CHALMERS
UNIVERSITY OF TECHNOLOGY

GAN-based water droplet removal

Master's thesis in Complex adaptive systems

Jiraporn Sophonpattanakit

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

GAN-based water droplet removal

JIRAPORN SOPHONPATTANAKIT

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and maritime sciences
Division of Vehicle engineering and autonomous systems
Chalmers University of Technology
Gothenburg, Sweden 2022

GAN-based water droplet removal

JIRAPORN SOPHONPATTANAKIT

Supervisor: Ola Benderius, Department of Mechanics and maritime sciences
Examiner: Ola Benderius, Department of Mechanics and maritime sciences

Master's Thesis 2022:68
Department of Mechanics and maritime sciences
Division of Vehicle engineering and autonomous systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Images on the top row are the original images with raindrops. The images on the bottom row are the output images generated from the droplet removal model.

Typeset in LaTeX
Printed by Chalmers Digitaltryck
Gothenburg, Sweden 2022

GAN-based water droplet removal
JIRAPORN SOPHONPATTANAKIT
Department of Mechanics and maritime sciences
Chalmers University of Technology

Abstract

This master thesis project studies how *generative adversarial networks* (GANs) can perform raindrop removal and reconstruct the scenes in single images. The models were trained by using synthetic droplet datasets, and real droplet datasets. The real droplet datasets were from Qian et al. datasets and the Reeds datasets [1]. The synthetic droplet datasets were generated from ground-truth images from prior datasets. The process was done by using OpenGL. Then the generated images were evaluated by full-reference quality metrics and non-reference quality metrics, such as SSIM and BRISQUE, then tested in object detection by a pre-trained DETR model and evaluated by *mean average precision* (mAP).

After comparing the quantitative quality of the images generated by models trained by real droplet datasets and synthetic datasets, the result showed that the real droplet datasets yield better image quality than the synthetic datasets. In the object detection task, though it can enhance image quality in comparison to the degraded images, the generated images did not improve the result in this aspect. Thus, it was concluded that the synthetic datasets need to be more realistic to be able to reach comparable results as the real droplets. In the object detection task, GANs generated images from the information in the latent space. As a result, there were some objects which were corrupted, and this made the object detection model miss classify the objects. Apparently, it may not be suitable for high precision and safety tasks.

In the aspect of the automated evaluation system, this thesis project investigated the design pattern for the conceptual design of the system. However, further functional and non-functional requirements are needed to be clarified for future implementation.

Keywords: GANs, computer graphic, machine learning, droplet removal, image inpainting, microservice architecture

Acknowledgements

I would like to thank my thesis supervisor Ola Benderius for guiding and giving advice to me in the whole process of making this thesis project. Furthermore, I would like to express my gratitude to Ted Sjöblom and the Reeds team for collecting data and providing the data used in this thesis. Additionally, this master thesis would not be able to be completed without the help of Christian Berger in setting up the environment for model training, algorithm testing and the storage of the data. Also, I would like to thank my colleagues for the fruitful discussion and the help they provided to me. Lastly, I would like to thank Andreas Ovnell for the informative and helpful suggestions in report writing.

Jiraporn Sophonpattanakit, Gothenburg, November 2022

Contents

List of Figures

List of Tables

1
Introduction

Adverse weather conditions, such as rain and snow, can negatively impact data quality gathered by sensors, causing data to be challenging to analyze. This is one of the significant issues for systems like surveillance cameras and perception systems of vehicles or robots where data are collected under outdoor conditions. The damaged data could impact the decision-making system of the vehicle and misinterpret the situation.

In *advanced driver-assistance systems* (ADAS), perception systems depend on information from sensors such as cameras, lidars, and radars. While radars and lidars measure the distance between vehicles and detected objects, cameras capture visual information of the surroundings of the vehicles, which would later be used in segmentation, detection and tracking tasks. According to one study, condensation on cameras under foggy conditions can reduce the contrast and consequently makes edge recognition more difficult [2]. Furthermore, information obtained under severe weather conditions, such as heavy rain, could be affected by rain streaks, blurring the edges of the objects in video frames. Apart from this, adherent water droplets also heavily degrade the quality of the data since the images or videos get distorted in areas where the droplets are located. Sometimes whole images can be blurred this way. As a consequence, these problems may lead to accidents due to poor performance when it comes to the perception system. For instance, the system might perform inefficiently on object detection tasks, which might lead to false classification, or the system may be unable to detect any objects at all.

The task of removing disturbances, such as droplets, from the degraded images could be done by using model-driven approaches based on mathematical or physical models of droplets. Data-driven approaches can also be used, such as generative models that feed data into the algorithm, making it learn to generate images free of raindrops. The main advantage of the first approach is that it does not require time to learn a large number of datasets. However, its ability to adapt to different shapes of droplets is limited. On the other hand, the data-driven approaches could be more robust but require a certain amount of data to yield desirable results and take a long time to train. Furthermore, pairing images with raindrops with images without raindrops is difficult since the alignment and lighting conditions may change during the data collection.

At the Chalmers research laboratory Revere, a large variety of research projects connected to autonomous vehicles, such as trucks, cars and boats, are conducted.

Specifically, the laboratory has, over the years, developed and successfully demonstrated its own microservice-based software framework OpenDLV to unify the development on various platforms. There were also cases where the data collected in the rain might have low quality due to the adherent droplets on the camera lens. Therefore, with the mentioned problem in perception systems due to adverse weather conditions, this master's thesis was conducted to study the quality of the generated images and how it can impact the object detection task.

1.1 Scope and limitations

The datasets dealt with in this thesis project were images taken only in daylight and under easy weather conditions. The raindrop removal algorithm's processing time is not included in this work and only focuses on single images. Furthermore, the images are only in monochrome, which the result may not be applied to the images with other colour channels such as RGB.

1.2 Research questions

The research questions of this thesis are:
RQ1: What will the quality of the output generated by the algorithms with synthetic datasets be?
RQ2: To what extent do the algorithms affect the performance of object detection tasks?
RQ3: How can the algorithms be structured based on the microservice architecture and be deployed as a cloud service?

1.3 Purpose and goal

This master's thesis aims to apply a deep learning algorithm based on *generative adversarial networks* (GANs) to remove artefacts such as adherent raindrops from images. Furthermore, investigate the quality of the generated images and the impact on object detection tasks when the training dataset is synthesized. Finally, the goal is to investigate how this algorithm can be used in automated evaluation systems, such as cloud services.

1.4 Thesis outline

In this work, there are six sections starting from the introduction. This section provides the research question, scope and limitations, and purpose. The next section, background, introduces previous research in the raindrop removal task. The third

section, method, explains how this thesis is conducted and evaluated. The result section presents the example images of ground truth images from each dataset, images generated by the algorithms and the quality of generated images. The discussion showed drawbacks and how this work can be improved. Finally, the conclusion summarizes and concludes answers to the research question.

2
Background

This section provides the background of this thesis. The content consists of related research, starting from the modelling of raindrops, then raindrop detection and raindrop removal.

2.1 Related works

Severe weather conditions like rain, snow and haze are problematic factors and can damage the quality of data, making perception systems potentially unreliable. Therefore, there are many studies on scene restoration and detection of artefacts due to adverse weather conditions, especially rain.

2.1.1 Adherent raindrops

Adherent raindrops can affect the quality of data in different ways. For example, reducing the visibility in vision systems by obstructing and distorting the scene in areas where there are droplets. In addition to reducing the visibility of vision systems, raindrops are also hard to detect since they are transparent and vary in shape and size. Moreover, adherent raindrops are usually out-of-focus and appear to be blurred, which can increase the difficulty of being detected correctly by the raindrop detection algorithm.

The task of raindrop modelling, detection and removal can be separated into two main categories based on input: (1) single-image-based method, and (2) video-based method [3]. Leading techniques used for this purpose are model-driven approach and data-driven approach. The main difference between a single-image-based method and a video-based method is that a single-image-based method uses information from one frame or image, while the video-based method uses temporal information between consecutive frames. The use of temporal information in the video-based method efficiently improves the quality of the output. In terms of technique, different from the model-driven approach, the data-driven approach does not need mathematical model to simulate the shape of raindrops. However, it requires a proper amount of training data so that the algorithms can perform accordingly.

2.1.2 Adherent raindrop modelling, detection and removal

Based on physical properties and the laws of physics, models of adherent raindrops are generally defined using mathematical models. Adherent raindrop models could also be used in model-driven raindrop detection and removal. Raindrop detection is actively studied in research on vision systems of vehicles [4][5][6]. The applications of raindrop detection are not just limited to image restoration but also extend to weather recognition [4][5][7]. The removal task generally means that artefacts like raindrops are removed from frames or images. Although adherent raindrop removal operations are usually performed after droplets are detected, there are a few approaches that do not require droplet detection to be done in advance.

2.1.2.1 Model-driven approach

Roser and Geiger proposed an algorithm for raindrop detection, including out-of-focus droplets in single images based on geometric relation and a photometric raindrop model [8]. Moreover, they suggested a method to restore partially occluded areas by fusing intensity information from neighbouring frames after image registration of consecutive frames. Furthermore, the raindrops model in this research was based on the assumption that droplets have spherical shapes [8]. Therefore, Roser et al. extended their study by proposing a model of adherent raindrop shape on inclined surfaces using Bézier curves [9]. They also carried out raindrop recognition tasks by implementing Canny edge detection to extract shapes of adherent droplets and then apply *random sample consensus* (RANSAC) to eliminate outliers. As a result, a raindrop shape model based on Bézier curves was able to reduce the deviation between actual droplets, and the model outperformed the model from previous research [8]. Since droplets are usually out of focus, Cord and Gimonet suggested algorithms for detecting unfocused raindrops to improve ADASs behaviour [6]. This project used image processing techniques called background subtraction and watershed together with photometric properties of raindrops to identify raindrops. A temporal filter was then used to increase the accuracy. However, the work did not include raindrop removal or background restoration methods.

You et al. worked with video-based modelling, detection and removal of raindrops in video based on the assumption that raindrops are quasi-static where the size of raindrops is around 40 pixels and motions are less than four pixels per second [10]. This state-of-the-art algorithm performs raindrop removal in partially occluded areas and completely occluded areas. Though this paper shows promising results in raindrop detection and removal, the approach fails when droplets are moving faster than four pixels per frame (a hundred pixels per second).

2.1.2.2 Data-driven approach

Data-driven approaches are often utilized for many reasons. Two of these are (1) scene restoration of occluded areas in a single-image based method and (2) restora-

tion with non-static droplet shapes in a video-based method. Machine learning algorithms, such as *convolution neural networks* (CNNs) and *generative adversarial networks* (GANs), are usually used in this approach [11]. Machot et al. proposed a combination of a cellular neural network and *support vector machine* (SVM) to detect adherent droplets on windshields to improve the performance of video-based ADASs in real-time constraints [12]. This paper used edges, colour in HSV colour space, histograms of oriented gradients and wavelets of raindrops to train and test the algorithms.

Apart from CNNs implementations, there have been many papers that implement GANs to restore the background of raindrops degraded images and video frames [1][13][14][15]. Qian et al. introduced image-to-image translation using attentive GAN to remove raindrops from single images [1]. This is one of the recent GAN-based raindrop removal algorithms which show excellent results in SSIM and PSNR metrics. Quan et al. proposed an algorithm using CNNs based on encoder-decoder architecture to remove raindrops artefacts from single images [11]. In the paper of Porav et al., the authors were not just proposed a deraining algorithm based on Pix2PixHD to improve the image segmentation task but also introduced a way to collect pair of clean-unclean data using a stereo camera [14][16].

3
Methods

The method section explains the methods used in this thesis project. This section consists of raindrop removal algorithms, datasets and, finally, an evaluation of the overall performance of the method.

3.1 Raindrop removal algorithms

The algorithms used to remove raindrops are based on the deep learning approach, which is a branch of machine learning algorithms. In this thesis project, GAN-based image inpainting was used to remove artefacts due to raindrops from single images.

3.1.1 Pix2Pix

The deep learning model used to remove raindrops is based on Pix2Pix, which is *conditional GANs* (cGANs) developed for image-to-image translation tasks, and it could be used in image inpainting tasks [17]. The image-to-image translation is a problem where one wants to transform an image to another style or edit the image but still maintain the main content of the original image, and image inpainting fills the areas of the image where the details were lost. Pix2Pix consists of a U-Net generator and a PatchGAN discriminator.

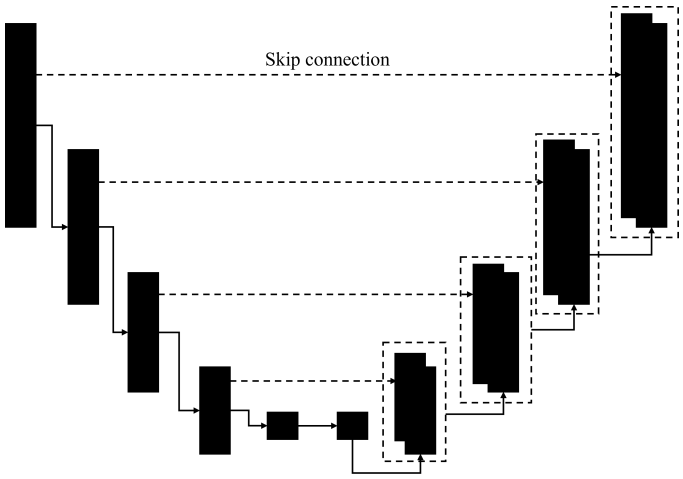

Figure 3.1: *The diagram of U-Net shows skip connections that allow the information to flow from an encoder to a decoder.*

The U-Net model, as shown in Figure 3.1, is usually applied in segmentation tasks. The network structure comprises an encoder and a decoder with skip connections that allow the information to flow directly from an encoder to a decoder of the same size. This connection help encoder to get information that may be lost during the contracting path of encoders and also reduce the problem of vanishing gradient when the model has many layers. The network used in this thesis adopted the main structure from Pix2Pix and added attention gates from the Attention U-net to the generator [18].

3.1.1.1 Attention gate

According to Oktay et al., the *attention gate* (AG) model was invented for segmentation tasks in medical imaging, which can be added to CNNs, and it does not add a considerable overhead to the main model [18].

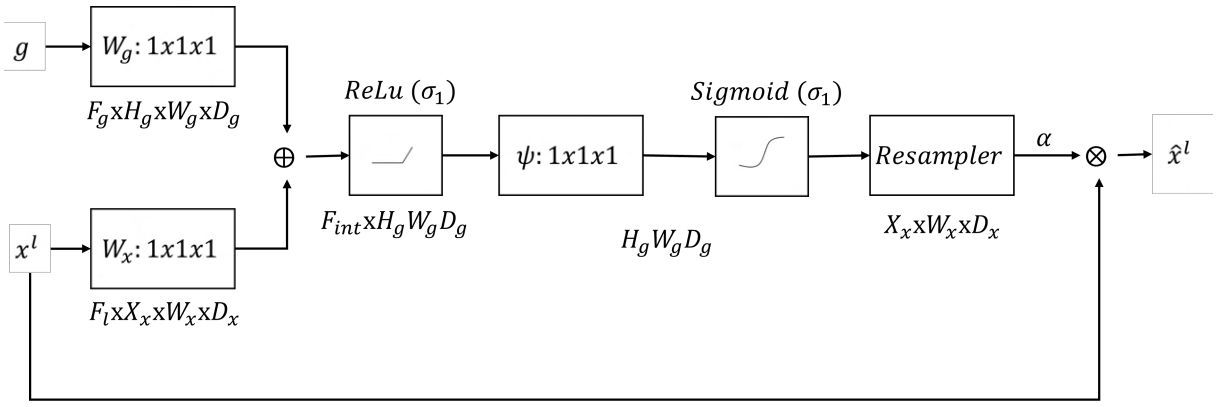

Figure 3.2: *The illustration shows computation in an attention gate where it uses information from skip connection and the next deeper layer to filter out irrelevant regions.*

In Figure 3.3, attention gates are added to the U-Net model. Before the information from the skip connection is concatenated with the features at the target layer, as shown in Figure 3.2, the AG model filter out irrelevant regions to make the model automatically learn where to focus and ignore areas that are unrelated. This can reduce the unnecessary activation of neurons in the model.

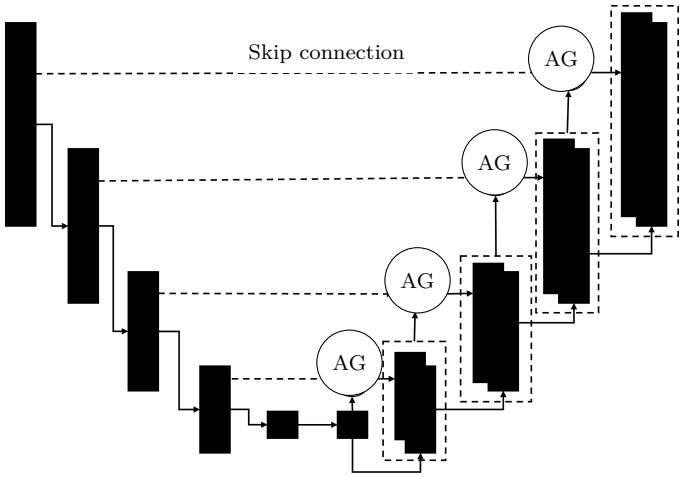

Figure 3.3: *The attention gates are implemented in the U-net model.*

3.1.1.2 PatchGAN model

The PatchGAN model is utilized as a discriminator where it classifies whether the images are real or fake. This model encoded each patch of the images. Thus instead of producing a single value, the PatchGAN generates the prediction in matrix format, as shown in Figure 3.4. Each value indicates how likely each patch is close to the actual image.

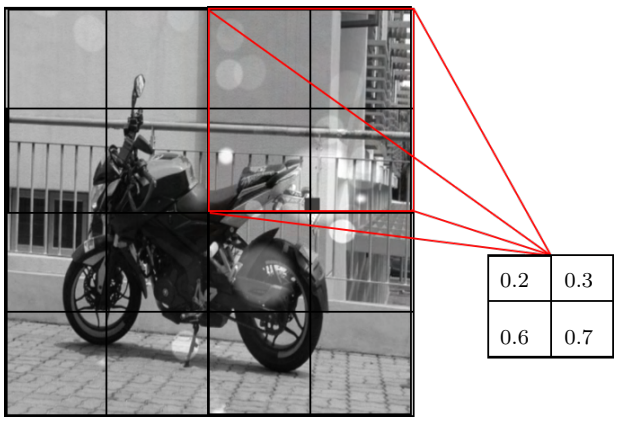

Figure 3.4: *This figure illustrates the output matrix produced by the PatchGAN. The value below 0.5 indicates the probability that the patch is fake and real otherwise.*

3.1.2 Loss function

Like any typical GANs model, the Pix2Pix model is trained in an adversarial manner. The generator tries to minimize the objective, and the discriminator competes against the generator to maximize the loss value.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}\left[log\, D(x, y)\right] + \mathbb{E}_{x,y}\left[log(1 - D(x, G(x), \theta))\right] \tag{3.1}$$

In addition to the adversarial loss, Pix2Pix use $L1$ distance to calculate the difference between real and generated images.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}\left[||y - G(x)||_1\right] \tag{3.2}$$

Finally, the total loss is:

$$G^* = arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G), \tag{3.3}$$

where λ is an arbitrary real number that performs as the weight for tuning.

3.1.3 Training

In this work, the model's configuration is as follows: the optimizer is Adam, the reconstruction parameter is 100, the batch size of four and the learning rate is 0.0004 for the discriminator and 0.0001 for the generator.

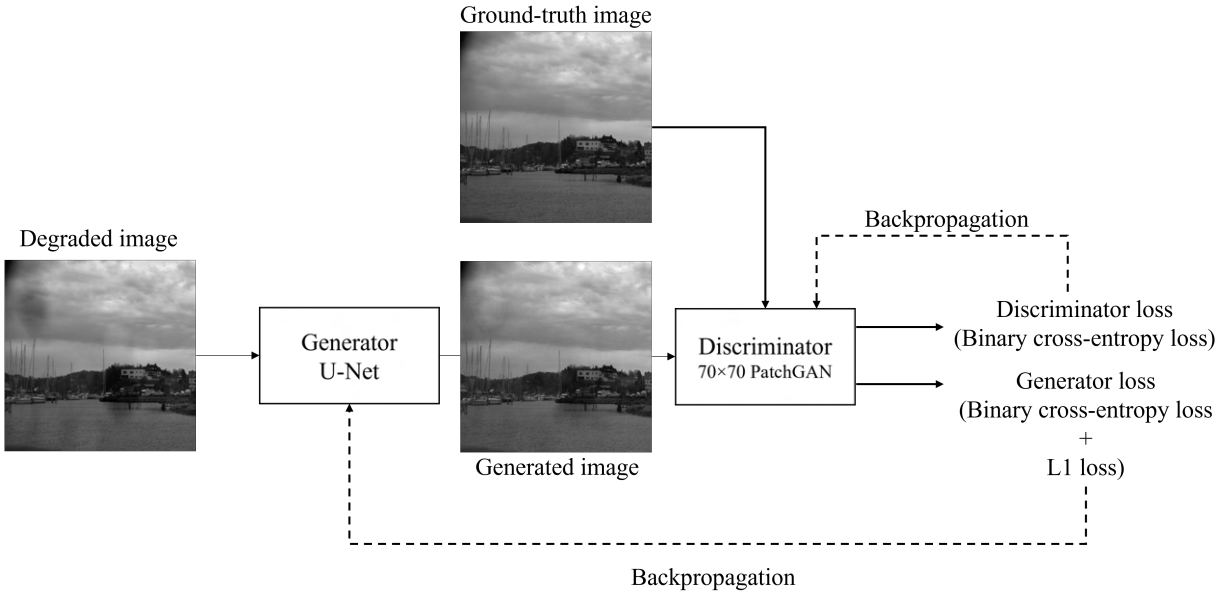

Figure 3.5: *This diagram illustrates the overview of the Pix2Pix model training.*

In Pix2Pix model training, as illustrated in Figure 3.5, the qualities of the outputs cannot be determined by objective function or loss function since GANs are models of two networks that compete against each other where the generator try to deceive the discriminator that the generated images are real. The values produced by the loss function are used to compare the performance of the discriminator and generator during the training process, whether one model overwhelms another or not. Additionally, the loss function is also used to determine when to stop the training process. Hence, it needs other metrics to evaluate the quality of generated images after the training is done.

The final model was selected based on the qualities of the generated images. The generated images are produced by feeding the degraded images from the test set that corresponds to the training set into the models. The metrics applied to measure the image qualities are SSIM, PSNR, BRISQUE and NIQE. Unlike traditional GANs, Pix2Pix does not feed noise as input to create diversity in the generated output, but it uses dropout at the first three layers of the decoder. Therefore, the input of the model is whole images.

3.2 Dataset

There are two types of datasets in this thesis: real droplets and synthetic droplets. These categories are further described in the following sections.

Datasets name	Droplets type	Size	Color Chanel	Amount	
				Train	Test
Q	Real	512x512	Monochrome, RGB	861	239
Reeds	Real	512x512	Monochrome	140	110
Reeds-gen	Synthetic	512x512	Monochrome	1028	-
Reeds-test	Real	512x512	Monochrome	-	316
Synthetic	Synthetic	512x512	Monochrome	1800	180

Table 3.1: *Metadata of datasets*

Table 3.1 shows the datasets used to train and test the models. The images with real raindrops are from the Reeds dataset and the Qian et al. dataset, which is represented in Q [1]. The images with synthetic droplets were generated using ground truth images from the mentioned sources as texture, then rendered the synthetic droplets onto them. Table 3.2 shows the quantitative quality of each dataset which is used as a reference for the generated images.

Dataset	BRISQUE	NIQE	PIQE	PSNR	SSIM
Q-clean	24.6217	3.3657	35.7552	-	-
Q-drop	26.0245	3.1312	34.5732	22.601	0.8375
Reeds-clean	34.9851	3.2699	28.6929	-	-
Reeds-sparse	25.8902	2.5568	28.0839	-	-
Reeds-dense	45.0369	4.4792	37.0778	-	-
Reeds-gen	33.604	3.5142	37.9919	29.3716	0.9916

Table 3.2: *The table illustrates the quantitative quality of each dataset.*

3.2.1 Real droplets

The pair of ground truth images and images degraded by raindrops were from the Qian et al. dataset and the Reeds dataset [1]. The Reeds dataset contained both clean images and unclean images, yet their alignment was not good. All datasets were converted to monochrome. The examples are shown in Figures 3.6 and 3.7.

Figure 3.6: *Example of images with real droplets from Qian et al. dataset [1]. Left: Degraded image. Right: Ground truth image.*

Figure 3.7: *Example of images with real droplets from the Reeds dataset. Left: Degraded image. Right: Ground truth image.*

3.2.2 Synthetic droplets

This dataset consisted of images with two types of scenes: urban streets and maritime scenes. Then water droplets were added to these ground truth images. There were three types of synthetic droplets: bokeh (Figure 3.8), distortion (Figure 3.8) and blur (Figure 3.9).

Figure 3.8: *Example of images with artificial droplets. The top row shows the droplets that distorted the covered area and its ground truth. The bottom row shows the droplets with the bokeh effect and its ground truth.*

3.2.2.1 Synthetic droplet generation

The synthetic droplets were generated using shader programming via *OpenGL shading language* (GLSL) and C++, which set the ground truth images as texture and rendered the artificial droplets over that texture, as shown in Figure 3.9.

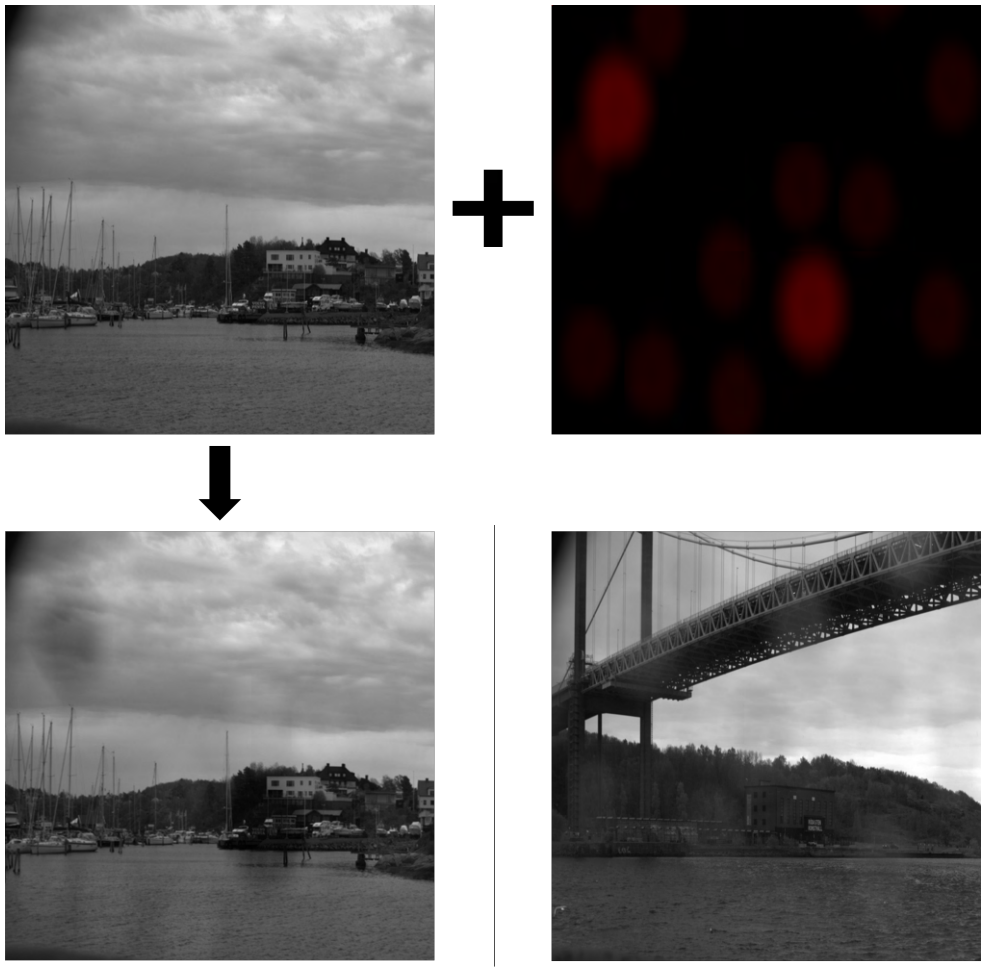

Figure 3.9: *Effect of synthetic water droplets on the image. Top-left, ground truth image. Top-right, synthetic droplets mask which blurs the covered areas. Bottom left, the result of adding the synthetic water droplet mask onto the ground truth image. Bottom-right, degraded image with real droplets.*

3.3 Evaluation

Evaluation of the generated images was done by measuring *structural similarity* (SSIM) index, *peak-signal-to-noise ratio* (PSNR) of generated images, *blind/referenceless image spatial quality evaluator* (BRISQUE) and *perception based image quality evaluator* (PIQE). The dataset from Qian et al. was used to train and test the model, then compared performance with the model trained with the synthetic images [1]. The Reeds-test dataset was used to evaluate the model that was trained by the Reeds-gen dataset. The quantitative quality of images was evaluated by the metrics in the no-reference quality metrics category since they do not have ground truth images as a reference.

3.3.1 Metric

The metrics are applied to measure the quantitative quality of the generated images. They are categorized into full-reference quality metrics and no-reference quality metrics.

Figure 3.10: *Both images are identical.*

Figure 3.11: *Left: reference image. Right: the reference image that had Gaussian noises added to it.*

Figure 3.12: *Left: Reference image. Right: Image taken from different angles.*

3.3.1.1 Full-reference quality metrics

The full-reference quality metrics are algorithms that require two images to compute the quantitative quality of the image. The quality of the target image is assessed against the reference or the original image with no distortions or degradations. Examples of this type of quality metric are the SSIM index and PSNR. They are the metrics used to evaluate the quantitative quality of the generated images. Both SSIM index and PSNR were implemented by using built-in functions from *scikit-image* [19]. The higher value of these metrics indicates better quality of the image. A more detailed description of the PSNR and SSIM follows below.

Pair of images	Metrics	
	PSNR	SSIM
Identical images	inf	1.0
Added noises	22.3654	0.8061
Different angle	15.5957	0.8685

Table 3.3: *The table shows the quantitative qualities measured by PSNR and SSIM of the images in Figures 3.10, 3.11 and 3.12.*

3.3.1.1.1 Peak-signal-to-noise ratio

PSNR is a metric which computes the ratio between the maximum possible value or power of a signal and the power of noise of two images in decibels. This metric usually measures the quality of the reconstructed or compressed images. PSNR can be defined as:

$$PSNR = 20log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right) \tag{3.4}$$

where MAX_f is the maximum signal power of pixel in the image.
MSE is:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}||f(i,j) - g(i,j)||^2 \tag{3.5}$$

From Eq. 3.5, $f(i,j)$ and $g(i,j)$ are two different images where f is the original image and g is the target image. i is the row index and j is the column index. m and n are the numbers of rows and columns, respectively. Therefore, the best score of MSE is 0. In other words, the two images are identical. For PSNR, if the result of MSE goes to zero, then PSNR become undefined and *scikit-image* returns the score as infinity. In fact, the *mean squared error* (MSE) is often used due to its simplicity in the calculation. It measures the average of the squared error. However, it does not perform well in measuring visual quality.

3.3.1.1.2 Structural similarity index

SSIM is a metric which takes the visual quality when perceived by human eyes into account. Thus, it is better than MSE and PSNR since it can provide a score relating to how humans judge good and bad images.

3.3.1.2 No-reference quality metrics

No-reference quality metrics will be selected if there is no original. Nonetheless, this type of metric yields an output that is more related to how humans evaluate the visual quality of the images since they are models that are trained on the images with good quality and do not contain any distortion, corruption or images that the types of degradation are known. The examples of no-reference quality metrics are BRISQUE, NIQE and PIQE. For these metrics, the lower value indicates the better quality of the image.

Images	Metrics	
	BRISQUE	PIQE
Original	37.2784	16.0396
Added noises	37.7933	21.2484
Change angle	34.8066	19.3034

Table 3.4: *The table shows the quantitative qualities measured by BRISQUE and PIQE of the images in Figures 3.10, 3.11 and 3.12.*

4
Results

This section shows the results obtained from the experiment described in the method section.

4.1 Droplet removal

In the droplet removal task, each model was trained for 75 epochs. The models were trained by using the reconstruction parameter of 100. The batch size was four, and the learning rate was 0.0004 for the discriminator and 0.0001 for the generator.

Models	Training sets	Test sets (SSIM)		
		Qian et al.	Qian et al. (ground truth)	Synthetic
Pix2Pix	Qian et al.	0.7849	0.9042	-
Pix2Pix with AG	Qian et al.	0.8355	0.9175	-
Pix2Pix	Synthetic	0.7849	0.8925	0.8666
Pix2Pix with AG	Synthetic	0.7944	0.8925	0.8701

Table 4.1: *SSIM values of the generated images. Each model in the model column was trained by the corresponding training set in the training set column.*

Table 4.1 and Table 4.2 present SSIM and PSNR values of generated images, respectively. The models used to perform droplet removal were Pix2Pix and Pix2Pix with attention agate. Each model was trained by the Qian et al. dataset, which contains images degraded by real droplets and the synthetic dataset to see the difference in droplet removal performance.

Models	Training sets	Test sets (PSNR)		
		Qian et al.	Qian et al. (ground truth)	Synthetic
Pix2Pix	Qian et al.	20.5971	21.4304	-
Pix2Pix with AG	Qian et al.	22.0186	22.6698	-
Pix2Pix	Synthetic	19.0978	21.4305	18.7407
Pix2Pix with AG	Synthetic	19.0704	21.4305	18.8928

Table 4.2: *PSNR values of the generated images. Each model in the model column was trained by the corresponding training set in the training set column.*

The generated images from the models trained by the Qian et al. dataset yielded higher SSIM and PSNR values than the models trained by the synthetic dataset.

Additionally, though the quality of the generated images when input images were ground truth images from the Qian et al. dataset were higher than the other test set, the models did not return the generated image that had the same properties as the input images. Furthermore, the Pix2Pix with attention agate model generated images with higher quality than the Pix2Pix model. Hence the Pix2Pix with attention agate model was selected to perform the droplet removal task for the Reeds test set and object detection task (see Tables 4.1 and 4.2).

Figure 4.1: *Left: The ground truth image. Right: The image degraded by real droplets.*

Regarding the quantitative quality of generated images, the results conformed with the quantitative quality in Table 4.1 and Table 4.2 as the image that was generated by the Qian et al. model and Pix2Pix with attention gate had fewer droplets than the rest of generated images. Moreover, these two models could reconstruct the images that were similar to the ground truth images. However, the models that were trained by synthetic were not able to remove droplets and could not reconstruct the areas that were covered by the real droplets. (see Figures 4.1 and 4.2)

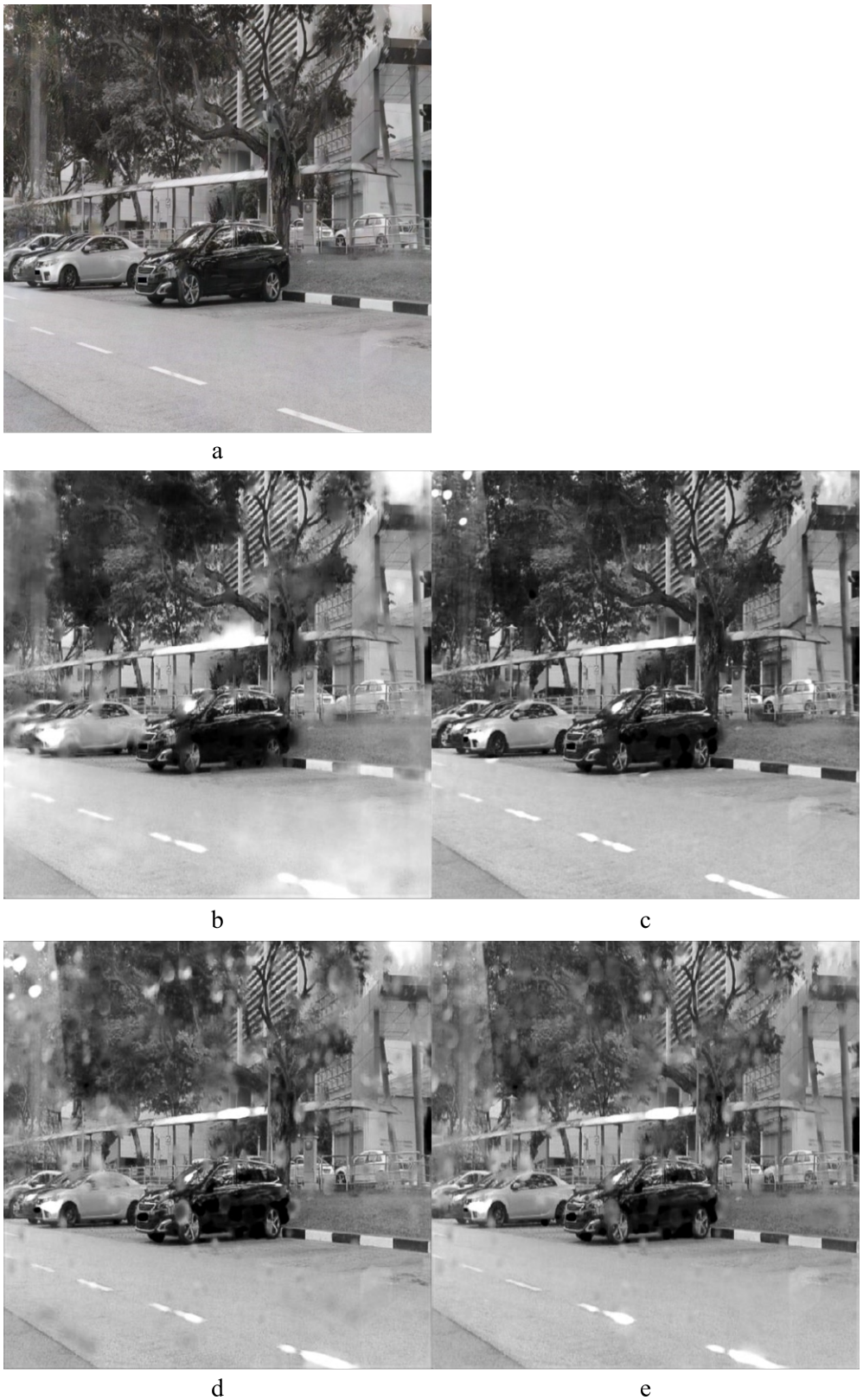

Figure 4.2: *Example of generated images. (a) Image generated by Qian et al. model. (b) Image generated by Pix2Pix and trained by actual droplets. (c) Image generated by Pix2Pix with attention gate and trained by real droplets. (d) Image generated by Pix2Pix and trained by synthetic droplets. (e) Image generated by Pix2Pix with attention gate and trained by synthetic droplets.*

4.1.1 The Reeds dataset

The Reeds training set can be separated into synthetic and augmented training sets. The synthetic training set was the images that contained only synthetic droplets, but the augmented training set consisted of images with actual droplets and synthetic droplets. In this experiment, evaluation was done by feeding images from the Reeds test set to the models.

Models	Training sets	Reeds test set (translucent)		
		BRISQUE	NIQE	PIQE
Pix2Pix with AG	Reeds-Synthetic	20.9924	3.1874	32.3405
Pix2Pix with AG	Reeds-Augmented	20.9536	3.1653	32.7896
Qian et al.	pre-trained	22.9683	2.6787	31.3793

Table 4.3: *The quantitative qualities of the images generated by the models in the first column. The test set was the Reeds images, where the characteristics of droplets were translucent.*

In terms of quantitative quality, all models produced images that had comparable quality to each other (see Tables 4.4 and 4.3). However, the image produced by the Pix2pix with AG, which was trained by using the synthetic the Reeds dataset, contained unwanted artefacts (see Figure 4.3). Besides, regarding the qualitative quantity of output images from translucent and opaque input images, the models generated better results in translucent test images than the opaque test images.

Models	Training sets	Reeds test set (opaque)		
		BRISQUE	NIQE	PIQE
Pix2Pix with AG	Reeds-Synthetic	38.0921	5.1369	29.1417
Pix2Pix with AG	Reeds-Augmented	41.5463	4.9844	31.3551
Qian et al.	pre-trained	29.0285	4.0908	47.2417

Table 4.4: *The table shows the qualities of the images generated by the models in the first column. The test set was the Reeds images, where the characteristics of droplets were opaque.*

Figure 4.3: *(a): The degraded image from the Reeds-test in translucent type. (b): Image generated by Qian et al. model. (c): Image generated by the Reeds-synth model. (d): Image generated by the Reeds-augmented model.*

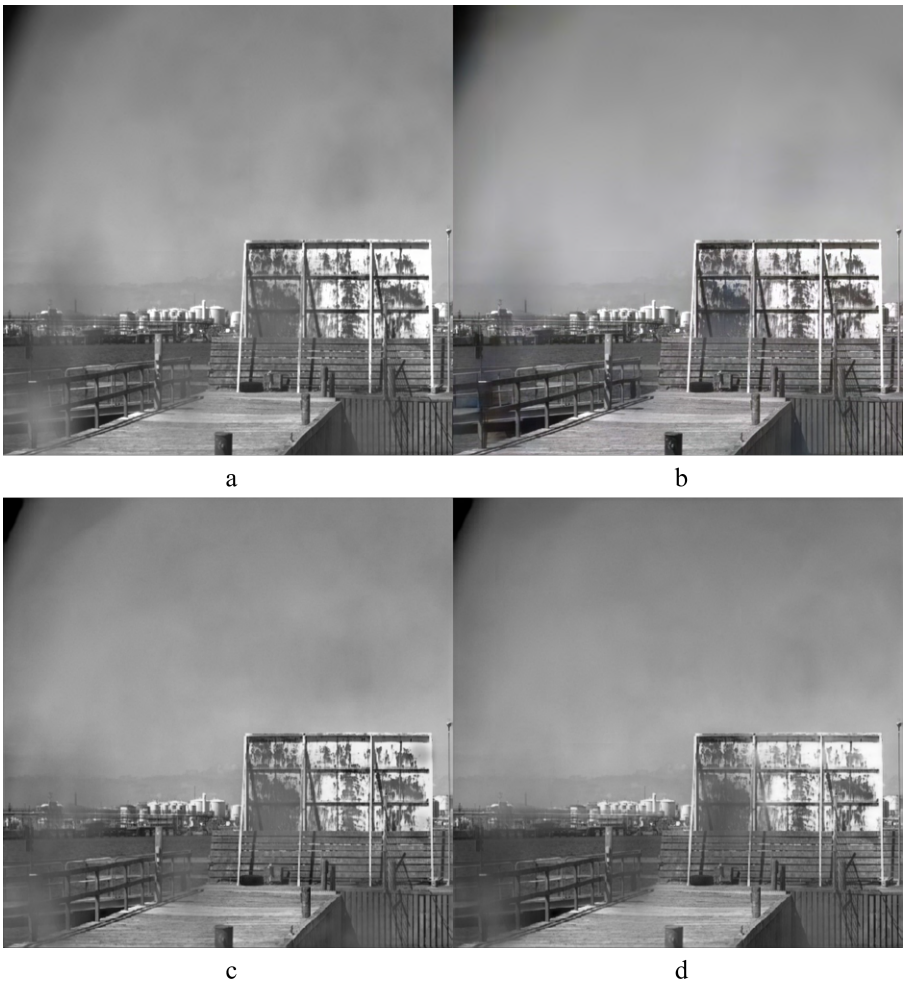

Figure 4.4: *(a): The degraded image from the Reeds-test in opaque type. (b): Image generated by Qian et al. model. (c): Image generated by the Reeds-synth model. (d): Image generated by the Reeds-augmented model.*

4.2 Object detection

In the object detection task, different types of generated images were fed into the object detection model (DETR) to see the impact of the generated model on the performance of the object detection model. The inputs were ground truth and degraded images of the Qian et al. dataset, generated images from Qian et al. model and generated images from Pix2Pix with the AG model.

Models	Training sets	BRISQUE	NIQE	PIQE	SSIM	PSNR
Pix2Pix with AG	Qian et al.	24.8773	3.4145	41.5406	0.8883	21.1898
Qian et al.	pre-trained	19.1434	3.0019	34.3537	0.9871	34.3653

Table 4.5: *The table shows the qualities of images generated by Qian et al. model and Pix2Pix with AG model [1].*

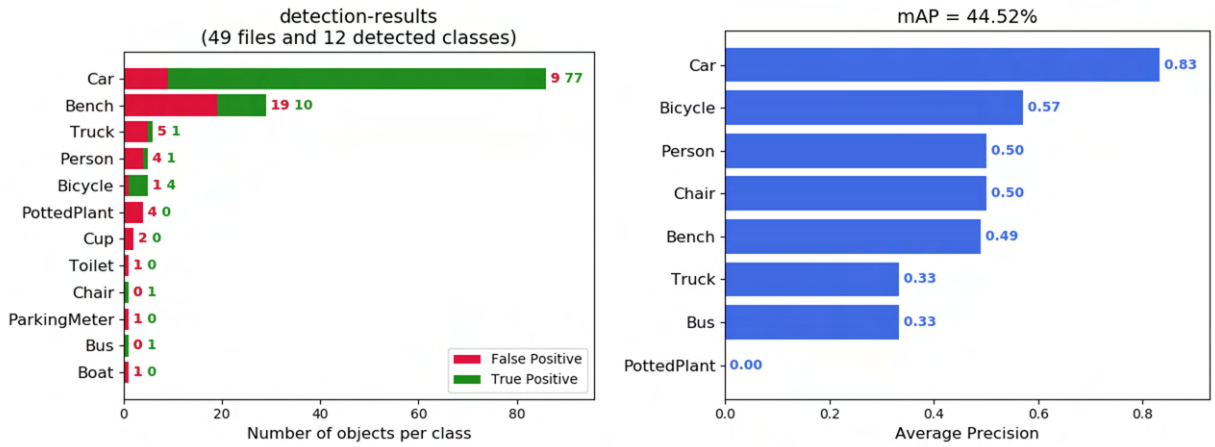

Figure 4.5: *mAP from ground truth images*

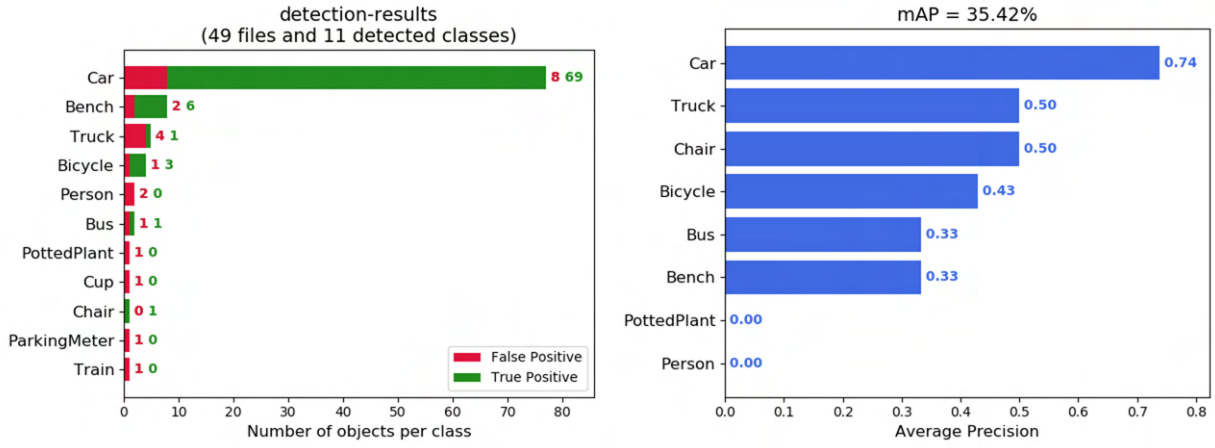

Figure 4.6: *mAP from degraded images*

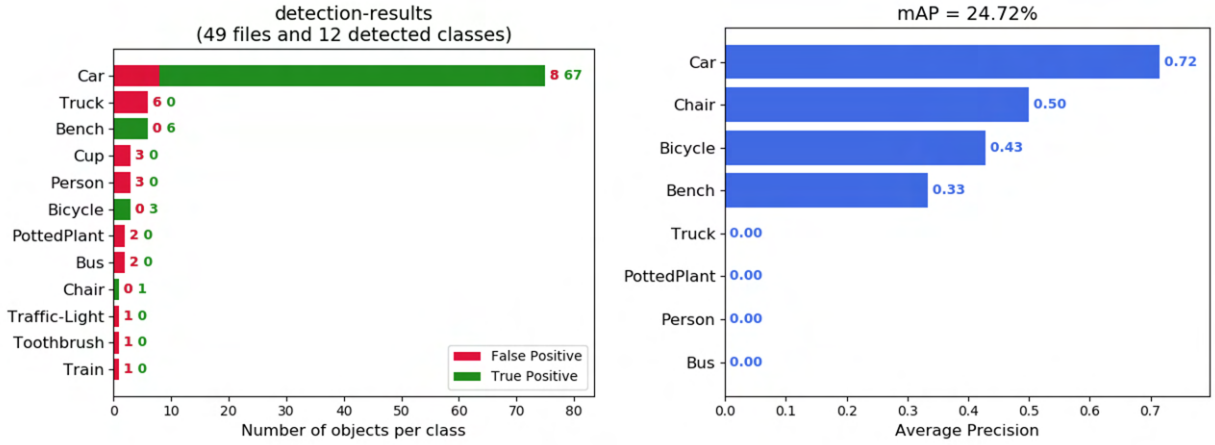

Figure 4.7: *mAP from images generated by Pix2Pix with AG model*

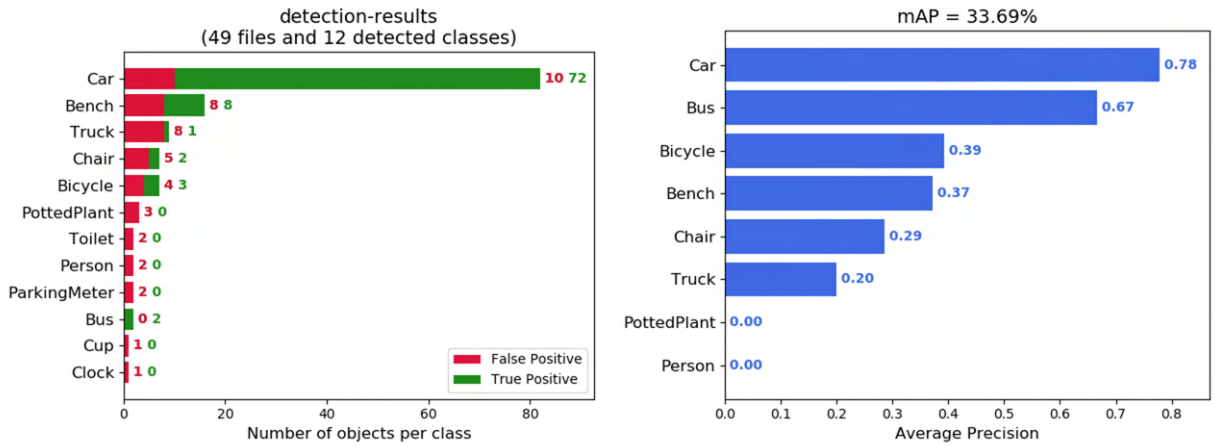

Figure 4.8: *mAP from images generated by Qian et al. model*

Figures 4.5, 4.6, 4.7 and 4.8 show that the generated images did not yield an mAP score higher than when inputs were ground truth images and degraded images. In other words, the DETR model can perform better in ground truth images and degraded images. However, in Table 4.5, Qian et al. model could produce images with an SSIM value of nearly one.

5
Discussion and improvement

This chapter discusses the failure of the models, the evaluation of the generated image quality and the possibility of implementing an automated evaluation system.

5.1 Generative adversarial network training

Generative adversarial networks were trained based on the concept that two models were trained simultaneously in an adversarial manner. Therefore, it is challenging to train this type of network since one might outperform another model and return unwanted feedback to its opposition model. The problem in GANs training found during this project were overfitting, generalization, and artefacts.

5.1.1 Overfitting and generalization

Overfitting is a phenomenon that can happen if a model learns too much from the training dataset. In the other type of model, for example, classification task, this could cause the model to classify the object wrongly because the model learnt noise and essential features of the training set too well and lost the ability to adapt to unseen data.

In the case of GANs, the input of the GANs model is usually a set of noise or a vector of random integers. The model will then process this input to generate the synthetic images. If the model was trained until it became overfitted, the generated images would have the same structure as the images in the training set. This problem is shown in Figure 5.1 where the model learnt from the dataset that has low diversity. This problem also leads to the model being unable to generalise the unseen data. Though the model could perform decently on the test set of similar images, it could not adapt to the unseen dataset and resulted in generated images that had the main structure from the images in the training dataset. However, in GANs, overfitting and mode collapse must not be understood as the same problem because mode collapse will be recognised if one type of training data is generated repeatedly while the rest are not produced.

Furthermore, in the Reeds test set, the models were not able to get rid of the droplets,

Figure 5.1: *The model shows signs of overfitting after training for 40 epochs. Left: Images from the train set. Right: Images from the evaluation set.*

but they were able to reconstruct a clearer scene instead. Thus the mixed model was fine-tuned to make it able to remove droplets from the Reeds dataset. This happened since the model cannot robustly acknowledge and locate the droplets; thus, the model recognized them as part of the scene where they should not be removed. Therefore, the model still requires further training to make it more adaptable to the new dataset. This problem is not just due to the shape or size of droplets but also because of the specification of the camera.

5.2 Artifacts

From the result in Sect. 4, the models generate satisfying results in terms of quantitative quality. However, some concerns should be raised after the inspection of qualitative quality, such as the model's reliability. In Figure 5.2, the area inside the blue circle shows the location of droplets. Though the model could generate realistic images without droplets, it extended the area of dark stains on the floor. The artefacts were undesirable because they could be misleading for the task that uses generated images in analyzing or making the decision. However, this problem

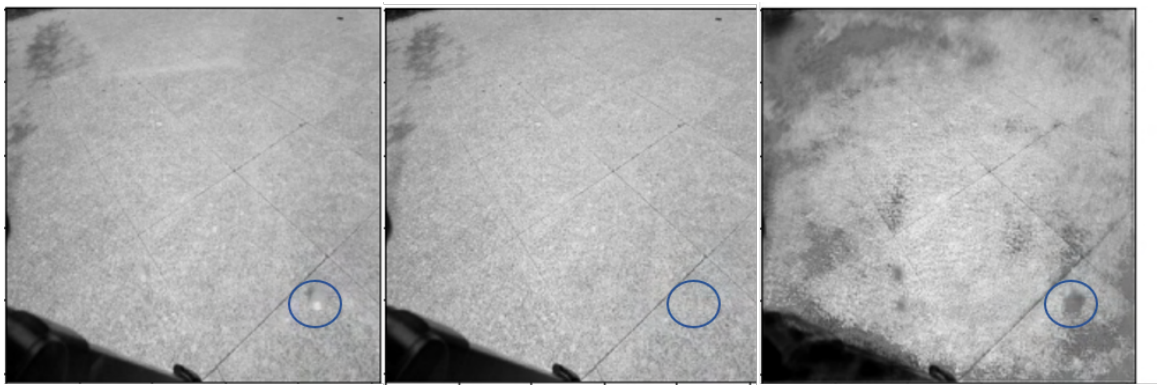

Figure 5.2: *The generator model generated images that contained features that did not exist in ground truth images and degraded images. Left: degraded image. Middle: ground truth image. Right: generated image.*

is not unexpected since GANs generated images from the data in the latent space that the model learned during the training.

5.3 Potential improvement

From the result section, we can see that there is room for improvement, for example, in the quality of synthetic droplets and the training strategy of the model. The details are explained in the following section.

5.3.1 Object detection

In this task, The models performed poorly on the images in which the droplets were removed. Though the PSNR and SSIM and qualitative quality of the images were acceptable to human eyes, the model could not recognize the object and tended to misclassify them. From the result, even though the quantitative quality of the generated images was better than the degraded images, the mAP of degraded images was better.

In Figure 5.3, it could be seen that the DETR model performed well on the ground truth images where there was only one failed detection, as shown in the pink bounding box. In the images (a) from Figure 5.4, the object detection model failed to detect one object while the generated images were able to recover the object.

As shown in the result, the higher-quality generated images could lead to better performance in object detection tasks than those with lower quality. However, there is also a possibility that the object detection model failed when the images were

interpretable and had acceptable qualities. The image (d) from Figure 5.4 was one example of when the model failed to detect the object even though the qualities of the image, as shown in Table 5.1, were better than the rest. Therefore, segmentation might be a tool to analyze further how the model classified each pixel of the generated images.

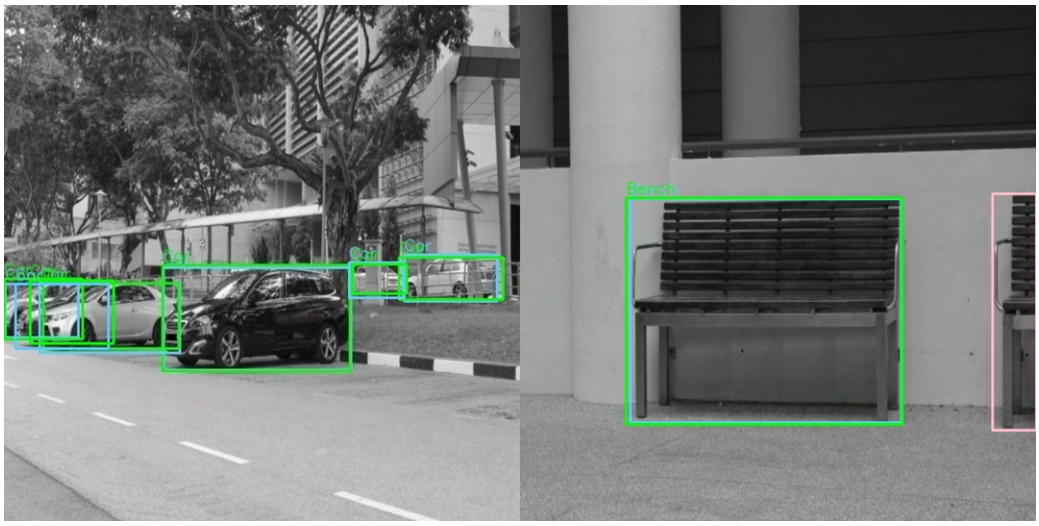

Figure 5.3: *Left and right: the result of object detection when the input images are ground truth images.*

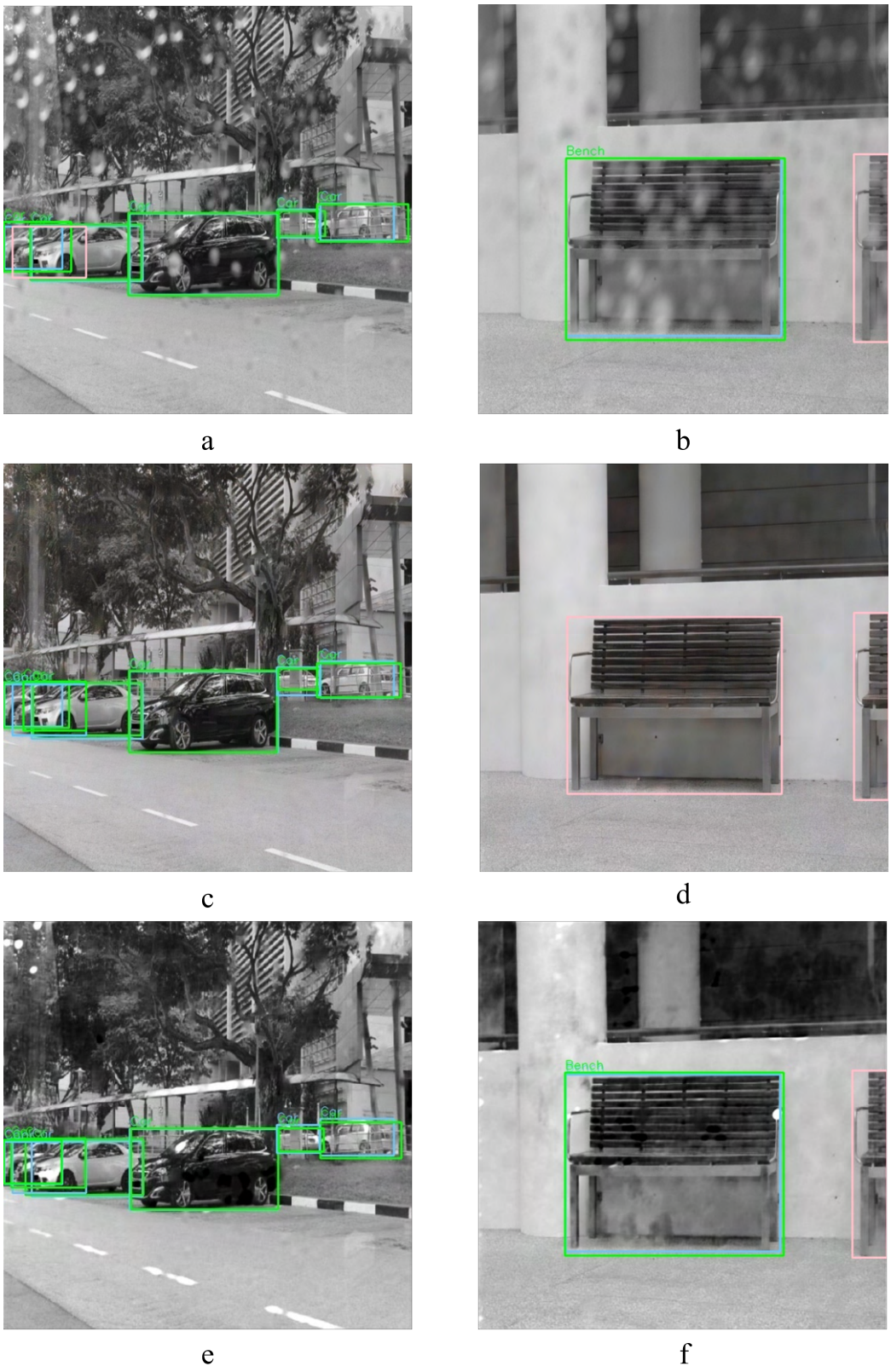

Figure 5.4: *(a) and (b): Images degrade by real droplets. (c) and (d): Images generated by Qian et al. model [1]. (e) and (f): Images generated by Pix2Pix+AG model*

Images	Models	Metrics	
		PSNR	SSIM
a	Degraded	31.1623	0.8232
c	Qian et al.	29.4913	0.7098
e	Pix2Pix+AG	31.5981	0.8568
b	Degraded	27.6103	0.8103
d	Qian et al.	27.5571	0.8873
f	Pix2Pix+AG	27.3576	0.8664

Table 5.1: *This table shows The qualities of the images in Figure 5.4*

5.3.2 Synthetic droplets

From the result section, the model that was trained by using the synthetic dataset can remove droplets and reconstruct the scene to an extent. This dataset could be used to evaluate the performance of the model if the actual dataset cannot be obtained. Thus, it still cannot replace the real droplet dataset. However, the synthetic dataset still holds one significant advantage over the real-world datasets where the alignment of ground truth images and degraded images are perfectly overlapping.

Therefore, improving the synthetic dataset, even though it may not replace the real-world dataset, could enhance the model's generalisation ability. From the method chapter in Sect. 3.3.2, there were three types of synthetic adherent droplets. The focused droplet had the main characteristic where it is transparent, and it could distort areas where it covered. However, in real droplets, these areas also blurred, and hazy and unfocused droplets increased opaqueness and blurred the images.

5.3.3 Model training strategy

In this work, the models were trained for 75 epochs with an initial learning rate of 0.0004 for the discriminator and 0.0001 for the generator, batch size of four and reconstruction constant was 100. To improve the stability of the model and make them able to be trained longer, the batch size should be increased so that the trainable parameters do not update too often, which could also lead to the models converging to any undesired local optima.

5.3.4 Automated evaluation

In real-world applications, the machine learning model or deep learning model is only a tiny part of the whole system. Thus automated evaluation could speed up the deployment and improve the life cycle management of the system. The automated evaluation system will take care of the tasks that can be handled automatically by the system instead of manually done by the developers.

The amount of data that the Reeds system has collected will keep growing, and these data will be difficult to process and annotate manually by researchers or developers. Therefore, automated evaluation is needed to accelerate the process of data pre-processing and data annotation.

The automated evaluation for the droplet removal system shall be able to detect droplets in images, annotate the images used in the model training and testing, and evaluate model performance. The users shall interact with the system via the web application's front end. This web application allows the users to see the result of the model evaluation, metadata of the datasets, manually annotate data and train the model.

5.3.4.1 Conceptual design

In a microservice architecture, several factors must be considered: communication between services, data management, security, scalability, and maintainability. This project was not to dive deep into the detail of these factors and the combination technology stack for the system. Besides the architecture, the systems may be deployed via CI/CD pipeline, which is connected to the cloud service. This will shorten the time for delivery since the pipeline can perform the task automatically, as stated in the script for the pipeline.

Figure 5.5 illustrates the conceptual design of the automated evaluation system. The design adopted the microservice architecture where the system consists of several services, and each service is dockerized and built into a deployable container. In a microservice architecture, each service should be self-contained and independent, resulting in a loosely-coupled system. In other words, when one service fails to operate, it should not impact other services, and each service should be able to be deployed individually. In addition, the system should be designed to fail fast. This means that the system must fail once an error occurs and notify the users or the system's admin about the error, making the system more stable since errors are visible to the developers.

All services will perform the operation as the name suggest. However, the droplet detection service will perform two subtasks within the service. In Figure 5.6, 5.8, the droplet detection service would be triggered if there are images coming to the service. If it cannot detect raindrops on the image, then that image would be stored as a ground truth image. This process continues as the new clean image overwrites the old one. In case the model can detect droplets on the image, the service generates annotation and saves it in storage. This dataset can later be used for the evaluation and training of the droplet removal model.
From Figure 5.5, the API gateway is an entry point for the clients. It will route the clients' requests to the corresponding services to perform the task. The cluster of microservices is managed by a microservices orchestration framework which mainly

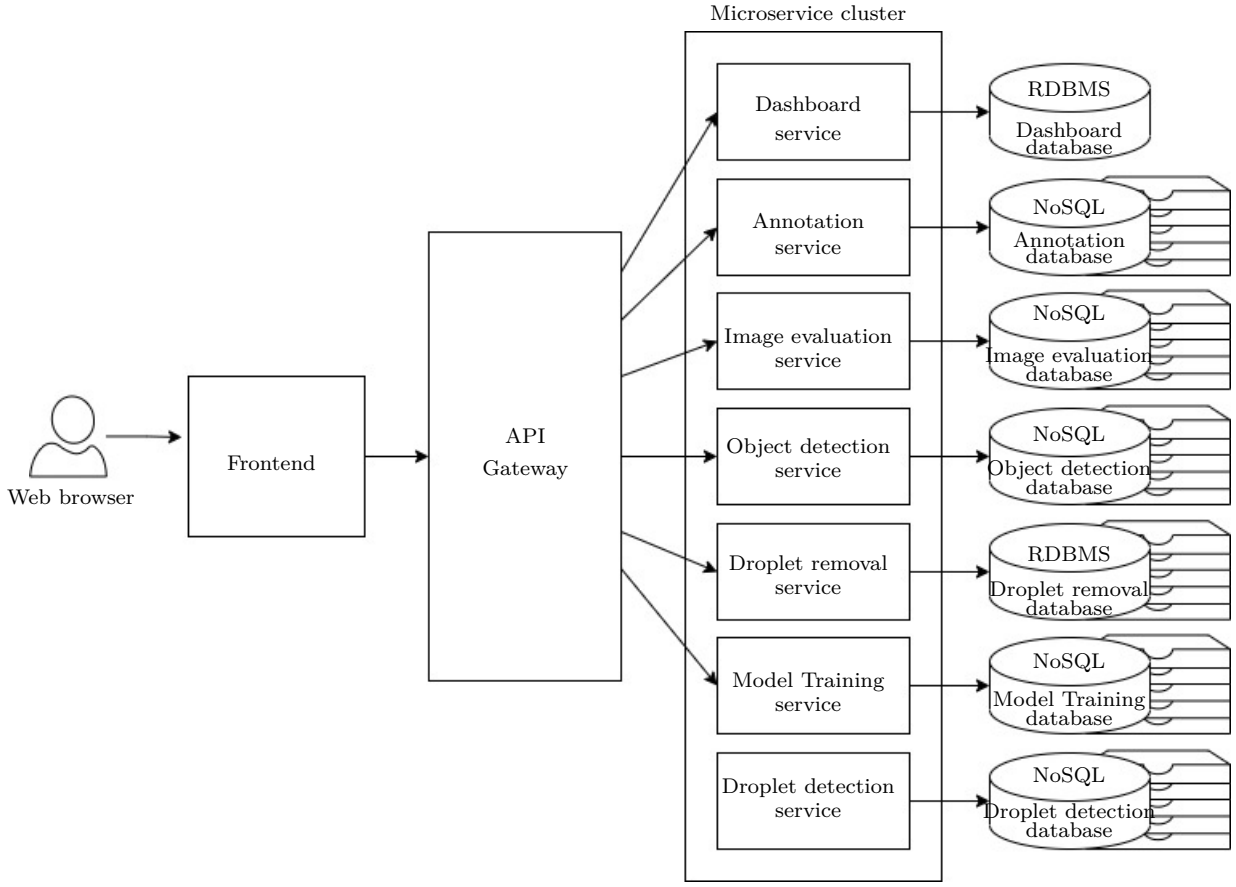

Figure 5.5: *This figure illustrates the conceptual design of the automated evaluation system. The structure of the system is based on the microservices architecture.*

manages the execution of service and resource utilization. The system follows the database-per-service pattern, where each service has its own database. Additionally, the design adopts polyglot persistence where each service can have its own data model and use specific technology that is most suitable for its task. For instance, in dashboard service, *relational database management system* (RDBMS) is chosen since the data can be simply modelled as in Figure 5.7. The *entity-relationship diagram* (ERD) in Figure 5.7 shows one-to-many relationship from MODEL_METADATA table to MODEL_SCORE table. The reason for one-to-many relationship is that a model may be tested by more than one metric. Thus MODEL_ID and METRIC_NAME are combined into a composite key to make a record unique. In other services like droplet removal service, droplet detection service, annotation service, image evaluation service, and model training service, NoSQL database are used since it would be a complex data model for RDBMS and it is more readable to a human when data are kept in the format of JSON or XML.

From Figure 5.8, Though the services have their own database, they still need to replicate data from other services. Therefore, the system needs another middleware to manage the data exchange between services. The mechanism or technology of the

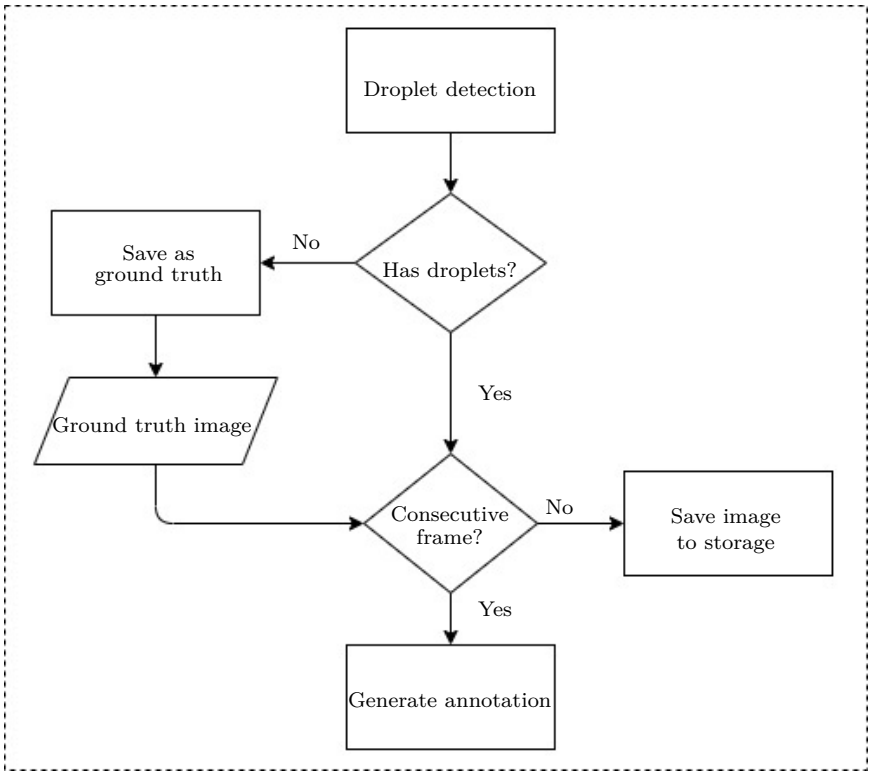

Figure 5.6: *This figure illustrates the flow diagram of the droplet detection service. It performs the droplet detection task and then creates annotation if the conditions are met.*

communication between services depends on the type, volume, frequency and characteristics of data that is being exchanged. The communication could be in the form of publish-subscribe, client-server or scheduled time to send data to the consumer. To prevent the system from having too much redundant data, the source system would retain the data for a limited period of time. After the period has passed and data are replicated to the consumer successfully, the data should be eliminated from the source system.

In the situation of an unstable network arises, the service shall be able to retry a couple of times. However, it must only keep trying for a while since it would consume an unnecessary amount of resources. Hence circuit breaker pattern shall be implemented to stop the services from attempting to communicate. This pattern allows the service to retry the sending request a limited number of times. Then, if it cannot commit the transaction successfully, the attempt will be paused for a time period. The circuit breaker mechanism would allow the service to try again after the paused period is ended.

In terms of deployment in the cloud environment, the system may adopt *software as a service* (SaaS) architecture. SaaS architecture can be utilized if the application is deployed in a remote server like the cloud server. Then, the application can be delivered over the internet.

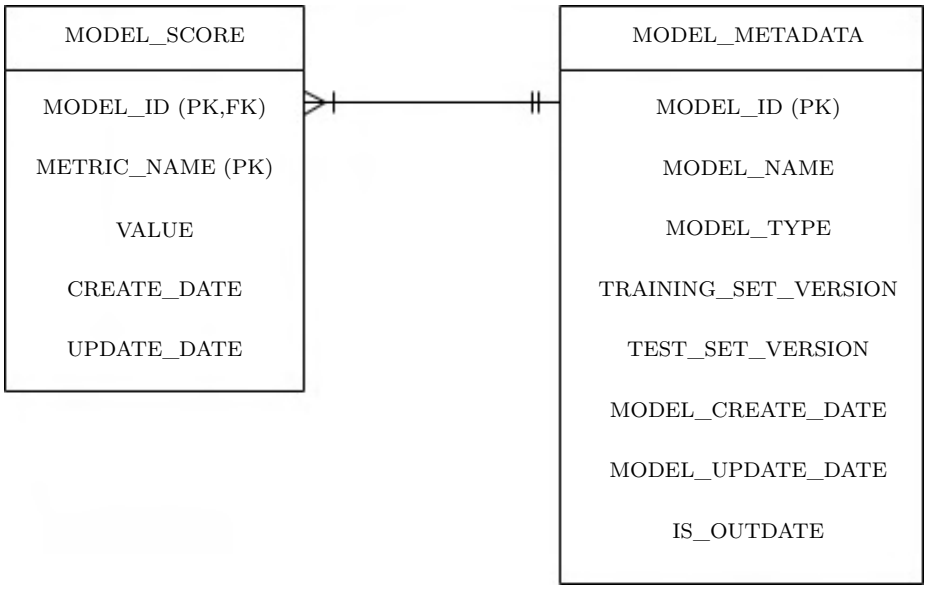

Figure 5.7: *The illustration of entity-relationship diagram for dashboard service. The PK and FK are primary key and foreign key, respectively. This diagram shows one-to-many relationship between MODEL_METADATA table and MODEL_SCORE table*

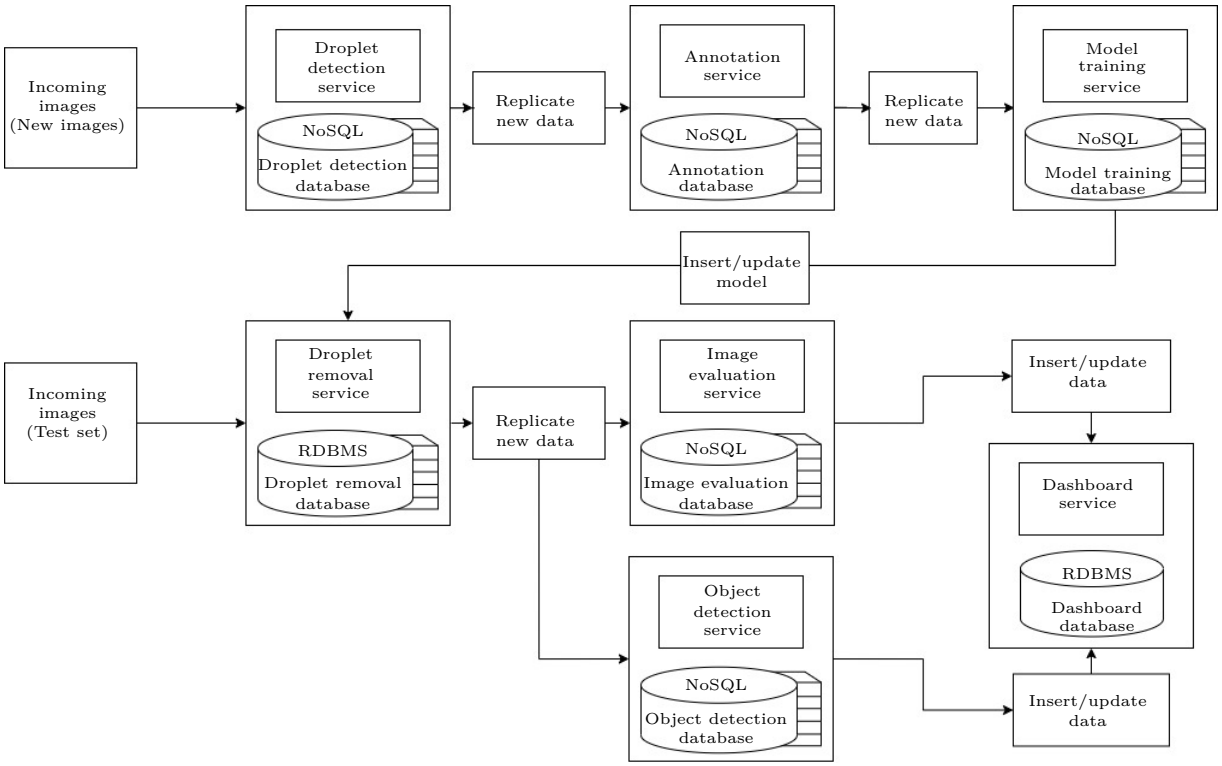

Figure 5.8: *This figure illustrates the flow of data in the automated evaluation system. A service replicates the data to the next service once they are successfully persisted in the database of the service that produced the data.*

6
Conclusion

This work was conducted to study the use of GANs model in the water droplet removal task. The main concept of this task was to leverage the image inpainting where the model fills the hole or missing areas of the image to reconstruct the image. Though the models produced acceptable outcomes in terms of quantitative quality and made the scene more understandable to the human eyes, they did not improve the performance of the object detection task. This problem could happen because the object detection model was more sensitive to how the pixels form the features such as edge, line and shade. Therefore, it does require higher quality of the generated image to overcome this problem which could be done by improving the training strategy or redesigning the model.

In terms of the synthetic dataset, in its current state, it can only be used in preliminary verification of the model's performance in case of scarcity of the training data. Thus, to improve its usability as the primary training set or as augmented data, the shape, colour, opacity, blurriness, refraction and reflection of the droplets must be taken into further improvement. For example, the current synthetic droplets did not blur and distort the areas at the same time. Furthermore, in real droplets, each drop did not have the same thickness, and it caused a wide variety of effects such as glare, bokeh and flare on the images rather than just distortion in the same image.

For the automated evaluation system, this part represented the conceptual design and the possibility of adopting different kinds of microservice design patterns. However, for a more detailed design, the requirement must be collected and analyzed further because the functional and non-functional requirements should be clarified in order to make the system reliable and stable. There are some aspects and design patterns that could be investigated further, for instance, the SAGA design pattern; this pattern could be applied for services communication and to manage data consistency of the services. In the aspect of security, the system could be utilized an authentication and authorization mechanism to limit the accessibility and keep track of the users who logged in to the system. Performance monitoring may be needed to study further as the system consists of many services. Thus, it would be beneficial to the system owner to notice the abnormality that could happen at any time. Furthermore, to implement the design in the real-world environment, there are a few more factors that must be investigated, such as the volume and frequency of the data produced by the source and the user interface of the system.

Bibliography

[1] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[2] Shizhe Zang, Ming Ding, David Smith, Paul Tyler, Thierry Rakotoarivelo, and Mohamed Ali Kaafar. The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE vehicular technology magazine*, 14(2):103–111, 2019.

[3] Wang, Hong and Wu, Yichen and Li, Minghan and Zhao, Qian and Meng, Deyu. A survey on rain removal from video and single image. *arXiv preprint arXiv:1909.08326*, 2019.

[4] Hiroyuki Kurihata, Tomokazu Takahashi, Ichiro Ide, Yoshito Mekada, Hiroshi Murase, Yukimasa Tamatsu, and Takayuki Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 205–210. IEEE, 2005.

[5] Xunshi Yan, Yupin Luo, and Xiaoming Zheng. Weather recognition based on images captured by vision system in vehicle. In *International Symposium on Neural Networks*, pages 390–398. Springer, 2009.

[6] Aurelien Cord and Nicolas Gimonet. Detecting unfocused raindrops: In-vehicle multipurpose cameras. *IEEE Robotics & Automation Magazine*, 21(1):49–56, 2014.

[7] Martin Roser and Frank Moosmann. Classification of weather situations on single color images. In *2008 IEEE Intelligent Vehicles Symposium*, pages 798–803. IEEE, 2008.

[8] Martin Roser and Andreas Geiger. Video-based raindrop detection for improved image registration. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 570–577. IEEE, 2009.

[9] Martin Roser, Julian Kurz, and Andreas Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Asian conference on computer vision*, pages 235–244. Springer, 2010.

[10] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detection and removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015.

[11] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2463–2471, 2019.

[12] Fadi Al Machot, Mouhannad Ali, Ahmad Haj Mosa, Christopher Schwarzlmüller, Markus Gutmann, and Kyandoghere Kyamakya. Real-time raindrop detection based on cellular neural networks for adas. *Journal of Real-Time Image Processing*, 16(4):931–943, 2019.

[13] Ülkü Uzun and Alptekin Temizel. Cycle-spinning gan for raindrop removal from images. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2019.

[14] Horia Porav, Tom Bruls, and Paul Newman. I can see clearly now: Image restoration via de-raining. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7087–7093. IEEE, 2019.

[15] Jinah Kim, Dong Huh, Taekyung Kim, Jaeil Kim, Jeseon Yoo, and Jae-Seol Shim. Raindrop-aware gan: Unsupervised learning for raindrop-contaminated coastal video enhancement. *Remote Sensing*, 12(20):3461, 2020.

[16] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[18] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

[19] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

Department of Mechanics and maritime sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY