Learning-based Image Conversion of Driving Scenes under Different Weather Conditions

Hanting Yang

Abstract

The safety performance of autonomous vehicles is closely related to their behavior under different weather conditions. In clear weather, sensors like cameras and radars can usually provide more accurate data. This enables the vehicle to better recognize traffic signs, obstacles, and other vehicles, thereby reducing the risk of accidents. Camera sensors are one of the dominant modalities in autonomous vehicles due to their high resolution, fast data rates, and direct correlation with human perception, which is crucial for understanding and navigating the environment. In contrast, rain, snow, and other adverse weather conditions will have a negative impact on data quality, such as low image contrast, color distortion, and reduced visibility.

This kind of data quality change due to weather variations can be considered a domain-shift problem, which refers to a situation where the distribution of target data is inconsistent with the distribution of source data. Domain adaptation techniques aim to reduce the domain gaps by applying knowledge from the target domain, thus improving the generalization ability of the pre-trained model.

However, there are several key issues that still need to be addressed in the context of autonomous driving. First, there is a lack of paired clear-weather data due to the difficulty of collecting data under different weather conditions while maintaining the same content. Secondly, the challenge of generating high-quality, realistic adverse weather data suitable for autonomous driving algorithms remains. Lastly, there is the problem of improving the

diversity and controllability of model outputs based on single data input to ensure model efficiency.

This thesis focuses on RGB images and defines the domain adaptation problem for autonomous driving in adverse weather conditions as an image conversion problem within the context of driving scenes. Furthermore, this thesis proposes corresponding solutions to address the aforementioned three challenges.

Firstly, to address the problem of unpaired image conversion, Chapter 3 proposes a framework based on Generative Adversarial Networks (GANs) and introduces the principle of cycle consistency principle. The idea is inspired by the concept of dual learning, which requires that an image converted into the target domain can be converted back to the input image, thereby preserving the texture of objects, roads, and buildings in the scene while changing the weather conditions. To ensure consistent weather effects in the conversion process, the chapter further introduces a weather layer to extract information during the forward conversion and input it into the reconstruction network through feature fusion. Additionally, apart from image conversion under two weather conditions, this chapter extends the original model to enable unpaired image conversion under multiple weather conditions.

Secondly, to generate more realistic driving scene images, Chapter 4 proposes an image conversion method that leverages semantic information as an additional input. The first step is to obtain the semantic segmentation map of the input image. Then, the second step is to crop image patches at the corresponding locations. The RGB image patches are input into the image conversion network, while the semantic segmentation image patches are input into the conversion network through a feature fusion layer. Using this approach, the proposed method achieves the generation of realistic snowy weather conditions.

Thirdly, to break the limitation of single projections in current image conversion methods, Chapter 5 proposes a framework that can provide multiple solutions and control the

direction of generation. The framework is based on a GAN which incorporates both a style encoder and a content encoder, specifically designed to extract relevant information from an image. The framework further employs a decoder to reconstruct an image using these encoded features, while ensuring that the generated output remains within a permissible range by applying a self-regression module to constrain the style latent space. By modifying the hyperparameters, the generator can generate controllable outputs with specific style codes.

The proposed methods are tested on public driving datasets and a self-captured dataset called Realistic Driving Scene under Bad Weather (RDSBW) dataset. The tests employ both traditional and novel image quality metrics, as well as metrics in autonomous driving perception tasks, to validate the effectiveness of the proposed methods. The results show that the proposed methods can achieve high-quality image conversion without pairing training data, and also that the degree of converted weather image can be controlled.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Domain Shift and Domain Varieties in Autonomous Vehicles

Autonomous Vehicles (AVs), also known as self-driving cars or driverless vehicles, are equipped with advanced sensors, cameras, and computing capabilities that enable them to navigate and operate without human intervention. Domain shift refers to the changes in environmental and operational conditions under which the AVs must function effectively. These changes can significantly impact the performance of an AV's perception, decision-making, and control systems. Domain varieties on the other hand represent the different types of environments or scenarios an AV might encounter.

1.1.1 Weather Domain

As the driving environment transitions, the distribution of the data collected by the sensors of AVs also changes, which is known as domain shift. This shift is greatly influenced by changes in weather conditions. Figure 1.1 lists common weather conditions a driving car may encounter.

During rain, the roads take on a visibly wet appearance, with a glossy sheen indicating surface wetness [5]. Individual raindrops create small, visible splashes, especially notable in puddles. This precipitation can lead to the accumulation of standing water, potentially

Figure 1.1: Driving scene images in different weather conditions.

causing puddles or flooding in lower areas. Rain also reduces visibility, creating a veil-like effect that obscures distant objects and landscapes. As vehicles move through wet roads, they displace water, generating splashes and spray that can further diminish visibility for nearby drivers.

In snowing conditions, the landscape is transformed into a white blanket, with snowflakes steadily falling and accumulating on the road surface, covering road markings and altering the road's texture. The snow creates slick surfaces, particularly when compacted by the passage of vehicles [6]. Heavy snowfall significantly reduces visibility, leading to a whiteout effect. Additionally, wind-driven snow can form drifts that alter the landscape and sometimes obstruct parts of the road.

Fog presents a unique driving environment, where a dense, misty veil blankets the surroundings, drastically reducing visibility [7]. Objects appear as hazy blocks or shapes, with their details obscured by the fog. Roads often become damp under these conditions, reflect-

ing light and adding to the visual complexity. Headlights and streetlights in foggy conditions become diffused, spreading light in a less focused manner, and vehicles or objects often emerge silently and suddenly from the fog at close range.

Overcast weather casts a uniform, diffused light over the driving scene, resulting in minimal shadows and a flattening of visual depth [8]. Colors appear more muted under the soft, even light, and any present shadow is softer and less defined. The low contrast environment under an overcast sky makes details appear less sharp, and there can be a mild glare, particularly when the cloud cover is not dense.

Under sunny conditions, the driving scene is characterized by intense reflections, particularly off wet or shiny surfaces, and sharply defined shadows that create a high-contrast environment [8]. The direct sunlight can cause significant glare, especially when reflecting off other vehicles, glass buildings, or wet roads. This intensity of light may lead to a squinting effect, temporarily impacting visibility. Sunlight also enhances the vividness of colors, making the environment appear more vibrant and dynamic.

1.1.2 Other Domain

Other domains with respect to geographical locations are shown in Fig. 1.2. Each of these environments can be considered as a distinct domain.

In urban areas, the driving scenes are characterized by complex road layouts, traffic signals, and a high density of road users, including pedestrians and cyclists [1, 9, 10]. The road networks in urban areas feature frequent intersections, one-way streets, and roundabouts. In districts of downtown and business centers, the main streets are packed with a variety of traffic signals, stop signs, and pedestrian crossings. Additionally, urban areas often involve scenarios such as parallel parking, navigating through tight spaces, and stopping for deliveries. Furthermore, regular construction activities and road maintenance can lead to

Figure 1.2: Driving scene images in different areas.

unexpected detours and obstacles.

The scene on highways is characterized by high-speed travel and streamlined traffic flow, primarily in straight lines with occasional gentle curves [11]. These roads typically have multiple lanes, promoting disciplined lane usage and minimal direct interaction between vehicles. Unlike urban areas, they do not have traffic signals but instead rely on signs for exits, distances, and services. Since they are designed for efficient long-distance travel, they have fewer stops and minimal presence of pedestrians or cyclists. Other features of highways include rest areas, toll booths, and emergency lanes. The surrounding environment often transitions from open rural landscapes to more urban settings as highways approach major cities, leading to changes in scenery and traffic patterns.

Rural driving scenes exhibit a combination of open landscapes, agricultural fields, and residential areas [12]. The roads in these areas vary greatly, ranging from well-paved main roads to unmarked or gravel paths. The presence of street lighting is often minimal, par-

ticularly in remote rural areas, resulting in darker roads at night. Rural roads may meander through hills, forests, and alongside rivers, offering diverse terrains and picturesque views. Additionally, the presence of wildlife or livestock crossing the road is a common occurrence in rural areas, introducing an element of unpredictability to driving.

Suburban areas blend the tranquility of residential neighborhoods with some characteristics of urban traffic, albeit on a smaller scale. In these areas, one may encounter school zones, local traffic intersections, and community centers. The overall pace is slower, with lower traffic density compared to urban settings, providing a more relaxed driving experience.

1.1.3 Differences between Weather Domains and Other Domains

Weather domains, including conditions like rain, snow, fog, overcast, and strong sunlight, create unique visual challenges that are inherently dynamic and often unpredictable. Unlike static urban or rural environments with consistent features like buildings or roads, weather conditions can drastically alter the appearance of these same features [13]. For instance, rain can add a reflective sheen to surfaces and obscure details, while fog significantly reduces visibility and contrast. These transient and variable conditions require adaptive and robust computer vision algorithms capable of interpreting a wide range of visual cues.

The focus on weather domains in this thesis is motivated by several compelling factors. Firstly, the dynamic complexity inherent in weather conditions is introduced. These conditions display a level of dynamic complexity not typically encountered in other domains. The rapid and unpredictable nature of weather changes presents a significant challenge for computer vision systems, necessitating quick adaptation to new visual environments. Secondly, the aspect of safety and reliability is considered paramount, particularly in applications such as autonomous driving and outdoor surveillance. The capability to accurately interpret visual

information under various weather conditions is critical for ensuring safety and reliability, making the mastery of weather-related visual challenges a pressing necessity. Thirdly, the existing technological gap in current computer vision systems is acknowledged, particularly in terms of performance under adverse weather conditions. This thesis is aimed at bridging this gap by developing algorithms specifically designed to manage the complexities presented by weather domains. Finally, the real-world applicability of these developments is underscored. Weather conditions, being a fundamental aspect of daily life and impacting sectors ranging from transportation to agriculture, present a field where enhancements in computer vision capabilities can have widespread and practical implications. This highlights the importance of advancements in this area.

In contrast, other domains, such as urban or rural landscapes, present more stable and predictable visual environments [1, 10, 12]. These domains are characterized by relatively fixed elements such as buildings, roads, trees, and consistent lighting conditions, except for natural changes between day and night. The challenge in these domains lies in handling the complexity of scenes, such as diverse architectural styles or natural landscapes. However, the lack of rapid, unpredictable change as seen in weather domains means that the algorithms developed for urban or rural areas can rely on more stable and consistent visual features.

The interplay between weather and other domains further highlights the complexity of visual interpretation in varying conditions [2, 3, 8]. For example, the transition from a clear, sunny day to a foggy evening involves a significant shift in visual perception. The algorithms must be able to recognize and adjust to these changes swiftly, understanding the same scene under different visual conditions. This requires not just advanced image processing techniques but also sophisticated machine learning models that can learn from diverse datasets encompassing various weather and environmental conditions. Therefore, the study of differences between weather domains and other domains in computer vision is crucial for devel-

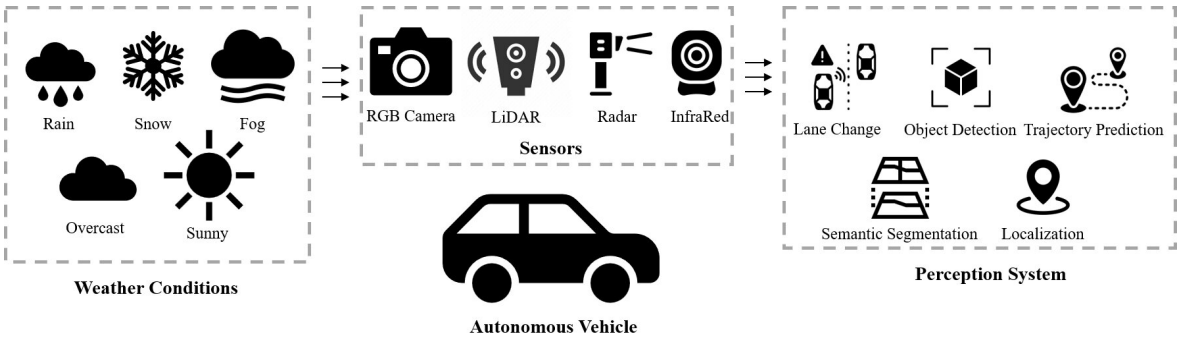


Figure 1.3: Relationship between weather conditions, sensors, and perception systems in autonomous vehicles. Perception systems play a critical role in identifying and interpreting the surrounding environment.

oping systems that can accurately interpret and interact with the world in a wide range of conditions.

1.2 Perception Systems under Adverse Weather Domains

Perception systems in modern technology, particularly in areas like robotics and computer vision, rely heavily on a variety of sensors to interpret and interact with their environment. These systems generally utilize RGB cameras, LiDAR, radar, and InfraRed (IR) sensors, each serving a distinct purpose. Figure 1.3 illustrates the relationship between weather conditions, sensors, and perception systems in AVs.

Camera sensors, with their high resolution and rapid data rates, are a key technology in AVs, closely mirroring human perception and playing a vital role in environmental interpretation. They capture visual information, providing detailed images of the surroundings. They are crucial for recognizing visual elements such as objects, signs, and signals. Computer vision techniques process these images to identify and classify various elements within the environment [9, 14].

LiDAR sensors emit light pulses and measure the time it takes for the light to reflect back. These data help create a three-dimensional map of the environment, useful for determining the shape and distance of nearby objects [15].

Radar sensors emit radio waves and measure their reflection off objects. This technology is especially effective in determining the distance and speed of objects, often used in scenarios where robust performance is needed in various weather conditions [16].

Thermal cameras are IR sensors that detect heat and are used to identify and measure the thermal signature of different objects. This is particularly useful in conditions where visual clarity is low, such as in fog or darkness [17].

Each sensor type contributes unique data, helping to build a comprehensive understanding of the surroundings. The integration and interpretation of data from these diverse sensors are crucial for accurate environmental perception, especially in applications that require high levels of autonomy and decision-making capabilities.

However, adverse weather conditions introduce significant challenges to the perception systems, particularly affecting the data captured by various sensors. Each sensor type encounters unique distortions and challenges under these conditions.

Cameras are essential in autonomous vehicles for object detection and tracking, offering advantages over other sensors like LiDAR, radar, and IR [18]. They provide high-resolution images and can capture color information, which is crucial for identifying and tracking various objects. While LiDAR is good for measuring distances and radar can penetrate fog or rain, neither can match the detailed visual data cameras offer. IR sensors help in low light conditions but do not provide the color or clarity of cameras during the day. This makes cameras a key tool for accurately detecting and tracking objects in many driving scenarios.

1.2.1 Characteristic of Data Captured under Adverse Weather Conditions

Cameras are highly susceptible to various weather-induced distortions [18]. Heavy rain and fog reduce visibility and contrast in images. This makes it difficult to distinguish objects from the background, as edges and contours become blurred. Varying lighting conditions can cause colors to appear washed out or overly saturated. For instance, low lighting in overcast conditions may lead to underexposed images, while intense sunlight can cause overexposure. Wet surfaces and direct sunlight create glare and reflections, which can lead to overexposed areas in the image where details are lost. High-speed rain or snow can cause motion blur in images, making it challenging to identify static and moving objects accurately.

LiDAR systems, although less affected by lighting conditions, face their own set of challenges [19]. Raindrops or snowflakes can scatter the LiDAR beams, resulting in noise within the point cloud. This manifests as false positives or ghost objects that are not actually present. In foggy conditions, the water droplets absorb and scatter the LiDAR signals, leading to attenuation and a reduction in the effective range of the sensor. This results in sparser point clouds and decreased accuracy in object detection. Wet and icy surfaces reflect LiDAR signals differently compared to dry conditions, which can lead to inaccuracies in interpreting the environment.

Radar, while robust, also faces weather-related issues. Similar to LiDAR, heavy rain or snow can attenuate its signals, reducing the clarity and range of detection [20]. Wet surfaces can cause unusual reflections of radar waves, while airborne particles like rain and snow can scatter the signals, leading to less precise readings. Different weather conditions can lead to varying levels of signal absorption by atmospheric particles, affecting the accuracy of the radar data.

IR sensors, commonly used for enhanced perception, particularly in low-light conditions,

face unique challenges when operating in adverse weather [21]. In conditions like rain or snow, IR light can be absorbed or scattered by water droplets or snowflakes in the air. This scattering reduces the effective range of the IR sensor and introduces noise into the captured data, making it difficult to accurately detect and identify objects. Fog presents a significant challenge due to the high water content in the air. The tiny water droplets in fog can absorb and scatter IR light, leading to a substantial reduction in visibility. This results in a loss of detail and contrast in the IR imagery, making it challenging to discern objects at a distance. Furthermore, since IR sensors rely on detecting thermal radiation, the thermal contrast between objects and their surroundings can be reduced in overcast or rainy conditions.

1.2.2 Challenges Faced by Perception Systems

Adverse weather can degrade the quality of features extracted by detectors [22]. For example, rain or fog can blur the edges and textures of objects, making it difficult for algorithms to identify and extract distinct features necessary for object detection. The appearance of features can vary significantly under different weather conditions. Snow may cover parts of objects, changing their shape and size in the sensor's view. This variability can confuse algorithms trained on data from clear weather conditions, leading to misclassification or missed detections. Figure 1.4 shows cases that the performance of pre-trained object detectors degrades in adverse weather conditions.

Classifiers rely on clear, distinct features to identify objects [23, 24]. In adverse weather, the altered appearance of objects can lead to classifier confusion, reducing accuracy in distinguishing between different types of objects like cars, pedestrians, and bicycles. Rapidly changing weather conditions can lead to sudden changes in object appearance. This requires classifiers to be highly adaptable and robust, a challenging task given the diversity and unpredictability of weather-induced changes.

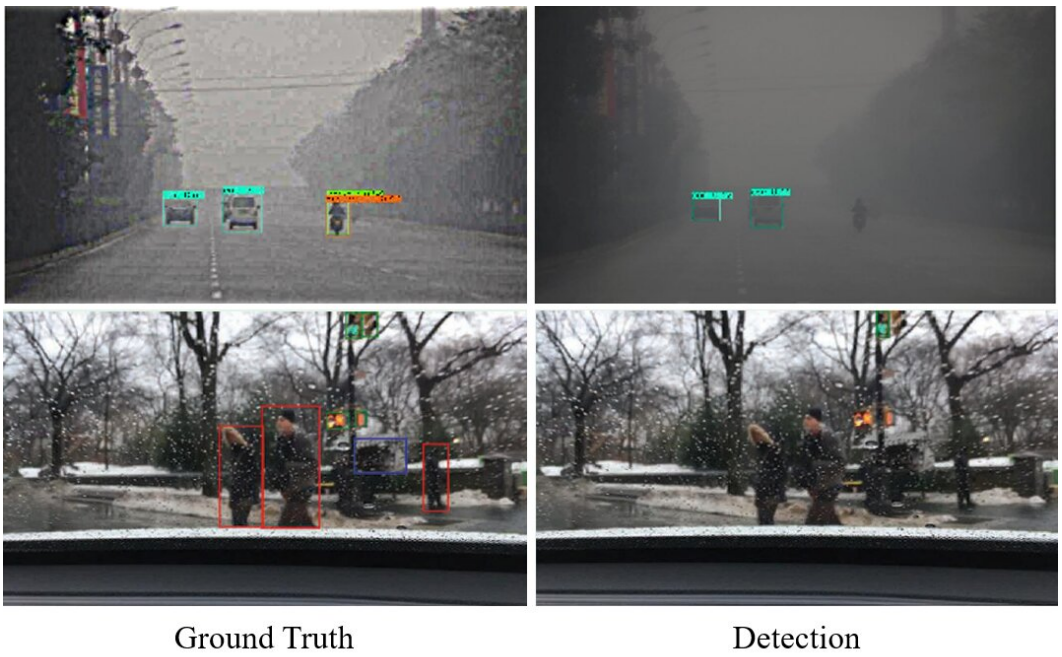


Figure 1.4: Performance degradation of pre-trained object detectors in adverse weather conditions.

The features used for environmental mapping, such as road edges, lane markings, and landmarks, can become obscured or distorted. This inconsistency leads to challenges in localization, as the vehicle's perception system struggles to match the observed data with its stored maps [25]. The dynamic nature of adverse weather can cause environmental features to change quickly over time, requiring continuous updates and recalibrations of the mapping system.

Lane detection algorithms often rely on edge detection to identify lane boundaries. However, in conditions like snow or heavy rain, these edges can become obscured or less pronounced, leading to inaccuracies [26]. The reduced contrast caused by adverse weather affects the perception system's ability to discern road boundaries and other critical road features, essential for maintaining lane discipline.

Adverse weather conditions like rain, snow, and fog blur the boundaries between different

segments in the vehicle's view, such as roads, sidewalks, and obstacles [27]. This blurring makes it challenging for algorithms to accurately segment the environment, a key step in understanding the scene. Weather conditions can alter the appearance and texture of surfaces. For example, a wet road looks different from a dry one, and a snow-covered car has a different texture than the same car in clear weather. These changes can lead to incorrect segmentation as the algorithms might fail to recognize familiar objects or surfaces under different conditions. In poor weather, the reduced visibility and altered lighting conditions can diminish the perception system's ability to discriminate colors. This affects semantic segmentation, which often relies on color cues to differentiate between various elements of the scene.

The prediction of object trajectories, essential for collision avoidance and path planning, becomes more complex in adverse weather. Reduced visibility and sensor noise can lead to inaccurate estimation of the speed and direction of other road users, increasing the risk of miscalculations in trajectory prediction [28]. Adverse weather adds a layer of complexity to the driving environment, with rapidly changing conditions and unexpected obstacles (like moving water or debris). This requires the prediction algorithms to be highly adaptive and quick to respond to new hazards that may not be present in clear weather conditions. The cues used for predicting the behavior of other road users or changes in the environment are often degraded in adverse weather. For example, the body language of pedestrians or the movement patterns of other vehicles can be obscured, making it challenging to predict their future actions accurately.

1.3 Purpose of Research

Domain adaptation in AV's perception systems is a methodological approach to address the discrepancy between the training data of the source domain and the real-world operat-

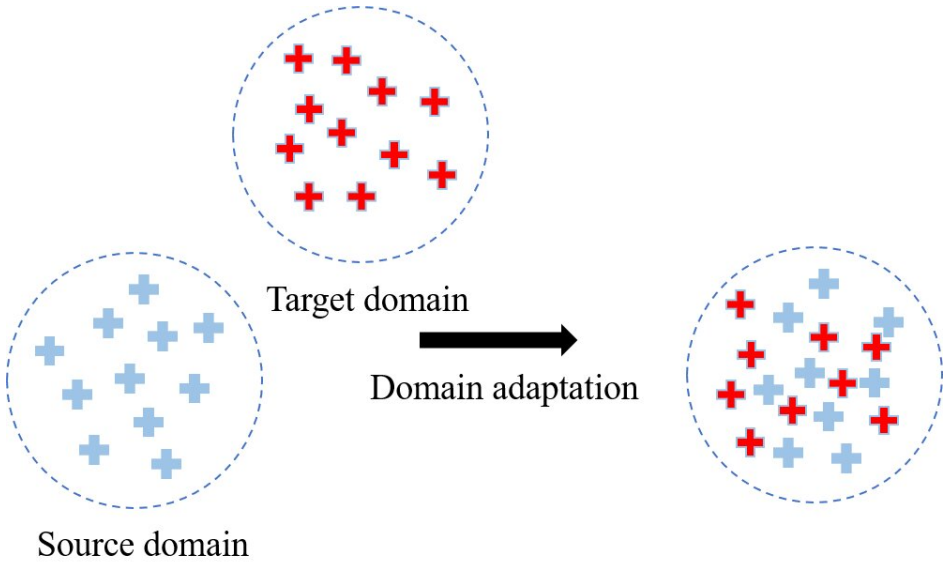


Figure 1.5: Domain adaptation tries to apply knowledge from a domain to a target domain with insufficient information.

ing conditions of the target domain as shown in Fig. 1.5. This discrepancy can be due to variations in lighting, weather, and other environmental factors.

The source domain typically consists of data collected under ideal conditions, such as clear weather and good lighting. However, the real-world data of the target domain presents more challenging conditions like rain, snow, fog, and nighttime driving.

Without domain adaptation, the performance of machine learning models degrades significantly when they encounter data that differ from their training set. This is due to the model's inability to recognize and correctly interpret features that appear differently in the target domain.

Image conversion as a technique in domain adaptation involves transforming images from the target domain to resemble those of the source domain as shown in Fig. 1.6. One of the primary goals of image conversion is to maintain feature consistency across domains. For instance, lane markings obscured by shadows in low-light conditions can be enhanced to ap-

Figure 1.6: Image conversion makes changes to an image while maintaining its original properties.

pear as they would in broad daylight, aiding in consistent feature extraction. The scarcity of certain types of weather-related or lighting-specific data can limit a model's exposure during training. Image conversion helps in augmenting the dataset with transformed images, providing a more comprehensive range of conditions for the model to learn from. Implementing image conversion typically involves techniques like style transfer, where the stylistic elements of one image domain are applied to another. This could mean applying the visual characteristics of daylight images to nighttime images or clear weather images to those taken in foggy conditions.

The strategic use of image conversion in domain adaptation directly contributes to the robustness of perception systems in AVs. By training on converted images, perception systems can better generalize across various environmental conditions. This enhances their ability to accurately detect and classify objects, regardless of the prevailing conditions. Image conversion can mitigate the adverse effects of environmental factors such as poor visibility or distorted object appearances. For example, converting a fog-affected image to resemble a clear day image can help in clearer object detection. The continuous process of converting and incorporating new images from different conditions into the training dataset allows for ongoing model refinement. This is crucial for adapting to the ever-changing driving environments and maintaining the relevance of the perception system. In challenging conditions, raw

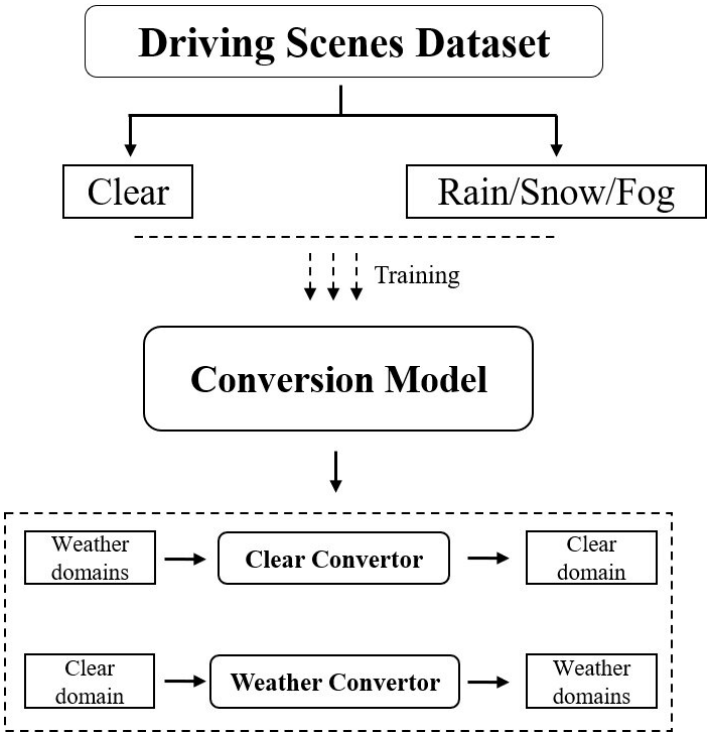


Figure 1.7: Research framework of this thesis.

sensor data may not be reliable due to noise and distortions. Preprocessing images through conversion techniques before feeding them into perception algorithms can significantly enhance data quality, leading to better decision-making.

In a more scientific context, domain adaptation through image conversion is a practical approach to address the inherent limitations of machine learning models trained on limited datasets. By enhancing data variability and ensuring feature consistency across diverse domains, image conversion plays a pivotal role in enabling AV perception systems to operate effectively in a wide range of real-world conditions. The research framework of this thesis employs image conversion to convert driving scene images from adverse weather conditions to clear weather conditions and vice versa, as illustrated in Fig. 1.7.

1.4 Overview of Proposed Methods

In this thesis, the complex task of image conversion for driving scenes is dissected into three distinct but interconnected subproblems: unpaired image conversion, realistic image conversion, and controllable image conversion. Each of these subproblems addresses a unique aspect of image conversion, employing advanced techniques from computer vision and machine learning.

The first part of the thesis focuses on unpaired image conversion, utilizing the concept of dual learning inspired by CycleGAN [29]. This approach is particularly useful for scenarios where paired training data (clear and adverse weather images of the same scene) are not available. A novel method is introduced where a weather layer is extracted by subtracting the converted image from the input images. This layer effectively captures the specific attributes of different weather conditions. The extracted weather layer is then fused with the features of a second generator using spatial feature transform techniques. This fusion enhances the ability of the model to incorporate weather-specific characteristics into the conversion process. The model is expanded to handle multiple weather conditions by introducing more generators. In this asymmetric framework, each adverse weather condition is assigned a dedicated generator, while a single generator is used for clear weather. This design allows for more specialized and accurate conversions for each weather condition.

The second part of the thesis enhances the realism of the converted images by using segmentation maps as additional input. This approach improves the semantic understanding of the model, allowing it to maintain the integrity of different objects and scenes during conversion. Deep supervision is employed to add supervisory signals to the intermediate layers of the generator. This technique helps in refining the feature extraction and transformation processes, leading to more realistic and semantically accurate image conversions.

In the third part, the generator is split into a style encoder and a content encoder. This

separation allows for more precise control over the style and content aspects of the converted images. A self-regression module is applied to constrain the style latent space. This ensures that the generated output remains within a permissible range, providing stability and consistency in the style aspects of the conversion. To further enhance the content encoder, a content feature discriminator is employed. This component helps in reinforcing the content-related attributes of the converted images, ensuring that the essential elements of the scene are preserved and accurately represented.

To support this in-depth research, "Realistic Driving Scenes under Bad Weather (RDSBW)" dataset is created. This dataset focuses on different weather conditions for driving scenes. Videos of driving scenes in different weather are recorded using a camera with a resolution of 1,920 $\times$ 1,080 pixels. The camera is mounted behind a car's windshield to obtain a clear view. From these videos, clear and distinct images are picked out. Then, they are sorted into categories based on the type of weather they showed. The dataset includes 4,171 images with rain, 4,777 images with snow, 2,052 images with fog, and 2,831 images of clear weather. It is ensured that the images are not sorted by location. This means each image is unique and not linked to a specific place.

This thesis uses the above wide range of images to train the models. This helps the models learn how to handle different weather conditions. The dataset is also good for testing the proposed models. How well they work with real-world images and are improved by them can be observed. In addition, this dataset can also be used by other people working on AVs and computer vision to test their own ideas and improve their technology.

1.5 Thesis Overview

The structure of this thesis is organized as follows. Chapter 2 provides a detailed overview of domain recognition and image conversion, classifying existing methods. Chapter 3 in-

troduces two unpaired image conversion methods, one for one-to-one and the other for one-to-many scenarios. Chapter 4 presents a real image conversion method based on semantic information. Chapter 5 discusses controllable graphic conversion methods based on latent space manipulation. Chapter 3 focuses on rainy and foggy weather images, while Chapters 4 and 5 focus on snowy images. Chapter 6 concludes this thesis and discusses future research directions in light of the identified shortcomings.

2 Related Work

2.1 Domain Adaptation

With the support of vast amounts of data, machine learning, particularly deep learning, has been widely applied in fields like computer vision and natural language processing, resulting in immense success. The ideal scenario for machine learning is to have a substantial number of labeled training instances, with the training and test data having the same distribution. However, in many real-world applications, obtaining sufficient labeled training data is often time-consuming, expensive, or even unfeasible. Additionally, the assumption of independent and identically distributed data, commonly made in machine learning, often does not hold in many tasks. Consequently, models trained using traditional machine learning algorithms often fail to achieve desired results in similar but new tasks, thus limiting their generalization and knowledge reuse capabilities.

Domain adaptation techniques can enhance the performance of machine learning models in cross-domain tasks. When there is a lack of labeled data in the target domain to train a well-performing machine learning model, one can consider pre-training the model in a different but related auxiliary domain with abundant labeled data, and then finetuning the pre-trained model to adapt and apply it to the target domain. This overcomes the challenge of obtaining labeled data in the target domain for practical applications. However, discrepancies in data distribution between domains pose a hurdle to model migration. Domain adaptation aims to learn a model that enables the knowledge acquired in an auxiliary domain

to be more effectively generalized in the target domain. By reducing the differences in data distribution, domain adaptation facilitates domain-invariant knowledge transfer and reuse. It is one of the cutting-edge research areas in machine learning and computer vision. Domain adaptation technologies are expected to address the issue of limited annotated data in the target domain and alleviate the high cost of training models from scratch, thereby improving the universality and knowledge transferability of machine learning models.

In a comprehensive review paper by Pan et al. [30], the research process of transfer learning is systematically explained. The paper provides a formal definition and classification of transfer learning and maps domain adaptation as one of its subfields.

In subsequent theoretical research, commonly used algorithms for shallow domain adaptation are divided into two main categories: instance-based domain adaptation and feature-based domain adaptation. Deep domain adaptation, as described by Wang et al. [31], is further divided into three categories: difference-based, confrontation-based, and reconstruction-based. Tan et al. [32] propose a classification into four categories: instance-based, mapping-based, network-based, and confrontation-based. Zhuang et al. [33] provide an overview of various representative methods of transfer learning and domain adaptation from the perspective of data and models. Zhao et al. [34] focus on single-source unsupervised domain adaptation scenarios, particularly, deep domain adaptation methods in this context. They classify deep domain adaptation methods into four categories based on the settings of domain offset loss and generation discrimination: difference-based methods, adversarial generation-based methods, adversarial discrimination-based methods, and self-supervision-based methods. Gautheron et al. [35] focus on the study of domain adaptation algorithms from the perspective of feature selection and feature space alignment. Wang et al. [36] expand and enhance the algorithm based on the idea of adversarial learning. Su et al. [37] combine the ideas of meta-learning, adversarial learning, and regularization to propose a weighted tem-

poral regularized domain adversarial network based on meta-learning. Fan et al. [38] review deep domain adaptation methods from four perspectives: domain distribution differences, confrontation, reconstruction and sample generation, and provide an overview of complex scenarios with different cross-domain label spaces.

Regarding the application of domain adaptation, Csurka [39] summarizes its usage in computer vision fields such as image classification, target detection, semantic segmentation, pose estimation, and video action detection, among others. Zhuang et al. [33] provide a summary of the application of domain adaptation methods in medical imaging and their use in computer-aided diagnosis, biological sequence analysis, traffic scene recognition, recommendation systems, and other domains. In addition, domain adaptation is widely employed in natural language processing tasks such as text classification, sentiment analysis, and machine translation [40].

2.1.1 Category of Domain Adaptation

This section conducts survey and analysis on the concept classification, representative methods, typical applications, and existing challenges of domain adaptation. The research scenarios of domain adaptation can be divided along different dimensions. This section classifies domain adaptation algorithms based on three dimensions: whether data labels are obtainable, the number of participating domains, and the composition of cross-domain data feature spaces.

2.1.1.1 Classification based on Supervision

Based on the availability and quality of data labels in the source and target domains, domain adaptation can be categorized into three classes: Unsupervised Domain Adapta-

tion (UDA), Semi-Supervised Domain Adaptation (SSDA), and Weakly-Supervised Domain Adaptation (WSDA).

UDA is currently receiving extensive research attention. It primarily focuses on scenarios where the source domain contains a large amount of labeled clean data, while the target domain has only a small amount of data with unavailable labels.

SSDA research focuses on scenarios where the source domain contains abundant labeled data, the target domain has data with unavailable labels, and there is also a small amount of labeled data in the target domain [41, 42]. The main difference between SSDA and UDA is that SSDA makes use of a small amount of labeled samples from the target domain during cross-domain adaptation. When there are very few labeled data samples available in the target domain, SSDA is sometimes referred to as Few-Shot Domain Adaptation (FSDA) [43].

WSDA takes into account scenarios where the source domain data may contain noise, relaxing the assumption that the source domain data is entirely clean. In WSDA, source domain data labels are obtainable, while target domain data labels are not available. However, it acknowledges that source domain data samples may contain noise in both their features and labels. The goal of WSDA is to train a model to mitigate the negative impact of source domain noise on the transfer and achieve positive transfer of clean source domain samples [44].

2.1.1.2 Classification based on the Number of Participating Domains

Based on the number of source and target domains, domain adaptation can be categorized into three classes: single-source domain adaptation, multi-source domain adaptation, and multi-target domain adaptation.

Single-source domain adaptation focuses solely on transferring knowledge from a single

source domain to a single target domain. Many traditional unsupervised and semi-supervised domain adaptation methods fall into this category.

Multi-source domain adaptation involves using labeled data from multiple different source domains. In this scenario, not only are there differences in data distribution between the source and target domains, but there may also be variations in data distribution among the multiple source domains. Mansour et al. [45] propose techniques like distribution-weighted combination, where a weighted combination of distributions from multiple source domains is used to construct the target distribution. Methods like Deep CockTail Network (DCTN) apply this rule in adversarial settings [46]. Additionally, techniques like Multi-Source Domain Adaptation (MSDA) networks are used to align the distribution differences between multiple source domains and the target domain [47].

Multi-target domain adaptation research involves transferring knowledge from a source domain to multiple unlabeled target domains, assuming that there are data distribution differences not only between the source and target domains but also among different target domains. In this context, certain approaches have been proposed. For example, Yu et al. [48] propose a method for model parameter adaptation, while Gholami et al. [49] use information-theoretic methods to identify shared feature subspaces across all domains to facilitate knowledge transfer from the source domain to multiple target domains.

2.1.1.3 Classification based on Feature Space

Domain adaptation can be categorized into two types based on whether the feature spaces of the source and target domain data are the same: homogenous domain adaptation and heterogeneous domain adaptation.

Homogeneous domain adaptation refers to scenarios where the source domain and target domain samples share the same feature space and label space, and have the same dimension-

ality. Such methods primarily focus on the same task across different domains, aiming to mitigate the performance drop caused by cross-domain data distribution shift. This enables cross-domain transfer and reuse of models or knowledge.

Heterogeneous domain adaptation refers to scenarios where the source domain and target domain have different feature spaces, which typically do not overlap. Source and target domains may not share feature labels, and their dimensionalities may also differ. Heterogeneous domain adaptation is more challenging than homogeneous domain adaptation because it requires addressing cross-domain data distribution differences while also performing transformations between feature spaces and label spaces to accommodate the need for knowledge transfer across domains [50].

2.1.2 Adversarial Learning-based Methods

Drawing inspiration from Generative Adversarial Networks (GANs) [51], adversarial approaches can be introduced into deep methods for domain adaptation. Adversarial-based methods can be categorized into two types: adversarial discriminative and adversarial generative.

From the perspectives of the type of loss function used, whether weight sharing is performed, and whether the underlying model is a generative or discriminative model, Tzeng et al. [52] categorizes domain adaptation methods based on adversarial learning and proposes a general framework.

2.1.2.1 Adversarial Discriminative-based Methods

Although adversarial discriminative domain adaptation methods employ different adversarial strategies, their fundamental idea is to impose an adversarial objective on a domain

discriminator. This transforms the measurement of domain distribution differences into domain confusion in the latent feature space. This approach is used to train the feature extractor, thus achieving feature-level domain adaptation.

The Domain Adversarial Neural Network (DANN) [53] is proposed based on the aforementioned general framework. Its architecture consists of a feature extractor, a classifier, and a domain discriminator. It incorporates the generative adversarial idea from GANs, and its training can be achieved by inserting specific Gradient Reversal Layers (GRL).

Tzeng et al. [52] introduce the Adversarial Discriminative Domain Adaptation (ADDA) method, which divides the optimization process into two separate objectives: generator and discriminator, using a label-flipping GAN loss.

In addition to aligning marginal distributions, Long et al. [54] introduce the Conditional Adversarial Domain Adaptation (CADN) method, which considers aligning conditional probability distributions to promote domain adaptation between two domains. Building upon DANN, CADN introduces the classifier's predictions as the condition upon which feature representations depend. It adds conditions to the domain discriminator by introducing joint variables related to classification predictions. Simultaneously, it models the cross-domain covariance between feature representations and classifier predictions, implicitly addressing multi-modal structure recognition issues, thereby enhancing the performance of cross-domain distribution adaptation.

2.1.2.2 Adversarial Generative-based Methods

GAN-based generative methods belong to pixel-level domain adaptation. They generate images from the source domain to the target domain and train them to be indistinguishable from images sampled from the target domain distribution, thus achieving domain confusion. Additionally, based on CycleGAN loss [29], researchers have proposed some effective do-

main adaptation methods.

Hoffman et al. [55] introduce the CyCADA (Cycle-Consistent Adversarial Domain Adaptation) method, which achieves cross-domain adaptation at both pixel-level and feature-level while ensuring semantic consistency. During adaptation, it uses cycle-consistency loss to match structure and semantic consistency while incorporating semantic loss based on a specific visual recognition task. The semantic loss guides the overall representation to be discriminative and maintains cross-domain semantic consistency before and after mapping. Similarly, Tzeng et al. [56] perform domain adaptation for target detection tasks using both pixel-level alignment and feature-level alignment.

Li et al. [57] extend previous research based on CycleGAN by combining conditional adversarial domain adaptation with cycle-consistency loss. They introduce the Cycle-Consistent Conditional Adversarial Transfer Network (3CATN) method to align two domains. They deploy a conditional domain discriminator using the covariance of features and their corresponding class predictions to capture complex multimodal structures embedded in the data. Additionally, considering that domain-invariant feature transformations are shared between two domains and can be mutually represented, they train two feature transformers: one that converts features from the source domain to the target domain and another that converts features from the target domain to the source domain. Cycle-consistency loss is calculated based on these two feature transformers. This approach captures complex multimodal structures in the data while avoiding negative effects caused by incorrect conditions.

2.2 Image Conversion

RGB images are essential data sources for perceptual systems. Eliminating domain gaps in images is a challenging problem. Image conversion, a domain adaptation method, aims to map source domain images to target domain images. It uses GANs and other deep learning

techniques for complex cross-domain and instance-level image conversions.

Image generation is a common task in computer vision, with the goal of creating images that are indistinguishable from real ones. In deep learning, GANs [51] have become a popular method for generating images due to their unique structure and learning approach. They consist of a generator and a discriminator, where the generator is responsible for producing images from input noise, and the discriminator's role is to distinguish between generated and real images. The generator and discriminator optimize their model parameters through adversarial learning. The generator attempts to confuse the discriminator, making it unable to differentiate between generated and real images. Meanwhile, the discriminator receives both generated and real images and classifies them as "0" or "1" to adjust the generator's performance. When the generated images closely resemble real ones, the discriminator cannot distinguish between them, achieving a balance in the adversarial process between the generator and the discriminator.

During training, the generator and discriminator are trained alternately. When updating the generator, the discriminator is kept fixed, and vice versa. As training progresses, the generated data become increasingly similar to real data. When the generator and discriminator reach a balance, the discriminator's output is around 0.5, indicating it cannot distinguish between real and generated data. At this point, the generator and discriminator have achieved their optimal models.

Unconditional GANs use noise as input and cannot control the generation of target images. Therefore, Mirza et al. [58] introduce Conditional Generative Adversarial Networks (CGANs), where image generation is conditioned on source domain images. They include additional conditions, where both the generator and discriminator receive conditional information to guide the generation process.

While GANs can generate realistic images, they still face issues such as training instability

and mode collapse. Researchers have addressed these problems by proposing improved models like Least Squares Generative Adversarial Networks (LSGAN) [59], Wasserstein Generative Adversarial Networks (WGAN) [60], and WGAN with Gradient Penalty (WGAN-GP) [61]. These enhancements have also found widespread applications in the field of image conversion. However, in the review literature on generative adversarial models [62, 63], the application of these models to image conversion is often described without detailed discussion. This section reviews recent image conversion methods and analyzes the latest research developments in this field.

2.2.1 Category of Image Conversion

Image conversion models are typically implemented in an encoder-decoder fashion. Some literatures refer to this approach as the generator. However, in feature-based conversion models, the encoding process is described as an encoder, and the decoding process is described as a generator. To maintain consistency in description, this chapter uses the terms encoder and generator to represent conversion models. The encoder downsamples input images, while the generator decodes the downsampled image features to generate the target image. Additionally, in conversion models based on cycle-consistency constraints, only a small amount of downsampling and upsampling is used, with the primary conversion accomplished by residual blocks. This model is referred to as a residual generator in this chapter. Typically, improvements to the generation process or the addition of optimization objectives are employed to enhance model performance. However, enhancing the generation process can make the network structure more complex and require more computational resources and time during training.

Image conversion can be divided into one-to-one mapping and one-to-many mapping, where one-to-many mapping includes multimodal mapping and multi-domain mapping. Mul-

timodal mapping refers to the generation of images where the style, such as color or texture, changes while still remaining within the same image domain. Meanwhile, multi-domain mapping refers to the conversion of source domain images into target images of multiple different domains with specific changes.

The conversion models of supervised learning have strong data dependencies and fewer forms. The conversion models of unsupervised learning, in order to break away from label dependence and enhance model processing capabilities, are complex and diverse. This section makes a more detailed division of such models, mainly including conversion models based on cycle consistency constraints, instance-level image conversion models, conversion models based on latent encoding, conversion models based on shared latent space, and conversion models based on feature separation. Among them, the conversion models based on cycle consistency constraints belong to one-to-one mapping, while the instance-level image conversion and conversion models based on latent encoding include both one-to-one and one-to-many mappings. The conversion models based on shared latent space and based on feature separation belong to one-to-many mapping.

2.2.1.1 Supervised Learning Methods

The training of supervised learning requires paired data, with strict one-to-one correspondence between source domain images and target domain images. A one-to-one mapping model refers to a model where a source domain image corresponds to a unique conversion result, including both generic conversion models and task-specific conversion models. Isola et al. [64] propose a general image conversion framework called pix2pix, which is compatible with various image conversion tasks such as image colorization, edge-to-image synthesis, and realistic image generation. It is based on Deep Convolutional Generative Adversarial Network (DCGAN) [65] and utilizes U-Net [66] to directly pass the features of the encoder

to the generator, bypassing the bottleneck layers of the encoder, allowing low-frequency features to be fully preserved. Additionally, they introduce the discriminator PatchGAN, which evaluates local regions of the generated image using receptive fields. Compared to the traditional approach of directly evaluating the entire image region, its local region evaluation improves the robustness and performance of the discriminator. However, in pix2pix, the L1 loss used can lead to the loss of high-frequency information, resulting in blurry converted images.

Perceptual Adversarial Network (PAN) [67] simulates the human perceptual process using a neural network model and replaces the L1 loss with a perceptual loss. The perceptual loss leverages a neural network model to extract image features and optimize deep-level abstract features. Traditional perceptual loss uses pre-trained models like Visual Geometry Group (VGG) [68] as feature extractors, and the quality of feature extraction depends on the dataset used to pre-train the model, limiting its generalization to other data. PAN uses a discriminator as a feature extractor to construct an adversarial perceptual loss, breaking away from pre-trained models and allowing for the extraction of perceptual features tailored to the specific dataset. This further enhances the effectiveness of the perceptual loss, but it still suffers from some image blurriness as it directly computes the distance between features at each layer.

Identical-Pair Adversarial Network (IPAN) [69] employs a perceptual similarity network [70] to create a discriminator that distinguishes between real and fake image pairs. Real image pairs consist of two real images, while fake image pairs consist of a generated image and a real image. Unlike PAN, IPAN uses the perceptual similarity network to establish a perceptual loss between the features of real and generated images. It first normalizes and scales the features at each layer and then computes the feature distance between real and fake image pairs, avoiding the image blurriness caused by directly calculating feature distances.

Regardless of whether the perceptual loss is calculated directly or indirectly, the problem of image blurriness cannot be completely resolved. Wang et al. [71] and He et al. [72] approach this issue from different perspectives by improving the quality of generated images through optimization of the generation process. Discriminative Region Proposal Adversarial Network (DRPAN) [71] uses score maps generated by the PatchGAN to identify the region with the lowest scores in the generated image and simultaneously constructs a corrector to rectify this region. During training, the corrector continuously repairs the region with the lowest scores while also fixing other regions affected by it, iteratively improving the entire image. Compared to PatchGAN, its judgment and correction of local regions are more direct and effective. He et al. [72] assume that a single transformation process cannot fully capture the transformation target, so a review process is needed for the generated images. To implement the review process, they add an inspector to the encoder-decoder structure. During the training process, the source domain image is encoded by the encoder, and the generator decodes it to generate the target image. This target image is then re-input to the encoder and combined with the source domain image features to produce the final output by the inspector.

Multimodal image translation enhances the diversity of generated images by obtaining various styles of generated images while keeping the source domain image unchanged. Existing methods acquire modal information separately from the source domain image and reference images.

Zhu et al. [73] introduce additional conditions to extend Variational AutoEncoder GAN (VAE-GAN) [74] and Layered Recursive GAN (LR-GAN) [75], leading to the development of BicycleGAN. This model maps source domain images and latent encodings to the target domain, allowing for the generation of different modes of target images by altering the latent encodings. In the Conditional VAE-GAN, latent encodings of target domain images

are extracted, and during the training phase, these latent encodings are mapped to a normal distribution. In contrast, Conditional LR-GAN focuses on the reconstruction of latent encodings to ensure they correspond to unique modes. Due to the stochastic nature of sampling from a normal distribution, BicycleGAN cannot explicitly specify the modal information of the generated images.

TextureGAN [76] uses the style information from reference images as a source of modes. It transfers this information to generated images using style loss and content loss [77], allowing the generated images to have different styles. This method effectively controls the modes of the target images. To ensure the quality of the generated images, it also incorporates local loss and global loss. With the influence of multiple losses, this model not only preserves the structure of the generated images but also transfers more detailed style information from the reference images. However, because content loss and style loss rely on pre-trained models, it may struggle to extract effective style features when there is a significant difference between the pre-trained model's data and the reference images. Additionally, balancing multiple losses requires manually setting hyperparameters, which increases the difficulty of training.

Albahar et al. [78] reduce the number of losses in their transformation model, optimizing it only with adversarial loss and L1 loss. Simultaneously, they build an additional encoder for reference images to avoid the pre-trained model from degrading the performance of the transformation model. To transfer the features from reference images, they define a parameter generator and a feature transfer layer. These components facilitate bidirectional feature transfer between the encoder of the source domain images and the encoder of the reference images. The parameter generator maps the features from each layer's output of the encoder into transfer parameters, and the feature transfer layer utilizes these parameters to transfer the features to the corresponding layers of the opposite encoder. Bidirectional feature transfer improves the efficiency of feature migration. However, the encoder for reference images

may lack optimization for feature extraction, potentially leading to uncontrollable feature quality.

2.2.1.2 Unsupervised Learning Methods

Unsupervised image transformation models use unpaired data, which do not require strict correspondences between source and target domains. This ease of obtaining non-paired data has led to a diverse range of transformation models in the unsupervised image transformation field.

Depending on the conversion purpose, unsupervised image conversion models can be divided into cycle-consistency-based conversion models, latent code-based conversion models, and shared latent space-based conversion models. These categories address various aspects of unsupervised image conversion, providing different approaches for different conversion objectives.

Cycle-consistency-based Conversion Models

Zhu et al. [29] address the limitations of supervised learning by proposing the unsupervised image transformation model CycleGAN, which allows for the use of non-paired data. This model utilizes any image from the target domain as a label for the corresponding image in the source domain, reducing the difficulty in obtaining data. To make use of unpaired data, they introduce a cycle-consistency constraint for image reconstruction. This constraint involves reconstructing the source and target domains through two generative paths: source domain to target domain and then back to source domain, and target domain to source domain and then back to target domain. In addition to the adversarial loss, it indirectly optimizes intermediate generative results through the reconstruction loss.

DiscoGAN [79] and DualGAN [80] employ a similar concept and to some extent reduce

the data requirements. While they achieve bidirectional transformations between two image domains, the mapping between these domains is overly flexible, making it difficult to obtain specific target domain images. Additionally, the use of pixel-wise L1 reconstruction loss in these models can result in issues like blurry generated images and information loss.

Increasing constraints is to make the generated images closer to the target domain images, thereby improving the quality of the generated images. Quality-aware GAN [81] uses image quality measurement methods to construct a quality-aware framework, optimizing original images and reconstructed images, reducing artifacts in generated images, and enhancing the clarity of generated images. The quality-aware framework consists of two types of losses: the quality-aware loss defined based on classical image quality measurement methods, which approximates the quality scores between the original images and the reconstructed images to optimize the reconstructed images, and the adaptive content loss defined based on deep networks, which optimizes the reconstructed images from deep abstract feature content. These two losses simulate the process of human perception of images, optimizing the reconstructed images at both the pixel level and the feature level, allowing generated images to capture more details.

Zhang et al. [82] pointed out that CycleGAN lacks effective constraints, leading to the loss of some information in generated images or the introduction of unnecessary changes. For example, in medical image conversions, images with tumors may lose the tumor part during the conversion process. To overcome these shortcomings, they introduce an additional smoothing term to optimize the adjacent regions of the image, ensuring that adjacent content in the source domain undergoes similar changes during the conversion process.

OT-CycleGAN [83] leverages Optimal Transport (OT) [84] to introduce additional constraints for achieving controllable one-to-one mappings that satisfy attribute transformations for specific tasks. OT calculates the minimal cost of transformations between different distri-

butions using a cost function. OT-CycleGAN computes the cost of attribute transformations in this manner and incorporates it into the optimization objective to achieve the desired transformations.

Improvements to the structure or mechanism include modifying the generation mechanism, enhancing the discriminator, and improving the generator. In the CycleGAN model, Lin et al. [85] introduce an auxiliary domain that lies between the source and target domains. For example, when converting brown hair into blonde hair, the auxiliary domain can represent black hair. With the help of the auxiliary domain, they construct multi-path consistency constraints to guide the generation of images using both source domain and auxiliary domain images, reducing the randomness in the conversion process. This enables generated images to obtain information from the auxiliary domain, thereby improving the quality of the generated images. However, it is important to note that introducing consistency constraints between the auxiliary domain and the target domain increases the complexity of the model by adding multiple generators and a larger number of parameters, making training more challenging.

Kim et al. [86] use an auxiliary classifier to obtain attention maps, which are represented as vectors and adaptively select features among different feature channels. Both the source domain image and the reference image are encoded simultaneously. The attention maps learn to differentiate features between the source domain image and the reference image, allowing the network to focus on learning the conversion part. Since the mapping between the auxiliary domain and the target domain remains a stochastic mapping, this approach is more targeted towards the conversion goal and avoids introducing additional generators. For controlling the shape and texture during the conversion process, the authors propose an adaptive selection mechanism for Instance Normalization (IN) [87] and Layer Normalization (LN) [88]. This mechanism allows the network to flexibly adjust the amount of change in

shape and texture, further enhancing control over the conversion target.

Stacked Cycle-consistent Adversarial Network (SCAN) combines multi-stage learning with CycleGAN to decompose the image generation process into multiple stages [89]. Each stage generates images at different resolutions, and the output of the current stage is combined with the output from the previous stage and passed to the next stage, allowing for progressive optimization. Compared to CycleGAN, it not only improves the quality of generated images in the same resolution transformation process but also enables the generation of higher-resolution images. Since it builds upon the basic CycleGAN framework, it still faces challenges such as random mappings and information loss. In addition to enhancing the quality of generated images, improvements in the structure or generation mechanism are also used to reduce the complexity of model training and the number of parameters.

CycleGAN++ removes the bidirectional consistency constraint and retains the unidirectional reconstruction loss [90]. It also adds domain information and classification loss to ensure the quality of generated images. This approach eliminates the circular structure of CycleGAN, reducing computational complexity during training and improving training speed.

Van der Ouderaa et al. [91] utilize reversible neural networks to construct a generator that enables reversible conversions. This method relies on a single generator to achieve bidirectional conversions between the source and target domains while maintaining model capacity and image quality. Reversible networks compute the output corresponding to each intermediate activation layer by reverse calculation through access to the output of the final activation layer. Consequently, there is no need to store the output of each activation layer during training, reducing the model's spatial complexity.

Latent Code-based Conversion Models

Latent encoding typically comes from images encoded by an encoder or noise sampling, representing information such as image content, attributes, and modes. Models based on

latent encoding usually involve networks that learn the required information from the latent encoding.

Chen et al. [92] define an interpolator to obtain interpolations between the latent encodings of the source and target domains. These interpolations describe the states and transformation paths occurring during the conversion process. The image transformation process includes multiple paths that cannot be represented in a single generated image. Using interpolated features allows for the generation of corresponding images for intermediate states, not only describing the transformation process but also achieving multi-domain and multi-modal mapping. However, using interpolation to achieve multi-domain and multi-modal mapping cannot edit specific attributes.

Xiao et al. [93] describe the latent encodings of source domain images and reference images as combinations of multiple attributes. These latent encodings are divided into multiple parts, with each part corresponding to different attributes. Fine-grained image transformations are achieved by swapping the attributes of source domain images and reference images.

Furthermore, SingleGAN [94] and InjectionGAN [95] achieve multi-domain and multi-modal mapping by using latent encoding to model the modes of the target domain under the guidance of domain encoding. Both of them employ a single generator to construct cyclic constraints, limiting the level of dissimilarity in the image domain, but they cannot be used for specific attribute editing. In conversion models based on cycle consistency constraints, Li et al. [96] describe the information difference between image domains as domain information imbalance, with information-rich domains representing rich content and information-poor domains representing the opposite. To balance the information difference, they introduce auxiliary variables in CycleGAN to construct AsymGAN. These auxiliary variables represent the latent encoding corresponding to the information-rich image domain and are used during training to map the information-poor image domain to the target do-

main based on this encoding. The auxiliary variables bridge the information gap between the source domain and the target domain, improving the quality of generation and enhancing the model's resistance to interference.

Almahairi et al. [97] use auxiliary variables to represent missing information during the conversion process. Unlike AsymGAN, this model considers transformations between arbitrary image domains, enhancing the model's data processing capabilities. In traditional unsupervised image conversion, complex cross-domain mappings typically rely on multiple generators and optimization objectives, making the model training complex and difficult to converge. To simplify the training process, Aiharbi et al. [98] rely on fundamental generative networks and optimization objectives. They use noise to model latent encoding and control the degree of transformation for each feature in the network layers. The latent encoding is embedded into the network layers using a fully connected approach, avoiding complex transformation processes and optimization objectives. This approach achieves multi-modal mapping but lacks control over image domains and cannot accomplish multi-domain mapping.

Shared Latent Space-based Conversion Models

Shared latent space [99] is based on the assumption that different image domains can be mapped to the same space, using shared information to establish relationships between domains and achieve cross-domain conversion. Conversion models based on shared latent space need to establish multiple optimization objectives to ensure a high degree of consistency in the shared portion. Similar to conversion models based on cycle consistency constraints, those based on shared latent space aim to explore how to effectively utilize non-paired data. While conversion models based on cycle consistency constraints optimize for simple domain-to-domain bidirectional conversions to generate target images, those based on shared latent space establish domain relationships through latent shared information. This

approach not only effectively utilizes non-paired data but also allows for the generation of target images by controlling different images, achieving diversity and determinism in the generated images.

Liu et al. [99] point out that without any assumption, it is impossible to obtain the joint distribution of different domains through the marginal distribution of image domains, which means that image conversion cannot be achieved. To address this issue, the hypothesis of a shared latent space is proposed. During the training phase, the latent encodings of source domain images and target domain images are mapped to the distribution of a shared latent space, ensuring that both latent encodings reflect shared information consistently, avoiding the influence of non-shared parts. However, when there is a significant semantic difference between two domains, this method cannot address domain semantic bias.

Taigman et al. [100], Royer et al. [101], and Murez et al. [102] also achieve image conversion based on the assumption of information sharing. Taigman et al. [100] fix the parameters of the encoder during the training process, making reverse conversion impossible. Royer et al. [101] and Murez et al. [102] add constraints on the latent encodings, further increasing the degree of information sharing and reducing the impact of domain differences.

While Liu et al. [99] achieve mapping between different domains of similar images, they could not accommodate multi-domain mapping for different types of domains. Anoosheh et al. [103] construct a multi-encoder-decoder model called ComboGAN, which consists of multiple encoders and generators, with each encoder and generator corresponding to different image domains. Encoders map images from different domains to a shared latent space, and generators use encodings from the shared latent space to generate images in the corresponding domains. This approach not only achieves cross-domain mapping between different types but also accommodates inputs from multiple different domains.

Lin et al. [104] simplify ComboGAN by using a single-encoder-multiple-decoder struc-

ture, reducing the need for domain determination and encoder parameters for input domains. However, as the number of domains increases, methods by Lin et al. [104] and Anoosheh et al. [103] require a large number of generators, which increases training difficulty and storage space.

2.3 Datasets

This section introduces the datasets used in this thesis. Among them, CityScapes [1] and EuroCity Persons [9] are public datasets, CityScapes Weather are synthetic datasets derived from the CityScapes dataset. They all contain images of driving scenarios.

Cityscapes Weather Datasets Cityscapes dataset is an annotated corpus of 5,000 driving scene images captured in urban areas. Researchers have also simulated various weather effects onto the dataset, using information such as depth maps based on atmospheric scattering models. Foggy Cityscapes [2] dataset includes three different fog densities for each image, representing visibilities of 150 m, 300 m, or 600 m, respectively. Rain Cityscapes [3] dataset is based on 295 images, which are used to generate 36 different foggy concentrations and rain types for each image. Snow Cityscapes [4] consists of 2,000 pairs of images with a resolution of 512×256 pixels for each of the training and testing sets.

EuroCity Persons Dataset The EuroCity Persons dataset [9] is a collection of images of pedestrians, cyclists, and other riders in urban traffic scenes, captured from a moving vehicle in 31 cities across 12 European countries. The dataset provides a large number of highly diverse, accurate, and detailed annotations for each image, including bounding boxes around pedestrians and cyclists, as well as additional attributes such as orientation, visibility, and occlusion. It is divided into daytime and nighttime sets with over 47,300 images. In the

following experiments, to make a comparable match to the snow dataset, 5,921 images are randomly selected from the daytime training set.

2.4 Conclusion

This chapter first provided a comprehensive overview of domain adaptation in machine learning. It discussed the challenges of applying models trained on source domain to a different target domain, especially when labeled data in the target domain is scarce. It covered various domain adaptation techniques, categorized based on supervision, the number of participating domains, and feature space composition. It detailed different methods like unsupervised, semi-supervised, and weakly-supervised domain adaptation, and delved into specific approaches such as distance-based methods and adversarial learning. Applications in computer vision, natural language processing, and other fields were also discussed.

Then, a detailed analysis of image conversion techniques in deep learning were provided, focusing on GANs and their variants. It discussed the challenges and advancements in image generation, emphasizing the roles of generators and discriminators in GANs. It covered conditional GANs for controlled image generation and addressed issues like training instability. Various categories of image conversion, including one-to-one and one-to-many mappings, were explored, alongside supervised and unsupervised learning methods for image transformation. It also delved into specific conversion models and their applications, highlighting the importance of features like cycle-consistency, instance-level conversion, and latent encoding in achieving effective image transformations. The last section introduced public datasets used in this thesis.

3 Unpaired Image Conversion

3.1 Introduction

In the realm of Autonomous Vehicle (AV) technology, the robustness and accuracy of perception algorithms play a pivotal role in ensuring safety and reliability. A significant challenge in this domain arises from the diverse and often unpredictable weather conditions that vehicles encounter on roads. When adverse weather occurs, retrieved images with low contrast and poor visibility can degrade the performance of the visual algorithms used in AV's perception systems, such as detection, tracking, and intention estimation. [18]

Unpaired image conversion is essential for converting images from one set of conditions, such as clear weather, into another, like rainy or foggy scenarios, without the need for corresponding paired images. This approach is particularly valuable because obtaining paired data in driving scenes is inherently challenging. Paired data requires capturing the exact same scene under different weather conditions, which is not only logistically complex but also time-consuming and often impractical. Factors such as changing natural light, moving objects, and evolving landscapes make it nearly impossible to acquire perfectly paired images in real-world driving scenarios.

The significance of unpaired image conversion lies in its potential to enhance the perception algorithms of AVs. By exposing these algorithms to a wider range of driving scenes recreated under various weather conditions, the vehicles can be better prepared for real-world scenarios.

This chapter aims to explore the methodologies in unpaired image conversion. The first part proposes a modified version of CycleGAN [29] for single weather image conversion by adding weather layer loss and information guidance. The model jointly learns the clear-to-weather conversion and its backward conversion in an end-to-end framework. The second part, to break the limitation of numbers in participant domains, extends the previous model with four generators and four discriminators. The four Generative Adversarial Networks (GANs) are trained to perform conversion between rain, snow, fog, and clear weather conditions. Once training is completed, the clear weather generator can convert the image from the source domain into the clear domain, no matter which of the three weather conditions is present.

3.2 Single Type Image Conversion

The proposed single-image conversion model is based on an improved version of CycleGAN. To effectively disentangle the weather layer from the source image, The model employs a weather layer loss inspired by the Joint Rain Generation and Removal (JRGR) method by Ye et al. [105]. The main idea is to regard images degraded by bad weather as the composition of a weather layer and a clean background. Therefore, the statistical distance between the removed layer and the generated layer is calculated. Then, the gradient feeds back to the network to ensure that generators are handled in the same weather conditions.

Besides, to enhance the supervision, the model utilizes the Spatial Feature Transform (SFT) [106] to fuse the weather layer feature into the network, so that it can act as an information guidance. Figure 3.1 shows the overall framework of the proposed model.

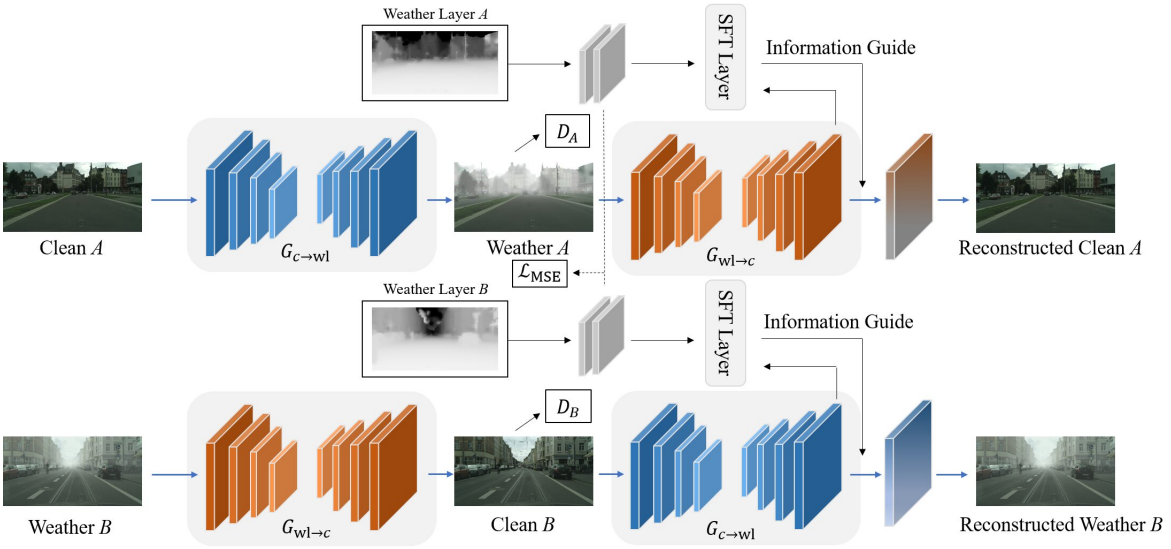


Figure 3.1: Architecture of the proposed bad weather removal model.

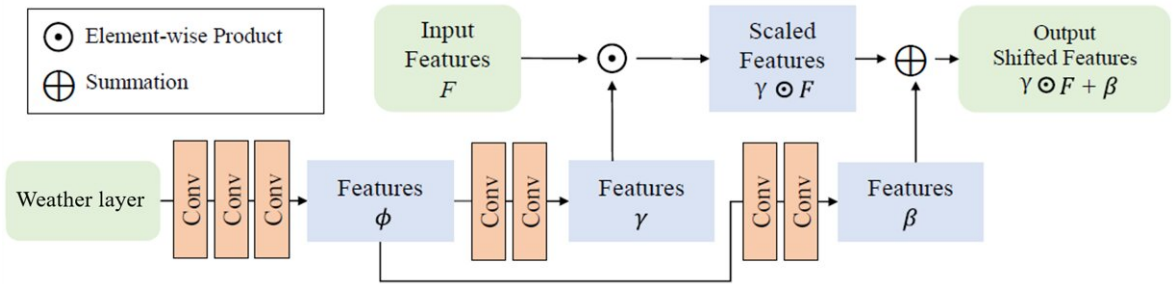


Figure 3.2: Spatial Feature Transform (SFT) layer in the proposed image conversion model.

3.2.1 Weather Information Guidance

In the image conversion process, the image of adverse weather is considered as a composition of weather layer and clear layer. The former contains the weather information which can be used to guide the cycle conversion process. To effectively use this information, SFT is used to combine weather layer and neural network features. Fig. 3.2 explains the detail of the SFT layer.

The main idea is to fuse the features of the middle layer into the original features in spatial dimensions through affine transformations. SFT is proposed by Wang et al. [106], originally applied to super-resolution reconstruction. Following the method of Shao et al. [107], a two-layer convolution module is used to extract the condition maps ϕ from the weather layer. Then the map is input to the other two convolution layers to predict the modulation parameters γ and β. Finally, the shifted features are obtained by:

$$\mathrm{SFT}(F \mid \gamma, \beta) = \gamma \odot F + \beta, \tag{3.1}$$

where F is the feature maps of the second to last convolution layer. Note that since the number of elements in the weather layer tensor is close to 0, if it is input to the convolution layer directly, the model will suffer from a vanishing gradient problem. Therefore, the tensor is normalized first when it is generated.

3.2.2 Loss Functions

The proposed model adds an overall loss to the architecture of CycleGAN, described as:

$$\begin{aligned} \mathscr{L}\left(G_{c\to \mathrm{wl}}, G_{\mathrm{wl}\to c}, D_A, D_B\right) =& \mathscr{L}_{\mathrm{GAN}}\left(G_{c\to \mathrm{wl}}, D_A, C, W\right) + \mathscr{L}_{\mathrm{GAN}}\left(G_{c\to \mathrm{wl}}, D_B, W, C\right) \\ &+ \lambda_c \mathscr{L}_{\mathrm{cyc}} + \lambda_{\mathrm{wl}} \mathscr{L}_{\mathrm{wl}}, \end{aligned} \tag{3.2}$$

where $\mathscr{L}_{\mathrm{GAN}}\left(G_{c\to w}, D_A, C, W\right)$ and $\mathscr{L}_{\mathrm{GAN}}\left(G_{w\to c}, D_B, W, C\right)$ are the adversarial losses of GAN. λ_c and λ_{wl} control the effects of cycle consistency loss and weather layer loss.

3.2.2.1 Weather Layer Loss

In order to prevent the removed weather from being different from the generated ones, such as input rain to generate fog, a weather layer loss is introduced. Now, there are many ways to measure the distance between two images. From the perspective of information

entropy, there are Kullback–Leibler (KL) divergence and Mutual Information (MI) methods, and from the perspective of distance, there are L1 and L2 norms. However, the amount of calculation involved in the former increases explosively with the growth of the image size, and the latter will make the generated weather layer lack change. Therefore, the Mean Square Error (MSE) is a more balanced choice. Assuming that the input of the first generator of each process is C_A, C_B, and the output is W_A, W_B, the definition of weather layer loss is provided as:

$$\mathscr{L}_{\mathrm{wl}}(\mathrm{wl}_A, \mathrm{wl}_B) = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{wl}_{A_i} - \mathrm{wl}_{B_i})^2, \tag{3.3}$$

where $\mathrm{wl}_A = W_A - C_A$, $\mathrm{wl}_B = W_B - C_B$.

3.2.2.2 Cycle Consistency Loss

The cycle consistency loss is introduced by CycleGAN [29] as:

$$\mathscr{L}_{\mathrm{cyc}} = \left\| G_{c\to\mathrm{wl}}\left(G_{\mathrm{wl}\to c}\left(C_A\right)\right) - C_A \right\| + \left\| G_{\mathrm{wl}\to c}\left(G_{c\to\mathrm{wl}}\left(W_B\right)\right) - W_B \right\|, \tag{3.4}$$

which calculates the L1 norm between the input image and the reconstructed image. The purpose is to prevent the second generator from generating random images of the target domain. To ensure the priority of background restoration, a higher weight than the weather layer loss is set. Note that, for weather removal, generators in the model handle image conversion between clear domain c and weather domain wl.

3.2.3 Experiment

This section evaluates the performance of the proposed method on public datasets and a self-collected dataset using image quality metrics as the measurement. Additionally, the performance of the pedestrian detector on both source and transformed images is also assessed.

3.2.3.1 Implementation Details

The model uses the Pytorch framework for training, testing, and image preprocessing. Two NVIDIA TITAN RTX GPUs are used for training. The training process performs 200 epochs on each dataset to ensure convergence. In the training phase, the Adam optimizer [108] and step learning rate schedule are used. In addition, the model sets λ_c as 10 and λ_{wl} as 2. The proposed model is similar to the original CycleGAN, except for the weather layer loss and information guidance layer. The image is randomly cropped into 360×360 pixels and input into the generator of ResNet18 [109] backbone.

3.2.3.2 Datasets

In the first experiment, 1,048 training images and 132 testing images are selected from Rain Cityscapes [3], and mixed with one-third of Foggy Cityspcapes [2] images with a visibility of 150 meters. This is called mixed weather CityScapes dataset. In the second experiment, the clear set and fog set of the Realistic Driving Scenes under Bad Weather Dataset (RDSBW) are used.

RDSBW is a newly created dataset that consists images of driving scenarios in four different weather conditions. Videos capturing different weather conditions were recorded using a camera mounted behind a car's windshield for clarity. High-quality images were picked out with a resolution of $1,920 \times 1,080$ pixels. All of these images were collected in China, with snow, fog, and clear weather scenarios collected in the northern regions, while rainy weather scenarios collected in the southern regions.

In detail, rainy weather scenes were captured in a small town, where the scenarios included a high number of pedestrians and vehicles, along with complex road conditions. Snow scenes were captured in the downtown, including main roads and urban highways. In the former, both pedestrians and vehicles were relatively densely populated, while in the case of high-

ways, only vehicles were present, and they moved at a fast pace. Fog and clear weather scenes were captured in the suburban areas of the city, featuring spacious roadways with a relatively sparse population of pedestrians. Due to adverse weather conditions during data collection, both fog and clear weather conditions resulted in great occlusion to objects around vehicles and the forward visibility.

It is important to note that the images are not organized by location, ensuring each image is unique and not associated with a specific place. The duration for each weather condition is over two hours. Figure 3.3 shows examples of the RDSBW dataset images from each set.

3.2.3.3 Experiment Results

Image Conversion and Detection Results on Mixed Weather CityScapes Dataset

Figure 3.4 shows qualitative results on the mixed dataset. Although faded color and small artifacts in the sky region can be observed, disentangled weather layers are removed from the foreground and benefit the ACSP detector. It is because the weather layer loss forces the model to remove and generate the same kind of weather. Meanwhile, the SFT layer transmits the weather information to the second generator, and the latent features learned by the first generator are kept. The quantitative results of Peak Signal-to-Noise Ratio (PNSR), Structural Similarity Index Measure (SSIM), and Feature Similarity Indexing Measure (FSIM) are presented in Table 3.1.

PSNR serves as a reliable objective measure for images, quantifying the discrepancy between corresponding pixel values. Higher PSNR values indicate reduced distortion in the generated images.

Figure 3.3: Sample scenes from the Realistic Driving Scenes under Bad Weather (RDSBW) dataset. Note that the images are uncorrelated by location.

Figure 3.4: Results on mixed weather CityScapes Dataset.

Table 3.1: Average PSNR, SSIM, and FSIM results on 100 sample images from mixed weather CityScapes dataset.

Method	**PSNR** ↑	**SSIM** ↑	**FSIM** ↑
Proposed	**47.28**	**0.97**	0.71
CycleGAN [29]	46.84	0.96	0.71
Dark Channel Prior [110]	43.55	0.90	**0.85**
FastCUT [111]	44.07	0.94	0.84
Haze-lines [112]	37.99	0.89	0.75

SSIM, on the other hand, assesses the similarity between two images by considering their luminance, contrast, and structure. An SSIM value of 1 indicates that the two compared images are identical in terms of structural information and quality, while a value of 0 indicates that the images are entirely different in these respects.

In Table 3.1, quantitative results were compared with other methods, including CycleGAN [29]. According to PSNR and SSIM metrics, the proposed model achieved the best result surpassing State-Of-The-Art methods [110, 111, 112]. Moreover, the model reached higher PSNR, SSIM, and FSIM values than CycleGAN, which means weather loss and information guidance improved the capability of the architecture on mixed weather CityScapes dataset.

An interesting finding is that previous methods that focus on physical rules showcased higher FSIM while the proposed method utilizing convolution features performed worse.

Adapted Center and Scale Prediction (ACSP) [113] makes some adaptations on the basis of Center and Scale Prediction (CSP) [114] which is an anchor-free detector. Compared with anchor-based methods, anchor-free methods sacrifice accuracy for speed. Therefore, it is more suitable for real-time applications on intelligent vehicles. The experiment tests the removal effects with vanilla ACSP, which achieved good results on Cityscapes [1]. The detection numbers of the ACSP detector were calculated on randomly selected 100 test images and shown as a boxplot in Fig. 3.5. It shows that the proposed model improves detection efficiency obviously by removing the visual effect caused by fog.

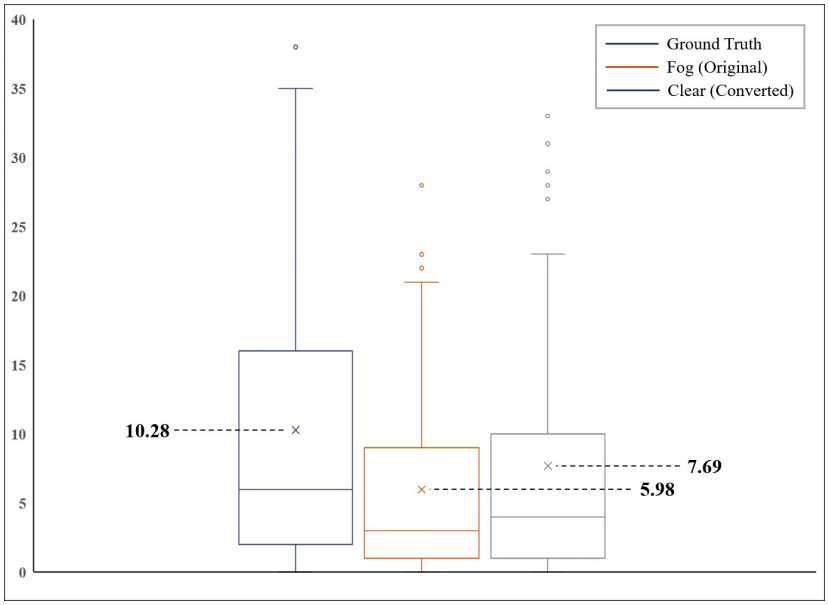


Figure 3.5: Box plot of ACSP detection numbers. "×" indicates the average value. After being processed by the proposed model, detection numbers increase.

Figure 3.6: Qualitative conversion results on RDSBW dataset compared to SOTA methods.

Image Conversion Results on RDSBW Dataset

Figure 3.6 shows the qualitative conversion results on the RDSBW dataset as there is no ground truth. We can clearly see that dark channel prior [110] suffers from high atmospheric light value. While CycleGAN [29] does not have a significant impact when the background contains only a flat color. Although the proposed model cannot remove all the foggy layers, it increases the contrast and visibility.

3.3 Multiple Type Image Conversion

This section proposes a novel model called Multiple Weather Conversion GAN (MWCG), inspired by CycleGAN [29]. The goal of this method is to convert clear weather images of traffic scenes into versions of these images with different types of weather degradation and then convert them back into clean ones. The proposed method can also be used to convert real-world, weather-degraded images into clearer ones. Overall, MWCG consists of three GANs for weather effect generation and one GAN for weather effect removal. The rationale for creating a multi-weather application is based on the observation that it would be convenient to be able to use a single model to remove various types of adverse weather effects that drivers are likely to encounter.

3.3.1 General Pipeline

To explain the theoretical basis of the proposed method in more detail, suppose rain, snow, and fog are three sub-domains of an adverse weather set ($X = x_1, x_2, x_3$) and that Y represents a clear weather domain. As shown in Fig. 3.7, there exist three mappings from adverse to clear weather: $x_1 \rightarrow Y$, $x_2 \rightarrow Y$, and $x_3 \rightarrow Y$. Furthermore, conversely, there also exist three mappings from clear to adverse weather: $Y \rightarrow x_1$, $Y \rightarrow x_2$, and $Y \rightarrow x_3$. In order to sim-

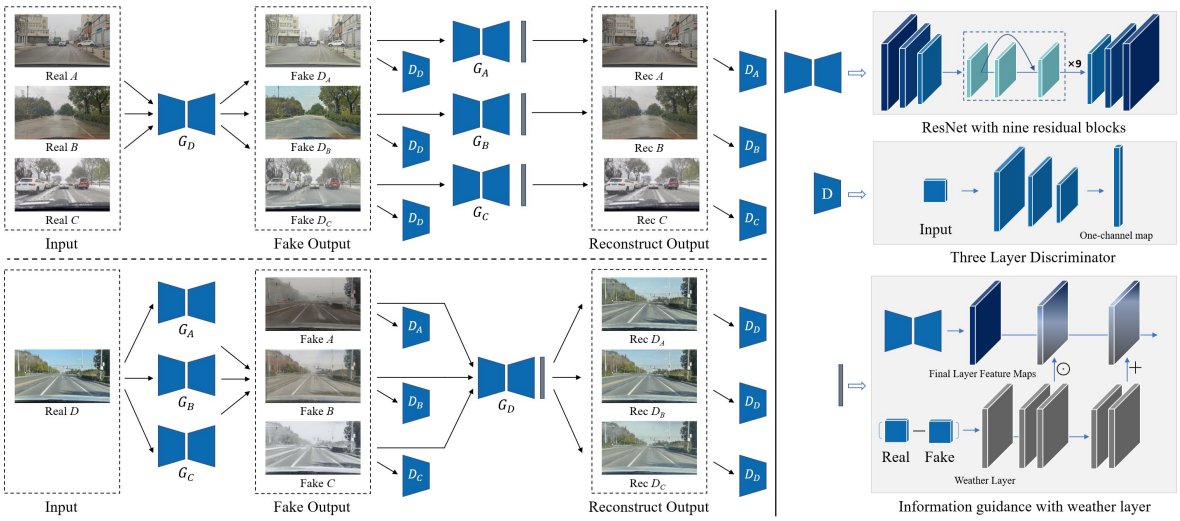


Figure 3.7: Architecture of the proposed Multiple Weather Conversion GAN (MWCG), consisted of four generators. All the generators have a ResNet encoder-decoder with nine residual blocks. Their associated discriminators are a three-layer CNN and a one-channel prediction map is output.

plify the mapping process, the model compresses the mapping $X \rightarrow Y$ into one network. Therefore, the proposed model requires four generators: G_A, G_B, and G_C for generating adverse weather effects (rain, snow, and fog respectively) and G_D for adverse weather removal. Correspondingly, four discriminators (D_A, D_B, D_C, and D_D) are introduced to distinguish real images from the generated, fake images. The pseudo-code of MWCG is provided in Algorithm 1.

Since the goal is to translate the unpaired, real-world weather images, MWCG borrows the cycle consistency principle from pioneering works [29, 115, 116, 117] to regulate the structure of the output images, so they remain the same as the input images. Therefore, after an image is translated by the weather removal/generation network, the model can translate it back into its original domain using the same generator.

Here A, B, C, and, D are used to represent sets of rain, snow, fog, and clear weather images, respectively. As part of a single processing step, the image data are simultaneously sorted

Algorithm 1 Multiple Weather Conversion GAN (MWCG)

Input: Training data pairs (A, B, C, D) ▷ In order of foggy, rain, snow, and clear

Output: Generator networks G_A, G_B, G_C, G_D

Initialize generators and discriminators
Define loss functions
Define optimizers for generator and discriminator
while $epoch \leq total\_epoches$ **do**
    for data pair (A, B, C, D) **in** data_loader **do**
        Generate fake images: $F_{D_A} = G_D(A)$, $F_{D_B} = G_D(B)$, $F_{D_C} = G_D(C)$ and $F_A = G_A(D)$, $F_B = G_B(D)$, $F_C = G_C(D)$
        Generate reconstruct images: $R_A = G_A(F_{D_A})$, $R_B = G_B(F_{D_B})$, $R_C = G_C(F_{D_C})$ and $R_{D_A} = G_D(F_A)$, $R_{D_B} = G_D(F_B)$, $R_{D_C} = G_D(F_C)$
        Update Discriminator D_A, D_B, D_C, D_D
        Update Generator G_A, G_B, G_C, G_D
    end for
end while

into two different places. On the one hand, real A, real B, and real C are input to G_D and the fake clear images are output. These fake images will then be input to G_A, G_B, and G_C to obtain the reconstructed adverse weather images. On the other hand, real D images will be simultaneously input to G_A, G_B, and G_C to obtain fake, adverse weather images. These results then go through G_D to obtain the reconstructed clear images.

3.3.2 Weather Generators and Discriminators

As the backbones of MWCG's four generators, ResNet [109] (with a Residual block) is used to maintain the previous output through a skip connection, a method which has been proven to be effective when training deeper neural networks. The input image will first be scaled down twice, using large convolutional filters. After obtaining the desired resolution, the first layer of feature maps of the image will go through nine ResNet blocks, generating denser representations with more channels. In a similar manner to the encoder-decoder architecture used in [118], two transpose convolutional layers then follow, to reverse the dense representations back into normal-size RGB images.

For the discriminators, the model uses simple, three-layer Convolutional Neural Networks (CNN) that gradually increase the number of filters. The last layer outputs a one-channel prediction map, which is the encoding input for the criterion function. Because the RDSBW dataset consists of high-resolution images, it would be time and memory-consuming to infer the entire images. Therefore, in the training stage, images are cropped into 480×480 pixel patches to reduce the calculation burden, which is then learned using PatchGANs [29, 64, 119].

3.3.3 Weather Information Guidance

To obtain better results, a disentangled training strategy [105, 120] is introduced that regards images degraded by adverse weather as composites of a weather layer and a clean background. the strategy can then calculate the numerical distance between the input and output of each generator and store those distance values in a tensor that has the same dimensions as the input image. This tensor is referred to as the weather layer. To provide additional input to the generator, SFT [106] is incorporated to combine the weather layer feature with

the extracted feature maps, allowing the weather layer to serve as guidance. The details of the SFT layer was described in 3.2.2.

The model then uses the feature maps of the penultimate convolutional layer of the GAN generator as input F to the SFT module. While the fake image output from the SFT module is similar to the input image, the values of the elements in the weather layer are close to 0, which is the consequence of the vanishing gradients. That is why the weather layer is normalized before it reaches the SFT module.

3.3.4 Loss Functions

Three kinds of loss functions are used when formulating an MWCG model: adversarial loss, cycle consistency loss, and identity loss. The overall objective function is formulated as:

$$\mathscr{L}_{\text{obj}} = \mathscr{L}_{\text{GAN}} + \lambda_c \mathscr{L}_{\text{cyc}} + \lambda_i \mathscr{L}_{\text{identity}}, \tag{3.5}$$

where λ_c and λ_i are weights that control the cycle consistency loss and identity loss, respectively.

3.3.4.1 Adversarial Loss

Adversarial losses are used to obtain four mappings, three for from clear to adverse weather ($Y \to x_1$, $Y \to x_2$, and $Y \to x_3$) and one for from adverse to clear weather ($X \to Y$). The first three mappings can be expressed as:

$$\sum_{i=A,j=1}^{C,3} \mathscr{L}_{\text{GAN}}\left(G_i, D_i, Y, x_j\right) = \mathbb{E}_{i \sim p_{\text{data}}(i)}\left[\log D_i(i)\right] + \mathbb{E}_{D \sim p_{\text{data}}(D)}\left[1 - \log D_i(G_i(D))\right], \tag{3.6}$$

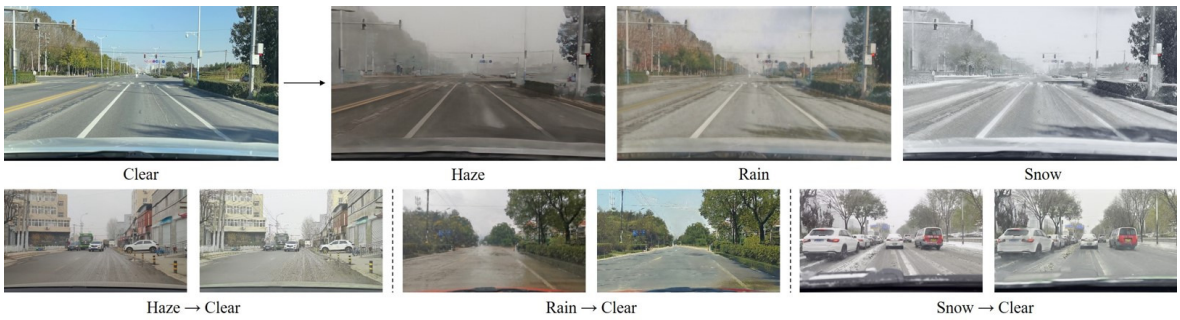


Figure 3.8: MWCG translates clear images into rainy, snowy, or foggy images using three different generators, creating a set of images representing different weather conditions (top row). In contrast, only one generator is needed to translate all three types of adverse weather images into clear ones (bottom row).

where G_A, G_B, and G_C try to generate images $G_A(D)$, $G_B(D)$, and $G_C(D)$ that look similar to images from domains x_1, x_2, and x_3 while D_A, D_B, and D_C aim to distinguish between the translated samples $G_A(D)$, $G_B(D)$, and $G_C(D)$ and real samples D, respectively. The base of the logarithm in the equation is usually set to 2 or e.

The transformation from adverse to clear weather involves three components, corresponding to each weather sub-domain. The mean values are calculated as:

$$\mathscr{L}_{\text{GAN}}(G_D, D_D, X, Y) = \frac{1}{3}\sum_{i=1}^{n} \mathscr{L}_{\text{GAN}}(G_D, D_D, x_i, Y), \tag{3.7}$$

where the $\mathscr{L}_{\text{GAN}}$ over x_i tries to enable G_D to generate better rain, snow, and fog images, while D_D needs to identify fake images after the generator is evolved.

3.3.4.2 Cycle Consistency Loss

The concept of "cycle consistency loss" is introduced by Zhu et al. [29], the paper proposing CycleGAN. It is calculated as the L1 norm between the input image and the reconstructed image and is used to prevent the second generator from generating random images of the target domain. An example of forward cycle consistency is shown in Fig. 3.8, where images of

each type of adverse weather is first translated into the "clear" domain before being restored to the original adverse weather images. This process can be formulated as:

$$\begin{aligned} A &\to G_D(A) \to G_A(G_D(A)) \approx A, \\ B &\to G_D(B) \to G_B(G_D(B)) \approx B, \\ C &\to G_D(C) \to G_C(G_D(C)) \approx C. \end{aligned} \tag{3.8}$$

Likewise, for backward cycle consistency, the clear image that is first translated into various weather domains should be restored to the same state as input:

$$\begin{aligned} D &\to G_A(D) \to G_D(G_A(D)) \approx D, \\ D &\to G_B(D) \to G_D(G_B(D)) \approx D, \\ D &\to G_C(D) \to G_D(G_C(D)) \approx D. \end{aligned} \tag{3.9}$$

To force the weather removal generator G_D to update at the same pace as the adverse weather generators, the average of the three cycle losses are calculated as the loss of G_D as:

$$\mathscr{L}_{\text{cyc}}(G,D) = \sum_{i=A}^{C} \mathbb{E}_{A\sim p_{\text{data}}(i)} \left[\|G_i(G_D(i)-i)\|_1\right] + \frac{1}{3}\sum_{j=1}^{C} \mathbb{E}_{D\sim p_{\text{data}}(D)} \left[\|G_D(G_j(D)-D)\|_1\right]. \tag{3.10}$$

3.3.4.3 Identity Loss

Identity loss is used to preserve the image color composition when applying painting transfer to realistic photo tasks. It is useful when dealing with large weather images that have obvious base color tones. The goal is to train the generator to learn to map the identities of the target domain images used as input. It can be expressed as:

$$\mathscr{L}_{\text{identity}}(G_A,G_B,G_C,G_D) = \sum_{i=A}^{D} \mathbb{E}_{i\sim p_{\text{data}}(i)} \left[\|G_i(i)-i\|_1\right]. \tag{3.11}$$

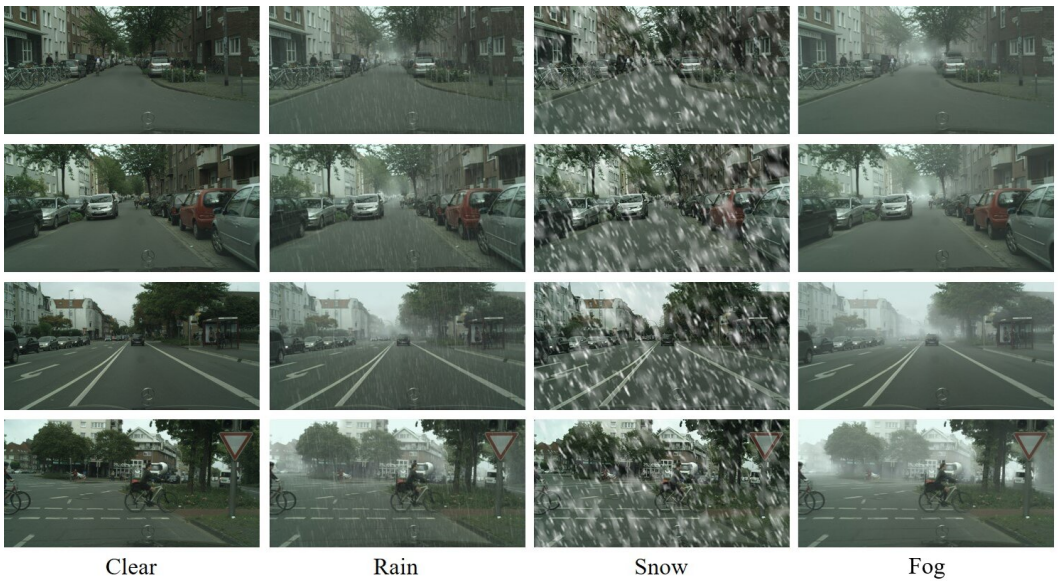


Figure 3.9: Sample scenes from the Cityscapes [1] and rearranged Cityscapes weather datasets. The latter combined images selected from Foggy Cityscapes [2], Rain Cityscapes [3], and Snow Cityscapes [4] datasets.

3.3.5 Experiment

This section evaluates the performance of the proposed extended conversion model using both the synthetic dataset and the full RDSBW dataset. In addition to assessing the conversion from adverse weather to clear conditions, conversions from clear to adverse weather conditions are also evaluated. Quantitative evaluation includes comparing the proposed method against single-weather removal techniques. Finally, the performance of pedestrian and object detectors on both the source and converted images is also assessed.

3.3.5.1 Implementation Details

The model uses the Pytorch framework for training, testing, and image preprocessing. Two NVIDIA RTX A6000 GPUs are used for training, with a batch size of 4. The MWCG model

is trained for 200 epochs on each dataset to ensure convergence, using the Adam optimizer [108] and a step learning rate schedule. In addition, the model set the λ_c and λ_i loss weights at 10 and the extra identity loss weight λ_{idt} at 2.

3.3.5.2 Datasets

To train the MWCG, Rain Cityscapes [3], Snow Cityscapes [4] and Foggy Cityscapes [2] are used to create a rearranged Cityscapes weather dataset, as shown in Fig. 3.9. The synthetic weather datasets use the same depth maps as the background more than once to simulate different weather intensities. Since low-intensity weather does not degrade visual applications very much and high-intensity weather occurs relatively infrequently, the experiment only uses the 300 meter foggy images from the Foggy Cityscapes dataset and 12 types of rain patterns from the Rain Cityscapes. To keep all the images in the training set at the same resolution, which is important to reduce domain difference, the Snow Cityscapes images are resized to $2,048 \times 1,024$ pixels using normal linear interpolation.

MWCG is also trained on RDSBW [121], with 4,171 rainy, 4,777 snowy, 2,052 fog and 2,831 clear images randomly selected in this experiment.

3.3.5.3 Experiment Results

Qualitative Evaluation Using RDSBW Data

The experiment first conducts a qualitative evaluation using MWCG with the RDSBW dataset. Samples of the weather generation results are shown in Fig. 3.10. MWCG can translate an unseen clear image into rain, snow, and fog images without changing the original background content. The proposed method seemed especially effective for adding rain and fog based on the following three aspects. First, the color of the image shifted based on the type of adverse weather effect being added. Second, the weather effects were similar

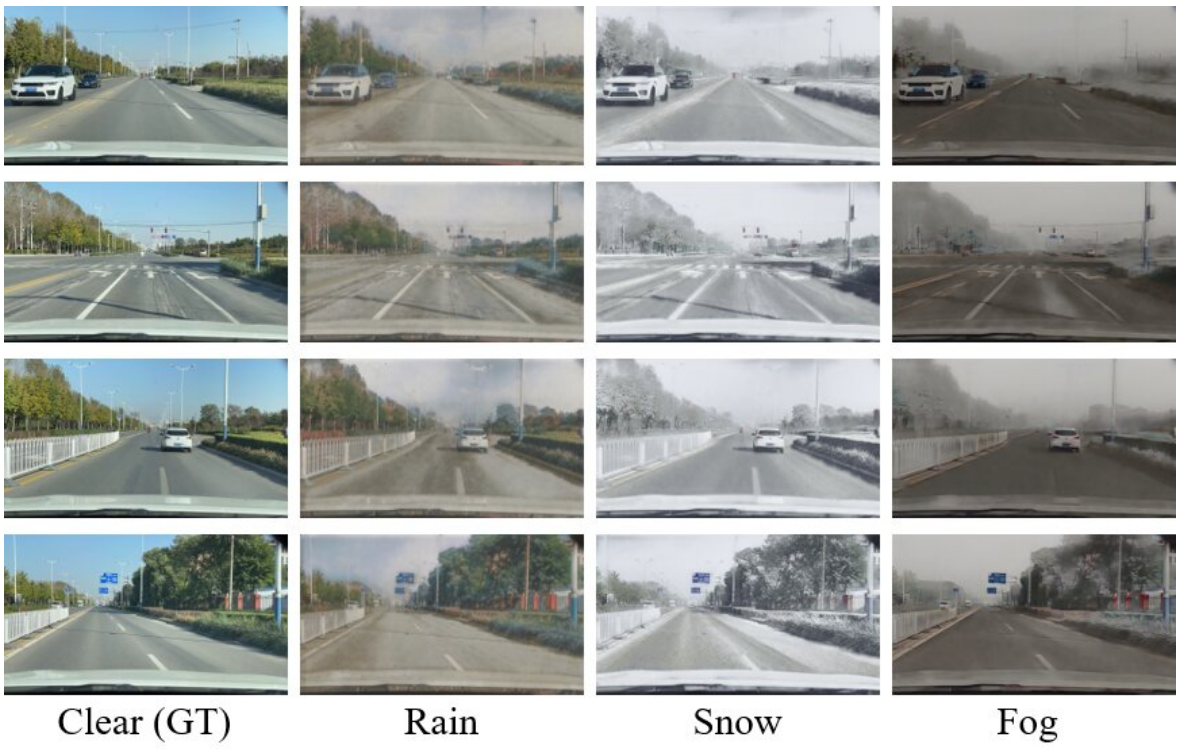


Figure 3.10: Weather generation results for RDSBW. Even when trained without paired image sets, MWCG still can translate clear images into images of three adverse weather conditions without corrupting the background content.

to those observed in real scenes, since MWCG did not simply add an extra layer to the input image but instead applied appropriately generated weather effects to each region of the image, to objects such as the sky, roads, and trees. Thirdly, MWCG was able to consider the semantic information. For example, the wires connecting the power and telephone poles were partly hidden under fog and the lane markings were covered by ice and snow under snowy. However, in the case of rain, the generated results were not ideal because the patterns in the rainy weather source images were not conspicuous enough for the model to learn them effectively. This problem could be addressed in the future by collecting more useful rainy weather image data.

Figure 3.11: Weather removal results for the RDSBW dataset, showing examples of MWCG's translation of adverse weather images into clear weather images. While MWCG was unable to recover objects and buildings hidden behind dense fog, it did not randomly insert fake objects.

Regarding the weather removal results, we can still observe accurate color transformation and realistic scene translation results, as shown in Fig. 3.11, but MWCG occasionally fabricated inputs, generating artifacts in some cases, most notably the insertion of fake grass in the middle of the road when removing rainy weather effects. This is due to limitations in the generation process and when weather effects are very extensive, the network was unable to

Figure 3.12: Weather removal results for the Cityscapes Weather datasets. Even if objects are occluded by fog, rain, or snowflakes, MWCG still can recover the original Cityscape images to generate clear images.

determine the original context without some guessing.

Qualitative and Quantitative Evaluation Using Cityscapes Weather Datasets

Next, Qualitative results are presented when using the Cityscapes Weather datasets. Since ground-truth images without any adverse weather phenomenon are included, quantitative results should be provided. As shown in Fig. 3.12, MWCG demonstrated high performance when removing weather effects during the qualitative experiment. This is because, in this setting, the only difference between the adverse and clear weather domains is the weather effects. Therefore, even though the data were unpaired during training, MWCG could still determine what is hidden behind the rain streaks, snowflakes, or fog and recover the original images. The generation performance of MWTG was also evaluated. The similarity between source images in Cityscapes Weather dataset and generated weather images is shown in Table 3.2.

Table 3.2: Generation Results on Cityscapes Weather dataset.

Type	Weather	PSNR↑	SSIM↑
Multi Task	Fog	21.09	0.92
	Rain	21.38	0.85
	Snow	19.02	0.68

Table 3.3: Comparison of image quality results with Rain Cityscapes dataset.

Type	Method	Venue	PSNR ↑	SSIM ↑
Specific Task	RCDNet [122]	CVPR2020	20.39	0.65
	MPRNet [123]	CVPR2021	20.10	0.68
	PReNet [124]	CVPR2019	20.48	0.66
	RESCAN [125]	CVPR2018	20.44	0.67
	Previous work [120]	VTC2022	22.46	0.89
Multi Task	MWCG (Proposed)	—	**25.16**	**0.91**

For quantitative evaluation, MWCG is compared with the State-Of-The-Art (SOTA) single weather removal methods, but only for their specific tasks. For de-fogging, the performance of the proposed MTWG method is compared with DehazeNet [126], Multi-Scale Convolutional Neural Network (MSCNN) [127], All-in-One Network for Dehazing (AODNet) [128], and GridDehazeNet [129]. For de-raining, MTWG was compared with Rain Convolutional Dictionary Network (RCDNet) [122], Multi-stage Progressive image Restoration Network (MPRNet) [123], Progressive Recurrent Network (PReNet) [124], and Recurrent Squeeze and Excitation Context Aggregation Net (RESCAN) [125].

Table 3.4: Comparison of image quality results with Snow Cityscapes dataset.

Type	Method	Venue	PSNR ↑	SSIM ↑
Specific Task	RESCAN [125]	ECCV2018	33.63	0.96
	SPANet [130]	CVPR2019	**35.73**	**0.97**
	DesnowNet [131]	TIP2018	33.58	0.94
	Previous work [120]	VTC2022	27.42	0.87
Multi Task	MWCG (Proposed)	—	25.23	0.86

Table 3.5: Comparison of image quality results with Foggy Cityscapes dataset.

Type	Method	Venue	PSNR↑	SSIM↑
Task Specific	DehazeNet [126]	TIP2016	14.97	0.49
	MSCNN [127]	ECCV2016	18.99	0.86
	AODNet [128]	ICCV2017	15.45	0.63
	GridDehazeNet [129]	ICCV2019	23.72	**0.92**
	Previous work [120]	VTC2022	**24.07**	**0.92**
Multi Task	MWCG (Proposed)	—	23.84	0.91

For de-snowing, MTWG is compared with RESCAN, Spatial Attentive Network (SPANet) [130], and DesnowNet [131]. Note that although MSCNN is listed in all the comparisons, it is still categorized as a single weather removal tool since it needs to be retrained for each removal task. In contrast, MWCG performs all these tasks using the same model.

PSNR and SSIM are used to compare the performance of each model when using the Cityscapes images. The results are shown in Tables 3.3~3.5.

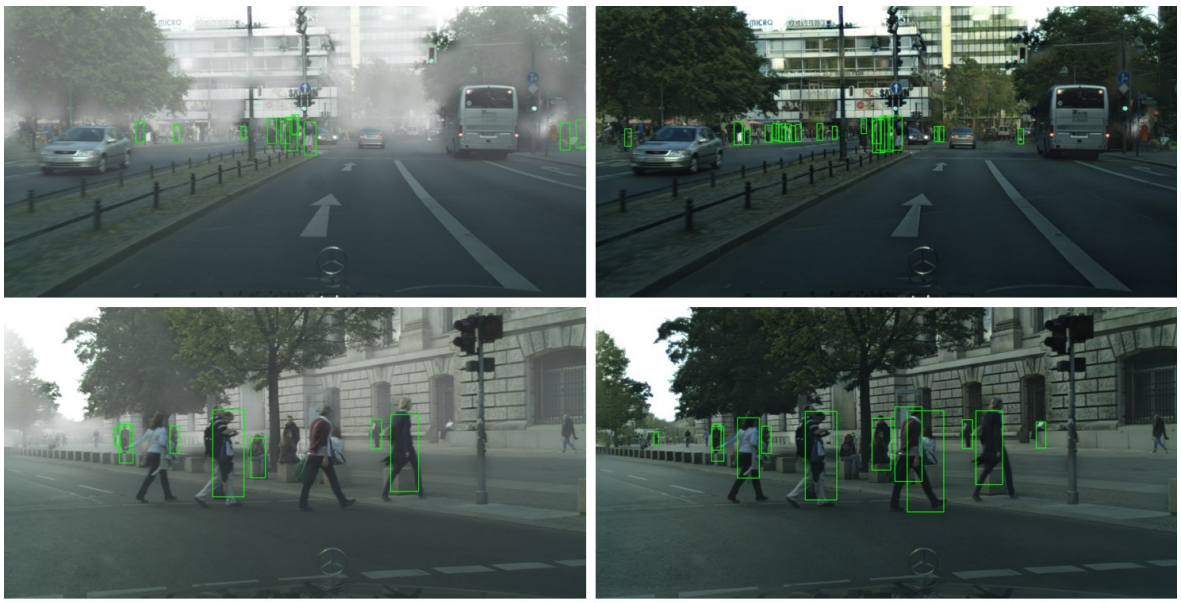


Figure 3.13: Application of MTWG on Foggy Cityscapes using a SOTA pedestrian detector ACSP.

From these results, we can see that MWCG achieved similar or better performance than the other de-raining or de-fogging methods, as measured using PSNR and SSIM. However, for the de-snowing task, MWCG was impaired by pixel resolution differences and, thus, did not achieve satisfactory performance. This is because the conventional methods were evaluated using the original Snow Cityscapes images, with an image resolution of 512×256 pixels, while these images were resized to match the resolutions of the other two datasets ($2,048 \times 1,024$ pixels) when training MWCG. Therefore, MWCG was dealing with images that are 16 times larger than the conventional methods.

Table 3.6: Log-average miss rate over False Positive Per Image (FPPI) results for ACSP pedestrian detector using Foggy Cityscapes dataset.

Data	Reasonable	Bare	Partial	Heavy
Foggy (before)	23.73%	16.32%	25.93%	58.70%
De-fogged (after)	20.65%	14.98%	21.30%	58.70%

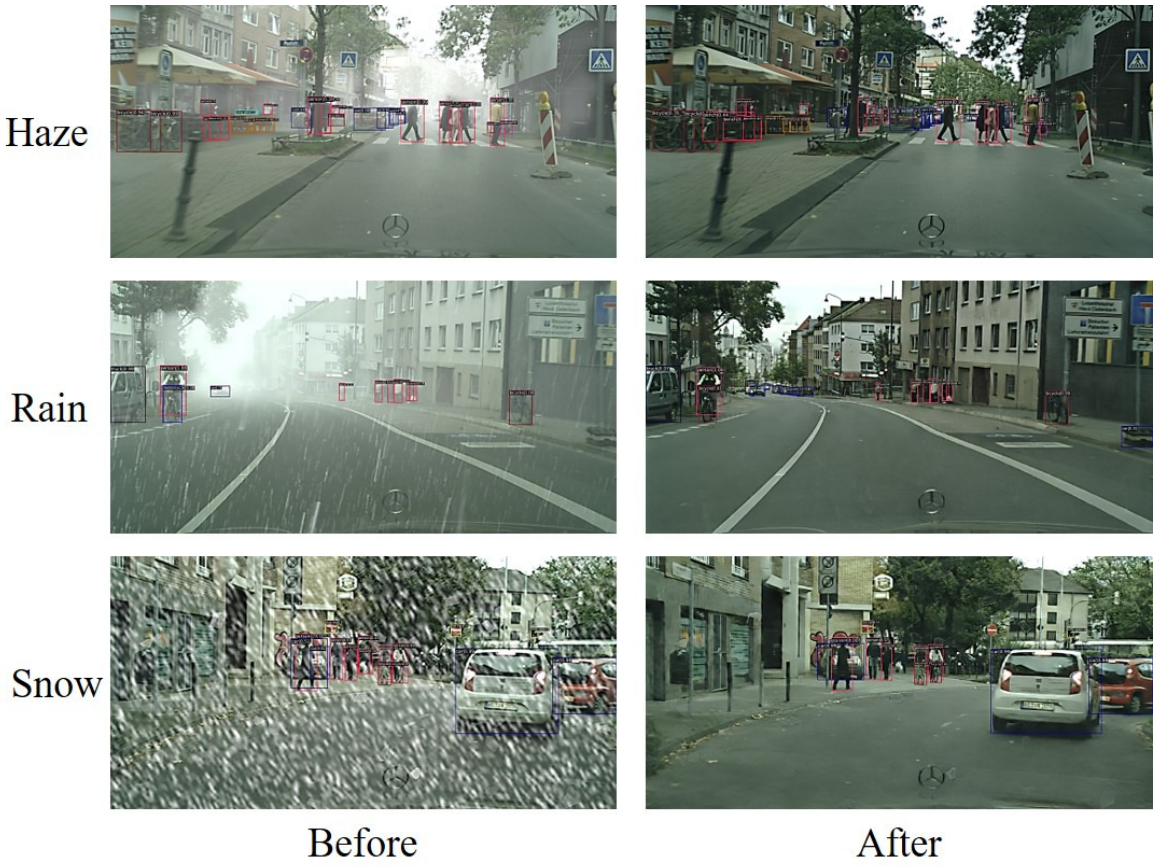


Figure 3.14: Application of MTWG on Weather Cityscapes using SOTA object detector Cascade-RCNN.

3.3.5.4 Evaluation Using Perception Algorithm

To verify the suitability of MWCG for visual applications, MWCG is tested using the SOTA pedestrian detector ACSP [113] on Foggy Cityscapes images. An example of ACSP detection applied to MWCG's de-fogging result is shown in Fig. 3.13. As we can see, the detection results clearly improved after the images were de-fogged using MWCG. For further investigation, ACSP was applied to the validation set of Foggy Cityscapes dataset. Results are shown in Table 3.6 in log-average Miss Rate over False Positive Per Image (FPPI).

A SOTA object detector was also applied, which uses Cascade-RCNN [14] as the backbone, on Cityscapes Weather datasets as shown in Fig. 3.14. We can observe the performance improvement in the detection numbers in different weather conditions.

3.3.5.5 Discussion

Based on the results of the evaluations, MWCG was confirmed to be able to translate images of multiple types of adverse weather into clear images since a constraint on cycle-consistency loss allowed the background context to remain unchanged. In this section, the capabilities and drawbacks of weather generation and removal using MWCG will be discussed in more detail.

In the experiment with the RDSBW dataset, the number of samples for each weather condition was unbalanced. In general, models would learn better conversion rules with more training samples. However, in the case of foggy weather, with only 2,000 training samples, the model was able to achieve satisfactory results. In contrast, in the case of images of rainy weather, even though the model was trained with 4,000 image samples, the translation results were less accurate. This is because rain creates more complex patterns in images. For example, streaks of rain in the air are spindly, so they are difficult for the camera to capture. Furthermore, rain is often accompanied by high humidity, thus there is often fog

in the background. When encountering finer and more varied distinctions, the proposed model tends to learn simpler representations, which is why the rainy images generated using MWCG were more similar to an intermediate output between fog and snow. We can also observe that MWCG's weather generation performance was superior to its weather removal performance. Since the CycleGAN model [29] is good at translation tasks involving color and texture changes, the proposed model MWCG based on it should inherit this ability. However, even though the proposed method intuitively considers the generation and removal of weather to be two separate tasks, the proposed model treats both as translation tasks. This means erasing noise, such as blocks of fog or flakes of snow from occluded objects, is not the primary target of the model but it is adding a layer of snow on the road or inserting a layer of fog in the distance, for example. Note that this difference in conversion performance is less obvious when using the Cityscapes dataset, where domain variance is minimal since the images are all synthesized using the same dataset.

3.4 Conclusions

The first work of this chapter proposed a disentangled adverse weather removal network for pedestrian detection. The model generated clear weather images from degraded images without extra physical parameters. Besides, the training process jointly learned the weather generation and removal in a disentangled manner. To keep weather information the same in the two generators, a weather layer loss was added to the CycleGAN [29] architecture. Further, an SFT layer was introduced to take the weather layer as information guidance and input to the reconstruct generator in each pipeline. Experimental results showed that the proposed model makes a SOTA detector such as ACSP [113] more effective on mixed Foggy CityScapes and RainCityScapes datasets.

In the second work of this chapter, a solution was explored to the visibility degradation

problem that AVs encounter when operating under adverse weather conditions, which can lead to malfunctioning of the perception module. The proposed, dual-purpose framework called MWCG was able to perform adverse weather generation and removal tasks simultaneously. In particular, the image translation model was trained using unpaired data. Three weather generators were used to create adverse weather effects on images of normal driving scenes obtained from video datasets, while a fourth clear weather generator was used to recover clear images by removing rainy, snowy, and foggy noise. To avoid translation deviation, a spatial feature transform layer was added to fuse the feature maps of the front-end network, as an information guide to the subsequent network.

A qualitative evaluation of MWCG using the RDSBW dataset and qualitative and quantitative evaluations using reorganized images from the Cityscapes and Cityscapes weather datasets showed that MWCG can achieve promising de-noising performance. Moreover, the results of a practical experiment showed that the proposed model boosted the performance of SOTA pedestrian detector ACSP when tested using the Foggy Cityscapes images.

4 Realistic Image Conversion

4.1 Introduction

This chapter focuses on enhancing the realism of image conversions for autonomous driving scenes, with an emphasis on achieving photo-realistic results. Using segmentation maps as additional input, the model gains a deeper understanding of different elements in the images, such as roads, vehicles, and buildings. This allows for more accurate maintenance of object integrity during the conversion process, ensuring each element is correctly represented under different weather conditions.

The importance of achieving photo-realistic image conversions cannot be overstated. For Autonomous Vehicles (AVs), the ability to interpret and react to their surroundings is crucial for safe navigation. Photo-realistic images provide a level of detail and accuracy that is essential for training AVs' perception systems. By experiencing a wide range of realistic scenarios, the vehicles can learn to recognize and respond to various environmental conditions more effectively.

Deep supervision is another technique employed in this chapter. It involves adding guidance to the intermediate layers of the generator within the image conversion model. This method improves the extraction and transformation of features in the images, leading to conversions that are not only realistic but also semantically accurate. The end result is that the model produces images that are not just visually convincing but also represent the scene's elements in a way that closely mimics real life.

By combining the use of semantic images and deep supervision, the model enhances its capability to create images that are indistinguishable from real photographs. This is vital for AVs, as their safety and reliability depend on accurate and realistic visual information from their surroundings.

4.2 Proposed Methods

This chapter shows that instead of removing noise from weather images, models based on CycleGAN [29] have more potential in synthesizing weather effects, even though they both are a subset of style transfer. When it comes to the task of removal, the generator will inevitably introduce artifacts to the original images, which are not suitable for subsequent perception algorithms. On the other hand, for synthesis, this tendency will help generate more natural scenes, such as the haze in the distance caused by snowflake accumulation. Additionally, this research suggests that by synthesizing weather effects with CycleGAN-based models, the proposed model can improve the diversity and realism of the images used for training perception models, particularly those designated to operate under adverse weather conditions. In addition, CycleGAN designs to handle larger images, e.g. $2,048 \times 1,024$ pixels resolution in the Cityscapes [1] dataset. Therefore, the proposed snow synthesis model takes CycleGAN as the backbone.

Two novel modules are further introduced: multi-modality module and deep supervision module. In the multi-modality module, an additional segmentation map is fed to the generator. The reason is to enable the generator to learn different translations across different regions. In detail, crop patches in the same position as input RGB images of the segmentation map are first taken. Then, a feature fusion technique called Spatial Feature Transform (SFT) [106] is employed to carry out deep fusion with the extracted RGB features. The reason is twofold: to achieve better optimization and to increase robustness to changes in

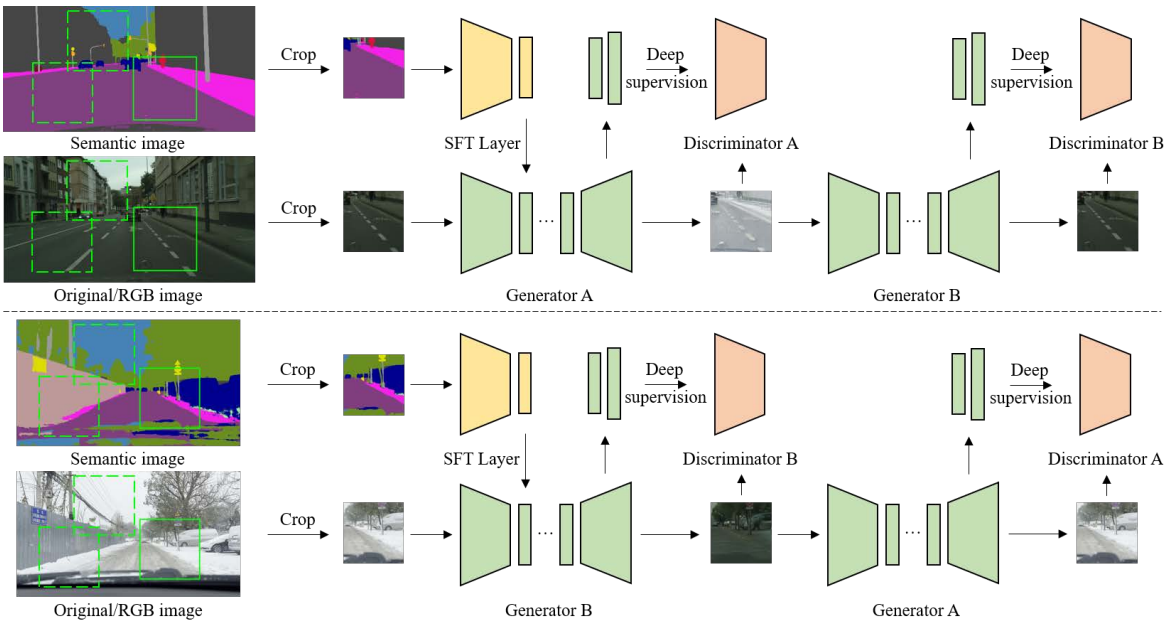


Figure 4.1: Algorithm of the proposed snow synthesis method.

scale. Meanwhile, in the deep supervision module, two extra side outputs are added to the discriminator. Together with the final output, these three feature maps are scaled down at the same proportion and represent different perception fields. The architecture of the proposed snow synthesis method is shown in Fig. 4.1.

4.2.1 Multi-modality Input with Segmentation Map

Physical snow simulation methods take into account the fact that snowflakes will be affected by wind and snow particles will accumulate and cause light scattering at a distance. While the deep learning model can mimic the dynamic behavior of snow, it is difficult to generate distance-dependent fog in various regions.

A segmentation map is used to represent multiple semantic segments in an image, which can simplify the process of identifying and analyzing the different regions. Incorporating

this semantic information into the generator can help the simulation to generate snow more accurately according to different regions, such as the sky, trees, and roads.

Training a high-quality semantic segmentation model from scratch can be a time-consuming and resource-intensive task. Instead, off-the-shelf models are usually more robust because they are trained on a large amount of data and have been fine-tuned to a variety of different tasks. Therefore, a State-Of-The-Art (SOTA) segmentation method Vision Transformer (ViT)-Adapter [132] is employed to calculate the segmentation maps used in this work.

By utilizing SFT, the proposed model can effectively fuse the segmentation map in a spatially-aware manner. The SFT layer learns a mapping function M, which outputs a modulation parameter pair (γ, β) based on a set of prior conditions Ψ. These learned parameters adaptively influence the outputs by applying an affine transformation to the intermediate feature maps in a spatial manner.

Once the (γ, β) parameters are obtained from the conditions, the transformation is executed by scaling and shifting the feature maps of a specific layer as:

$$\mathrm{SFT}(F \mid \gamma, \beta) = \gamma \odot F \oplus \beta, \tag{4.1}$$

where $\odot$ and $\oplus$ indicate element-wise product and summation, respectively.

4.2.2 Deep Supervision

Deep supervision is a concept that involves adding supervision to the intermediate layers of a neural network. The added supervision allows the network to learn more discriminative features and improve its performance by mitigating the gradient vanishing problem. This enables the creation of deeper networks and more efficient learning. During the forward propagation phase, deep supervision does not alter the flow of information through the network. The overall loss is a combination of the loss at the final output layer and the intermediate

supervision loss defined as:

$$\mathscr{L} = \mathscr{L}_L + \alpha \mathscr{L}_K, \tag{4.2}$$

where $\mathscr{L}_L$ is the final layer, $\mathscr{L}_K$ is the K^{th} layer, and α is the weight.

In this case, the last three convolutional layer outputs of the generator are input to the discriminator. It is the upsampling stage that restores the spatial size of extracted features to its original size.

4.2.3 Loss Functions

Three kinds of loss functions are used when formulating the proposed snow synthesis model: adversarial loss $\mathscr{L}_{\text{GAN}}$, cycle consistency loss $\mathscr{L}_{\text{cyc}}$, and identity loss $\mathscr{L}_{\text{identity}}$.

The overall objective function is formulated as:

$$\mathscr{L}_{\text{obj}} = \lambda_a \mathscr{L}_{\text{GAN}} + \lambda_c \mathscr{L}_{\text{cyc}} + \lambda_i \mathscr{L}_{\text{identity}}, \tag{4.3}$$

where λ_a, λ_c, and λ_i are weights that control each loss.

4.2.3.1 Adversarial Loss

Adversarial loss $\mathscr{L}_{\text{GAN}}$ is used to obtain two mappings, clear to snow and recover. The mappings can be expressed as:

$$\mathscr{L}_{\text{GAN}}(G_C, D_S, C, S) = \mathbb{E}_{s \sim p_{\text{data}}(s)}[\log D_S(s)] + \mathbb{E}_{c \sim p_{\text{data}}(c)}[1 - \log D_S(G_C(c))], \tag{4.4}$$

$$\mathscr{L}_{\text{GAN}}(G_S, D_C, S, C) = \mathbb{E}_{c \sim p_{\text{data}}(c)}[\log D_C(c)] + \mathbb{E}_{s \sim p_{\text{data}}(s)}[1 - \log D_C(G_S(s))], \tag{4.5}$$

where C indicates the clear domain, S indicates the snow domain and related low case indicates specific sample. G_C tries to generate images that look similar to snow scenes while D_S aims to distinguish between translated samples $G_C(c)$ and real samples S. In the opposite

direction, the target is changed from snow to clear. The base of the logarithm in the equation is usually set to 2 or e.

4.2.3.2 Cycle Consistency Loss

The cycle consistency loss function compares the output of a forward translation (e.g., clear to snow) with the output of a backward translation (e.g., snow to clear). The goal is for the output of the backward translation to be similar to the original input, ensuring that the network has learned a meaningful translation rather than simply memorizing the training data. The loss $\mathscr{L}_{\text{cyc}}$ is calculated as the L1 or L2 distance between the original input and the output of the backward translation as:

$$\mathscr{L}_{\text{cyc}} = \mathbb{E}_{s\sim p_{\text{data}}(s)}[\|G_C(G_S(s)) - s\|_1] + \mathbb{E}_{c\sim p_{\text{data}}(c)}[\|G_S(G_C(c)) - c\|_1]. \tag{4.6}$$

The lower this distance is, the better the network is at maintaining the original content of the image.

4.2.3.3 Identity Loss

Identity loss $\mathscr{L}_{\text{identity}}$ is used to preserve image color composition when applying painting transfer to realistic photo tasks. It is also useful when dealing with large weather images that have obvious base color tones. The goal is to train the generator to learn to map the identities of the target domain images used as input. It is expressed as:

$$\mathscr{L}_{\text{identity}}(G_C, G_S) = \mathbb{E}_{s\sim p_{\text{data}}(s)}[\|(G_S(c)) - c\|_1] + \mathbb{E}_{c\sim p_{\text{data}}(c)}[\|(G_C(s)) - s\|_1]. \tag{4.7}$$

Figure 4.2: Examples of real snow dataset from the RDSBW dataset.

4.3 Experiment

This section conducts qualitative and quantitative evaluation of the proposed conversion method with respect to snow synthesis. SOTA conversion methods are used to make comparisons and common image quality metrics are calculated. Further, pedestrian detection is applied to images before and after conversion.

4.3.1 Implementation Details

The model uses ResNet [109] as the generation network, including two layers of downsampling, three layers of identity mapping, and two layers of upsampling. Each image before input to the generator is randomly cropped into 256×256 patches. The corresponding segmentation map is cropped at the same position. For the discriminator, the model uses a three-layer convolutional network to extract a 128-dimension embedding.

The model is trained using the AdamW [133] optimizer with an initial learning rate of 0.0002. The batch size is set to 16 for training 200 epochs without segmentation maps and full training of 100 epochs. The parameters λ_a, λ_c, and λ_i are set to 1, 10, and 1 respectively.

4.3.2 Datasets

To provide a realistic learning target, the snow set of Realistic Driving Scenes under Bad Weather (RDSBW) dataset introduced in 3.2.3.2 is used. High-quality images are picked out and resized to a resolution of 960×540 pixels. The final snow dataset contains 6,814 images and a few examples in various scenes are presented in Fig. 4.2.

The Cityscapes [1] and EuroCity Persons [9] datasets are used as the source of the clear set. The latter is divided into daytime and nighttime sets with over 47,300 images. To make a comparable match to the snow dataset, the experiment randomly selected 5,921 images in the daytime training set.

4.3.3 Results

Fig. 4.3 presents results of snow synthesis on the Cityscapes and EuroCity Persons datasets. Compared to CycleGAN-based methods, one-way methods including Fast Contrastive Unpaired Translation (FastCUT) [134] and Multimodal Unsupervised Image-to-Image Translation (MUNIT) [135] lack a bijection relationship between the two domains. The lightweight FastCUT only preserves the structure as we can not observe snow effects but more of a whitening process because maximizing mutual information alone is not enough for changing the appearance of the snow synthesize task. In contrast, CycleGAN completely changes the appearance of snow scenes while maintaining structure and texture, but the far-end snow representation is vague and resembles dense fog, particularly in the sky region. The proposed method, using the information from segmentation maps, learns to generate snow by region and can preserve building outlines in distance.

Given the fact that the subjects of evaluation were synthesized snow images without ground truth, the experiment selected Peak Signal-to-Noise Ratio (PSNR), Structural Simi-

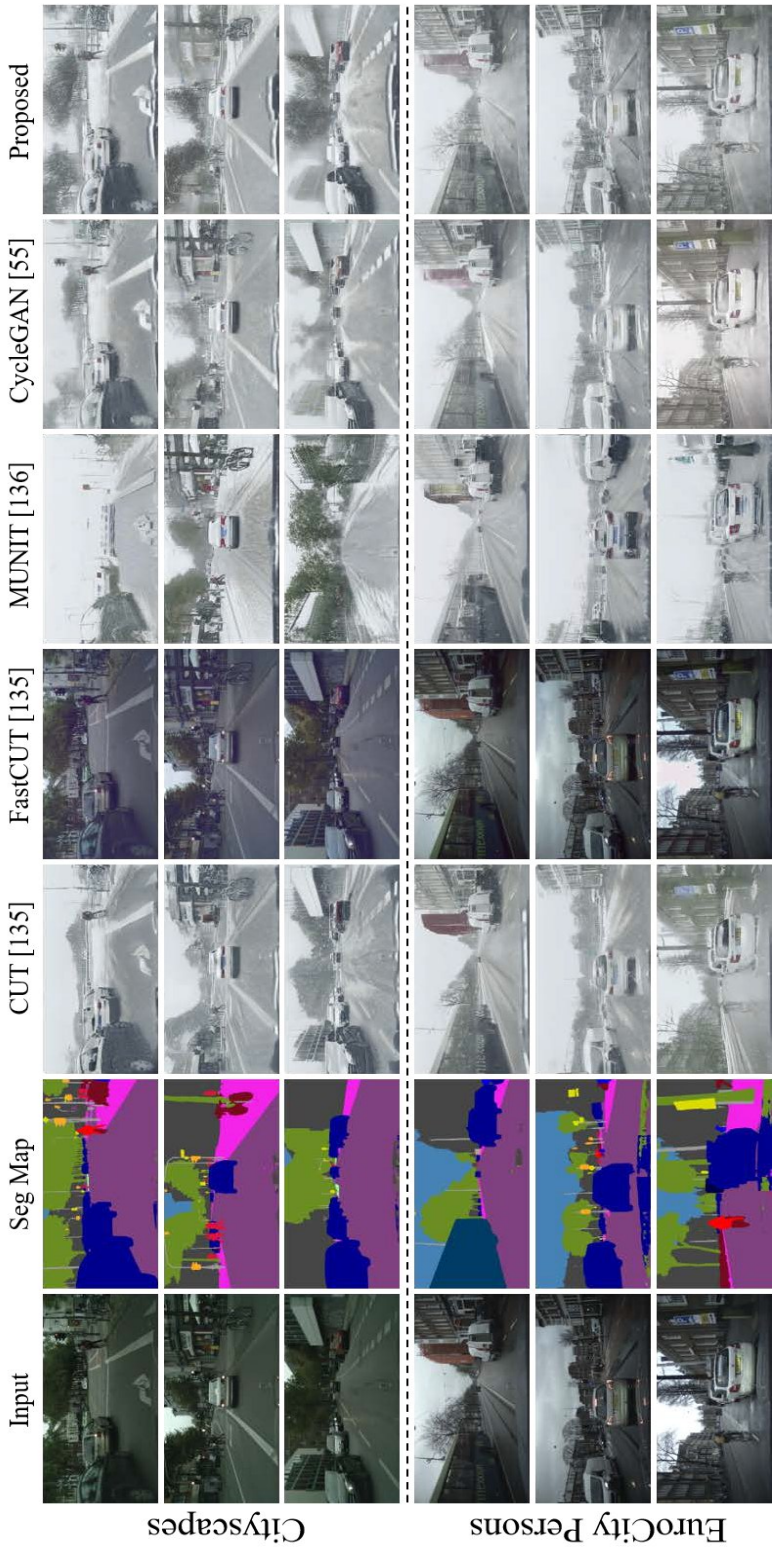


Figure 4.3: Qualitative results of snow image synthesis.

Table 4.1: Image quality metrics between synthesized and real images on Cityscapes dataset.

Method	**PSNR↑**	**SSIM↑**	**FID↓**
CUT [134]	7.04	0.37	247.22
CycleGAN [29]	7.10	0.40	217.24
MUNIT [135]	6.07	0.26	258 .57
Proposed	**7.32**	**0.43**	**196.18**

larity Index (SSIM), and Fréchet Inception Distance (FID) [136] as quantitative evaluation metrics to measure the similarity between the snow and original images. In particular, higher PSNR indicates the robustness of the representation against corrupting noise, while higher SSIM signifies a close similarity in structural information between the snow and original images, and lower FID shows proximity in data distribution. 100 generated fake images are randomly selected for testing.

FID calculates the Fréchet distance between two sets of images based on the features extracted by the inception network [137]. It provides a measure of similarity between the generated images and their respective benchmarks. A lower FID value suggests that the generated images closely resemble the benchmark images.

The other metrics were introduced in 3.2.3.3.

Tables 4.1 and 4.2 show the results of similarity between the synthesized and real images. The high values of PSNR and SSIM not only suggest good verisimilitudes of the proposed synthesized snow images but also lower noises. What is more, the obvious lower FIDs of the proposed model state that the data distribution between the target and source is relatively close. The overall statistics demonstrate strong capability on the snow synthesis of the proposed model and superior performance over other SOTA methods.

Table 4.2: Image quality metric between synthesized and real images on EuroCity Persons dataset.

Method	**PSNR↑**	**SSIM↑**	**FID↓**
CUT [134]	8.78	0.45	206.37
CycleGAN [29]	8.38	0.50	175.74
MUNIT [135]	7.66	0.33	235.40
Proposed	**9.35**	**0.54**	**138.43**

An interesting discovery is that the EuroCity Persons dataset captures scenes under cloudy weather with wet roads, providing a better match to the snow dataset compared to the Cityscapes dataset. This highlights the importance of controlling environmental variables when dealing with large-sized images for improved translation results.

4.3.4 Impact on Object Detection Performance

In addition to the qualitative and quantitative analyses previously conducted, a new set of experiments is carried out to evaluate the impact of the proposed realistic image conversion method on object detection algorithms. These experiments are specifically designed to assess how the addition of synthetic snow to the original Cityscapes dataset and EuroCity Persons images affects the performance of standard object detection models as shown in Figs. 4.4 and 4.5.

Figure 4.4: Results of applying SOTA object detector Cascade RCNN on the source and converted images of EuroCity Persons dataset.

To conduct this experiment, a SOTA object detection algorithm is applied to two sets of images: the original Cityscapes images and their counterparts with realistically converted snow scenes, same for the EuroCity Persons dataset. The primary objective is to evaluate change in the detection accuracy due to the altered environmental conditions, specifically the presence of snow.

Figure 4.5: Results of applying SOTA object detector Cascade RCNN on the source and converted images of CityScapes dataset.

The results revealed a noticeable decline in the object detection performance on the snow-converted images compared to the original Cityscapes dataset. The decline can be attributed to the added complexity and visual changes introduced by the snow, which create new challenges for the detection models. These challenges include reduced visibility of objects, alterations in their appearance due to snow coverage, and changes in lighting and contrast.

This experiment highlights the realistic nature of the snow conversion in the proposed method, as it accurately mimics the real-world difficulties that autonomous driving systems

Table 4.3: Comparison of log average miss rate (↓) of Adapted Center and Scale Prediction (ACSP) with converted snow images as additional training data.

Training data	**Reasonable**	**Bare**	**Partial**	**Heavy**
Cityscapes	14.80%	10.32%	14.69%	51.38%
Cityscapes + synthetic snow images	**13.08%**	**8.13%**	**14.50%**	**50.39%**

face in snowy conditions. The results underscore the need for further refinement of object detection algorithms to adapt to diverse weather conditions, emphasizing the importance of training these models on a wide range of environmental scenarios, including those with adverse weather elements like snow.

4.3.5 Enhanced Pedestrian Detection with Synthetically Augmented Data

As a further extension of this research, another set of experiments is conducted to assess the effectiveness of using synthetically generated snow images as additional training data for pedestrian detection algorithms. Specifically, the experiment compares the performance of the Adapted Center and Scale Prediction (ACSP) algorithm [113] trained solely on the Cityscapes dataset against the same algorithm trained with an augmented dataset that includes the realistically converted snow images.

This experiment focuses on evaluating pedestrian detection accuracy under various levels of occlusion, categorized as Reasonable, Bare, Partial, and Heavy. The key metric used to measure performance is the log-average miss rate, a standard benchmark in pedestrian detection studies.

The results as shown in Table 4.3 were significant: ACSP trained with the augmented dataset (Cityscapes plus synthetic snow images) showed a marked improvement in detection

accuracy across all occlusion categories compared to ACSP trained only on the original Cityscapes dataset. The log-average miss rate was notably lower for the algorithm trained with the additional synthetic snow data. This improvement indicates that the inclusion of diverse weather conditions, like snow, in the training dataset can enhance the algorithm's ability to detect pedestrians in challenging visibility conditions.

These findings demonstrate the value of synthetic data augmentation in training more robust pedestrian detection algorithms. By exposing the ACSP algorithm to a wider range of scenarios, including adverse weather conditions simulated by the proposed realistic snow conversion, the model should become better equipped to handle real-world challenges. This is particularly relevant for autonomous driving systems, where reliable pedestrian detection under various environmental conditions is crucial for ensuring safety.

4.4 Conclusion

This chapter proposed a novel snow synthesis model for generating realistic snow scenes on existing driving datasets. The proposed model was based on cycle-consistent adversarial networks, which learn the mapping between two domains. To address the problem of artifacts and inadequate scene understanding, a multi-modality input was introduced by incorporating segmentation maps. The additional information was deeply fused with the previously extracted feature maps to improve the performance of the model. Additionally, deep supervision was used to govern the generation of features at different scales, further enhancing the realism of the generated snow images. The proposed model was evaluated with typical image quality metrics on the Cityscapes and EuroCity Persons datasets and its strong ability to generate realistic snow images with respect to the image quality metric and the impact on object detection algorithms were demonstrated. The experiment also validated that with an augmented dataset that includes the realistically converted snow images, the pedestrian

detector can achieve better performance on log-average miss rate.

5 Controllable Image Conversion

5.1 Introduction

Controllability in image conversion is a critical aspect, especially in applications like autonomous driving where precision and customization are key. This chapter introduces an innovative approach in image conversion, focusing on enhancing this controllability. By gaining precise control over the stylistic and content aspects of image conversions, the proposed method can tailor the outputs to fit specific environmental and operational requirements, which is crucial for the effective training and performance of Autonomous Vehicles (AVs).

In this advanced approach, the generator within the image conversion model is divided into a style encoder and a content encoder. This separation enables more nuanced control over both the aesthetic and structural elements of the images. The style encoder is responsible for capturing and manipulating the visual style of the image, such as its color scheme and texture. To maintain consistency in style, a self-regression module is applied to the style latent space, ensuring that the style variations remain within acceptable limits and do not detract from the image's realism.

Conversely, the content encoder focuses on the scene's semantic and structural aspects, such as the layout and objects within the image. Enhancing the content encoder is a content feature discriminator, a tool that strengthens the content-related attributes of the converted images. This ensures that key elements of the original scene are preserved and accurately

depicted after conversion, a necessity for autonomous driving systems that rely on precise visual information for navigation and decision-making.

The combination of a style encoder and a content encoder in the image conversion process allows for an unprecedented level of control. This controllability is essential for producing conversions that are not only realistic and accurate but also tailored to specific requirements of style and content. This chapter will detail these technical components and their integration, highlighting how they collectively enhance the controllability of the image conversion process.

To synthesize realistic snow on the driving datasets, the proposed method focuses on the Generative Adversarial Networks (GANs) with cycle consistency. The goal is to learn the mapping between the snow domain and the clear weather domain. In previous unpaired image conversion methods [120, 138], two generators were employed to transfer images into the expected domain. Two corresponding discriminators were employed to differentiate real images and fake images. The cycle consistency ensured that translated images can be reconstructed into original input images.

Recently, when researchers use similar methods for weather removal or synthesis, they follow an assumption that weather images can be decomposed into content partition and weather partition [139, 140, 141]. The partition could be any mathematical format, such as vectors or tensors. In general image translation tasks, the weather partition refers to the style representation. This technique will disentangle the translation process and preserve the structural feature of the background. Therefore, the model follows the assumption and splits the generator into three networks: style encoder, content encoder, and decoder.

In the field of representation learning, incomplete disentanglement is often more prevalent. This concept suggests that images from varying domains have a shared content representation space, but the style representation space remains unique to each domain. This idea is

also known as the shared latent space assumption. In this task, the style is related to snow, and different classifications detail the attributes of weather events.

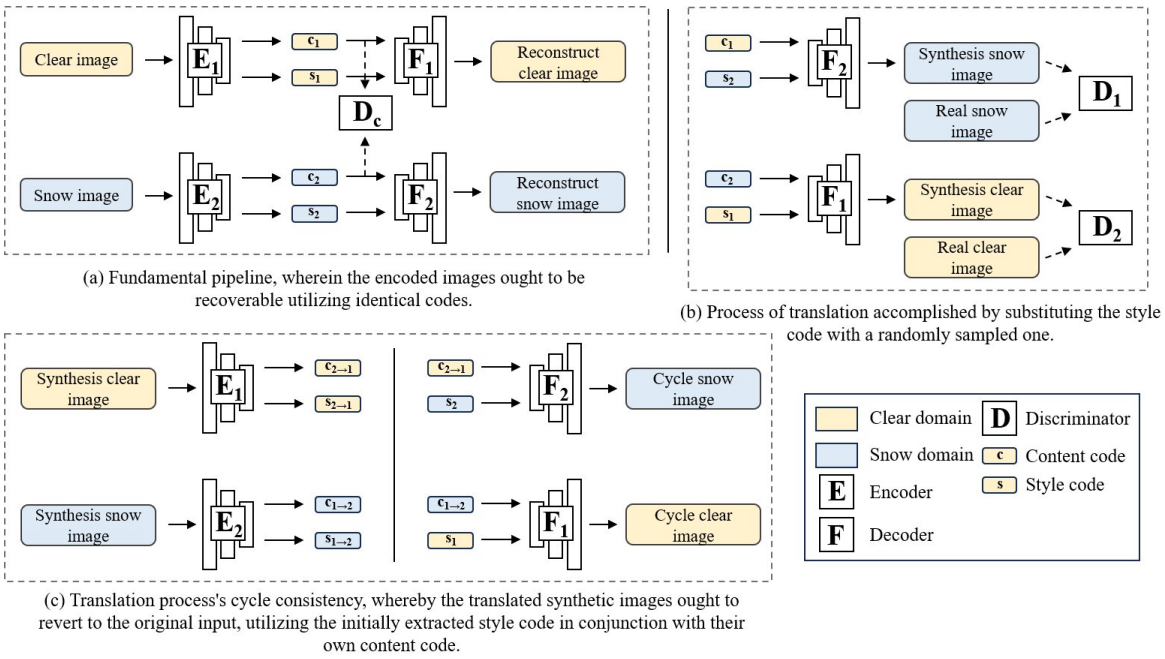
Intuitively, the content codes and style codes should be disjoint in the representation space. To better achieve representation disentanglement, a content discriminator is applied to distinguish the domain membership of the encoded content features. The goal is to force content encoders to generate features that cannot be identified, which means the content code does not contain style details.

Further, to make the degree of synthesized snow controllable, it is needed to explore the space S of style partition. Inspired by the work of Zhang et al. [139], the proposed model converts the snow domain into a continuous space by associating the style code vectors with a linear manipulation. With the help of the content discriminator, the style code will not contain information on image attributes. Ideally, the interpolated style code should represent an intermediate snow density.

5.2 Controllable Unsupervised Snow Synthesis

5.2.1 Fundamental Basis

To illustrate the framework of the proposed Controllable Unsupervised Snow Synthesis (CUSS) method in an intuitive way, suppose that $x_1 \in X_1$ and $x_2 \in X_2$ are images from the clear domain and the snow domain, respectively. In statistics, the images belong to two marginal distributions, $p(x_1)$ and $p(x_2)$. The joint distribution $p(x_1, x_2)$ is inaccessible due to lack of paired data. The goal is to learn an image conversion model that can estimate two conditionals probabilities, $p(x_{1\to2}|x_1)$ and $p(x_{2\to1}|x_2)$, where $x_{1\to2}$ is a sample of synthesized snow images and $x_{2\to1}$ is a sample of synthesized clear images (recovered from real snow samples). In general, the synthesized outputs do not fall into a single mode.



(a) Fundamental pipeline, wherein the encoded images ought to be recoverable utilizing identical codes.

(b) Process of translation accomplished by substituting the style code with a randomly sampled one.

(c) Translation process's cycle consistency, whereby the translated synthetic images ought to revert to the original input, utilizing the initially extracted style code in conjunction with their own content code.

Figure 5.1: Architectural design of the proposed Controllable Unsupervised Snow Synthesis (CUSS) network. Solid arrows show the forward process of the generators and dashed arrows show the input to the discriminators.

To obtain other possible solutions, the partially shared latent space assumption from Multimodal Unsupervised Image-to-Image Translation (MUNIT) [135] is adopted to produce diverse snow effects. This theory posits that each image $x_i \in X_i$ originates from a content latent code c_i, shared across both domains, and a unique style latent code s_i tied to its respective domain. For snow synthesis, a matching pair of clear and snow images (x_1, x_2) from the combined distribution is created by $x_1 = F_1(c_1, s_1)$ and $x_2 = F_2(c_2, s_2)$, with F_1 and F_2 as the foundational generators with the inverse encoders E_1 and E_2, with $E_1 = (F_1)^{-1}$ and $E_2 = (F_2)^{-1}$.

Algorithm 2 Controllable Unsupervised Snow Synthesis (CUSS)

Input: Training data pairs (X_1, X_2) ▷ In order of clear and snow

Output: Encoders E_1, E_2, Decoders F_1, F_2 ▷ F_1 generate clear images, F_2 generate snow images

1: Initialize encoders, decoders, and discriminators
2: Define loss functions
3: Define optimizers for generator and discriminator
4: **while** $epoch \leq total\_epoches$ **do**
5: **for** data pair (X_1, X_2) **in** data_loader **do**
6: Get content codes and style codes of input images: $(c_1, s_1) = E_1(x_1)$, $(c_2, s_2) = E_2(x_2)$, $(s_{n1}, s_{n2}) \sim \mathcal{N}(0,1)$ ▷ s_{n_i} means style code sampled from normal distribution
7: Generate fake images: $x_{1\to2} = F_2(c_1, s_{n1})$, $x_{2\to1} = F_1(c_2, s_{n2})$
8: Generate reconstruct images: $x_{1\to1} = F_1(c_1, s_1)$, $x_{2\to2} = F_1(c_2, s_2)$
9: Get content codes and style codes of fake images: $(c_{21}, s_{21}) = E_1(x_{2\to1})$, $(c_{12}, s_{12}) = E_2(x_{1\to2})$
10: Generate cycle translation images: $x_{1\to2\to1} = F_1(c_{12}, s_1)$, $x_{2\to1\to2} = F_2(c_{21}, s_2)$
11: **Update** Discriminator D_1, D_2, D_c
12: **Update** Generator E_1, E_2, F_1, F_2
13: **end for**
14: **end while**

The structure of the CUSS model is depicted in Fig. 5.1. As displayed in Fig. 5.1 (a), the proposed conversion model has an encoder E_1 and a decoder F_1 for the clear domain X_1, and an encoder E_2 and a decoder F_2 for the snow domain X_2. Each image fed into the encoder becomes converted into a content code c and a style code s, represented as $E(x) = (c, s)$. The translation between images occurs by interchanging encoder-decoder pairs, as depicted in

Fig. 5.1 (b). For instance, to transform a clear image $x_1 \in X_1$ to X_2, the CUSS first captures its content latent code $c_1 = E_1^c(x_1)$ and draws a style latent code s_2 from the normal distribution $q(s_2) \sim \mathcal{N}(0, I)$. Then, if employs F_2 to generate the ultimate snow image $x_{1\to2} = F_2(c_1, s_2)$.

Earlier research [142] harnesses the cycle consistency loss [29], measured by the L1 norm of the input image. This aims to deter the secondary generator from producing arbitrary target domain images. However, Huang et al. [135] demonstrates that if cycle consistency is imposed, the translation model becomes deterministic. As a result, a style-enhanced cycle consistency is integrated into the image-style joint spaces, which aligns better with multi-modal image conversion. As illustrated in Fig. 5.1 (c), the CUSS derives the content code $c_{1\to2}$ and style code $s_{1\to2}$ from the synthetic snow image $x_{1\to2}$. Then the content code $c_{1\to2}$ and the identical style latent code s_2 are fed to the clear decoder F_1. The result image is named cycle clear image $x_{1\to2\to1}$. The idea behind style-enhanced cycle consistency is that by translating an image to a target domain and then back to the original style, the model should retrieve the initial image. The CUSS does not apply explicit loss measures to ensure this style-enhanced cycle consistency, but it is suggested by the bidirectional reconstruction loss. The pseudo-code of CUSS is shown in Algorithm 2.

5.2.2 Disentanglement of Content and Style

A disentangled representation captures the underlying structure of the data so that individual factors can be modified independently without affecting others. The goal is to achieve complete disentanglement, where both content and style features are extracted independently. To achieve this, a content discriminator D_c is used to remove style information from the content feature. At the same time, self-supervised style coding is used to reduce content information from the style feature.

To enhance the content encoder, the content feature discriminator proposed by Lee et al. [143]

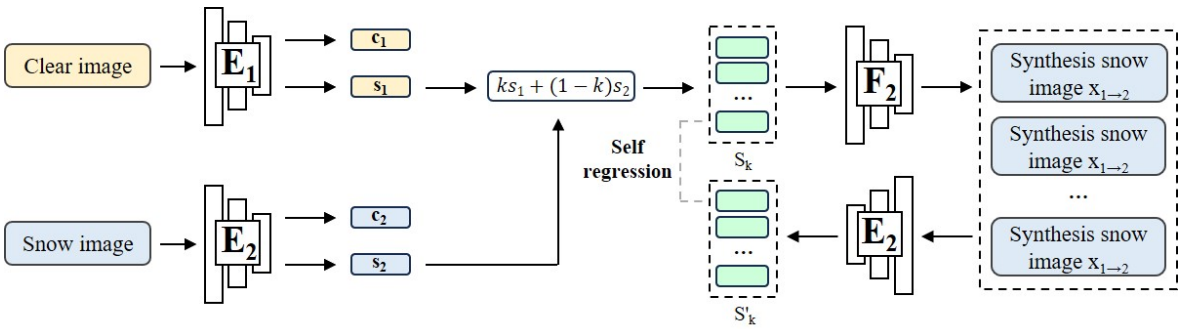


Figure 5.2: Snow sizing through self-regression style coding.

is employed. Initially, the content encoder extracts content codes, denoted as c_1 and c_2, from the respective inputs x_1 and x_2. The content discriminator D_c takes input images and classifies their source domain. Then one objective of the content encoder is to deceive D_c with distinct features. As a result, the content encoder, and discriminator refine each other through adversarial training. Once equilibrium is reached, the extracted content features no longer retain any stylistic information about the image.

When this game of generators and discriminators stabilizes at the Nash equilibrium [144], it becomes impossible for D_c to ascertain the image domain of the content feature, implying an absence of snow details in the content feature. A successful separation of style from content is achieved when the content encoder exclusively captures the image's content characteristics.

It is proved that utilizing the content discriminator can prevent content codes from containing style details [143]. Naturally, the next step is to remove content details from style codes. The purpose is to make the generation more stable without being affected by other factors. Consequently, the self-supervised style coding is implemented to remove any excess content detail from the style codes as illustrated in Fig. 5.2.

Using a nonlinear function f to denote the style encoder, interpolation on two style codes is initially performed; one from the clear domain and the other from the snow domain to

acquire s_k as:

$$s_k = f\left(kx_1 + (1-k)x_2\right). \tag{5.1}$$

The proposed methodology involves extracting style codes from two distinct domains. By using linear interpolation between these domains, guided by the parameter k, CUSS can generate a range of snow sizes. Throughout the training process, a randomly selected k value from the interval [0, 1] is used to derive a novel style code. This newly generated style code serves as a self-supervised pseudo-label. In the scope of this problem, the need to disentangle the style feature from the content feature makes sure that the operation on the style code is consistent with those on the input image.

According to the derivation of Zhang et al.'s work [139], because s_k is calculated by the linear projection of s_1 and s_2, it should contain snow detail that is also a linear relation of x_1 and x_2. By using s_k and a content code of a clear input c_1, a new snow image x_k can be generated. The style encoder then encodes s_k again to obtain its style code, which will be supervised by s_k itself. The loss function is defined as:

$$\mathscr{L}_s\left(E_1^c, E_2^s, F_2\right) = \quad \mathbb{E}_{x_1,x_2}\left[\left\|E_2^s\left(F_2\left(E_1^c\left(x_1\right), s_k\right)\right) - s_k\right\|_1\right]. \tag{5.2}$$

Even though the function f is nonlinear, the decoder continually generates x_k and the model optimizes the encoder with s_k in the training phase to maintain a consistent relationship between input images and corresponding style codes linearly. In the early stages, the style code will contain an extra content detail because of latent space entanglement. At every forward iteration, the extra details become separated from the style codes. Instinctively, the style encoder will identify content details and ignore them. In the absence of manually assigned labels, the process relies on s_k as a self-generated label to guide the updates of networks. The desired situation is that the style code will generate snow according to each object distribution at every distance and not decrease the information density of traffic sign areas.

Due to the stochastic choice of k at each forward iteration, the encoder is compelled to project the snow-related detail into a linear space. As a result, linear adjustments to style codes are used to generate images with varying snow densities. For example, the k value presenting the $k \times 100\%$ snow density of the input snow image can be specified. Then the encoders extract the style code and content code of the input clear image. After that, the content code and interpolated style code are fed to the decoder to obtain the output. The factor k governs the snow density; Since s_2 originates from the baseline snow, it can be scaled up or down using k to yield a background invariant image featuring different levels of snow density as:

$$x_k = F_2\left(E_1^c\left(x_1\right), kE_1^s\left(x_1\right) + (1-k)E_2^s\left(x_2\right)\right). \tag{5.3}$$

5.2.3 Loss Functions

The comprehensive loss function discussed in this chapter comprises several components: adversarial loss $\mathscr{L}_{\text{adv}}$, image reconstruction identity loss $\mathscr{L}_{\text{id}}$, content reconstruction loss $\mathscr{L}_{\text{recon}}^{\text{c}}$, style regression loss $\mathscr{L}_{\text{regre}}$, cycle consistency loss $\mathscr{L}_{\text{cc}}$, and content loss $\mathscr{L}_{\text{cont}}$. The overall objective function is formulated as the weighted sum of these individual loss components:

$$\mathscr{L} = \lambda_{\text{adv}}\mathscr{L}_{\text{adv}} + \lambda_{\text{id}}\mathscr{L}_{\text{id}} + \lambda_{\text{recon}}^{c}\mathscr{L}_{\text{recon}}^{c} + \lambda_{\text{regre}}^{s}\mathscr{L}_{\text{regre}}^{s} + \lambda_{\text{cc}}\mathscr{L}_{\text{cc}} + \lambda_{\text{cont}}\mathscr{L}_{\text{cont}}, \tag{5.4}$$

where variables λ act as the model's hyperparameters, modulating the significance of each loss component.

5.2.3.1 Adversarial Loss

Adversarial loss $\mathscr{L}_{\text{adv}}$ is employed in both the clear and snow domains to enhance the realism of the generated images. In the domain of clear images, the adversarial loss $\mathscr{L}_{D_1}$ is

specified as:

$$\mathscr{L}_{D_1} = \mathbb{E}_{x_1 \sim P_{X_1}} \left[\log D_1 \left(x_1\right)\right] + \mathbb{E}_{x_2 \sim P_{X_2}} \left[\log \left(1 - D_1 \left(F_1 \left(E^c \left(x_2\right), s_1\right)\right)\right)\right]. \tag{5.5}$$

Here, D_1 serves the purpose of differentiating real clear images from their synthesized counterparts, striving to maximize the aforementioned loss function. On the other hand, F_2 aims to reduce the loss in order to make the generated clear images appear more authentic. Likewise, $\mathscr{L}_{D_2}$ for the snow domain is defined as:

$$\mathscr{L}_{D_2} = \mathbb{E}_{x_2 \sim P_{X_2}} \left[\log D_2 \left(x_2\right)\right] + \mathbb{E}_{x_1 \sim P_{X_1}} \left[\log \left(1 - D_2 \left(F_2 \left(E^c \left(x_1\right), s_2\right)\right)\right)\right]. \tag{5.6}$$

Both of these adversarial losses are considered to have equal impact, and they are straightforwardly summed up to compose the final adversarial loss as:

$$\mathscr{L}_{\text{adv}} = \mathscr{L}_{D_1} + \mathscr{L}_{D_2}. \tag{5.7}$$

5.2.3.2 Identity Loss and Latent Space Reconstruction Loss

When provided with a snow image and a clear image, the encoders are required to recreate the input image based on the same content code and style code. As such, the disparity between the reassembled image and the initial image serves as the identity loss $\mathscr{L}_{\text{id}}$, adding additional constraints to the encoder as:

$$\mathscr{L}_{\text{id}} = \mathbb{E}_{x_1 \sim P_{X_1}} \left[\left\|F_1 \left(E_1^c \left(x_1\right), E_1^s \left(x_1\right)\right) - x_1\right\|_1\right] + \mathbb{E}_{x_2 \sim P_{X_2}} \left[\left\|F_2 \left(E_2^c \left(x_2\right), E_2^s \left(x_2\right)\right) - x_2\right\|_1\right]. \tag{5.8}$$

Additionally, the aim is for the decoded images to have content and style features closely resembling those in the original images. As a result, the following losses for the reconstruction of content code and style code are defined as:

$$\mathscr{L}_{\text{recon}}^c = \mathbb{E}_{x_1 \sim P_{X_1}} \left[\left\|E_1^c \left(x_{2\to 1}\right) - E_1^c \left(x_1\right)\right\|_1\right] + \mathbb{E}_{x_2 \sim P_{X_2}} \left[\left\|E_2^c \left(x_{1\to 2}\right) - E_2^c \left(x_2\right)\right\|_1\right], \tag{5.9}$$

$$\mathscr{L}^{s}_{\text{recon}} = \mathbb{E}_{x_1 \sim P_{X_1}} \left[\| E^{s}_{1}(x_{2\to 1}) - E^{s}_{1}(x_1) \|_1 \right] + \mathbb{E}_{x_2 \sim P_{X_2}} \left[\| E^{s}_{2}(x_{1\to 2}) - E^{s}_{2}(x_2) \|_1 \right]. \quad (5.10)$$

It is important to note that the reconstruction loss of a style code is treated as falling under the umbrella of the self-supervised style coding loss. These two are summed up, with the same weight applied to both, to arrive at the final style coding loss as:

$$\mathscr{L}^{s}_{\text{regre}} = \mathscr{L}^{s}_{\text{recon}} + \mathscr{L}_{s}, \quad (5.11)$$

where $\mathscr{L}_s$ indicates the loss between the interpolated style codes and the style codes extracted from the newly generated snow images as defined in Eq. 5.2.

5.2.3.3 Cross-Cycle Consistency Loss

The proposed model incorporates the cross-cycle consistency loss $\mathscr{L}_{\text{cc}}$, as in [143], to facilitate the learning of domain mappings. For the generated snow image $x_{1\to 2}$, its corresponding clear image x_2 can be recovered through a desnowing transformation. The cross-cycle consistency loss constrains the scope of the generated image while maintaining the background information of the input images. The L1 distance between the cyclically reconstructed image and the original image serves as the measure for this cross-cycle consistency loss. The image conversion process, which involves converting the clear image to the snow image and the other way around, proceeds as:

$$\begin{aligned} x_{1\to 2} &= F_2\left(E^{c}_{1}(x_1), E^{s}_{2}(x_2)\right), \\ x_{2\to 1} &= F_1\left(E^{c}_{2}(x_2), E^{s}_{1}(x_1)\right). \end{aligned} \quad (5.12)$$

The reverse translation operation, which entails reconstructing the original input from the generated image as:

$$\begin{aligned} x_{1\to 2\to 1} &= F_1\left(E^{c}_{2}(x_{1\to 2}), E^{s}_{1}(x_{2\to 1})\right), \\ x_{2\to 1\to 2} &= F_2\left(E^{c}_{1}(x_{2\to 1}), E^{s}_{2}(x_{1\to 2})\right). \end{aligned} \quad (5.13)$$

The formulation of the cross-cycle consistency loss for both the snow and clear image domains is:

$$\mathscr{L}_{\text{cc}} = \mathbb{E}_{x_1 \sim P_{X_1}} \left[\| x_1 - x_{1\to 2\to 1} \|_1 \right] + \mathbb{E}_{x_2 \sim P_{X_2}} \left[\| x_2 - x_{2\to 1\to 2} \|_1 \right]. \tag{5.14}$$

5.2.3.4 Content Loss

The goal of content loss $\mathscr{L}_{\text{cont}}$ is to make content discriminator D^c not able to determine whether an image is from the snow weather domain or the clear weather domain. It is adversarial training between the content encoder E^c and content discriminator D^c. Once the final balance is reached, the extracted content features do not contain any style information of the image. The equation is formulated as:

$$\mathscr{L}_{\text{cont}}(E^c, D^c) = \mathbb{E}_{x_1} \left[\log \left(D^c \left(E^c \left(x_1 \right) \right) \right) \right] + \mathbb{E}_{x_2} \left[\log \left(1 - D^c \left(E^c \left(x_2 \right) \right) \right) \right]. \tag{5.15}$$

5.3 Experiments

To validate the proposed CUSS method, this section delves into the influence of various modules and loss functions on the generated outcomes. It also benchmarks these outcomes against existing methods through both quantitative and qualitative metrics. Initially, an overview of the implementation details and the datasets used are provided. Subsequently, the proposed model is examined in-depth, and comparisons are made with current methodologies in the field. The model's effectiveness is further supported by showcasing visualizations of intermediate outcomes and conducting a generalization analysis. The final portion of this section focuses on ablation studies to scrutinize the model's components.

5.3.1 Implementation Details

The proposed model's network comprises two encoders, two decoders, two discriminators, and a content encoder. Among the encoders, one is designed for style, and the other for content. Their structure aligns with what is described in [143]. Breaking it down:

- The content encoder has five convolutional layers.
- The style encoder includes an initial residual layer, two downsampling layers, and one adaptive average pooling layer.
- The content encoder features an initial residual layer, two downsampling layers, and four residual blocks.
- Each decoder is made up of four residual blocks and two upsampling layers. It employs adaptive instance normalization, while the encoders use standard instance normalization.
- All the discriminators take specific image patches with the same resolution as input, which is inspired by Demir's work [145]. This structure includes five convolutional layers.

For training, minibatch stochastic gradient descent is implemented with a batch size of 12, using the Adam optimization technique [108] with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate is initially set at 0.0001 and is linearly reduced from the 100th epoch onwards. In the training phase, the input is cropped to a resolution of 256×256 pixels. The weight of each loss function is set as: $\lambda_{\text{adv}} = 1, \lambda_{\text{id}} = 10, \lambda^{c}_{\text{recon}} = 1, \lambda^{s}_{\text{regre}} = 1, \lambda_{\text{cc}} = 1$, and $\lambda_{\text{cont}} = 1$. All testing and experimentation are performed on an NVIDIA RTX A6000 GPU equipped with 24 GB of memory.

Figure 5.3: Collection of self-captured videos depicting urban driving amidst intense snowfall. The footage includes various road users, including cyclists, automobiles, buses, and pedestrians.

5.3.2 Datasets

All images in the Cityscapes [1] dataset are used as a clear source to train the model, given that the number of images is close to the snow set. To maintain consistency with the snow dataset, 5,921 EuroCity Persons [9] snapshots are handpicked from the daytime training segment.

The curated RDSBW [121] snow dataset comprises a total of 6,814 meticulously curated photographs. The number of intercepted images is the same as for Cityscapes dataset because GAN training is prone to problems such as mode collapse, which leads to training failure. Some of the examples are shown in Fig. 5.3. Training and testing both use the same dataset. The difference is that patches of input images are used during training, while the complete images are used during testing.

In the experimental setting, the Cityscapes and the EuroCity Persons datasets are taken as the target set. These two datasets contain over 5,000 road scenarios under different urban and weather conditions. There are also variations of road users, such as pedestrians, cyclists, and moving vehicles.

5.3.3 Performance Assessment

This section presents a comprehensive analysis comparing the outputs of CUSS with the current State-Of-The-Art (SOTA) image conversion methods. Deep analysis of the impact of disentanglement is carried out, and a meticulous examination of the uniqueness and significance of each module is concluded based on ablation studies.

5.3.3.1 Assessment Criteria

Initially, the quality of image synthesis is evaluated using traditional computer vision measures, namely, the Peak Signal-to-Noise Ratio (PSNR) and the Structural SIMilarity index (SSIM). Additionally, Metrics that specifically focus on the depth and perception features of the images are employed, including the Fréchet Inception Distance (FID) [144], the Learned Perceptual Image Patch Similarity (LPIPS) distance [70], and the Visual Geometry Group (VGG) distance [68].

LPIPS is a metric used to evaluate perceptual differences between images [73]. Unlike traditional metrics, such as Mean Squared Error (MSE), which measure pixel-level differences or structural similarities, it employs deep learning to better align with human visual perception. In essence, it offers a more perceptually meaningful measure of image similarity, especially useful in tasks like image synthesis, where the objective is not just to reproduce pixel-accurate outputs but to generate outputs that are perceptually indistinguishable or pleasing to humans.

VGG distance refers to a perceptual loss metric based on the VGG network that was originally designed for image classification tasks [68]. Similar to LPIPS, it is used to measure the difference between two images in a feature space. The activities from one or more layers of the VGG network capture higher-level content and texture information about the images.

The other metrics were introduced in 4.3.3.

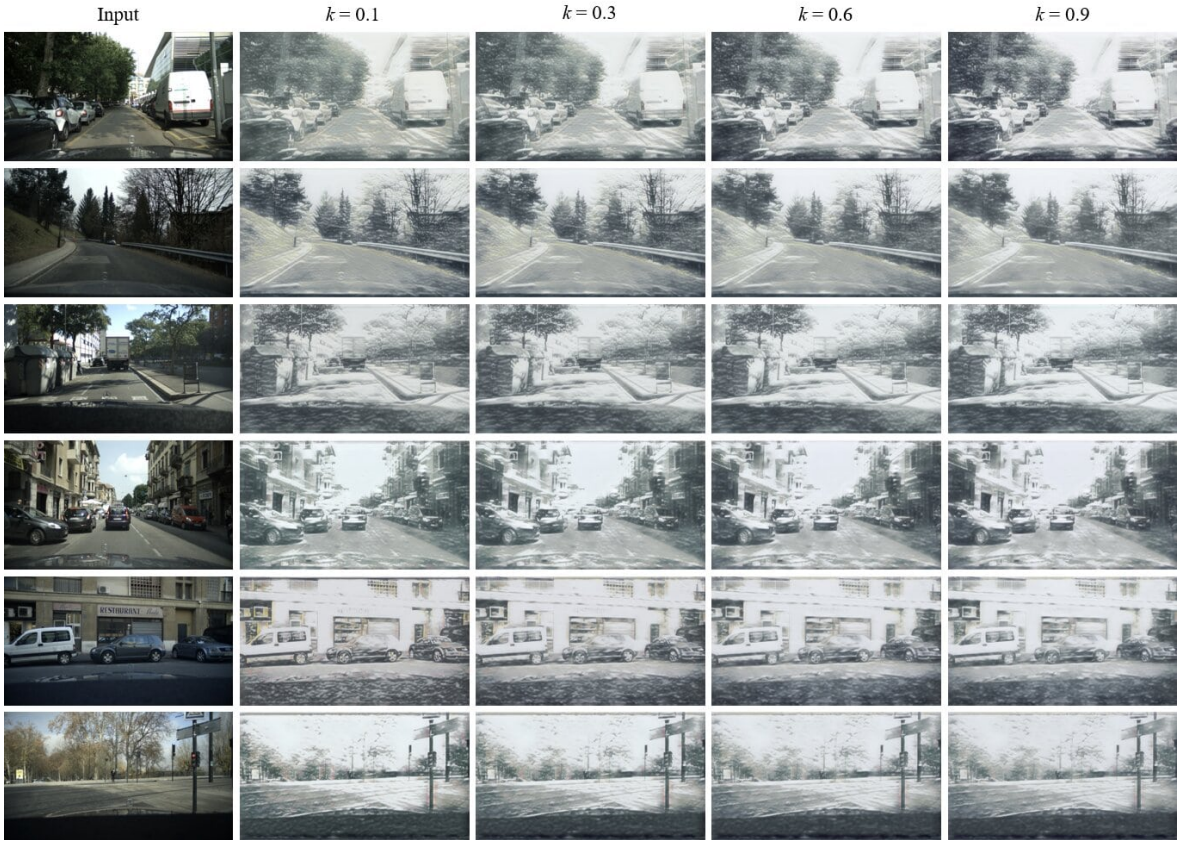


Figure 5.4: Synthesis of multi-density results on EuroCity Persons dataset by adjusting the parameter k. From the left to the right column, objects such as vehicles, people, and trees are covered with snowflakes and haze that gradually increase in size.

5.3.3.2 Qualitative Results

Snow images in varying sizes are generated by adjusting the previously discussed parameters, and the proposed model is contrasted against the leading SOTA methods.

Figure 5.4 displays images with varying amounts of snow. It is important to note that the k value of the input clear image is 0, while the value of the input snow image is 1. The snow feature is then adjusted within the range of 0 to 1. The differences in snow density across images with distinct parameter values demonstrate success in differentiating the generated snow through manipulation. As an illustration, as the value increases, objects at the far end

Figure 5.5: Comparisons between the synthesized snow images produced by the proposed method and SOTA unsupervised image translation methods. In particular, CUT deviates from the utilization of cycle consistency and the associated loss, as observed in CycleGAN. Conversely, the remaining models including the proposed CUSS, incorporate a form of partial style cycle consistency.

of the image become less distinguishable. Due to the impact of the style feature coding, the style encoders can identify between large and small snowflakes.

The results of the qualitative comparison are shown in Fig. 5.5. This experiment compares the generated snow images with mainstream image conversion methods (CycleGAN [29], Contrastive Unpaired Translation (CUT) [134], MUNIT [135], and DiveRse Image-to-image Translation (DRIT) [143]). The first two models use single projections, while the last two can produce diverse outcomes. For a more accurate comparison, ResNet [109] is consistently used as the backbone for the generator in all methods. The training uses the Cityscapes and EuroCity Persons datasets as clear source and the curated RDSBW snow set as target sources. The methods used for comparison all require no paired data.

The qualitative comparison shows that models such as CUT, MUNIT, and DRIT mainly exhibited three primary defects. First, after transforming images to represent snowy scenes, the original colors were often distorted, diminishing the natural appearance of the scene. Second, these models sometimes introduced artifacts that were not present in the original image, leading to inconsistencies and jarring visual outcomes. Lastly, they inadequately handled the far end and sky regions, resulting in uneven or unrealistic snow representation in these areas.

In contrast, the proposed method offered several advantages. The proposed method naturally integrated snow, ensuring that its boundaries fade out seamlessly across the image, providing an authentic representation in both the foreground and background. By distinguishing between snow style and actual image content, the proposed method was able to capture and reproduce the intrinsic properties of snow, resulting in a synthesis that feels genuine and consistent throughout the image. Moreover, while other models rendered trees or other objects as if they were buried under unnatural snow formations, the proposed technique retained the original structure and detail, providing a more balanced and realistic representation.

5.3.3.3 Quantitative Results

As reported in other image conversion works, the constraint of cycle consistency is strong so that the ability to generate diverse outputs is suppressed. However, the output image will retain a high similarity to the original image, which explains why CycleGAN achieved the best SSIM value as shown in Tables 5.1 and 5.2. Compared with other methods, CUSS combines the content discriminator and style code manipulation, and both turned out effective for high-quality synthesizing. Therefore, CUSS achieved better results on those metrics.

Table 5.1: Comparison on Cityscapes dataset between SOTA image translation techniques through numerical evaluation. Images with $k = 1$ are generated.

Methods	**SSIM**↑	**PSNR** ↑	$d_{\mathbf{VGG}}$ ↓	**FID**↓	$d_{\mathbf{LPIPS}}$ ↑
CUT [134]	0.32	15.85	6.06	26.16	**0.05**
CycleGAN [29]	**0.47**	16.07	5.89	26.34	**0.05**
MUNIT [135]	0.45	16.12	5.92	25.45	**0.05**
DRIT [143]	**0.47**	16.43	5.43	25.87	**0.05**
CUSS (Proposed)	**0.47**	**16.91**	**5.12**	**25.10**	**0.05**

Table 5.2: Comparison on EuroCity Persons dataset is made between SOTA image translation techniques through numerical evaluation. Images with $k = 1$ are generated.

Methods	**SSIM**↑	**PSNR** ↑	$d_{\mathbf{VGG}}$ ↓	**FID**↓	$d_{\mathbf{LPIPS}}$ ↑
CUT [134]	0.40	15.91	6.12	26.25	**0.05**
CycleGAN [29]	**0.50**	16.23	5.93	26.58	**0.05**
MUNIT [135]	0.48	16.48	6.01	25.85	**0.05**
DRIT [143]	0.47	16.74	5.76	26.01	**0.05**
CUSS (Proposed)	0.49	**16.98**	**5.39**	**25.40**	**0.05**

CUT is the only method that does not employ any format of cycle generation pipeline, but instead uses contrastive learning. The data used in the experiment cannot satisfy the requirement of a large batch size, which cannot make full use of contrastive loss. The results of CUSS proved that the proposed method was available even when the data were not sufficient.

To understand the individual contribution of different components of CUSS, an ablation study with respect to the loss functions is conducted. Since loss functions reflect the direc-

Table 5.3: Results from quantitative model comparisons after eliminating various loss factors are presented. Images are generated with three sets of k values, and the impact of the content discriminator, cross-cycle consistency loss, reconstruction losses, and style regression loss is examined.

Module	**SSIM**↑	**PSNR** ↑	$d_{\mathbf{VGG}}\downarrow$	**FID**↓	$d_{\mathbf{LPIPS}}\uparrow$
w/o $\mathscr{L}_{\text{id}}^{x}$	0.46	16.87	6.04	26.16	**0.05**
w/o $\mathscr{L}_{\text{recon}}^{c}$	0.46	16.83	6.05	26.13	**0.05**
w/o $\mathscr{L}_{\text{recon}}^{s}$	**0.47**	16.84	6.03	26.09	**0.05**
w/o $\mathscr{L}_{\text{cc}}$	0.46	16.86	6.06	26.11	**0.05**
w/o $\mathscr{L}_{\text{cont}}$	**0.47**	16.80	6.02	26.13	**0.05**
CUSS ($k = 1.0$)	**0.47**	16.91	6.00	26.09	**0.05**
CUSS ($k = 0.6$)	**0.47**	16.93	5.96	26.04	**0.05**
CUSS ($k = 0.3$)	**0.47**	**16.97**	**5.93**	**25.98**	**0.05**

tion of model optimization, the validation of the new module of the content discriminator and self-supervised style coding is performed, along with testing the improvement from reconstructing the image, style code, and content code. Table 5.3 shows that each component is crucial to the CUSS model presented in the decrease of the metrics. Images are generated with three sets of k values (0.3, 0.6, 1.0). Smaller k values indicate the small size of the synthesized snow, i.e., closer to the input clear image. The results show that the model produced the best quality output at the smallest k values.

5.3.3.4 Object Detection with Controlled Snow Size in Images

An important set of experiments is conducted to evaluate the impact of controlled snow size on object detection performance as shown in Fig. 5.6. This experiment involves a comparison using SOTA object detection algorithms Cascade R-CNN [14] on both original

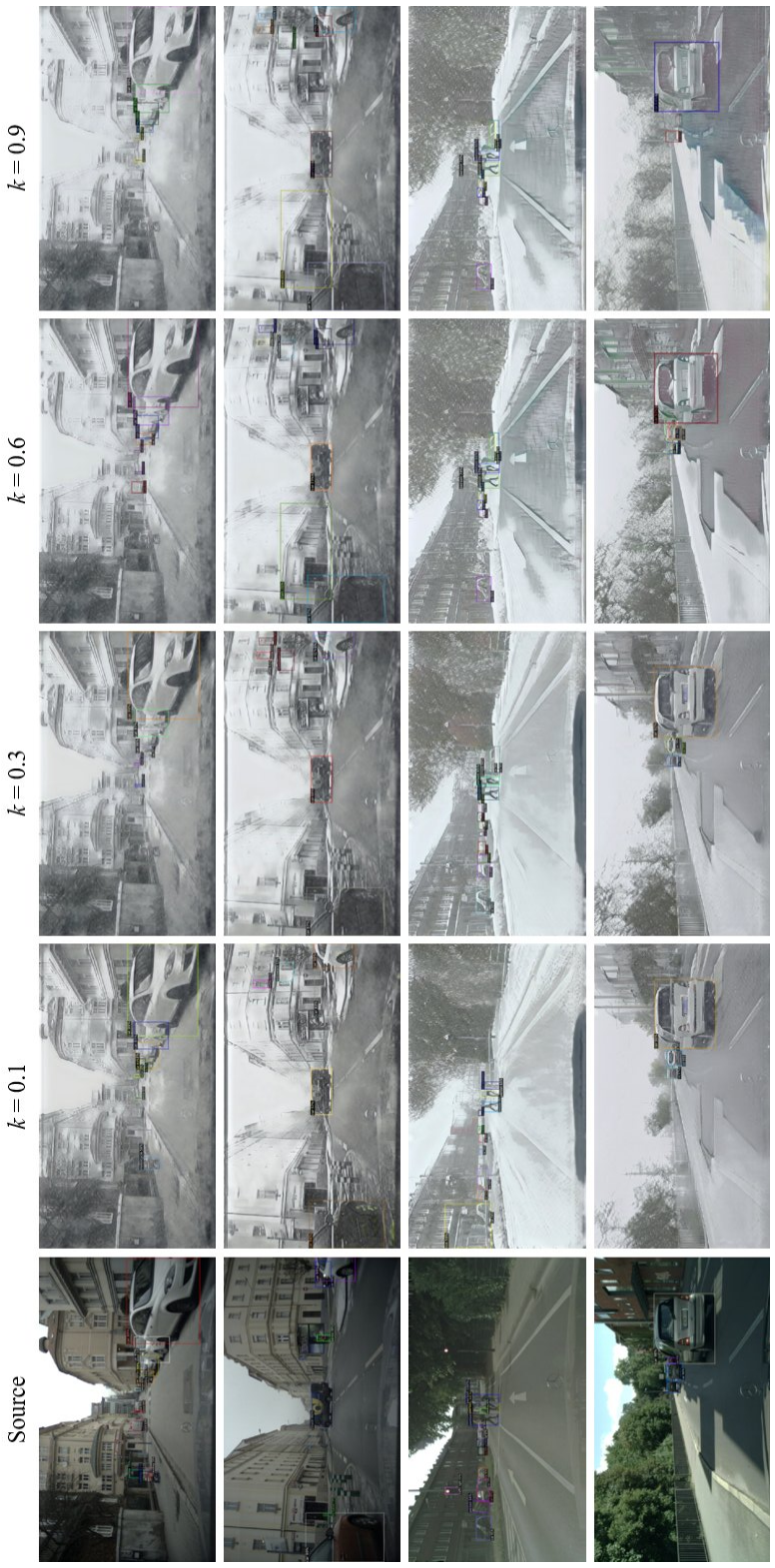


Figure 5.6: Object detection results on the source and converted snow image with different k values.

images from the Cityscapes and EuroCity Persons datasets, and on images with varying sizes of synthetic snow added. The synthetic snow is controlled by adjusting a parameter known as the "k value", which directly influences the extent of snow coverage in the images.

The primary objective of this experiment is to assess how changes in snow size, controlled via the k value, affect the number of objects detected by the algorithms. This serves as a crucial test of the realism and quality of the snow-enhanced images generated by the proposed method, as well as the impact of varying weather conditions on object detection systems.

The results of this experiment indicated a clear trend; as the size of the snow in the images increased, the overall number of objects detected by the SOTA detectors decreased. This reduction in detected objects can be attributed to the increased occlusion and reduced visibility caused by larger snowflakes or more intense snowfall in the images. It mirrors real-world conditions where heavy snow can obscure vision and make object detection more challenging.

These findings are significant as they demonstrate two key points. Firstly, the proposed method successfully allowed for controlled manipulation of snow size in images, showcasing the flexibility and precision of the proposed image conversion approach. Secondly, the fact that object detection performance varied with snow size is indicative of the realistic quality of the generated images. The ability to produce such realistic variations in weather conditions within images is vital for training and testing autonomous driving systems, ensuring they can adapt and perform accurately under different environmental scenarios.

5.3.3.5 Discussion

Image conversion methods such as CycleGAN, which uses the principle of cycle consistency, produce deterministic outputs. For a given clear image, it will produce the same translated snow image every time. To produce diverse outputs, researchers manipulate the

latent space of extracted image features by dividing them into style codes and content codes. In the experiments, it was found that latent space manipulations inevitably split the translation network into two or more parts, which lead to performance degradation. In this work, the solution was to use a content discriminator to distinguish the content code from different domains. With the requirement of generating an indistinguishable content code, the encoder could achieve better disentangled representations.

When obtaining the disentangled style code, the operation on it will reflect on the output snow images [146]. Therefore, the style codes of the input clear image and the snow image were interpolated. Since the input snow image represents the maximum snow size, the degree of snow effects can be controlled. It is important to note that snow effects larger than the input image cannot be obtained.

The controllable output was high quality and reasonable; the scenes were gradually covered with stronger snow effects. However, the generated snow was not invariant to objects. The snow covering the trees and the snow covering the building should be different; the snow effects should change appearance according to scenario changes. However, it looked similar in Fig. 5.4. To improve CUSS, the consideration of semantic information in the latent space is necessary.

The proposed method basically belongs to domain translation, which learns knowledge from the source domain and transfers it to the target domain. In the case of snow generation, the output will only contain a similar snow effect with input snowy images. To obtain more variety of snow like in real snow scenes, more snow data with different snow shapes and sizes should be added. However, if there are too many modes in the GAN training process, it can lead to mode collapse. In addition, the data should be collected in the same region to avoid large domain gaps, such as Asian driving scenes and European driving scenes.

5.4 Conclusion

This chapter presented a method for unsupervised snow synthesis, wherein a controllable method was introduced that incorporated latent space manipulation. To effectively separate the features of snow style and content, an additional content discriminator was incorporated along with a self-regression style coding module. To transition smoothly from clear to snow-affected images, a partial style cycle consistency loss was employed to refine the latent representation space. Furthermore, comparative analyses were conducted to comprehend the impact of each loss component or module within the model on the outcomes. When subjected to quantitative and qualitative evaluation against various techniques using the Cityscapes and EuroCity Persons datasets, the proposed approach consistently produced diverse and high-quality traffic scenes under snowy conditions. The assessment also included an examination of how changes in snow size, controlled via the k value, affect the number of objects detected by the SOTA detector. Moving forward, future research endeavors can be classified into two distinct paths:

- Expanding the proposed technique to tackle generation tasks in more demanding driving conditions, such as heavy rain, dense fog, nighttime, and strong light.
- Delving deeper into the relationship between generative methods and latent space manipulation for image conversion tasks by integrating existing insights from self-supervised and contrast learning methodologies.

6 Conclusions

6.1 Summary of the Thesis

Autonomous driving needs to operate in various scenarios, and some scenarios, such as severe weather conditions, can lead to a decline in the performance of the onboard perception system. This is because the data collected in different scenarios have domain shift problems. The solution is to use domain adaptation. This thesis built a framework to convert driving images taken in different severe weather conditions, using RGB images as data and image conversion as the domain adaptation method,

This thesis has divided the image conversion for driving scenes under adverse weather conditions into three sub-problems. Firstly, when obtaining paired data is difficult, how to train a conversion model with unpaired data. Secondly, how to obtain realistic and high-quality results after conversion. Thirdly, how to control the conversion process to ensure that each conversion consistently produces high-quality results.

Chapter 1 introduced the research background of this thesis. Firstly, it began with the data collected from Autonomous Vehicles (AVs) and analyzed the differences between various data domains, highlighting the distinctions in the weather domain compared to others. Secondly, it outlined the characteristics of data collected under different adverse weather conditions and the challenges these adverse weather conditions pose to perception systems. Thirdly, it presented the objective of this research, which was to apply knowledge from one domain to another through domain adaptation and solve the domain gaps between different

weather data. Fourthly, it introduced image conversion as one method of domain adaptation, and briefly discussed the issues present in driving scenarios and the proposed image transformation approach. Finally, it described the structure of this thesis.

Chapter 2 provided a comprehensive overview of domain adaptation in machine learning. It discussed the challenges of applying models trained on the data of the source domain to the data of the target domain, especially when labeled data in the target domain is scarce. It covered various domain adaptation techniques, categorized based on supervision, the number of participating domains, and feature space composition. Then, a detailed analysis of image conversion techniques in deep learning was provided, focusing on Generative Adversarial Networks (GANs) and their variants. It discussed the challenges and advancements in image generation, emphasizing the roles of generators and discriminators in GANs.

Chapter 3 first introduced an unpaired image conversion network based on GANs. This network incorporated the principle of cycle consistency and achieved the conversion of driving scene images from clear weather to foggy weather without paired data. To enhance the capability of the conversion network, a weather layer was introduced and merged into the reconstruction network through feature fusion. The effectiveness of this method was validated on public datasets. Then, this method was extended to handle multiple weather conditions. Specifically, the number of networks for converting from clear to adverse weather conditions was increased, while one network for converting from adverse weather to clear conditions was kept. This multi-weather joint learning approach benefits feature extraction in the conversion network and performs well on both public datasets and the self-collected Realistic Driving Scenes under Bad Weather (RDSBW) datasets.

Chapter 4 aimed to generate snow in driving scene images and proposed a method for realistic image conversion. This method was built upon the conversion network from the previous chapter and utilized additional semantic information to create snow effects. Fur-

thermore, deep supervision was implemented by incorporating intermediate outputs from the last two convolutional layers in the generator as multi-scale supervision signals during training. The generated images were compared with those produced by various network architectures, and the results were assessed both qualitatively and quantitatively using public datasets. The experimental results demonstrated that the proposed model was capable of synthesizing realistic snow effects in driving images. Meanwhile, the performance decline of State-Of-The-Art (SOTA) object detectors on converted snow images and the improved performance of pedestrian detectors trained with additional converted snow images indirectly confirmed that the proposed model was capable of generating realistic images.

Chapter 5 continued to focus on snow as the proposed research subject and introduced a controllable image conversion method. This method leveraged latent space manipulation. To effectively separate the features of snow style and content, an additional content discriminator was incorporated along with a self-regression style coding module. To transition smoothly from clear to snow-affected images, a partial style cycle consistency loss was employed to refine the latent representation space. Furthermore, comparative analyses were conducted to comprehend the impact of each loss component or module within the model on the outcomes. When subjected to quantitative and qualitative evaluation against various methods using public datasets, the proposed method consistently produced diverse and high-quality traffic scenes under snowy conditions. Additionally, the observed decrease in performance of SOTA object detectors on converted snow images generated by the proposed model suggested that, even with the enhanced capability to control the size of the snow in the images, the proposed model retained its proficiency in creating highly realistic visuals.

6.2 Future Work

Although the proposed model achieved image conversion of driving scene images under different weather conditions, a number of challenges still remain. The future research endeavors can be classified into four distinct paths

Enhanced Framework for Diverse Conditions

This framework would go beyond merely adding more converters. It would incorporate advanced algorithms capable of understanding and adapting to a wide range of environmental, lighting, and atmospheric conditions. The framework might also integrate context-aware processing, where the converters understand the scene's context (e.g. urban, rural, indoor, and outdoor) and adjust the conversion process accordingly.

Advanced Weather Effect Creation Technique

This technique would not be constrained by the weather conditions present in the input image. It could involve a sophisticated model that can artificially generate a variety of weather effects, like rain, snow, fog, or sunny, in a realistic manner. This could be done by analyzing the existing elements in the image and then rendering weather effects that are coherent with the scene's geometry, lighting, and perspective. The technique might also allow for user-defined weather conditions, enabling the creation of scenes with customized weather.

Perceptual Model for Guided Image Conversion

This model would involve developing a perceptual understanding of images to guide the conversion process. The model would use performance metrics such as detection accuracy and prediction reliability to guide the image conversion process. This approach ensures that the converted images are optimized for the specific needs of autonomous driving systems, enhancing their ability to accurately perceive and interpret real-world scenarios.

Holistic Environmental Simulation and Response System (HESRS)

HESRS would not only convert images but simulate entire environments with highly realistic weather and lighting conditions. It would replicate real-world physics and environmental interactions in a comprehensive virtual setting. This system would be capable of generating any imaginable weather scenario, from common situations like rain or snow to rare phenomena like solar eclipses or extreme meteorological events, in a highly realistic manner. The core of HESRS would be a predictive Artificial Intelligence (AI) that can forecast potential weather-related challenges and adapt the vehicle's driving strategy accordingly. This AI would learn from past scenarios and continuously evolve, becoming more adept at handling unforeseen weather conditions. This ambitious vision of HESRS represents a confluence of advanced AI, environmental science, and simulation technology. While it is a dream with current technology limitations, its realization could revolutionize not just autonomous driving but several other fields, aligning closely with the evolving needs of a climate-impacted world.

References

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes dataset for semantic urban scene understanding," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[2] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, pp. 973–992, 2018.

[3] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8022–8031.

[4] K. Zhang, R. Li, Y. Yu, W. Luo, and C. Li, "Deep dense multi-scale network for snow removal using semantic and depth priors," *IEEE Transactions on Image Processing*, vol. 30, pp. 7419–7431, 2021.

[5] J. Jin, A. Fatemi, W. M. P. Lira, F. Yu, B. Leng, R. Ma, A. Mahdavi-Amiri, and H. Zhang, "RaidaR: A rich annotated image dataset of rainy street scenes," in *Proceedings of the 18th IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2951–2961.

[6] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," *The International Journal of Robotics Research*, vol. 40, no. 4–5, pp. 681–690, 2021.

[7] M. A. Kenk and M. Hassaballah, "DAWN: Vehicle detection in adverse weather nature dataset," *Computing Research Repository arXiv Preprints,* arXiv:2008.05402, 2020.

[8] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2636–2645.

[9] M. Braun, S. Krebs, F. Flohr, and D. M. Gavrila, "The EuroCity Persons dataset: A novel benchmark for object detection," *Computing Research Repository arXiv Preprints,* arXiv:1805.07193, 2018.

[10] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi autonomous driving dataset," *Computing Research Repository arXiv Preprints,* arXiv:2004.06320, 2020.

[11] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The comma2k19 dataset," *Computing Research Repository arXiv Preprints,* arXiv:1812.05752, 2018.

[12] J. Mao, M. Niu, C. Jiang, H. Liang, J. Chen, X. Liang, Y. Li, C. Ye, W. Zhang, Z. Li, J. Yu, H. Xu, and C. Xu, "One million scenes for autonomous driving: ONCE dataset," *Computing Research Repository arXiv Preprints,* arXiv:2106.11037, 2021.

[13] K. Li, Y. Li, S. You, and N. Barnes, "Photo-realistic simulation of road scene for data-driven methods in bad weather," in *Proceedings of the 16th IEEE International Conference on Computer Vision Workshops*, 2017, pp. 491–500.

[14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2019.

[15] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. de la Escalera, "BirdNet: A 3D object detection framework from LiDAR information," in *Proceedings of the 21st International Conference on Intelligent Transportation Systems*, 2018, pp. 3517–3523.

[16] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian, "The rise of radar for autonomous vehicles: Signal processing solutions and future research directions," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 20–31, 2019.

[17] X. Dai, X. Yuan, and X. Wei, "TIRNet: Object detection in thermal infrared images for autonomous driving," *Applied Intelligence*, vol. 51, no. 3, pp. 1244–1261, 2021.

[18] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023.

[19] Y. Zhang, M. Ding, H. Yang, Y. Niu, Y. Feng, K. Ohtani, and K. Takeda, "L-DIG: A GAN-based method for LiDAR point cloud processing under snow driving conditions," *Sensors*, vol. 23, no. 21, pp. 8660_1–19, 2023.

[20] D. Ma, N. Shlezinger, T. Huang, Y. Liu, and Y. C. Eldar, "Joint radar-communication strategies for autonomous vehicles: Combining two key automotive technologies," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 85–97, 2020.

[21] J. Kim, S. Hong, J. Baek, E. Kim, and H. Lee, "Autonomous vehicle detection system using visible and infrared camera," in *Proceedings of the 12th IEEE International Conference on Control, Automation and Systems*, 2012, pp. 630–634.

[22] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 713–724, 2003.

[23] Q. A. Al-Haija, M. Gharaibeh, and A. Odeh, "Detection in adverse weather conditions for autonomous vehicles via deep learning," *AI*, vol. 3, no. 2, pp. 303–317, 2022.

[24] F. J. Abdu, Y. Zhang, and Z. Deng, "CCA-based fusion of camera and radar features for target classification under adverse weather conditions," *Neural Processing Letters*, vol. 55, no. 6, pp. 7293–7319, 2023.

[25] Y. Dong, W. Guo, M. Li, F. Zha, B. Shao, and L. Sun, "Framework of degraded image restoration and simultaneous localization and mapping for multiple bad weather conditions," *Optical Engineering*, vol. 62, no. 4, pp. 48 102_1–30, 2023.

[26] M. M. Yusuf, T. Karim, and A. S. Saif, "A robust method for lane detection under adverse weather and illumination conditions using convolutional neural network," in *Proceedings of the 2020 International Conference on Computing Advancements*, 2020, 8 pages.

[27] N. Reddy, A. Singhal, A. Kumar, M. Baktashmotlagh, and C. Arora, "Master of all: Simultaneous generalization of urban-scene segmentation to all adverse weather

conditions," in *Proceedings of the 17th European Conference on Computer Vision*, vol. 39, 2022, pp. 51–69.

[28] S.-H. Huang, "An application of neural network on traffic speed prediction under adverse weather conditions," Ph.D. dissertation, The University of Wisconsin, Madison, WI, USA, 2003.

[29] 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 16th IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[31] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[32] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *Proceedings of the 27th International Conference on Artificial Neural Networks*, vol. 3, 2018, pp. 270–279.

[33] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[34] S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J. E. Gonzalez, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and K. Keutzer, "A review of single-source deep unsupervised visual domain adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 473–493, 2020.

[35] L. Gautheron, I. Redko, and C. Lartizien, "Feature selection for unsupervised domain adaptation using optimal transport," in *Proceedings of the 2018 European Conference on Machine Learning and Knowledge Discovery in Databases*, vol. 2, 2019, pp. 759–776.

[36] G. Wang, T. Guo, Y. Yu, and H. Su, "Unsupervised domain adaptation classification model based on generative adversarial network," *Acta Electronica Sinica*, vol. 48, no. 6, pp. 1190–1197, 2020.

[37] P. Su, S. Tang, P. Gao, D. Qiu, N. Zhao, and X. Wang, "Gradient regularized contrastive learning for continual domain adaptation," *Computing Research Repository arXiv Preprints,* arXiv:2007.12942, 2020.

[38] F. Cangning, L. Peng, and X. Ting, "A review of depth domain adaptation: General and complex situations," *Acta Automatica Sinica*, vol. 47, no. 3, pp. 515–548, 2020.

[39] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *Computing Research Repository arXiv Preprints,* arXiv:1702.05374, 2017.

[40] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 5, 46 pages, 2020.

[41] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8050–8058.

[42] T. Kim and C. Kim, "Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation," in *Proceedings of the 16th European Conference on Computer Vision*, vol. 16, 2020, pp. 591–607.

[43] F. Liu, J. Lu, B. Han, G. Niu, G. Zhang, and M. Sugiyama, "Butterfly: One-step approach towards wildly unsupervised domain adaptation," *Computing Research Repository arXiv Preprints,* arXiv:1905.07720, 2021.

[44] Y. Shu, Z. Cao, M. Long, and J. Wang, "Transferable curriculum for weakly-supervised domain adaptation," in *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 4951–4958.

[45] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," *Advances in Neural Information Processing Systems*, vol. 21, pp. 1041–1048, 2008.

[46] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[47] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1406–1415.

[48] H. Yu, M. Hu, and S. Chen, "Multi-target unsupervised domain adaptation without exactly shared categories," *Computing Research Repository arXiv Preprints,* arXiv:1809.00852, 2018.

[49] B. Gholami, P. Sahu, O. Rudovic, K. Bousmalis, and V. Pavlovic, "Unsupervised multi-target domain adaptation: An information theoretic approach," *IEEE Transactions on Image Processing*, vol. 29, pp. 3993–4002, 2020.

[50] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1381–1391, 2018.

[51] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, pp. 139–144, 2014.

[52] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.

[53] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[54] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," *Advances in Neural Information Processing Systems*, vol. 31, pp. 1647–1657, 2018.

[55] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CYCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1989–1998.

[56] E. Tzeng, K. Burns, K. Saenko, and T. Darrell, "SPLAT: Semantic pixel-level adaptation transforms for detection," *Computing Research Repository arXiv Preprints,* arXiv:1812.00929, 2018.

[57] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and Z. Huang, "Cycle-consistent conditional adversarial transfer networks," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 747–755.

[58] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computing Research Repository arXiv Preprints,* arXiv:1411.1784, 2014.

[59] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proceedings of the 16th IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.

[60] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 214–223.

[61] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5769–5779, 2017.

[62] W. Shaoqian and L. Ximing, "Survey on research progress of generating adversarial networks," *Journal of Frontiers of Computer Science & Technology*, vol. 14, no. 3, pp. 377–388, 2020.

[63] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3313–3332, 2021.

[64] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.

[65] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Computing Research Repository arXiv Preprints,* arXiv:1511.06434, 2015.

[66] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 3, 2015, pp. 234–241.

[67] C. Wang, C. Xu, C. Wang, and D. Tao, "Perceptual adversarial networks for image-to-image transformation," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4066–4079, 2018.

[68] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Computing Research Repository arXiv Preprints,* arXiv:1409.1556, 2014.

[69] T. L. Sung and H. J. Lee, "Image-to-image translation using identical-pair adversarial networks," *Applied Sciences*, vol. 9, no. 13, pp. 2668_1–15, 2019.

[70] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[71] C. Wang, W. Niu, Y. Jiang, H. Zheng, Z. Yu, Z. Gu, and B. Zheng, "Discriminative region proposal adversarial network for high-quality image-to-image translation," *International Journal of Computer Vision*, vol. 128, no. 10–11, pp. 2366–2385, 2020.

[72] T. He, Y. Xia, J. Lin, X. Tan, D. He, T. Qin, and Z. Chen, "Deliberation learning for image-to-image translation." in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 2484–2490.

[73] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," *Advances in Neural Information Processing Systems*, vol. 30, pp. 465–476.

[74] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proceedings of the 33th International Conference on Machine Learning*, 2016, pp. 1558–1566.

[75] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *Computing Research Repository arXiv Preprints,* arXiv:1605.09782, 2016.

[76] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: Controlling deep image synthesis with texture patches," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8456–8465.

[77] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423.

[78] B. AlBahar and J.-B. Huang, "Guided image-to-image translation with bi-directional feature transformation," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9016–9025.

[79] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1857–1865.

[80] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proceedings of the 16th IEEE International Conference on Computer Vision*, 2017, pp. 2849–2857.

[81] L. Chen, L. Wu, Z. Hu, and M. Wang, "Quality-aware unpaired image-to-image translation," *IEEE Transactions on Multimedia*, vol. 21, no. 10, pp. 2664–2674, 2019.

[82] R. Zhang, T. Pfister, and J. Li, "Harmonic unpaired image-to-image translation," *Computing Research Repository arXiv Preprints,* arXiv:1902.09727, 2019.

[83] G. Lu, Z. Zhou, Y. Song, K. Ren, and Y. Yu, "Guiding the one-to-one mapping in CycleGAN via optimal transport," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, vol. 01, 2019, pp. 4432–4439.

[84] L. Kantorovitch, "On the translocation of masses," *Management Science*, vol. 5, no. 1, pp. 1–4, 1958.

[85] J. Lin, Y. Xia, Y. Wang, T. Qin, and Z. Chen, "Image-to-image translation with multi-path consistency regularization," *Computing Research Repository arXiv Preprints,* arXiv:1905.12498, 2019.

[86] J. Kim, M. Kim, H. Kang, and K. Lee, "U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation," *Computing Research Repository arXiv Preprints,* arXiv:1907.10830, 2019.

[87] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings*

of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6924–6932.

[88] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *Computing Research Repository arXiv Preprints,* arXiv:1607.06450, 2016.

[89] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang, "Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks," in *Proceedings of the 14th European Conference on Computer Vision*, vol. 4, 2018, pp. 184–199.

[90] J. Zhang and Y. Hou, "Image-to-image translation based on improved cycle-consistent generative adversarial network," *Chinese Journal of Electronics*, vol. 42, no. 5, pp. 1216–1222, 2020.

[91] T. F. van der Ouderaa and D. E. Worrall, "Reversible GANs for memory-efficient image-to-image translation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4720–4728.

[92] Y.-C. Chen, X. Xu, Z. Tian, and J. Jia, "Homomorphic latent space interpolation for unpaired image-to-image translation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2408–2416.

[93] T. Xiao, J. Hong, and J. Ma, "ELEGANT: Exchanging latent encodings with GAN for transferring multiple face attributes," in *Proceedings of the 14th European Conference on Computer Vision*, vol. 5, 2018, pp. 168–184.

[94] X. Yu, X. Cai, Z. Ying, T. Li, and G. Li, "SingleGAN: Image-to-image translation by a single-generator network using multiple generative adversarial learning," in *Proceedings of the 14th Asian Conference on Computer Vision*, vol. 5, 2019, pp. 341–356.

[95] W. Xu, K. Shawn, and G. Wang, "Toward learning a unified many-to-many mapping for diverse image translation," *Pattern Recognition*, vol. 93, pp. 570–580, 2019.

[96] Y. Li, S. Tang, R. Zhang, Y. Zhang, J. Li, and S. Yan, "Asymmetric GAN for unpaired image-to-image translation," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5881–5896, 2019.

[97] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville, "Augmented CycleGAN: Learning many-to-many mappings from unpaired data," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 195–204.

[98] Y. Alharbi, N. Smith, and P. Wonka, "Latent filter scaling for multimodal unsupervised image-to-image translation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1458–1466.

[99] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *Advances in Neural Information Processing Systems*, vol. 30, pp. 700–708, 2017.

[100] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," *Computing Research Repository arXiv Preprints,* arXiv:1611.02200, 2016.

[101] A. Royer, K. Bousmalis, S. Gouws, F. Bertsch, I. Mosseri, F. Cole, and K. Murphy, "XGAN: Unsupervised image-to-image translation for many-to-many mappings," in *Domain Adaptation for Visual Understanding*. Springer, Cham, Switerland, 2020, pp. 33–49.

[102] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4500–4509.

[103] A. Anoosheh, E. Agustsson, R. Timofte, and L. Van Gool, "ComboGAN: Unrestrained scalability for image domain translation," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 783–790.

[104] Y. Lin, K. Fu, S. Ling, and P. Cheng, "An efficient multi-domain framework for image-to-image translation," *Computing Research Repository arXiv Preprints,* arXiv:1911.12552, 2019.

[105] Y. Ye, Y. Chang, H. Zhou, and L. Yan, "Closing the loop: Joint rain generation and removal via disentangled image translation," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2053–2062.

[106] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 606–615.

[107] Y. Shao, L. Li, W. Ren, C. Gao, and N. Sang, "Domain adaptation for image dehazing," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2808–2817.

[108] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computing Research Repository arXiv Preprints,* arXiv:1412.6980, 2014.

[109] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[110] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[111] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3522–3533, 2015.

[112] D. Berman, T. Treibitz, and S. Avidan, "Non-local image dehazing," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1674–1682.

[113] W. Wang, "Adapted center and scale prediction: More stable and more accurate," *Computing Research Repository arXiv Preprints,* arXiv:2002.09053, 2020.

[114] W. Liu, I. Hasan, and S. Liao, "Center and scale prediction: Anchor-free approach for pedestrian and face detection," *Computing Research Repository arXiv Preprints,* arXiv:1904.02948, 2019.

[115] Q.-X. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," *Computer Graphics Forum*, vol. 32, no. 5, pp. 177–186, 2013.

[116] F. Wang, Q. Huang, and L. J. Guibas, "Image co-segmentation via consistent functional maps," in *Proceedings of the 14th IEEE International Conference on Computer Vision*, 2013, pp. 849–856.

[117] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3D-guided cycle consistency," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 117–126.

[118] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[119] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.

[120] H. Yang, A. Carballo, and K. Takeda, “Disentangled bad weather removal GAN for pedestrian detection,” in *Proceedings of the IEEE 95th Vehicular Technology Conference*, 2022, 6 pages.

[121] H. Yang, “RDSBW dataset: Road damage segmentation and benchmarking dataset,” https://sites.google.com/g.sp.m.is.nagoya-u.ac.jp/rdsbw-dataset, 2021, (Accessed: 2023-12-19).

[122] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3103–3112.

[123] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, “Multi-stage progressive image restoration,” in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 821–14 831.

[124] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, “Progressive image deraining networks: A better and simpler baseline,” in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3937–3946.

[125] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proceedings of the 16th European Conference on Computer Vision*, vol. 7, 2018, pp. 254–269.

[126] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[127] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Proceedings of the 14th European Conference on Computer Vision*, vol. 1, 2016, pp. 154–169.

[128] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "AOD-Net: All-in-one dehazing network," in *Proceedings of the 16th IEEE International Conference on Computer Vision*, 2017, pp. 4770–4778.

[129] X. Liu, Y. Ma, Z. Shi, and J. Chen, "GridDehazeNet: Attention-based multi-scale network for image dehazing," in *Proceedings of the 17th IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7314–7323.

[130] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 270–12 279.

[131] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "DesnowNet: Context-aware deep network for snow removal," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3064–3073, 2018.

[132] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, "Vision transformer adapter for dense predictions," *Computing Research Repository arXiv Preprints,* arXiv:2205.08534, 2022.

[133] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *Computing Research Repository arXiv Preprints,* arXiv:1711.05101, 2017.

[134] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proceedings of the 18th European Conference on Computer Vision*, vol. 4, 2020, pp. 319–345.

[135] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the 15th European Conference on Computer Vision*, vol. 3, 2018, pp. 172–189.

[136] Y. Niu, Y. Zhong, W. Guo, Y. Shi, and P. Chen, "2D and 3D image quality assessment: A survey of metrics and challenges," *IEEE Access*, vol. 7, pp. 782–801, 2018.

[137] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[138] H. Yang, A. Carballo, Y. Zhang, and K. Takeda, "Framework for generation and removal of multiple types of adverse weather from driving scene images," *Sensors*, vol. 23, no. 3, pp. 1548_1–16, 2023.

[139] C. Zhang, Z. Lin, L. Xu, Z. Li, W. Tang, Y. Liu, G. Meng, L. Wang, and L. Li, "Density-aware haze image synthesis by self-supervised content-style disentanglement," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4552–4572, 2021.

[140] M. Wang, T. Su, S. Chen, W. Yang, J. Liu, and Z. Wang, "Automatic model-based dataset generation for high-level vision tasks of autonomous driving in haze weather," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 9071–9081, 2022.

[141] S. Ni, X. Cao, T. Yue, and X. Hu, "Controlling the rain: From removal to rendering," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6328–6337.

[142] H. Yang, M. Ding, A. Carballo, Y. Zhang, K. Ohtani, Y. Niu, M. Ge, Y. Feng, and K. Takeda, "Synthesizing realistic snow effects in driving images using GANs and real data with semantic guidance," in *Proceedings of the 2023 IEEE Intelligent Vehicles Symposium*, 2023, 6 pages.

[143] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the 15th European Conference on Computer Vision*, vol. 1, 2018, pp. 35–51.

[144] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6629–6640, 2017.

[145] U. Demir and G. Unal, "Patch-based image inpainting with generative adversarial networks," *Computing Research Repository arXiv Preprints,* arXiv:1803.07422, 2018.

[146] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *Computing Research Repository arXiv Preprints, arXiv:*1812.02230, 2018.

Acknowledgement

I would like to express my deepest gratitude to Professor Kazuya Takeda of Nagoya University for his invaluable guidance, patience, and support throughout my doctoral journey. His expertise and insights have been instrumental in shaping my research and academic growth.

My sincere thanks go to my direct supervisor Professor Alexander Carballo and other mentors in this lab Professor Ding Ming, Professor Keisuke Fujii, Professor Kento Ohtani, and Professor Eijiro Takeuchi, for their relentless dedication and invaluable advice. Their expertise in the field of autonomous driving technology has been a constant source of inspiration and learning for me.

I am deeply indebted to my friends and colleagues Yuxiao Zhang, Maoning Ge, Niu Yingjie, and Yan Feng for their selfless help, collaborative spirit, and the countless hours we spent together in academic pursuits. Their friendship and support have been a great source of comfort and motivation.

A special thanks to my partner, Ruchen Zhang, whose understanding, and companionship have been a beacon of light in my life. Her presence has greatly enriched my doctoral experience and brought immense happiness to my life.

To my seniors Patiphon Narksri and Jacob Lambert, and my colleagues Atsushi Kuribayashi, and Aoki Takanose, I am thankful for the joyful moments and the cultural exchanges we shared during our time at the Friday night beer meeting.

I am grateful to the entire team at the Behavior Signal Processing Lab, including my initial remote mentors during the pandemic, especially Tomoki Hayashi for his guidance in server management, and my peers at Pix Moving for the invaluable industry experience.

Finally, my heartfelt appreciation goes to my parents, whose unwavering love, encouragement, and sacrifices have been the cornerstone of my achievements. Their belief in me has been my greatest strength.

This journey would not have been possible without the collective support and encouragement of everyone mentioned above, and I am eternally grateful for their presence in my life.

List of Publications

Journal Papers

[1] Yang, Hanting, Carballo, Alexander, Zhang, Yuxiao, and Takeda, Kazuya, "Framework for generation and removal of multiple types of adverse weather from driving scene images," *Sensors*, vol. 23, no. 3, pp. 1548_1–16, 2023, DOI: 10.3390/s23031548

[2] Yang, Hanting, Carballo, Alexander, Zhang, Yuxiao, and Takeda, Kazuya, "Controllable unsupervised snow synthesis by latent style space manipulation," *Sensors*, vol. 23, no. 20, pp. 8398_1–19, 2023, DOI: 10.3390/s23208398

[3] Zhang, Yuxiao, Carballo, Alexander, Yang, Hanting, and Takeda, Kazuya, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146–177, 2023, DOI: 10.1016/j.isprsjprs.2022.12.021

[4] Zhang, Yuxiao, Ding, Ming, Yang, Hanting, Niu, Yingjie, Feng, Yan, Ohtani, Kento, and Takeda, Kazuya, "L-DIG: A GAN-based method for LiDAR point cloud processing under snow driving conditions," *Sensors*, vol. 23, no. 21, pp. 8660_1–19, 2023, DOI: 10.3390/s23218660

International Conferences

[5] Yang, Hanting, Carballo, Alexander, and Takeda, Kazuya, "Disentangled bad weather removal GAN for pedestrian detection," in Proceedings of the *IEEE 95th Vehicular Technology Conference*, Helsinki, Finland, June 2022, 6 pages, DOI: 10.1109/VTC2022-Spring54318.2022.9860865.

[6] Yang, Hanting, Ding, Ming, Carballo, Alexander, Zhang, Yuxiao, Ohtani, Kento, and Takeda, Kazuya, "Synthesizing realistic snow effects in driving images using GANs and real data with semantic guidance," in Proceedings of the *2023 IEEE Intelligent Vehicles Symposium*, Anchorage, AK, USA, June 2023, 6 pages, DOI: 10.1109/IV55152.2023.10186565.

[7] Zhang, Yuxiao, Ding, Ming, Yang, Hanting, Niu, Yingjie, Feng, Yan, Ge, Maoning, Carballo, Alexander, and Takeda, Kazuya, "LiDAR point cloud translation between snow and clear conditions using depth images and GANs," in Proceedings of the *2023 IEEE Intelligent Vehicles Symposium*, Anchorage, AK, USA, June 2023, 7 pages, DOI: 10.1109/IV55152.2023.10186814.