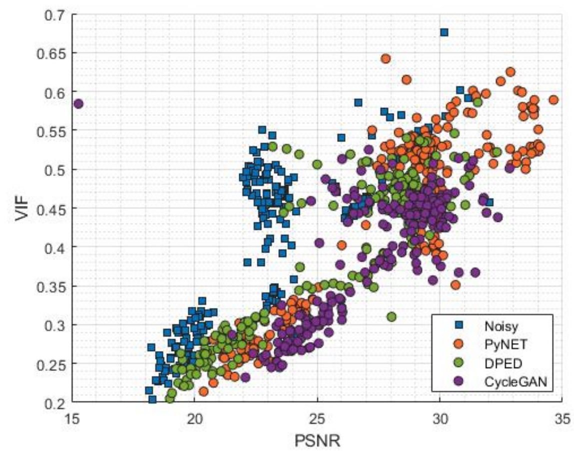
In this section, we perform another experiment to test all enhanced images on the semantic segmentation tasks to find the best enhancement method. Figure 13 shows some results from the semantic segmentation task using the PyNET image enhancement method.

Figure 10 ISM scores for several sample night images using PyNET, DPED, and CycleGAN image enhancement.

According to Figure 13, for fog weather and night conditions, the semantic segmentation result slightly improved, indicated by a better recognition of buildings, pedestrians, and pavement. However, for rainy weather, we could still see many rain streaks on the semantic segmentation result, and by visual observation, there is no significant improvement in semantic segmentation. The overall result of semantic segmentation on each weather condition will be discussed in subsequent sub-sections.



(a)



(b)

Figure 11 Comparison between (a)VIF-SAM and (b)PSNR-VIF scatter plots showing the distribution of image quality before and after image enhancements for rainy images.

Table 4 Image quality assessment for rainy image enhancement

Image Quality Assessments	PyNET	DPED	CycleGAN
PSNR	**28.00**	26.02	27.57
SSIM	53.89	**61.22**	47.96
RMSE	0.80	**0.75**	0.81
UQI	**0.98**	0.95	0.97
VIF	**0.44**	0.40	0.39
SAM	**0.08**	0.10	0.09
Proposed ISM (BE)	1.29	1.29	1.29
Proposed ISM (AE)	**1.65**	1.48	1.57

*Bold values indicate the best image enhancement method based on image assessment metrics.

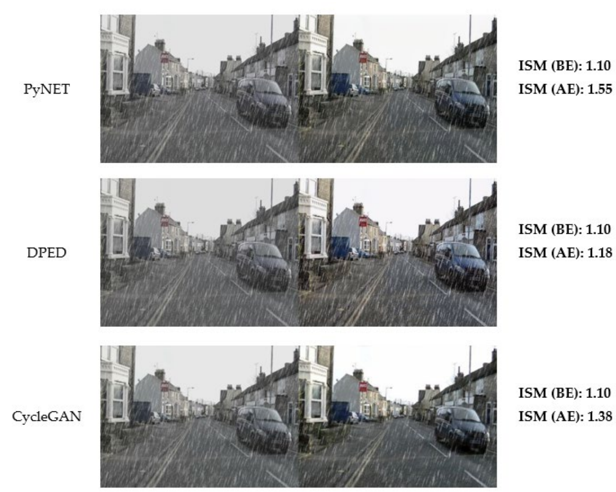


Figure 12 ISM scores for several sample rainy images using PyNET, DPED, and CycleGAN image enhancement.

3.4.1 Semantic Segmentation Of Foggy Weather Images

Figure 14 shows the calculated mean accuracy, mean intersection over union (IoU), and accuracy for each class for the fog. IoU is calculated by dividing the overlap between the predicted and ground truth annotation over the union. At the same time, mean accuracy represents the correctly classified pixels in the image. The blue bars represent the noisy image or image before enhancement and are also the benchmark for minimum improvement performance for the image enhancement methods.

From the mean accuracy and mean IoU chart, PyNET has improved the pixel classification, and there is a slight improvement in semantic segmentation based on the IoU. This performance improvement has also been shown earlier by ISM scores. From Figure 14, we can see that the accuracy of PyNET is the highest at 79%, followed by DPED at 76% accuracy, and CycleGAN with 72% accuracy. The baseline accuracy for the semantic segmentation on noisy fog images without any image enhancement is 68%. This result agrees with the ISM calculation, where PyNET also delivers the best ISM score. In terms of mean IoU, PyNET slightly improved the mean IOU, however, DPED and CycleGAN decreased the mean IoU of the segmentation results.

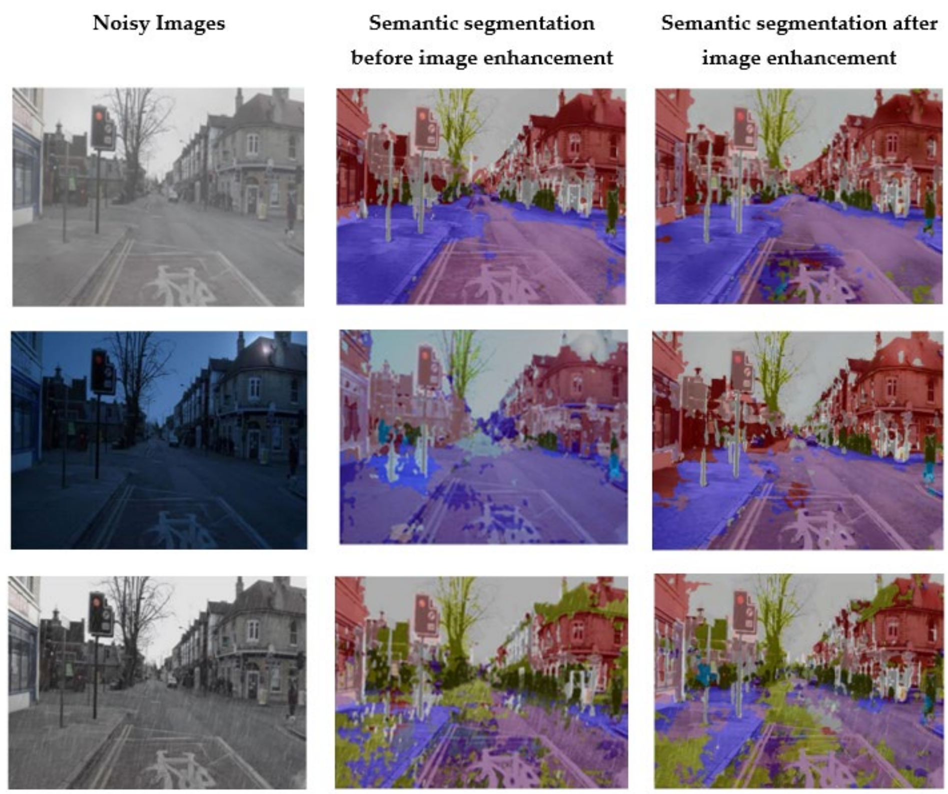


Figure 13 Example results of semantic segmentation for foggy weather, night conditions, and rainy weather images.

PyNET has managed to improve the accuracy of the semantic segmentation for fog images by 11% which is from 68% to 79% accuracy when compared against the semantic segmentation without any image enhancement, as shown in Figure 14 (a). However, based on the accuracy for each class, we can see that seven out of eleven classes have PyNET scores with the highest accuracy. PyNET scores the highest accuracy for the class of cars, pedestrians, and bicyclists. In the fog experiment, classes like building, sky, and road are the only classes that do not have any improvement in semantic segmentation. For the class of car and sky, PyNET scores more than 90% accuracy for each class. We also found that CycleGAN performs better than PyNET in pole, tree, sign symbol, fence, pedestrian, and bicyclist classes.

3.4.2 Semantic Segmentation of Night Weather Images

Figure 15 shows the calculated mean accuracy, mean intersection over union (IoU), and accuracy for each class for the semantic segmentation of night condition images. In terms of mean accuracy, PyNET obtains the best performance with 76% accuracy, followed by DPED with 71% accuracy and CycleGAN with 68% accuracy. The baseline accuracy for the semantic segmentation on night images without any image enhancement is 59%. According to this figure, the improvement delivered by PyNET is 17% when compared to the semantic segmentation of noisy night images at 59% accuracy.

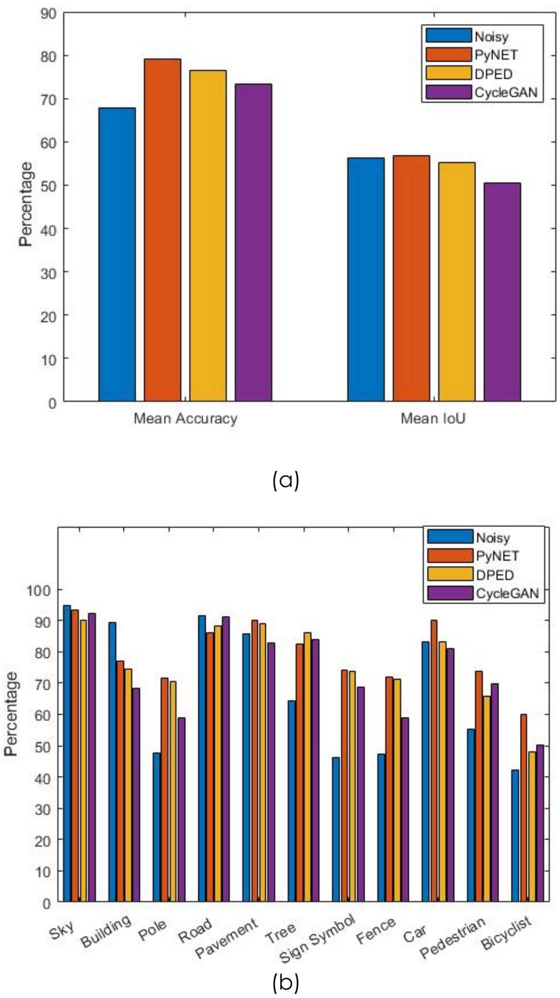


Figure 14 Semantic segmentation results in terms of (a) Mean accuracy and Mean IoU, and (b) per class accuracy for foggy weather images.

This finding again agrees with the ISM score that we obtained in the earlier experiment. In terms of mean IoU, PyNET significantly improved the mean IOU, while DPED and CycleGAN also improved the mean IoU of the segmentation results. Regarding accuracy for each class, all image enhancement methods have improved the accuracy for each class except for the car, sign symbol, and pedestrian classes.

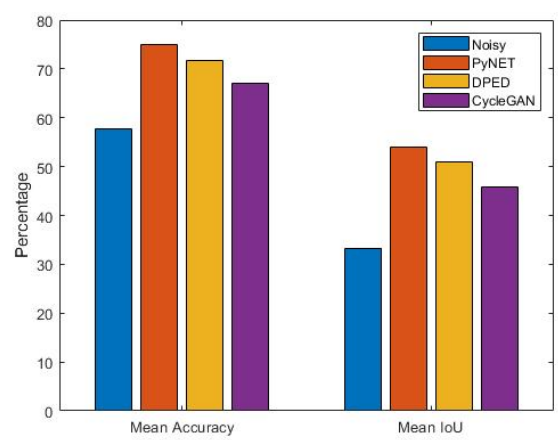
The biggest improvement is observed in the pavement class. Notably, PyNET enhancement does improve the segmentation accuracy for all semantic segmentation classes. We notice that small objects such as a pole, fences, pedestrians, and bicyclists achieve at least 10% improvement regardless of the method used. About five classes scored more than 80% accuracy in the semantic segmentation task, where the classes are the sky, road, pavement, tree, and car. Based on Figure 15, we can see that before the enhancement, buildings are segmented as a sign symbol, the pedestrian is segmented as a bicyclist, and the most obvious changes are in the segmentation of sky and road.

3.4.3 Semantic Segmentation Of Rainy Weather Images

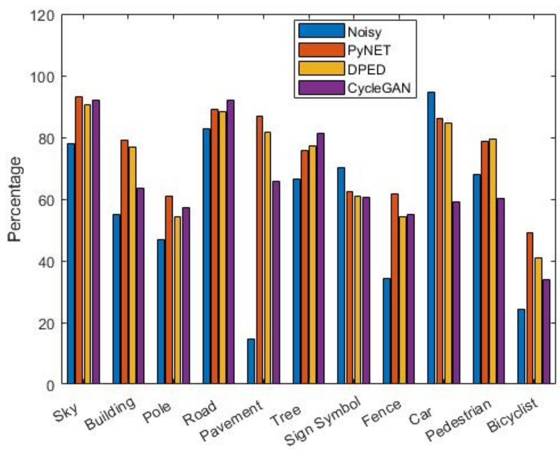
Figure 16 shows calculated mean accuracy, mean intersection over union (IoU), and accuracy for each class for the semantic segmentation of rainy weather images. In terms of mean accuracy and mean IoU for rain images, PyNET and CycleGAN significantly improve the accuracy of classifying, localizing, and segmentation of ResNet-18 for rain images. From the overall performance for rain in accuracy for each class, we can see that the CycleGAN method does perform slightly better than PyNET.

In terms of mean accuracy, Cycle-GAN obtains the best performance at 66% accuracy, and PyNET at 61% accuracy. The baseline accuracy for the semantic segmentation on noisy rainy weather images without any image enhancement is 50%. Thus, for the rain-weather images, PyNET manages to improve the semantic segmentation accuracy by 11% when compared to the baseline accuracy without any image enhancement. Furthermore, PyNET and CycleGAN performance on accuracy for each class does not have a significant difference. PyNET performs best at improving the segmentation of sign symbols, trees, fences, and pedestrians, while CycleGAN is better at improving the segmentation of buildings, poles, roads, pavement, cars, and bicyclists. Still, even though PyNET does not score the highest accuracy in each class, the accuracy gap with the highest performance is not significant.

From the results presented throughout this work, the images enhanced by PyNET has shown the best performance in terms of the ISM score derived from PSNR, SSIM, UQI, VIF, and the inverse of RMSE and SAM. The higher ISM metrics performance for PyNET indicates that this method manages to enhance the noisy images better than DPED and CycleGAN. This finding is further supported by the ResNet-18 semantic segmentation performance where the images enhanced by PyNET delivered the highest segmentation accuracy. The reason for this is mainly due to the PyNET architectures that uses a pyramidal structure of multiple convolutional blocks with different receptive field sizes. This hierarchical approach allows the network to capture features at multiple scales, from fine details to larger contextual information. This is beneficial for image enhancement proposed in this work because it enables the model to focus on both local and global features, which is generally essential for tasks like denoising, super-resolution, and image deblurring.



(a)

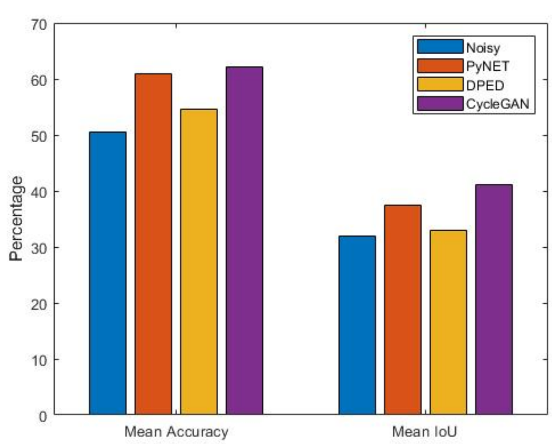


(b)

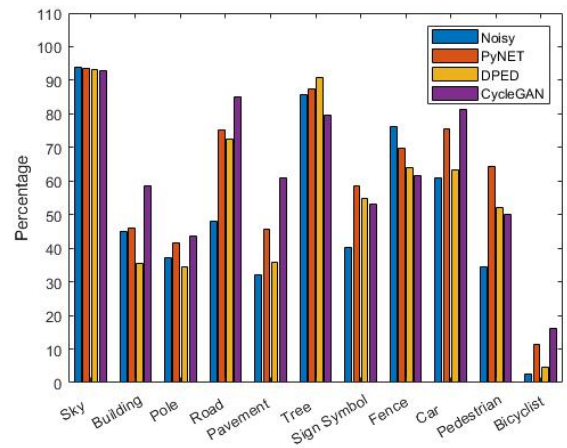
Figure 15 Semantic segmentation results in terms of (a) Mean accuracy and Mean IoU, and (b) per class accuracy for night images.

4.0 CONCLUSION

We aim to improve the traffic images under the effect of several weather conditions, such as fog, rain, and night. Therefore, we proposed a PyNET image enhancement method where the deep learning architecture model has an inverted pyramid, and each level has its function, and we compared PyNET performance to DPED and CycleGAN. Based on the image assessment result, different types of image quality assessment metrics will deliver different results in terms of the best-performing image enhancement method. Thus, to calculate the overall performance, we propose using ISM as a single value metric to represent image quality enhancement produced by the tested deep learning methods. From the ISM, overall results show that PyNET performs better than DPED and CycleGAN in all weather or conditions. In foggy weather, PyNET delivers ISM = 1.52 compared to DPED at 1.50 and CycleGAN at 1.50; whereby for rainy weather, the PyNET ISM is 1.53 whereas DPED has ISM = 1.47 and CycleGAN has ISM = 1.43. Finally, for the night condition, PyNET delivers ISM = 1.65, which surpassed DPED and CycleGAN at 1.48 and 1.57 respectively.



(a)



(b)

Figure 16 Semantic segmentation results in terms of (a) Mean accuracy and Mean IoU, and (b) per class accuracy for rainy weather images.

To prove the validity of ISM, we test the enhanced image on semantic segmentation using ResNet-18. The results show that the quality of the image harms semantic segmentation. However, PyNET has improved and increased the accuracy of the overall semantic segmentation in ResNet-18 by as much as 11% for foggy weather conditions, 17% for night conditions, and 11% for rainy weather conditions. The validity of the ISM score is proven by the semantic segmentation results. We also found that DPED and CycleGAN reduce the mean IoU for the fog weather images. Furthermore, among the tested weather effects, semantic segmentation on rain gives the lowest mean IoU. For future work, we will improve the method to remove the rain streak for better improvement of semantic segmentation results.

Acknowledgment

The author would like to thank Universiti Teknologi MARA for the support given in this work through the Micro-Industry Hub grant MIH-(001/2020).

References

[1] P. P. Shinde and S. Shah, 2018. "A Review of Machine Learning and Deep Learning Applications," *International Conference on Computing Communication Control and Automation*. 1–6, DOI: 10.1109/ICCUBEA.2018.8697857.

[2] A. A. Khan, A. A. Laghari, and S. A. Awan, 2018., "Machine Learning in Computer Vision: A Review," *EAI Endorsed Transactions on Scalable Information Systems,* 8: 1–11. DOI: 10.4108/eai.21-4-2021.169418.

[3] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, 2019., "A review of machine learning and IoT in smart transporttion," *Future Internet*, 11: 1–23. DOI: 10.3390/FI11040094.

[4] Y. Xin 2018. "Machine Learning and Deep Learning Methods for Cybersecurity," IEEE Access, 6: 1–23. DOI: 10.1109/ACCESS.2018.2836950.

[5] P. Gogas and T. Papadimitriou, 2021. "Machine Learning in Economics and Finance," *Computational Economics,* 57: 1–4. DOI: 10.1007/s10614-021-10094-w.

[6] F. Femling, A. Olsson, and F. Alonso-Fernandez 2018., "Fruit and

Vegetable Identification Using Machine Learning for Retail Applications," *International Conference Signal Image Technology and Internet Based System (SITIS),* 9–15. DOI: 10.1109/SITIS.2018.00013.

[7] B. T.k., C. S. R. Annavarapu, and A. Bablani 2021. "Machine learning algorithms for social media analysis: A survey," *Computer Science Review* 40: 1–32. DOI: 10.1016/j.cosrev.2021.100395.

[8] H. I. Suk, 2017., An Introduction to Neural Networks and Deep Learning, 1st ed. Elsevier Inc.

[9] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, 2018 "Recent advances in convolutional neural network acceleration," Neurocomputing, 1–27. DOI: 10.1016/j.neucom.2018.09.038.

[10] K. Li, Y. G., Q. Y., C. X., and D. 2020, "Low-Light Image Enhancement for Autonomous Driving System using DriveRetinex-Net," 1: 1–5. Elsevier B.V. DOI: doi.org/10.1016/j.knosys.2020.106617.

[11] K. Zhang, W. Zuo, S. Gu, and L. Zhang 2017 "Learning deep CNN denoiser prior for image restoration," IEEE Conference on Computer Vision and Pattern Recognition.. 1–10. DOI: 10.1109/CVPR.2017.300.

[12] S. Ghosh, N. Das, I. Das, and U. Maulik 2019. Understanding deep learning techniques for image segmentation," *ACM Computer Surveys* 52: 1–35. DOI: 10.1145/3329784.

[13] C. Wang, Y. Han, and W. Wang 2019. "An end-to-end deep learning image compression framework based on semantic analysis," Applied Sciences, 9: DOI: 10.3390/app9173580.

[14] D. H. Kim and H. Y. Lee 2017. "Image manipulation detection using convolutional neural network," *International Journal Applied Engineering Research*. 12: 1–7,.

[15] A. Noguchi and T. Harada, "Image generation from small datasets via batch statistics adaptation," International Conference on Computer Vision (ICCV), 1–9. DOI: 10.1109/ICCV.2019.00284.

[16] A. R. Pathak, M. Pandey, and S. Rautaray. 2018. "Application of Deep Learning for Object Detection," *Procedia Computer Science,* 132: 1–12. DOI: 10.1016/j.procs.2018.05.144.

[17] B. Chitradevi and P. Srimanthi. 2014., "An Overview on Image Processing Techniques," *International Journal of Innovative Research in Computer and Communication Engineering* 2: 1–7. [Online]Available:http://www.hindawi.com/isrn/sp/2013/496701/.

[18] S. Gollapudi 2018., "Learning For Computer Vision Tasks: A review," *International Conference on Intelligent Computing and Control,* 1–5. DOI: 10.1007/978-1-4842-4261-2_3.

[19] Y. Qi 2021., "A Comprehensive Overview of Image Enhancement Techniques," *Archives of Computational Methods in Engineering.* 29: 583–607. DOI: 10.1007/s11831-021-09587-6.

[20] S. Rahman, M. M. Rahman, K. Hussain, S. M. Khaled, and M. Shoyaib. 2014. "Image enhancement in spatial domain: A comprehensive study," Conf. Comput. Inf. Technol. ICCIT, 1–6. DOI: 10.1109/ICCITechn.2014.7073123.

[21] G. Singh and A. Mittal 2014. "Various Image Enhancement Techniques- A Critical Review,*" International journal of innovation and Scientific Research*, 10: 1–8

[22] Z. Shi, Y. Feng, M. Zhao, and L. He 2019, "A joint deep neural networks-based method for single nighttime rainy image enhancement," Neural Computing and Applications. 1–14. DOI: 10.1007/s00521-019-04501-5.

[23] S. Joshi, R. Arindom, T. Dikshit, B. Anish, A. G. Deep, and P. Pallav. 2015. "Image Enhancement Techniques: A Study," *Indian Journal of Science and Technology* . 8: 1–12. DOI: 10.17485/ijst/2015/v8i.

[24] S. A. Nossier, J. Wall, M. Moniri, C. Glackin, and N. Cannings 2014. "An experimental analysis of deep learning architectures for supervised speech enhancement," Electron, 10: 1–32. DOI: 10.3390/electronics10010017.

[25] A. Ignatov, L. Gool, and R. Timofte 2020. "Replacing mobile camera isp with a single deep learning model," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops Work*, 1–11. DOI: 10.1109/CVPRW50498.2020.00276.

[26] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Gool 2017., "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks," *IEEE International Conference on Computer Vision* (ICCV), 1–16. DOI: 10.1109/ICCV.2017.355.

[27] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. 2017. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *IEEE International Conference on Computer Vision* 2242–225. DOI: 10.1109/ICCV.2017.244.

[28] D. Ngo, S. Lee, Q. H. Nguyen, T. M. Ngo, G. D. Lee, and B. Kang 2020 "Single image haze removal from image enhancement perspective for real-time vision-based systems," Sensors (switzerl.,. 1–21. DOI: 10.3390/s20185170.

[29] F. Hussain and J. Jeong 2016. "Visibility Enhancement of Scene Images Degraded by Foggy Weather Conditions with Deep Neural Networks," *Journal of Sensor.* 2016: 1–10. DOI: 10.1155/2016/3894832.

[30] H. Wang, Y. Li, and H. Moon 2021 "Robust Korean License Plate Recognition Based on Deep Neural Networks," Sensors (swit-zerl., 1–18. DOI: 10.3390/s21124140.

[31] C. Li, J. Zhu, L. Bi, W. Zhang, and Y. Liu 2022. "A low-light image enhancement method with brightness balance and detail preservation," PLoS ONE, 17.

[32] L. H. Pham and ; Duong Nguyen-Ngoc Tran; Jae Wook Jeon 2020 "Low-Light Image Enhancement for Autonomous Driving Systems using DriveRetinex-Net," IEEE Int. Conf. Consum. Electron. - Asia, pp. 1–5. DOI: 10.1109/ICCE-Asia49877.2020.9277442.

[33] S. Akintayo, G. L. Adedotun, and S. Kin 2016. "LLNet: A Deep Autoencoder approach to Natural Low-light Image Enhancement," arXiv. 1–11,. [Online]. Available:https://arxiv.org/pdf/1511.03995.pdf

[34] L. Ma, T. Ma, R. Liu, X. Fan, and Z. Luo 2022 "Toward Fast, Flexible, and Robust Low-Light Image Enhancement," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 1–10,.

[35] W. Ren 2019. "Low-Light Image Enhancement via a Deep Hybrid Network," *IEEE Transactions on Image Processing*. 28: 1–12. DOI: 10.1109/TIP.2019.2910412.

[36] J. Mukhtarjee, K. Praveen, and V. Madumbu 2018. "Visual Quality Enhancement of Images under Adverse Weather Conditions," *21st International Conference on Intelligent Transportation Systems (ITSC),* ITSC, 1–8. DOI: 10.1109/ITSC.2018.8569536.

[37] J. Younkwan, J. Lee, Y. Jeon, B. Ko, and M. Jeon 2021., "Task-Driven Deep Image Enhancement Network for Autonomous Driving in Bad Weather," *IEEE International Conference on Robotics and Automation (ICRA*), 1–8. DOI: 10.1109/ICRA48506.2021.9561076.

[38] T. Matsui, T. Fujisawa, T. Yamaguchi, and M. Ikehara 2018. "Single-Image Rain Removal Using Residual Deep Learning," *25th IEEE International Conference on Image Processing (ICIP).* 1–5. DOI: 10.1109/ICIP.2018.8451612.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, E. P. Simoncelli, and S. Member 2004., "Image Quality Assessment : From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, 13: 1–13. DOI: 10.1109/TIP.2003.819861.

[40] U. Sara, M. Akter, and M. S. Uddin. 2019, "Image Quality Assessment through FSIM , SSIM , MSE and PSNR — A Comparative Study," *Journal of Computer and Communications,* 9: 1–11. DOI: 10.4236/jcc.2019.73002.

[41] Z. Wang and A. C. Bovik 2002 "A universal image quality index," *IEEE Signal Process,* 9: 1–4. DOI: 10.1109/97.995823.

[42] D. Asamoah, E. Ofori, S. Opoku, and J. Danso 2018. "Measuring the Performance of Image Contrast Enhancement Technique," *International Journal of Computer Applications,*. 181: 1–8,.

[43] S. Minaee, Y. Boykov, and F. Porikli 2022., "Image Segmentation Using Deep Learning : A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, DOI: 10.1109/TPAMI.2021.3059968.

[44] Z. Zhou. 2019. "Attention based Stack ResNet for Citywide Traffic Accident Prediction," *IEEE International Conference on Mobile Data Management (MDM)*, 0–2. DOI: 10.1109/MDM.2019.00-27.

[45] A. K. Agrawal, K. Agarwal, J. Choudhary, A. Bhattacharya, and S. Tangudu 2020. "Automatic Traffic Accident Detection System Using ResNet and SVM," *Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)* pp. 1–6. DOI: 10.1109/ICRCICN50933.2020.9296156.

[46] Z. Zhang 2021. "ResNet-Based Model for Autonomous Vehicles Trajectory Prediction," *IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE),* pp. 1–4. DOI: vv10.1109/ICCECE51280.2021.9342418.

[47] G. J. Brostow, J. Fauqueur, and R. Cipolla 2009, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters,*. 1–10. DOI: 10.1016/j.patrec.2008.04.005.

[48] M. R. Shortis, J. W. Seager, E. S. Harvey, and S. Robson 2005 "Influence of Bayer filters on the quality of photogrammetric measurement," VideometricsVII: 1–9. DOI: 10.1117/12.588217.

[49] R. M. Vogel 2020., "The geometric mean?," Communication in Statistics- Theory and Methods, 1–14. DO: 10.1080/03610926.2020.1743313.

[50] G. G. S. 2023. "Relationship Between Arithmetic Mean and Geometric Mean: Types, Differences, and Solved Examples," EMBIBE.