Analysis of the influence of de-hazing methods on vehicle detection in aerial images

Khang Nguyen, Phuc Nguyen, Doanh C. Bui, Minh Tran, Nguyen D. Vo
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam

***Abstract*—In recent years, object detection from space in adverse weather, incredibly foggy, has been challenging. In this study, we conduct an empirical experiment using two de-hazing methods: DW-GAN and Two-Branch, for removing fog, then evaluate the detection performance of six advanced object detectors belonging to four main categories: two-stage, one-stage, anchor-free and end-to-end in original and de-hazed aerial images to find the best suitable solution for vehicle detection in foggy weather. We use the UIT-DroneFog dataset, a challenging dataset that includes a lot of small, dense objects captured in various altitudes, as the benchmark to evaluate the effectiveness of approaches. After experiments, we observe that each de-hazing method has different impacts on six experimental detectors.**

***Keywords*—foggy weather, vehicle detection, DWGAN, Two-Branch, YOLOv3, Sparse R-CNN, Deformable DETR, Cascade R-CNN, CrossDet, adverse weather**

I. Introduction

Nowadays, with the significant development of information and communication technology, especially the surging growth of the deep learning era, vehicles detection and surveillance have become extremely important and necessary. Along with that development, the prevalence of UAVs (Unmanned Aerial Vehicles) makes vehicle surveillance extremely simple and effective. Traffic surveillance from aerial images is a particular interest to researchers because it serves many essential different purposes, such as being used in the military or monitoring traffic conditions, urban management, or simply helping us know where to park in the parking lot. Compared to detecting ground view vehicles, detecting vehicles from the aerial image is more complicated and challenging due to multiple frames, various backgrounds, small objects appearing with a variety of shapes. In addition, there is a critical factor that directly affects the model's performance: the weather.

Recently, by applying deep convolution neural network CNN [1], a significant number of studies have been done to tackle this problem, such as the proposal of the Double focal loss convolutional neural network framework (DFL-CNN) model [2] with the combination of feature CNN and the focal loss function [3] or Lu *et al.* [4] carried out experiments on YOLO [5] for vehicle detection based on aerial images datasets COWC [6], VEDAI [7] and DOTA [8]. These articles have one thing in common: they were all experimented with in clear-weather conditions and achieved excellent results.

However, in reality, the weather conditions are constantly changing (rain, snow, smog, night, thunderstorms, fog, etc.), which significantly affects the accuracy and learning ability of the models.

We choose the scope of our research to detect vehicles from aerial images under foggy weather. We choose the fog scene because it is often seen in the early morning. Sometimes, the appearance of dense fog may seriously affect the ability to monitor the traffic situation. Prior studies such as SFA-Net [9] used to detect objects in the rain or solve the problem of semantic foggy scene understanding (SFSU) by de-fogging of convolutional neural networks [10].

In this study, we focus on investigating advanced object detection methods Cascade R-CNN [11], Casdou [12], YOLO-v3 [13], CrossDet [14], Deformable DETR [15], Sparse R-CNN [16] for object detection in foggy conditions that limit visibility. At the same time, the evaluation of visibility improvement through 2 de-fogging (Image Dehazing) methods, which are DW GAN [17] and Two-branch Dehazing [18], also implemented on the UIT-DroneFog [12].

We summarize our contributions in this paper as:

- Using image dehazing methods to filter foggy images from the challenging dataset UIT-DroneFog.
- Experimenting with one-stage, two-stage, and the latest end-to-end deep learning methods on dehazed datasets.
- Providing in-depth evaluation of dehazing methods on experimental deep learning models to choose the best models.

The rest of the paper is II Related work; III is Experimental Methods; IV is Experimental Results; and finally, the Conclusion and Future Work.

II. Related Work

Object detection in the foggy condition in particular and in adverse weather conditions has been addressed in two directions: domain adaption and condition-based object detection. In domain adaptation approaches, many studies proposed to improve the more robust detector based on Faster R-CNN [19, 20, 21] to help it be adapted with other domains of images, which could be different from the original dataset. These domain-adapted detectors were trained on the source dataset, which might include images with normal, original conditions. Then, the detectors were then evaluated the performance on images with foggy, rainy or other adverse weather conditions. The characteristic of this approach was focusing on the model's architecture and did not affect the images. On the other hand, condition-based approaches tended to propose the specific detectors in concrete contexts [22, 23]. Therefore, these detectors include processing modules such as rainy and foggy removal to transform condition-adverse images into normal images. Then the transformed images were then used for training and testing.

A. Domain Adaptation In Adverse Weather

Chen *et al.* [19] tackled the adaptive domain problem by focusing on two levels: the image-level shift and the instance-level shift. The image-level shift could be described as style, illumination, etc., while the instance-level shift was object appearance and size. Based on Faster R-CNN, the authors designed H-divergence theory-based adaption module on image-level shift and instance-level shift to decrease domain discrepancy. In detail, a domain classifier was built and trained in an adversarial training manner at each level. Those two classifiers then incorporated a consistency regularizer to train a domain-invariant Regional Proposal Network in Faster R-CNN, which then could be adapted with other domains of images. The authors used the Cityscapes dataset as one of the experimental datasets. They trained on the normal version of Cityscapes and evaluated on its foggy version. Experiments proved that the performance increased by combining the proposed adaption module.

Khodabandeh *et al.* [20] proposed a robust Faster R-CNN and the Noisy Labeling strategy for domain adaptation. In detail, the authors split the pipeline into three phases. In the first phase, the Faster R-CNN was trained on the source dataset then used as the base detector. Next, the authors used the base detector to obtain noisy bounding boxes. At the same time, all ground-truth instances in the source dataset were also extracted. After that, noisy bounding boxes and all extracted ground-truth boxes were fit into a classification module. The classification module would be trained on the exact ground-truth boxes from the source dataset and refine the class categories of the noisy boxes. Finally, the upgrade version of Faster R-CNN was trained on both datasets: the source dataset with human-annotated ground-truth boxes and the target dataset with annotations refined via a classification module. The proposed approach was evaluated by training on the Cityscapes and evaluating on Foggy Cityscapes dataset. Through experiments, the authors proved that using the Noisy Labeling strategy and their improved Faster R-CNN could help achieve better results than normal training, original Faster R-CNN, and other approaches 36.45% AP. Notably, this result was 7.07% lower than the result obtained via training Faster R-CNN on the Foggy Cityscapes Dataset.

Sindagi *et al.* [21] defined a novel prior-adversarial loss function that utilized the additional knowledge from images in foggy, rainy images to correlate the amount of degradation directly. In detail, the proposed loss was used to train a prior estimation network to predict condition-specific prior from features map and minimize the weather information present in the features. This approach helped the main detection network features become invariant, decreasing the effects of adverse weather such as foggy or rainy. Furthermore, to avoid distortions caused by weather-based degradation, the authors proposed a set of residual feature recovery blocks in the detection network for de-distorting features leading to better performance. The proposed method was evaluated by training Cityscapes and evaluating on Foggy-Cityscapes dataset, the highest result was 39.3%, which extremely outperformed the previous approaches.

B. Condition-based Object Detection Methods

To handle the problem of vehicles in foggy images that are difficult to recognize, Huang *et al.* [22] proposed a feature recovery module, which was considered as a restoration subnet and integrated with the main backbone of the detection model. The feature recovery (FR) module was designed by sharing feature extraction layers with the backbone. So, this proposed module learned to restore the clean image from the foggy image via the MSE loss in the training time. Notably, the aim was not fitting the restored image to the detection model. They trained the external FR module to help improve the weights in the main backbone, which helped enhance the quality of the features extracted from foggy images. The proposed approach was evaluated on the FOD and Foggy Driving dataset, proving the effectiveness of other detection methods.

Sen *et al.* [23] proposed an encoder-decoder U-shaped network with residual connection from one layer to another for fog removal. Then, the authors employed the PP-YOLO [24] detection model for training object detection. The authors also created a dataset of foggy images by synthesizing them based on the existing dataset. Their ablation studies proved that the detection performance increased when including the proposed fog removal module.

Tran *et al.* [12] proposed a synthesized foggy dataset named UIT-DroneFog based on a normal-weather dataset. Then, they experimented with the existing detectors on the new dataset to observe the performance. In their study, no fog removal techniques were recommended or used, but the authors proposed the combination of Double Heads and Cascade R-CNN, which achieved better results compared to the other methods.

C. Discussion

The adaptive domain approaches encourage object detectors to operate well on cross-domain datasets, which is suitable if we can not collect the images belonging to the domain we expect. Furthermore, adaptive object detectors also help us not to retrain the model for the new domain dataset, which is time-consuming. However, adaptive detectors still have limitations because they can not perform well as their counterparts trained and evaluated on the same domain data. Therefore, we suppose that if we have the images belonging to the domain data we need, it is unnecessary to employ adaptive detectors. So, in this study, we focus on exploring the performance of normal two-stage, one-stage and end-to-end object detection methods combined with de-hazing methods on the foggy dataset.

III. Experimental Methods

A. Object Detectors

Object Detection is a complex, challenging problem in computer vision used to localize and classify objects based on images and videos. With the surging growth of Deep Learning in recent years, feature extraction from data is straightforward to implement and time-efficient, leading to the emergence of intensive studies of Object Detection. In addition to improvements in classification, researchers conduct many in-depth studies in localization to reduce computational costs and memory to make the model work more efficiently. Therefore,

the research on the following detection frameworks happens to meet nowadays's needs: Two-stage, One-stage, Anchor-free and End-to-end object detection.

1) Two-stage Detectors

The framework is also known as the region-based framework. Region proposals generated from input images pass through CNN where features are extracted. Then, based on the extracted features, category-specific classifiers are used to classify labels for the region proposals. There are many well-established two-stage methods that we are going to carry out the experiment on, like Cascade RCNN and Casdou.

Cascade R-CNN. In 2018, Cai and Vasconcelos proposed the high-quality detection model Cascade R-CNN. Multi-stage object detection architecture with set detectors trained in turn with the current detector's output as the input to the next detector used to solve not only the mismatch in quality between the output and the detector but also the overfitting problem caused by the sensitive IoU threshold(when the IoU is large). However, creating a high-quality detector is not simply increasing the IoU during the training phase. As we increase the IoU threshold, it also means that a significant decrease is witnessed in the number of active training samples. Different heads in the architecture designed for a particular IoU threshold, from small to large, are used at different stages (H1, H2, H3). Cascade regression is a resampling process, providing positive samples for further processing stages:

$$f(x, b) = f_T \circ f_{T-1} \circ ... \circ f_1(x, b)$$

T: total number of refining bounding box stages. Each f_T regressor in the cascade optimized for the respective distribution b_T. Figure 1 illustrates the architecture of Cascade R-CNN.

CasDou (Cascade R-CNN and DoubleHead). In the year 2021, Tran *et al.* proposed Casdou with the combination of Cascade R-CNN, Double Heads [25], and Focal Loss [3]. The method tested on the high-quality aerial foggy outdoor vehicle dataset UIT-DroneFog achieved 34.70% on the mAP score. The author used the Cascade R-CNN backbone instead of the Faster R-CNN because it helps the model attain high-quality detection with structural cascade regression. What's more, the Double Heads detector can be flexibly attached to various models to achieve higher detection results. In addition, based on their analysis and assumption, the author uses Focal Loss to help the model converge and have a more apparent distinction between class objects than Cross-Entropy Loss. Focal Loss is defined as follows:

$$L_{FL}(p_T) = -\alpha(1 - p_T)^{\gamma} log(p_T)$$

With (α) is the balanced form of the Focal Loss function, and (γ) is used to calculate the modulating factor.

Figure 2 illustrates the architecture of CasDou.

2) One-stage Detector

Although achieving high accuracy, the two-stage methods are computationally expensive and have high resources-consumption. One-stage method - the YOLO was born for incredible processing speed while maintaining high accuracy to achieve real-time object detection. One-stage architecture directly predicts class probabilities and regress bounding-box offset values with a single feed-forward CNN network instead of heavily depending on generated region proposals.

YOLO-v3. (You Only Look Once v3) YOLO-v3 was proposed by Redmon and Farhadi with the help to improve the accuracy of the object detection problem while keeping the interference time at the appropriate speed. YOLOv3 uses Darknet-53 (ImageNet [26] Trained Network) and Residual Network (ResNet)[27] consisting of 53 convolutional layers built with consecutive convolutional 3x3, and 1x1 layers, followed by ResNet connection skips to activate propagating through deeper layers without diminishing the gradient. Finally, the average pooling layer, 1000 fully connected layers, and Softmax activation function are added to perform classification. With this robust structure, DarkNet-53 has a much higher speed than other platforms like ResNet-101 or ResNet-152. Initially, the image is passed through a block of convolutional layers to extract features. Then, it is divided into a grid of size S × S. When the image is divided into a grid, each cell in the image is responsible for detecting the object whose center locates on that cell. After selecting the anchor box, the model uses the Direct Position Prediction formula to regress the size of the true bounding box. The model then predicts the bounding box's label using Multiple Classification. The illustration of the YOLOv3 method is shown in Figure 3.

3) Anchor-free Detector

Anchor-free detectors find the object without the preset anchors. It helps the model become less dependent on anchor-related hyperparameters and generalize more easily. CrossDet stands out to be the effective anchor-free detector that considers continuous object information and reduces noise interference.

CrossDet. CrossDet was proposed by Qiu *et al.* in 2021. Instead of using anchor-based methods and point-based methods that are inclined to produce noise feature output, the author proposed the CrossDet helping extract the information continuously and accurately. CrossDet-an anchor-free detector using a set of cross lines to represent objects consists of two main phases: (1) Generating the coarse crossline representation (2) Refining the crossline representation based on the extracted features on the horizontal and vertical lines. Based on trained cross lines features, feature maps $I \in R^{C \times W \times H}$ generated from backbone FPN are processed to crossline extract module (CEM). CEM is then trained to extract horizontal and vertical features. The model uses decoupled regression mechanism to optimize cross lines growth along with vertical or horizontal features. CrossDet yields the results on the two data VOC2007 [28] and MS-COCO [29], with the results, are 52.8 and 48.4 respectively on the AP score. Figure 4 shows the architecture of CrossDet method.

4) End-to-end Detectors

The End-to-end approach aims to build a complex deep learning model, removing hand-designed components like pre-processing (anchor generation) and post-processing (Non-maximum suppression) and integrating them into a single model. Today's famous End-to-end methods are Deformable DETR or Sparse R-CNN. They work on sparse candidates which are progressively processed and refined through various stages.

Deformable DETR. In 2021, Zhu *et al.* proposed the

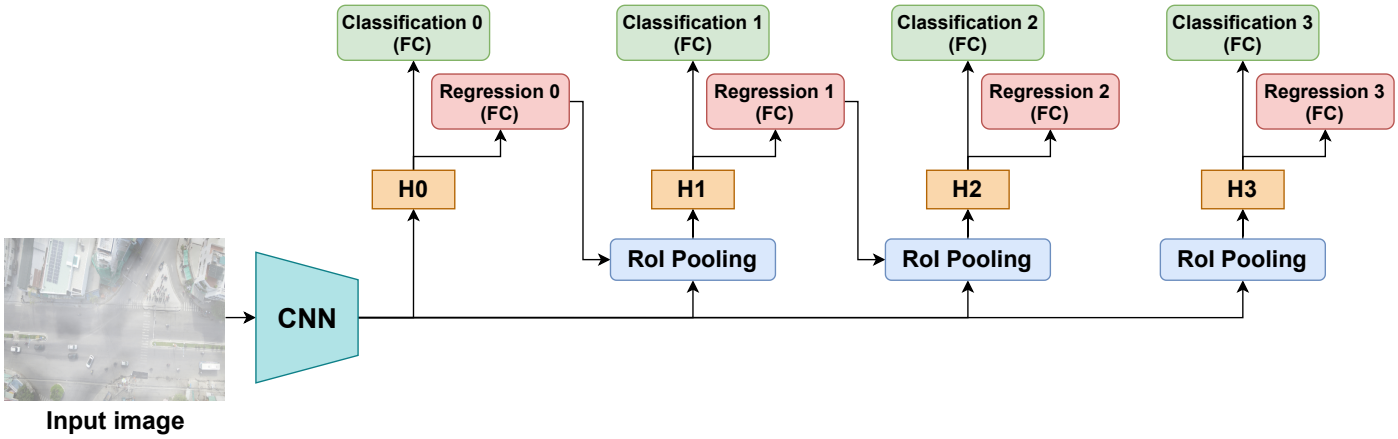


Figure 1: Illustration of two-stage Cascade R-CNN object detection method

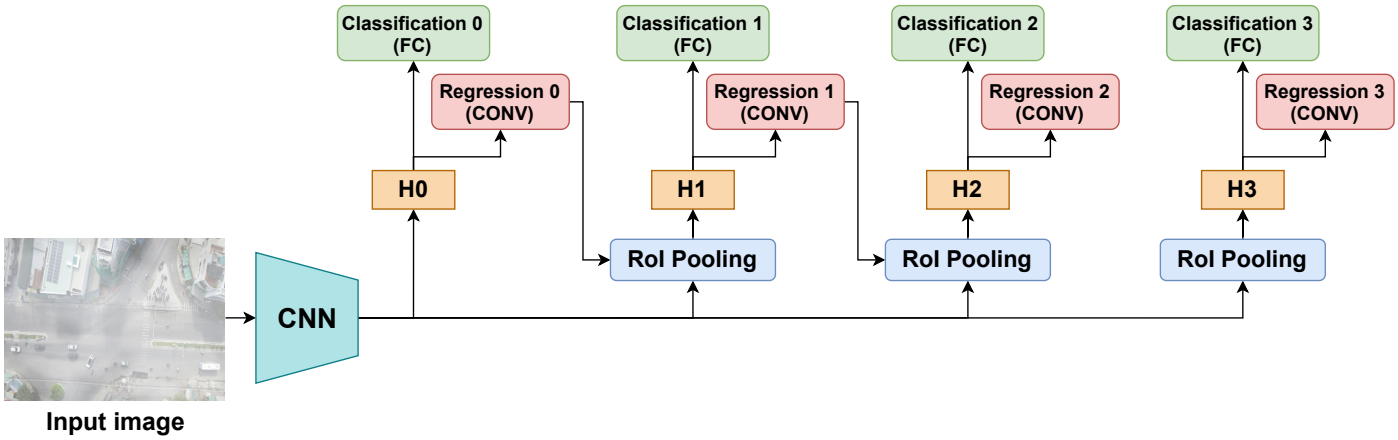


Figure 2: Illustration of two-stage CasDou object detection method

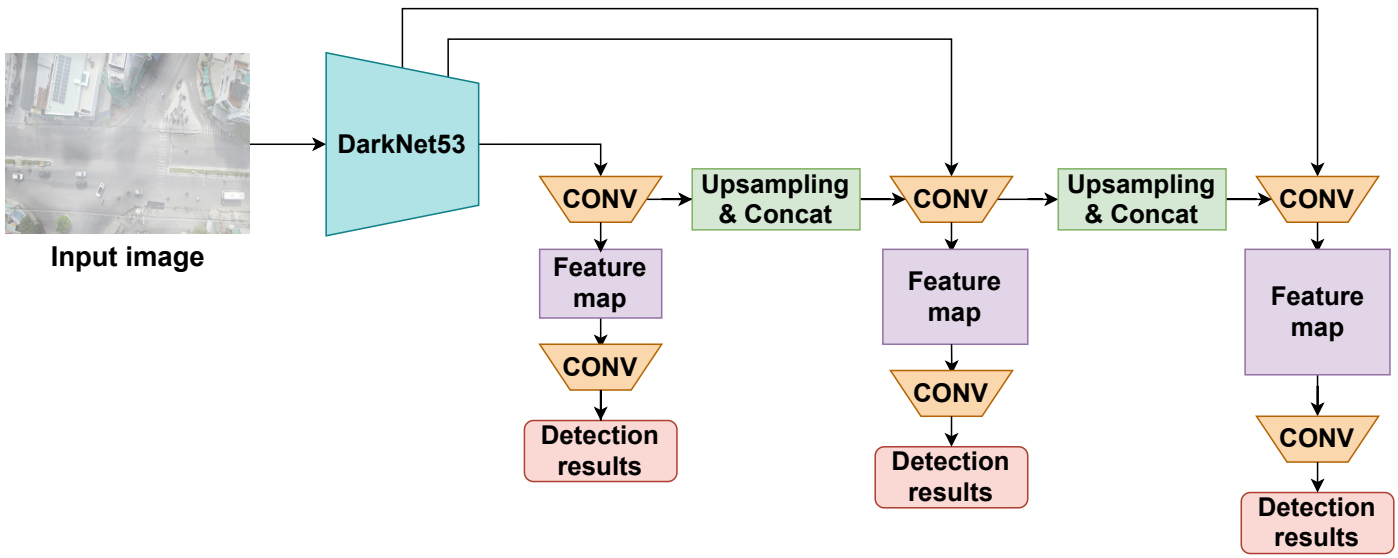


Figure 3: Illustration of one-stage YOLOv3 object detection method

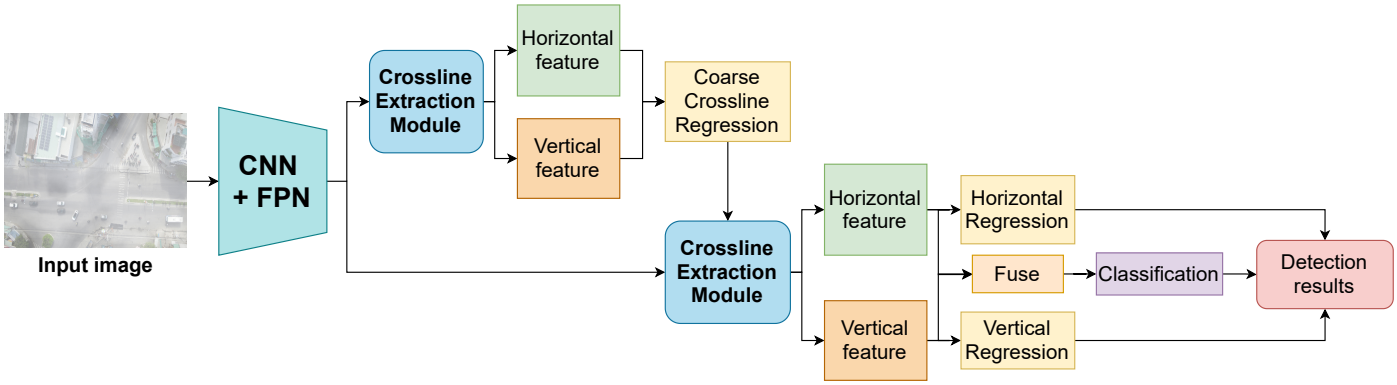


Figure 4: Illustration of anchor-free CrossDet object detection method

Deformable DETR method as an improved version of the prior DETR [30]. However, achieving results as high as Faster R-CNN, DETR witnesses slow convergences and problems in detecting small objects due to the limitation of the attention mechanism in the Transformer. The Deformable DETR is a combination of a Deformable convolution [31] and a Transformer [32]. It combines the effectiveness of the sparse spatial sampling of Deformable convolution and the relation modeling capability of Transformers. From there, it was formed into a Deformable attention module to focus on the part of "sampling spatial" as a pre-filter to focus on prominent areas instead of every location on the feature maps. Moreover, thanks to the fast convergence and flexibility of Deformable DETR, the authors also experimented with some methods to optimize the predict bounding box on the MS COCO dataset such as iterative bounding box refinement mechanism on region proposals proposed by the model. Experiments show that Deformable DETR converges faster and gives more accurate results than DETR with x10 fewer training epochs. Figure 5 shows the architecture of Deformable DETR method.

Sparse R-CNN. In 2021, the Sparse R-CNN method, proposed bySun *et al.*, is a sparse object detection method. Sparse R-CNN is an End-to-End object detection method because it eliminates the post-processing mechanism of non-maximum suppression, which is different from prior R-CNN models. Object detection methods such as the Dense method in the YOLO family where locations of anchor boxes densely cover spatial positions, scales, and aspect ratios in a single-shot way. A typical Dense-to-sparse approach is Faster R-CNN, which uses RPN [33] to derive region proposals from the dense region of candidates and refine those bounding boxes and class-specific features. The sparse method applied by Sparse R-CNN replaces RPN with a set of learnable region proposals and proposal features. Input is an image with a set of region proposals and proposal features that are randomly initialized and optimized with other parameters in the whole network. Features are extracted and fed into the backbone along with region proposals and proposal features, eventually generating outputs classification and localization. Sparse R-CNN shows its accuracy, run-time and convergence in training on the challenging COCO dataset, yielding a 45.0 AP score at 22 fps using the ResNet-50 FPN backbone. Figure 6 shows the architecture of Sparse R-CNN object detection method.

B. Image Dehazing Methods

1) DW GAN (Discrete Wavelet Transform GAN)

DW-GAN was proposed by Fu *et al.* in 2021. This method was designed to tackle two problems that some existing CNN-based dehazing methods have when working with non-homogeneous cases. These problems are the loss of texture details when images are being dehazed due to the complicated haze distribution and over-fitting problems because of the lack of training data. The architecture of this method is a novel two-branch generative adversarial network. For the first branch, called the DWT branch, they proposed the idea of directly embedding the frequency domain knowledge into the dehazing network by utilizing wavelet transform. Therefore more high-frequency knowledge in the feature map can be retained. In terms of the knowledge adaptation branch, the Res2Net was employed with the pre-trained ImageNet weights as initialization in order to prevent overfitting and improve the generalization ability of the network. Finally, they add a basic 7 × 7 convolution layer as a fusion operation to map the features from two branches to clear images. Furthermore, they also introduced the final loss blend function shown in Equation 1. (L_1) is L1 loss, (L_{SSIM}) denotes MS-SSIM [34] loss ,($L_{perceptual}$) represents perceptual loss [35] and, for the adversarial loss (L_{adv}), the discriminator in [36] is employed.

$$\mathcal{L}_{total} = L_1 + \alpha L_{SSIM} + \beta L_{perceptual} + \gamma_4 L_{adv} \quad (1)$$

Where (α) = 0.2, (β) = 0.001 and (γ) = 0.005 are the hyper-parameters weighting for each loss functions.

2) Two-branch Dehazing

Yu *et al.* proposed another two-branch neural network for non-homogeneous dehazing via ensemble learning. The authors found that a carefully built CNN frequently fails on a non-homogeneous dehazing dataset introduced by NITRE challenges [37] even though it performs well on large-scaled dehazing bench-marks. Therefore, they introduced a two-branch neural network to deal with the aforementioned problems separately, followed by a learnable fusion tail to map their different features. The first branch, the transfer learning sub-net, is based on an ImageNet pre-trained Res2Net. This branch extracts robust global representations from input images

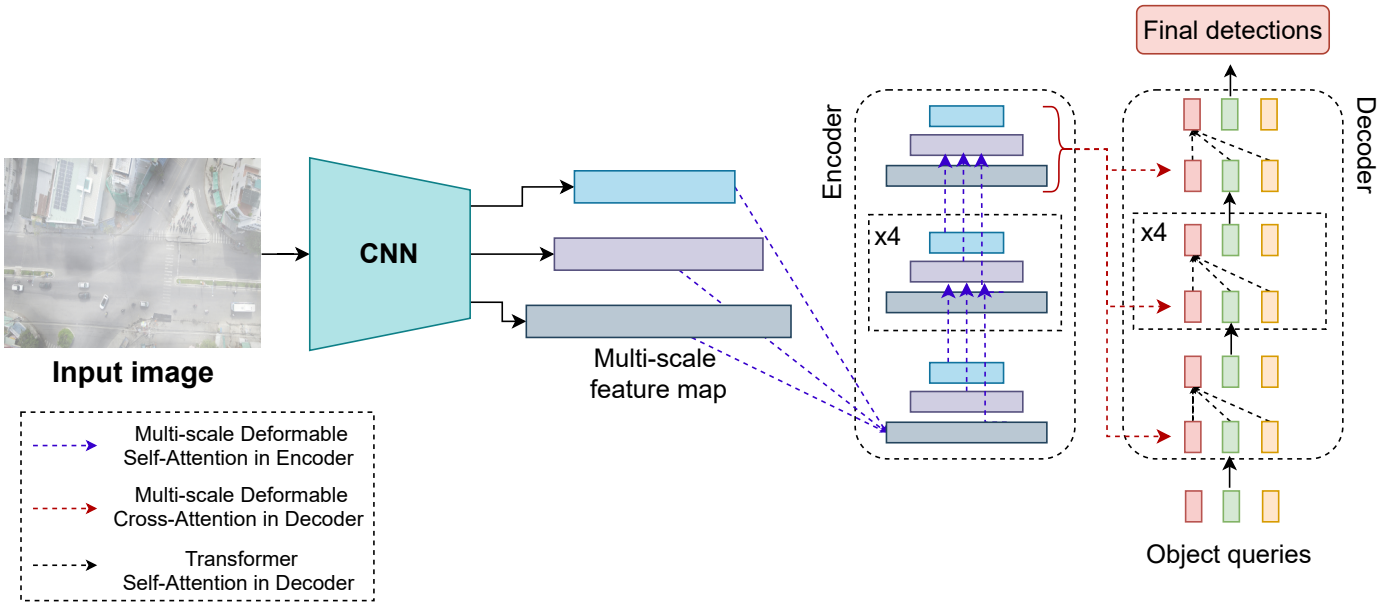


Figure 5: Illustration of end-to-end Deformable DETR object detection method

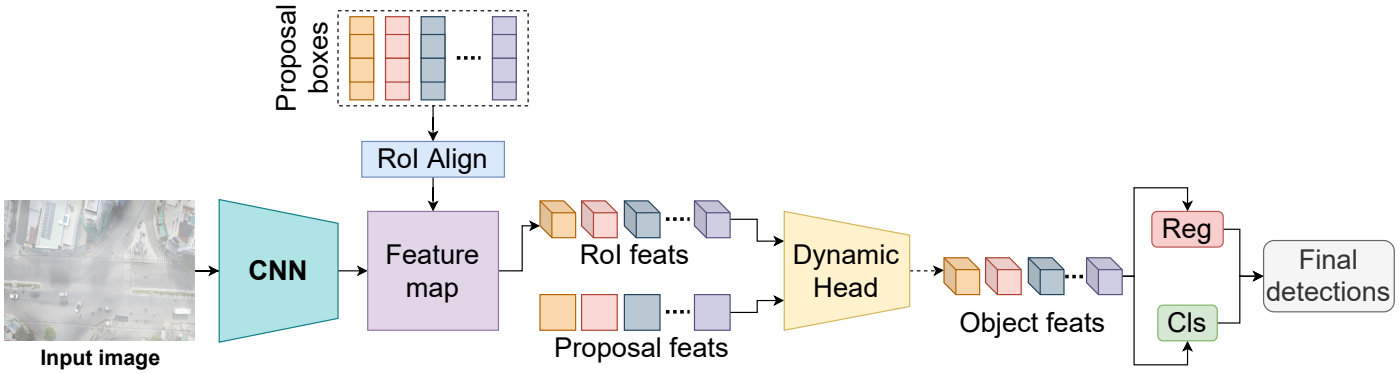


Figure 6: Illustration of end-to-end Sparse R-CNN object detection method

with pre-trained weights and then helps the network address the problem of lacking training data. In the second branch, Yu *et al.* used residual channel attention network to design the current data fitting sub-net. This branch has five residual groups; each has ten residual blocks. However, the second branch always maintains the input image's original resolution and avoids using any downsampling operation. Finally, a fusion layer generates the entire network's final output. The fusion layer, in particular, takes the concatenation of features from the branches and maps them to clear outputs. Besides, they also applied the adversarial loss with the discriminator in [36] because of its effectiveness in helping restore photo-realistic photos [38], especially for a small-scaled dataset.

IV. Experimental Results

A. Benchmark Suite

In this study, the UIT-DroneFog dataset, which was created by Tran *et al.*, was employed to evaluate the performances of detectors in foggy aerial images. This dataset consists of 15,370 foggy aerial images captured by drones with about 0.6 million bounding boxes of various means of transportation and pedestrians. This dataset has four classes: Pedestrian, Motor, Car, and Bus. We also used the default subsets provided by the authors, which include: Training set (8,580 images), Validation set (1,061 images), and Testing set (5,729 images). The numbers of each class are shown in Figure 7.

There are several reasons why we choose this dataset. Firstly, the images in this dataset are of high quality, which helps the detectors work more efficiently. Secondly, the context of these images is diverse and especially, there is an imbalance in this dataset with a vast majority of motor objects. This can be a tough challenge for our detectors.

Example images of this dataset are shown in Figure 8.

B. Experimental Settings

The experimental processes were conducted on a GeForce RTX 2080 Ti GPU with 11018 MiB memory. We trained the models by employing the MMDetection framework

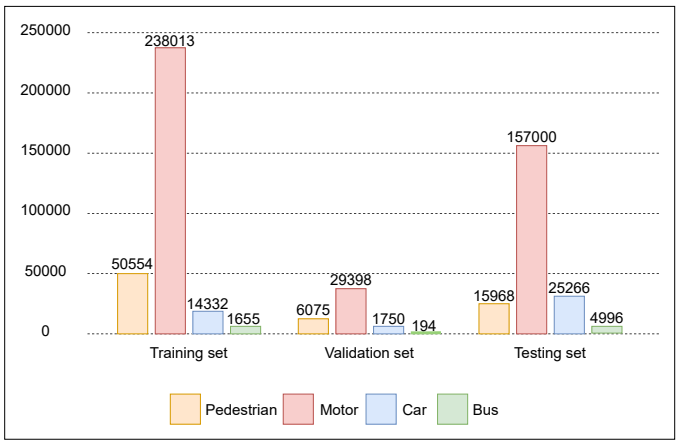


Figure 7: Statistics of UIT-DroneFog dataset.

V2.10.0[39]. For each model, we used the highest mAP score configuration, provided on the MMDetection GitHub website[1] or author's GitHub. We conducted every training process on a single GeForce RTX 2080 Ti GPU.

To evaluate the detectors, we used the best weights of each model on the validation set to predict and report the results on the testing set via the mAP measure to evaluate the performance of models, which is the same as the object detection contest on the MS COCO dataset. The AP score was calculated for 10 IoU varied from 50% to 95% with steps of 5%. Besides, the results of two specified values of 50% and 75% were also reported.

C. Experimental Results

The experimental results are reported in Table I, Table II, Table III. In Table 1, we report the performance of five experimental object detectors on original foggy images. Overall, CasDou - a two-stage method - shows the best performance (34.7% AP) while YOLOv3 performs the worst. These results reflect the characteristic of two-stage and one-stage detectors correctly: two-stage methods often perform better than one-stage methods about accuracy. However, Deformable DETR, an end-to-end detector, has the highest results on Pedestrian and Motor class objects, which are 0.5% and 2.8% higher than CasDou. Moreover, Deformable DETR is exceptionally competitive with CasDou about AP score (33.7% AP compared to 34.7% AP), proving that end-to-end detectors that may have higher FPS can perform as well as two-stage detectors.

In Table 2, we report the results of five object detectors but use DW-GAN to de-haze images before training. In general, there is a variation in the order of accuracy between the methods. Cascade R-CNN, again a two-stage method, becomes the detector that shows the best performance among the experimented detectors, but the AP score is not higher than CasDou trained on original foggy images. However, Cascade R-CNN trained on images de-hazed by DW-GAN has a noticeable improvement than its counterpart trained on synthetic hazy images (+1.3% AP, +2.5% AP@50, +1.9% AP@75). Furthermore, DW-GAN also helps YOLOv3 and Sparse R-CNN enhance the AP score compared to their results in Table 1 (+1.4% AP and +0.3% AP, respectively). Therefore, de-hazing images using DW-GAN significantly and positively affect the performance of five object detectors.

Table 3 reports the results using Two-Branch as the de-hazing method. It can be seen that this de-hazing method can not help object detectors improve the results compared to their counterparts trained on original hazy images. However, the accuracy between the methods is the same as Table 1: CasDou shows the best performance (33.1% AP) and YOLOv3 performs the worst (20.4% AP); CasDou also indicates the best AP on Car and Bus classes (57.4% AP and 40.1% AP) while Deformable DETR shows the best AP on Pedestrian and Motor (2.7% AP and 35.8% AP). Notably, the CrossDet detector using the Two-Branch de-hazing method shows a slight improvement compared to its results in Table 1 (+0.5% AP, +0.3% AP@50, +1.3% AP@75).

Through three experimental results, we can notice that two-stage methods, especially CasDou and Cascade R-CNN, have the ability to perform better than one-stage and end-to-end detectors on both original and dehazed datasets. Besides, from these experiments, it can be seen that resurfacing objects from hazy aerial images by dehazing them is a tough challenge and not always effective due to color deviation and the loss of information compared to haze-free images. In fact, the Two-branch dehazing method significantly reduces the detection results of CasDou's detection result (−1.6% AP), while the DW-GAN is proved to be more effective when helping Cascade R-CNN improve its detection results were improved. This could be explained by the fact that DW-GAN has higher results than Two-branch dehazing when both of these two methods use the same dataset [37]. In addition, although the AP result is not so high, the DETR shows that this method can outperform all other methods by a large margin when detecting small objects, which are Pedestrian and Motor, on all three datasets (shown in Figure 9. This means that although the small objects are blurred by fog, this method learns the features of this kind of object better. On the other hand, when detecting objects in big sizes such as Bus and Car - many times bigger than Motor and Pedestrian, a two-stage method will be an appropriate choice because the AP scores of Car and Bus detected by two-stage methods in three report tables always ranks top. Figure 10 and Figure 11 show the detection results of CasDou and Cascade R-CNN methods with and without using de-hazing techniques.

V. Conclusion

This study has provided experimental results and a thorough analysis of condition-based approaches to the problem of vehicle detection on aerial images in foggy weather. In short, we evaluate the effectiveness of advanced object detectors on aerial images with and without foggy removal. DW-GAN and Two-Branch are used for de-hazing. CasDou achieves the highest performance among experiments when the AP is recorded at 34.7% on original images. Experimental results also show that detectors trained on de-hazed methods can not achieve the best results. Still, with some detectors such as YOLOv3, CrossDet, Sparse R-CNN, foggy removal help them slightly improve the detection performance compared to their counterparts trained on original images. As we reported,

[1]https://github.com/open-mmlab/mmdetection

Figure 8: Example images of UIT-DroneFog dataset.

Table I: Experimental results with the default configuration. The best performance is marked in boldface. (not dehazed)

Method	Pedestrian	Motor	Car	Bus	AP	AP@50	AP@75
Cascade R-CNN	2.10	34.50	56.80	38.40	32.90	45.80	38.50
CasDou	2.70	34.20	**59.30**	**42.50**	**34.70**	**50.20**	**40.30**
CrossDet	1.60	27.30	51.10	31.80	27.90	45.60	30.30
YOLOv3	1.10	21.50	41.10	15.80	19.90	32.10	21.10
Deformable DETR	**3.20**	**37.00**	56.30	38.30	33.70	51.60	38.00
Sparse R-CNN	2.70	23.80	30.80	28.80	21.50	32.90	23.20

Table II: Experimental results on UIT-Drone21 dehazed by DWGAN. The best performance is marked in boldface.

Method	Pedestrian	Motor	Car	Bus	AP	AP@50	AP@75
Cascade R-CNN	2.20	32.90	**58.80**	**42.80**	**34.20**	48.30	**40.40**
CasDou	**2.30**	32.70	58.30	39.60	33.20	47.80	38.10
CrossDet	1.30	27.20	50.60	29.40	27.10	44.00	29.70
YOLOv3	1.20	21.10	41.50	21.30	21.30	33.30	23.60
Deformable DETR	2.10	**35.10**	55.40	36.20	32.20	**48.60**	37.30
Sparse R-CNN	**2.30**	22.70	33.90	28.30	21.80	32.50	24.20

the approaches using a GAN-based model to remove foggy images somehow can not achieve the results of using original images. Therefore, in the future, we plan to conduct more experiments on other recent GAN-based methods to explore their effectiveness in de-hazing. Besides, we also plan to propose a new approach that can adaptively predict foggy images using a model trained on original images.

ACKNOWLEDGEMENT

This research is funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number DS2021-26-01. We also would like to show our gratitude to the UIT-Together research group for sharing their pearls of wisdom with us during this research.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

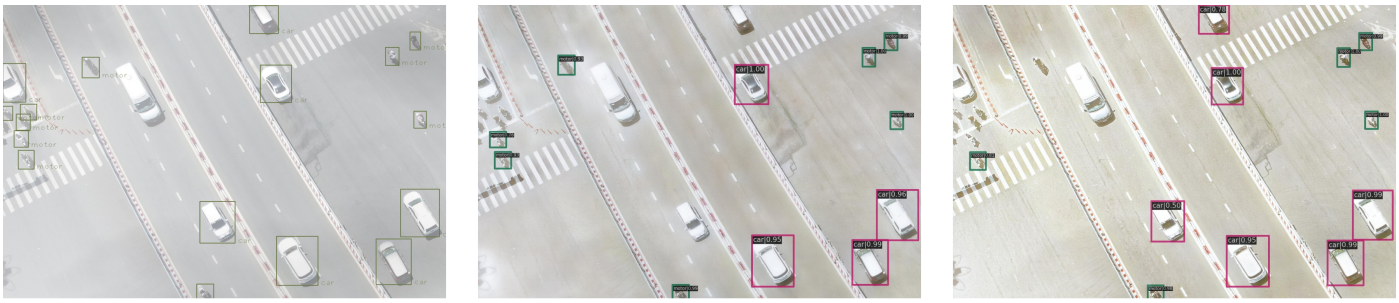
[2] M. Y. Yang, W. Liao, X. Li, Y. Cao, and B. Rosenhahn, "Vehicle detection in aerial images," *Photogrammetric Engineering & Remote Sensing*, vol. 85, no. 4, pp. 297–304, 2019.

Table III: Experimental results dehazed by Two-Branch. The best performance is marked in boldface.

Method	Pedestrian	Motor	Car	Bus	AP	AP@50	AP@75
Cascade R-CNN	1.90	32.40	57.30	38.60	32.60	45.50	38.40
CasDou	1.90	33.00	**57.40**	**40.10**	**33.10**	46.60	**39.10**
CrossDet	2.00	25.60	51.00	33.50	28.00	44.40	31.00
YOLOv3	8.00	20.20	40.30	20.20	20.40	32.40	22.40
Deformable DETR	**2.70**	**35.80**	54.70	32.80	31.50	**48.40**	35.30
Sparse R-CNN	2.40	23.20	32.20	27.90	21.40	32.70	23.40

Figure 9: Visualization images of detectors on UIT-DroneFog dataset. (In order to see the image clearly, please zoom in 2×)

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.



Original DW-GAN Two-branch Dehazing

Figure 10: CasDou on 3 types of dataset. The dark green bounding boxes are Motor, the purple bounding boxes are Car, light green bounding boxes are Pedestrian and Bus are cyan. (In order to see the image clearly, please zoom in 2×)

Original DW-GAN Two-branch Dehazing

Figure 11: Cascade R-CNN on 3 types of dataset . (In order to see the image clearly, please zoom in 2×)

[4] J. Lu *et al.*, "A vehicle detection method for aerial image based on yolo," *Journal of Computer and Communications*, vol. 6, no. 11, pp. 98–107, 2018.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[6] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *European conference on computer vision*, Springer, 2016, pp. 785–800.

[7] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2016.

[8] G.-S. Xia *et al.*, "Dota: A large-scale dataset for object detection in aerial images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3974–3983.

[9] S.-C. Huang, Q.-V. Hoang, and T.-H. Le, "Sfa-net: A selective features absorption network for object detection in rainy weather conditions," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[10] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018.

[11] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[12] M. T. Tran, B. V. Tran, N. D. Vo, and K. Nguyen, "Uit-dronefog: Toward high-performance object detection via high-quality aerial foggy dataset," in *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, IEEE, 2021, pp. 290–295.

[13] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[14] H. Qiu *et al.*, "Crossdet: Crossline representation for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3195–3204.

[15] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[16] P. Sun *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 454–14 463.

[17] M. Fu, H. Liu, Y. Yu, J. Chen, and K. Wang, "Dw-gan: A discrete wavelet transform gan for nonhomogeneous dehazing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 203–212.

[18] Y. Yu, H. Liu, M. Fu, J. Chen, X. Wang, and K. Wang, "A two-branch neural network for non-homogeneous dehazing via ensemble learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 193–202.

[19] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3339–3348.

[20] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready, "A robust learning approach to domain adaptive object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 480–490.

[21] V. A. Sindagi, P. Oza, R. Yasarla, and V. M. Patel, "Prior-based domain adaptive object detection for hazy and rainy conditions," in *European Conference on Computer Vision*, Springer, 2020, pp. 763–780.

[22] S.-C. Huang, T.-H. Le, and D.-W. Jaw, "Dsnet: Joint semantic learning for object detection in inclement weather conditions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2623–2633, 2020.

[23] P. Sen, A. Das, and N. Sahu, "Object detection in foggy weather conditions," in *International Conference on Intelligent Computing & Optimization*, Springer, 2021, pp. 728–737.

[24] X. Long *et al.*, "Pp-yolo: An effective and efficient implementation of object detector," *arXiv preprint arXiv:2007.12099*, 2020.

[25] Y. Wu *et al.*, "Rethinking classification and localization for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 186–10 195.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[28] D. Hoiem, S. K. Divvala, and J. H. Hays, "Pascal voc 2008 challenge," *World Literature Today*, vol. 24, 2009.

[29] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[31] J. Dai *et al.*, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[32] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[33] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2016. arXiv: 1506.01497 [cs.CV].

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[35] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, Springer, 2016, pp. 694–711.

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[37] C. O. Ancuti, C. Ancuti, F.-A. Vasluianu, and R. Timofte, "Ntire 2021 nonhomogeneous dehazing challenge report," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 627–646.

[38] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[39] K. Chen *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.