GENERATION OF CORNER CASES IN INFRARED AUTONOMOUS DRIVING DATASET WITH STABLE DIFFUSION

OTONOM SÜRÜŞ İÇİN UÇ DURUMLARIN KARARLI DİFÜZYON İLE KIZILÖTESİ GÖRÜNTÜLERDE ÜRETİMİ

MERT YİĞİT

PROF. DR. AHMET BURAK CAN

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering

January 2025

ABSTRACT

GENERATION OF CORNER CASES IN INFRARED AUTONOMOUS DRIVING DATASET WITH STABLE DIFFUSION

Mert Yiğit

Master of Science, Computer Engineering

Supervisor: Prof. Dr. Ahmet Burak CAN

January 2025, 78 pages

The main priority of autonomous driving is situational awareness. To achieve this, autonomous vehicles are equipped with many sensors such as cameras, radar, and LIDAR. With the developments in deep learning, algorithms working with these sensors show very successful results in the detection of popular classes such as pedestrians, cyclists, and traffic signs. However, detecting unexpected objects not encountered in the scene is also very important for safe autonomous driving. Although there are some studies on detecting such situations, which are called corner cases, in the visible domain, these studies assume that these objects are visible. In real life, corner cases can also be encountered in low light conditions such as fog and darkness where visibility is poor. Therefore models developed with the assumption that these objects are always visible fail to detect them in real life, causing the relevant safety precaution systems not to react early and posing a serious risk to traffic safety. Failure to detect or late detection of such situations, even milliseconds are important, causes accidents. On the other hand, since infrared cameras are sensitive to thermal radiation, they can give healthy input without being affected by weather conditions such as darkness, fog, haze or night where visibility is poor, and they enable detection of such corner cases even from long distances. However, no existing infrared autonomous driving

dataset contains corner cases and the detection of corner cases in the infrared domain has not been studied before. Therefore, for the first time, we present a high-quality dataset for autonomous driving by generating corner case scenarios in the infrared domain with stable diffusion. In addition, another novel aspect of this study is that we train a detection model on the dataset we have created and perform the detection of corner cases in the infrared domain and present a baseline performance for the model.

Keywords: Corner case, autonomous driving, thermal infrared, stable diffusion

ÖZET

OTONOM SÜRÜŞ İÇİN UÇ DURUMLARIN KARARLI DİFÜZYON İLE KIZILÖTESİ GÖRÜNTÜLERDE ÜRETİMİ

Mert Yiğit

Yüksek Lisans, Bilgisayar Mühendisliği

Danışman: Prof. Dr. Ahmet Burak CAN

Ocak 2025, 78 sayfa

Otonom bir sürüşün temel önceliği durumsal farkındalıktır. Bunu sağlamak için otonom araçlar kameralar, radar, LIDAR gibi çok farklı sensörle donatılmışlardır. Derin öğrenme alanındaki gelişmelerle birlikte bu sensörlerle çalışan algoritmalar yaya, bisikletli ve trafik işaretlerinin tespiti gibi popular konularda oldukça başarılı sonuçlar vermektedirler. Fakat sahnede daha önce karşılaşılmamış, beklenmeyen objelerin tespiti de güvenli bir otonom sürüş için oldukça önemlidir. Uç durum olarak adlandırılan bu gibi durumların görünür alanda tespitine yönelik bazı çalışmalar olsa da bu çalışmalar bu objelerin görülebilir olduğu varsayımına dayanmaktadır. Gerçek hayatta ise uç durumlar görüşün zayıf olduğu sis, karanlık gibi hava şartlarında da bulunabilirler. Bu yüzden bu nesnelerin her zaman görünür olduğu varsayımıyla geliştirilen modeller gerçek hayatta bu objelerin tespiti konusunda başarısız olurlar, ve ilgili güvenlik sistemlerinin erken reaksiyon vermemesine sebep vererek güvenlik konusunda ciddi bir risk oluştururlar. Milisaniyelerin bile önemli olduğu böyle durumların tespit edilememesi veya geç tespit edilmesi kazalara sebep olur. Buna karşın kızılötesi kameralar termal radyasyona duyarlı olduğu için görüşün zayıf olduğu karanlık, sis, pus gibi hava durumlarından etkilenmeden sağlıklı bir görüntü verebilmekte ve bu gibi uç durumların tespitine uzak mesafelerden bile olanak sağlamaktadırlar. Fakat daha önce

hiçbir kızılötesi otonom sürüş veri seti uç durumları içermemekte ve uç durumların kızılötesi alanda tespiti daha önce çalışmamış bir konudur. Bu yüzden bu çalışma ile ilk kez kızılötesi alanda uç durum senaryolarının kararlı difüzyon ile üretildiği otonom sürüş için bir veri seti sunuyoruz. Ayrıca bu çalışmanın başka bir özgün yönü ise oluşturduğumuz veri seti üzerinde bir tespit modeli eğiterek uç durumların kızılötesi alanda tespitini gerçekleştirdik ve modelin referans performansını sunduk.

Keywords: Uç durum, otonom sürüş, termal kızılötesi, kararlı difüzyon

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks to my advisor, Prof. Dr. Ahmet Burak Can, for his continuous support, valuable guidance, and insightful feedback throughout my research journey. His expertise and encouragement have greatly influenced the direction of this thesis, and I feel truly privileged to have had the opportunity to work under his mentorship. His confidence in my abilities gave me the motivation to persevere and complete this work.

I am profoundly thankful to my family for their steadfast love, endless patience, and unwavering support, which have been the cornerstone of my resilience. Their constant encouragement has illuminated my path, even in the darkest moments, and their faith in my potential has fueled my determination. They have been my confidants, offering not just advice but inspiration, and lifting me with their unwavering belief. Their sacrifices and unwavering presence have been the silent force driving my journey forward. To them, I owe a debt of gratitude beyond words, for they have been the foundation upon which this accomplishment stands.

I would also like to express my heartfelt appreciation to my team leader, Alican Eslek, for his leadership and encouragement, and to my colleague, Burak Şenyiğit, for his collaboration and friendship.

CONTENTS

Page

ABSTRACT i
ÖZET iii
ACKNOWLEDGEMENTS v
CONTENTS vi
TABLES viii
FIGURES ix
ABBREVIATIONS xi
1. INTRODUCTION 1
1.1. Scope Of The Thesis 4
1.2. Contributions 5
1.3. Organization 5
2. BACKGROUND OVERVIEW 6
2.1. Autonomous Vehicles 6
2.2. Corner Cases in Autonomous Driving 7
2.3. Infrared Imaging 9
2.4. Generative Models 12
2.5. Stable Diffusion 14
2.6. LORA 17
2.7. Object Detection 19
3. RELATED WORK 23
3.1. Corner Case Creation with Stable Diffusion 23
3.1.1. DriveDreamer: Towards Real-World-driven World Models for Autonomous Driving 23
3.1.2. WEDGE: A multi-weather autonomous driving dataset built from generative vision-language models 24
3.1.3. Time to Shine: Fine-Tuning Object Detection Models with Synthetic Adverse Weather Images 24

3.2. Corner Case Detection on Visible Domain 25
3.2.1. Pixel-wise Anomaly Detection in Complex Driving Scenes 25
3.2.2. Conditioning Latent-Space Clusters for Real-World Anomaly Classification 25
3.2.3. HazardNet: Road Debris Detection by Augmentation of Synthetic Models 26
3.3. Thermal datasets for autonomous driving 27
3.3.1. KAIST Multispectral Pedestrian Detection Benchmark Dataset 27
3.3.2. FLIR ADAS 27
3.3.3. SODA 28
4. PROPOSED METHOD 30
5. EXPERIMENTAL RESULTS 38
6. DISCUSSION 50
7. CONCLUSION 52

TABLES

Page

Table 4.1 Environmental parameters. 37

Table 5.1 Model performance metrics for different classes on the test dataset. 45

FIGURES

Page

Figure 1.1 Comparison of images from visible and infrared cameras at night in low light conditions, the diagram is taken from [1]. 3

Figure 2.1 A modern autonomous vehicle, the diagram is taken from [2]. 7

Figure 2.2 The deer darted from the darkness, the diagram is taken from [3]. 8

Figure 2.3 The image highlights the infrared portion of the electromagnetic spectrum, the diagram is taken from [4]. 11

Figure 2.4 This image compares the schematic structures of four generative models, the diagram is taken from [5]. 14

Figure 2.5 This diagram illustrates the architecture of a diffusion-based generative model, the diagram is taken from [6]. 15

Figure 2.6 This diagram illustrates the architecture of a diffusion-based generative model, the diagram is taken from [7]. 17

Figure 2.7 This diagram represents the architecture of YOLOv8 [8], showcasing its convolutional layers (Conv), C2f modules, Bottlenecks, Spatial Pyramid Pooling-Fast, and detection head for object detection across different scales, the diagram is taken from [9]. 22

Figure 3.1 Example image from DriveDreamer, the diagram is taken from [10]. . . 23

Figure 3.2 Example images from the WEDGE dataset, the diagram is taken from [11]. 24

Figure 3.3 Anomaly Segmentation Framework, the diagram is taken from [12]. . . 25

Figure 3.4 VAE-based method is proposed in for real-world anomaly classification, the diagram is taken from [13]. 26

Figure 3.5 HazardNet, the diagram is taken from [14]. 26

Figure 3.6 A sample infrared image from the KAIST [15] dataset. 27

Figure 3.7 A sample infrared image from the FLIR ADAS [16] dataset. 28

Figure 3.8 A sample infrared image from the SODA [17] dataset. 29

Figure 4.1 GISD workflow. 31
Figure 4.2 The process of masking where the object will be generated during inpainting with stable diffusion. 35
Figure 5.1 This image represents the augmentation of deer LORA with different poses, and sizes on a single background with the text prompt: ”Deer at standing on the road,<lora:DeerLORA:1.5>”. 39
Figure 5.2 This image represents scenes with different distances, sizes, and numbers produced from 4 different classes. 40
Figure 5.3 Generation of deer with stable diffusion. 41
Figure 5.4 Generation of dog with stable diffusion. 42
Figure 5.5 Generation of trash bin object with stable diffusion. 43
Figure 5.6 Generation of traffic cone object with stable diffusion. 44
Figure 5.7 This image represents the detection results of the model trained with the generated dataset on real-world corner case images. 45
Figure 5.8 This image represents the results of the predictions on the test images. 46
Figure 5.9 This figure presents F1 score of the detection model. 47
Figure 5.10 This figure presents, the precision performance of the detection model. 48
Figure 5.11 This figure presents model accuracy of the detection model. 48
Figure 5.12 This figure presents confusion matrix metrics of the detection model.. 49

ABBREVIATIONS

LIDAR	:	**L**ight **I**nterference **D**etection **A**nd **R**anging
YOLO	:	**Y**ou **O**nly **L**ook **O**nce
mAP	:	**m**ean **A**verage **P**recision
IoU	:	**I**ntersection **o**ver **U**nion
III	:	**I**nformation **I**ntegration **I**nitiative
LWIR	:	**L**ong **W**ave **I**nfrared
MWIR	:	**M**edium **W**ave **I**nfrared
FIR	:	**F**ar **I**nfrared
NIR	:	**N**ear **I**nfrared
FLIR	:	**F**orward **L**ooking **I**nfrared
RGB	:	**R**ed **G**reen and **B**lue
ADAS	:	**A**dvanced **D**river-**A**ssistance **S**ystems
LORA	:	**L**ong **R**ange **A**rea
ALVINN	:	**A**utonomous **L**earning **V**ehicle **I**ntelligent **N**avigation **N**etwork
DARPA	:	**D**efense **A**dvanced **R**esearch **P**rojects **A**gency
VAE	:	**V**ariational **A**uto**E**ncoder
GANs	:	**G**enerative **A**dversarial **N**etworks
AI	:	**A**rtificial **I**ntelligence
RNN	:	**R**ecurrent **N**eural **N**etwork
LLM	:	**L**arge **L**anguage **M**odel
HOG	:	**H**istogram **o**f **O**riented **G**radients
SVM	:	**S**upport **V**ector **M**achine
CNN	:	**C**onvolutional **N**eural **N**etwork
SPFF	:	**S**ingle **P**oint **F**usion **F**ramework
CLI	:	**C**ommand **L**ine **I**nterface

1. INTRODUCTION

Autonomous vehicles offer a revolutionary change in transportation. The artificial intelligence algorithms running in these vehicles are at the core of this change. It is undeniable that developments in the field of artificial intelligence have directly enhanced autonomous vehicles. The developments in areas such as increasing driving safety and reducing traffic density are especially promising. Visual perception is critical for autonomous vehicles. Artificial intelligence applications provide autonomous vehicles with perception capability and aim to improve this capability. With perception capability, autonomous vehicles have a variety of abilities such as classifying and detecting objects around them and understanding traffic signs and lane markings. These abilities are crucial for safe autonomous driving. Autonomous vehicles run many different algorithms throughout the drive to ensure safety through sensors such as multi-spectrum cameras, radar, and LIDAR. The vast amount of data collected from these sensors is analyzed, processed, and given to artificial intelligence algorithms and the results are obtained. These vehicles are developing day by day in areas such as decision-making, and collision avoidance and show better results [18]. Object detection can be considered a critical component of the detection system in autonomous vehicles. Modern object detection models, such as YOLO [19] and R-CNN [20], process the images from the vehicle camera in real time to identify the objects and provide the necessary information to the vehicle control system to take the necessary actions. The development of detection algorithms for autonomous vehicles usually starts with identifying target objects. These objects usually include classes such as pedestrians, vehicles, cyclists, and traffic signs, which are highly popular in this area. After collecting data about these objects, these objects are then annotated to become a dataset for the study, and the selected model is trained using the annotated dataset. Model performance is evaluated through parameters such as mAP, IoU, Precision, and Recall. If necessary, optimization is performed on the model and the model is ready to be deployed after these processes. The models learn the patterns and features of these classes and recognize these objects in a real drive. Among these various sensors, especially visible domain cameras are essential for tasks that use visual information

about the vehicle environment by capturing high-resolution images. With the detailed color and texture information that they provide, these cameras have been and continue to be used in many studies [21–24] on object detection and segmentation. On the other hand, it is equally important to detect unexpected objects that may be encountered during a drive. These unexpected situations are called corner cases, which are found in a few driving datasets in the literature. Models trained on a closed set dataset, which have not encountered these objects, fail to detect these objects when they encounter them during autonomous driving, thus endangering driving safety. While it is important to detect these objects quickly and react early, not detecting these objects at all leads to major safety risks. The corner case in the visible domain is a topic that has been slowly being studied in recent years. Studies [13, 14, 25, 26] in this area focus on detecting the corner case scenario detected by the visible camera during autonomous driving. In the visible domain, both synthetic and real-life corner case scenarios are generated. Since visible cameras work in a light-dependent method, the amount of light in the environment directly affects the quality of the captured image. In other words, the better the lighting in the environment, the more distinguishable the objects are in the camera and this provides a healthier image. The detection and segmentation models running on these cameras were also trained with this data and then used to detect objects at similar illumination levels. However, studies in the visible domain for the corner casework on the assumption that the objects that create the corner case are always visible. The problem is that in real-world autonomous driving, objects may not always be visible. The main reasons for this are weather conditions such as fog and haze that may limit visibility or driving in the dark at night. In such cases, the visible domain camera cannot give a healthy input to the artificial intelligence model it is working with, and the model that is not trained with this type of data cannot detect these objects. In contrast, infrared cameras operate in low light or complete darkness. Long Wave Infrared (LWIR) cameras operate at a wavelength of 8-14 nanometers in the infrared spectrum. Unlike visible cameras, they work on thermal radiation emitted by objects instead of light. This capability allows them to operate in low light or dark conditions. As can be seen in Figure 1.1 during a night drive in a low light environment, while the visibility distance in the visible camera does not allow pedestrian and vehicle detection, pedestrians and vehicles in the same scene can be distinguished with

the infrared camera and the detection of these classes is successfully performed.

Figure 1.1 Comparison of images from visible and infrared cameras at night in low light conditions, the diagram is taken from [1].

When infrared autonomous datasets are evaluated, there are studies [27–31] on the detection of popular classes such as pedestrians, vehicles, and cyclists in a similar way in the visible domain. Although infrared cameras can give a very reliable input in such conditions, no work has previously addressed corner cases in the infrared domain. No previous study has utilized the advantage of the infrared domain to detect corner cases. The use of such an effective technology only for the detection of a limited number of objects in autonomous driving is considered to be a major drawback in this field. Creating a dataset containing a corner in real life is both a dangerous and time-consuming process. Creating corner case scenarios jeopardizes other elements in the traffic and corner case scenarios are found in many different variants. These variants include the location, size, angle, and appearance of the object and its diversity within the object class. In addition, when the autonomous datasets available in the literature are examined, it is observed that the data collection process is collected at a certain time and region and does not include scenarios that endanger traffic. As a result, models trained with only certain objects at certain times cannot detect an object they have not encountered before. A model that has not been trained with a dataset that does not include corner case scenarios cannot give correct input to the control mechanism of the autonomous vehicle. For example, when an autonomous vehicle encounters a trash

bin in the middle of the road in an environment where the amount of light is sufficient, the decision-making mechanism decides to slow down and pass around the object, but when the same scenario is considered in foggy weather where the amount of light is not sufficient, the visible domain camera will fail and this will cause the decision-making mechanism to fail and result in an accident. When the same scenario is performed with an infrared camera, the corner case object can be detected from very long distances and accidents can be prevented by taking action early in the decision-making mechanism. Therefore, our main motivation in this thesis is to accurately model the thermal characteristics of objects that may cause corner cases and to create a new thermal infrared dataset by leveraging stable diffusion to provide a foundation for future studies in the infrared field.

1.1. Scope Of The Thesis

The motivation of this thesis is to teach the thermal characteristics of the objects that create corner cases to the generative model and to create a dataset by leveraging the stable diffusion using the inpainting process of these cases. With this work, a dataset containing corner cases for the first time in the infrared domain is presented in the literature. This dataset contains 1000 different images in 512x512 size using stable diffusion on the FLIR ADAS [16] dataset with objects belonging to 4 different classes (deer, dog, trash bin, traffic cone). The objects that create the corner case are augmented to the scenes with the correct thermal characteristics in many different variants in terms of location, size, number, and diversity within the class. With LORAs created for each object, objects can be customized on a scene basis with inputs to the text prompt. We also trained a YOLOv8 [8] based detection model using the dataset we created to detect corner cases both in the test part of our data set and in real-world images and created a baseline performance. The model trained on our dataset can correctly detect corner case objects with high prediction scores. The dataset and LORA models we created are publicly available for academic studies in the field of infrared autonomous driving.

1.2. Contributions

We have three novel contributions in this thesis.

- To the best of our knowledge, no study has been done before using stable diffusion in the infrared domain to create corner case scenarios for autonomous driving. We create and share LORA models for the infrared domain to support future research.
- We create a dataset containing corner cases in thermal long-wave infrared (LWIR) images and make it publicly available. To the best of our knowledge, this is the first dataset in the literature containing the corner cases in infrared images.
- Furthermore, we train a model for the detection of objects in corner cases. This is the first study on detecting corner cases in infrared images.

The LORA models, dataset, and detection models created as part of this thesis can be accessed via the following link: https://github.com/mert-yigit/GISD

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents a brief introduction to our work, motivation, contribution, and organization.
- Chapter 2 gives comprehensive background information on our work.
- Chapter 3 provides an overview of the related works.
- Chapter 4 introduces our methodology.
- Chapter 5 includes the results of our thesis.
- Chapter 6 comprises a discussion of our research.
- Chapter 7 states the conclusion of the thesis.

2. BACKGROUND OVERVIEW

2.1. Autonomous Vehicles

The history of autonomous vehicles dates back to the 1920s. The first example of this technology was considered a remote radio-controlled vehicle named "American Wonder" [32] in 1925. Although there was no significant development until the mid-1980s, Carnegie Mellon developed an autonomous driving vehicle for the Navlab project in 1986. This vehicle is considered a critical milestone in the field of autonomous driving. In the 1990s, studies in this field were further developed with an autonomous vehicle called Autonomous Land Vehicle in a Neural Network (ALVINN) [33] by Carnegie Mellon. This vehicle is considered the first application of artificial intelligence in an autonomous vehicle. The turning point in autonomous vehicles was the competitions organized by DARPA in the 2000s. In 2004, the vehicles participating in the DARPA Grand Challenge accelerated the work in this field. Vehicles participating in this competition were expected to complete a 150-mile route in the Mojave Desert without human interaction. However, although no vehicle was successful in this competition, the vehicle named Stanley [34] from Stanford University successfully completed this route in the competition organized the next year. This success has attracted the attention of many researchers from both industry and academia, increasing the number of studies carried out in this field. In the 2010s, Waymo started to test autonomous vehicles that include multiple sensors such as LIDAR, camera, and radar. In this period, companies such as Tesla and Uber, which are major players in the industry, have also increased their investments in this field. Today, autonomous vehicles are one of the areas at the forefront of technological development. As can be seen in Figure 2.1, a modern autonomous vehicle is equipped with multiple sensors such as radar, LIDAR, and camera. Especially in the field of sensor technology, developments such as LIDAR technology becoming smaller and more accessible, and infrared cameras becoming more cost-effective make these vehicles safer in complex environments. It is worth noting that the main factor leading to the rapid development of these tools today is the work in the field of

artificial intelligence. Artificial intelligence algorithms improve the detection algorithms of these vehicles, and this is achieved through better object detection, classification, and scene understanding by training deep learning models on large datasets. For example, Waymo [35] is developing very advanced artificial intelligence models to enable its vehicles to drive safely in complex driving scenarios. Regulatory and safety standards continue to work on the control of autonomous vehicles, and governments are seeking to achieve a balance between innovation and safety.

Figure 2.1 A modern autonomous vehicle, the diagram is taken from [2].

2.2. Corner Cases in Autonomous Driving

As autonomous vehicles, equipped with a wide range of sensors and artificial intelligence models, become more and more present on public roads today, there is a very important but under-emphasized problem that these vehicles face more and more. This problem is the real-time handling of uncommon and unexpected scenarios called corner cases. Corner

case scenarios can cause accidents if not handled properly. As can be seen in Figure 2.2, a deer jumping out onto the road in limited visibility is very hazardous for traffic safety. Deep learning methods developed for visual perception fail to detect corner cases because corner cases are not encountered during training. In autonomous driving, corner cases are encountered at many different levels. In ref [36] categorize corner cases into the most detailed classes in the literature. It is possible to analyze at 5 different levels: scenario level, scene level, object level, domain level, pixel level. Scenario level corner case scenarios can be divided into 3 different classes. Anomalous scenarios, for example, are dangerous and unknown situations that are not encountered during training and have a high risk of accident, for example, a pedestrian who suddenly appears in front of us from the side of a large truck blocking our line of sight on the road.

Figure 2.2 The deer darted from the darkness, the diagram is taken from [3].

Novel scenarios, unknown situations that do not pose a danger, for example, we are going to another city, we need to cross the motorway and our autonomous vehicle knows only lane changes and turns in the training dataset, there is no information about access to the motorway. Risky scenarios, potentially dangerous and known situations, for example, we are passing through a narrow street and another car is coming towards us from the opposite side,

two vehicles approach each other while passing and there is no distance between them. At the scene level, there are 2 sub-categories: collectively, contextual anomaly, multiple people protesting on the road in the city center, and as a contextual anomaly, oil being scattered on the road and traffic cones being placed at the location of the incident as it is no longer visible. The presence of a bear in the center of the street at the object level in the settlement. At the domain level, for example, when our autonomous vehicle passes from Great Britain to Europe, the direction of traffic flow changes. It can be classified as a local outlier where there is an effect on the autonomous vehicle at the pixel level and situations that affect our global outlier sensors outside of us. As a local outlier, bad or dead pixels in our camera, as a global outlier, it can be said that the sun creates an overexposure effect on our camera after leaving the tunnel.

2.3. Infrared Imaging

Infrared imaging enables the detection and visualization of thermal radiation emitted by objects depending on their temperature. Unlike visible light, the infrared band cannot be detected by the human eye. Infrared technology is used in multiple fields from autonomous surveillance, medical diagnosis, and military applications to industrial monitoring. Infrared imaging works based on the principle of thermal radiation. Accordingly, all objects with a temperature above absolute zero emit thermal radiation. The density and wavelength of this radiation are related to the temperature of the object. Infrared cameras, also known as thermal cameras, detect this radiation and convert it into an electronic signal, which is then processed through a series of processes to provide a visual representation. The most common bit depths for infrared images, which are usually in digital form, are 8-12-16 bits. The color information is that if the camera is white-hot, hot objects are shown as white, and cold objects are shown as black, and if the camera is black-hot, hot objects are shown as black, and cold objects are shown as white. The electromagnetic spectrum includes a very wide range of bands and the infrared band range extends from 700 nanometres Near-infrared (NIR) to 1 millimeter Far infrared (FIR). Infrared imaging is subdivided into 5 sub-bands:

- NIR: is the infrared band closest to the visible spectrum in the range from 700nm to 1.4 μm and is used in applications such as night vision surveillance where illumination is required.
- MWIR: 3-8 μm range Used in military and industrial applications where very hot objects provide good thermal contrast.
- LWIR: is available in the range of 8-15 μm and is the most widely used for thermal imaging in the infrared band. It provides very successful results in areas where the amount of light is low or none.
- SWIR: is available in the range of 8-15 μm and is the most widely used for thermal imaging in the infrared band. It provides very successful results in areas where the amount of light is low or none.

The image highlights the infrared portion of the electromagnetic spectrum, dividing it into Near Infrared (NIR), Short-Wave Infrared (SWIR), Mid-Wave Infrared (MWIR), and Long-Wave Infrared (LWIR), with wavelengths ranging from 0.78 μm to 12 μm. It also shows that InGaAs detectors are sensitive in the NIR to SWIR regions (0.78–1.7 μm), making them useful for infrared imaging and sensing.

One of the main problems encountered during autonomous driving is to ensure safe driving in low light conditions such as fog, snow, and night. In such cases, infrared cameras are successful in detecting obstacles by emitting thermal radiation even in low light, unlike ordinary cameras and radars. Today, autonomous vehicles already make use of infrared cameras to navigate in low-light conditions. By monitoring and analyzing the images provided by these cameras, central decision-making units prevent collisions with objects and increase vehicle safety. Systems such as LIDAR in autonomous vehicles cannot work properly due to adverse weather conditions such as snow, rain, fog, and excessive sunlight. Input data may contain noise that cannot be analyzed. Infrared cameras, however, are less affected by such conditions because they work on thermal radiation, unlike visible light or radio signals. A car with a thermal imaging camera in the sensor suite gains the ability to gain

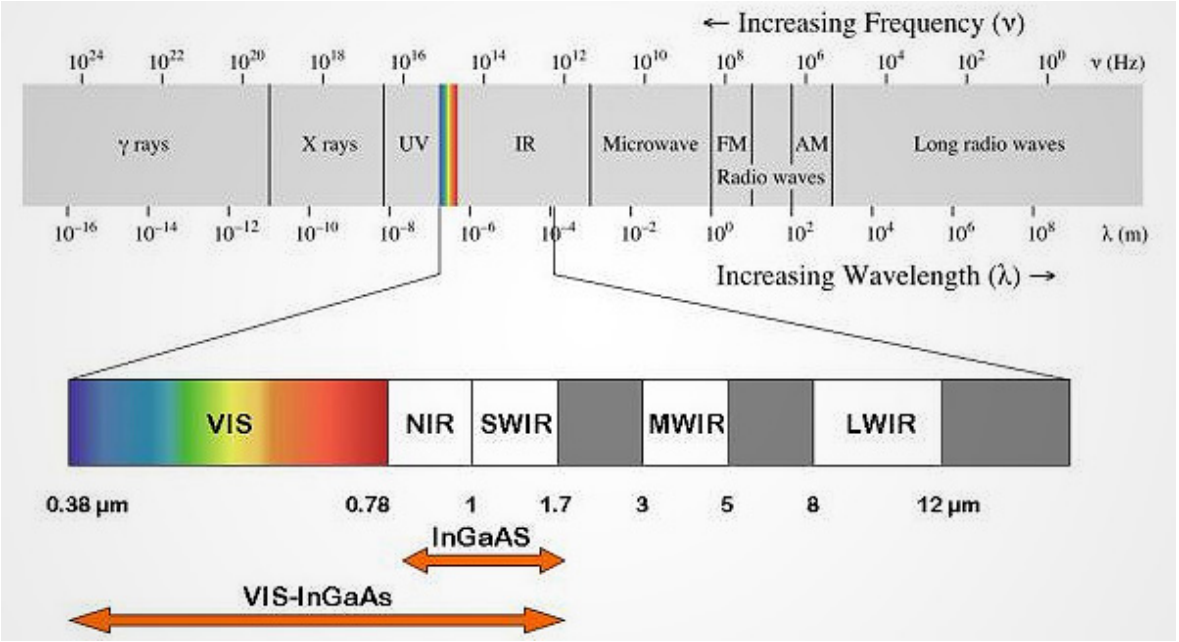

Figure 2.3 The image highlights the infrared portion of the electromagnetic spectrum, the diagram is taken from [4].

a deeper understanding of the vehicle environment. As this technology matures and becomes more affordable, autonomous vehicles will provide even higher levels of reliability and reliability and will become even more fully autonomous. In the future, especially with the widespread use of sensor fusion, autonomous vehicles will better utilize data from multiple sources to compensate for each other's weaknesses and provide a more accurate vision for the vehicle. With the advances in material technology, infrared cameras will become cheaper and will be more widely used in autonomous vehicles.

In particular, by performing sensor fusion, the autonomous vehicle improves its sensing and decision-making capabilities by combining information from multiple sensors. By combining the detailed color information provided by visible cameras with the image information provided by infrared cameras in low-light situations, both sensors complement each other's deficiencies in very different scenarios and increase the level of situational awareness.

2.4. Generative Models

Generative models are machine learning models designed to generate similar to the given training data. Unlike discriminative models, they aim to create new samples from the original data by learning the underlying probability distribution of the given training data, as opposed to being trained to classify and predict. Some common examples of generative models are Variational Autoencoders (VAEs) [37], Generative Adversarial Networks (GANs) [38], and Transformers [39]. Generative models are based on learning the relationship between latent space (z) and observed data (x). Latent space (z) is the representation of data at lower orders and the generative model aims to produce realistic images by learning to map from latent space to data space. Generative AI models are aimed at generating new original data from this data by recognizing the patterns in the data already obtained.

Figure 2.4 shows a Generative Adversarial Network (GAN) at the top, followed by a Variational Autoencoder (VAE), a Flow-based model, and finally a Sequential (autoregressive) model. Each model illustrates how data (x) is processed through latent space (z) and then transformed back, highlighting different architectures for generating data from latent variables.

GANS: First [38] introduced in 2014, GANs were the most popular generative AI models before diffusion models emerged. GANs are based on 2 neural networks competing against each other. While the generator creates new examples, the discriminator tries to distinguish whether the created content is real or fake. When the two models are trained together, with each iteration the generator starts to produce better and better content to mislead the discriminator and improve the results. Although GANs provide high-quality and fast results, the variability in the outputs produced is not very high, so GANs are generally preferred for domain-specific tasks.

VAE: It [37] consists of two neural networks, an encoder and a decoder. The encoder network aims to convert the training data into a smaller and denser form. The encoder and decoder work together to provide a more efficient and simpler latent data representation. While

VAE models produce images faster, the images produced are not as high quality as diffusion models.

Diffusion Models: Also known as diffusion probabilistic models generative models with a two-stage process in latent space in the training phase. The first of these two stages is forward diffusion and the second is reverse diffusion. In the forward diffusion stage, random noise is added to the training data, while in the reverse diffusion process, the noise-added data is tried to be reconstructed back. Diffusion models are slower to train than VAE generative models, but they provide better quality outputs due to the two-stage training process.

It should be noted that it is the transformer architecture that underlies the effective and rapid development of Generative AI models. Transformer networks are similar to RNNs [40], but they are developed based on non-sequential processing of sequential input data. It is suitable for Generative AI models with text input because it is based on two structures called self-attention and positional encoding. With these two structures, the model allows the model to focus on the relationship between the words in the given text even if there are very long distances between them. The self-attention layer assigns a weight to each part of the input, where the weight indicates the importance of the given part relative to the rest of the input.

Generative AI models can take very different inputs such as text, image, audio, video, and even code and convert them into very different forms. For instance, they can convert a text to an image, an image to a song, or a video to a text.

Language: The origin of many generative AI models, text is also the most studied and advanced in this domain. One example of this is large language models, known as LLMs, such as GPT-2 [41], and BERT [42] which have multiple uses from code development to language translation.

Voice: Generative AI models have been used in recent years in multiple areas such as song development, voice assistants [43, 44], and voice recognition.

Visual: Generative AI is one of the most popular areas of use. There are multiple uses [45, 46] such as the creation of 3D images, and the production of artistic artifacts.

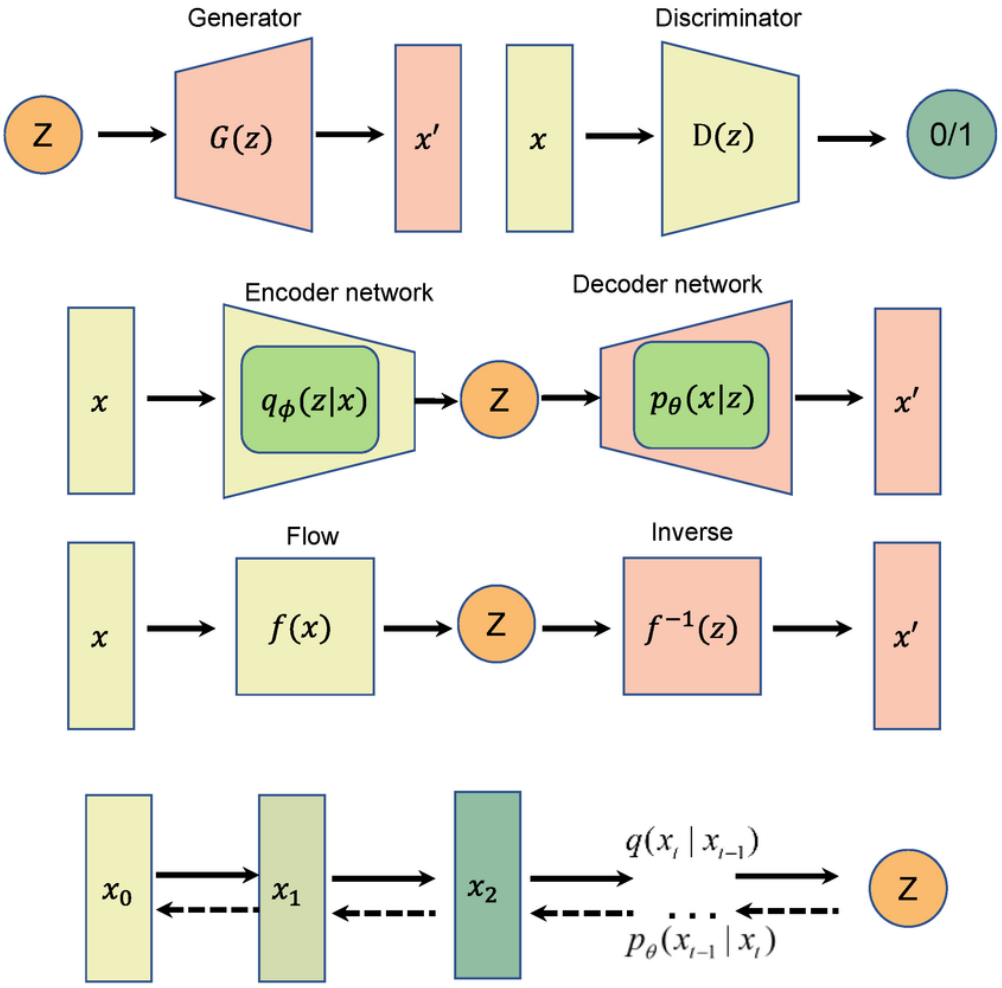

Figure 2.4 This image compares the schematic structures of four generative models, the diagram is taken from [5].

Synthetic data: It is especially effective when we do not have data to train AI models. It is a very advantageous area of use for conditions where it is very difficult or costly to generate data. There are also areas of use such as creating new medical images [47], drug discovery [48], and chemical molecules.

2.5. Stable Diffusion

Stable diffusion is a relatively new and recently popular machine learning model among generative AI models. Stable diffusion, one of the diffusion models, is very interesting because it produces high-quality, realistic pictures, sound, and even text. The diffusion

process at the core of stable diffusion aims to gradually transform noise into a meaningful output over a series of iterative steps. With this feature, it differs from classic GANs. The diffusion process consists of two stages:

- Forward Diffusion Process: Over a period of time, small amounts of noise are gradually added to a meaningful input, which may be a picture. This means that the meaningful data is iteratively corrupted with noise over time. After some iteration, the meaningful data resembles noise. In short, in forward diffusion, the data is converted into noise in a controlled manner. As can be seen in Figure 2.5, noise is added to the image repeatedly in the forward diffusion part.
- Reverse Diffusion Process: After the meaningful data is completely converted to noise, the model trained for the reverse process tries to make the noisy output meaningful.

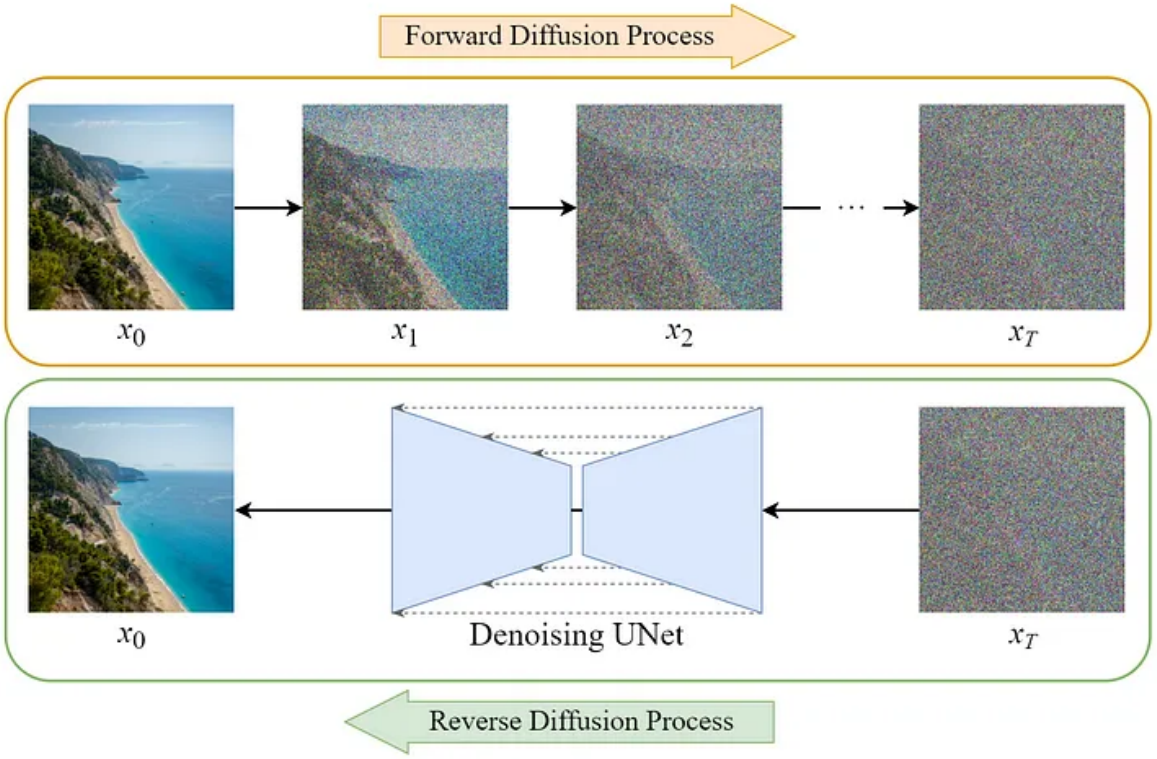

Figure 2.5 This diagram illustrates the architecture of a diffusion-based generative model, the diagram is taken from [6].

For this two-step process to be successful, stable diffusion models are trained on very large datasets. In the training phase, the model learns the complex patterns in the data and in this way, it provides very high-quality outputs from noise during the reverse diffusion process.

Stable diffusion is based on the U-Net [49] architecture. As it is known, U-Net consists of two parts: encoder and decoder.

- Encoder stage: The noisy input is downsampled through a series of convolution layers, and at each layer, the features of the input are extracted. From the spatially scaled input, the model learns global features.
- Latent Space Representation: The model learns the latent representation of the data after the encoding phase. Since the latent space contains the essential information of the input data, the decoder can perform the reverse diffusion process.
- Decoder: the process of gradually converting the latent representation into a high-resolution image.

Conditioning: A critical feature that Stable diffusion has is that it generates images depending on extra inputs. This process is achieved through cross-attention. In cross-attention, the model provides conditioning by checking whether the image produced aligns with the text prompt provided while the image is being generated.

Stable diffusion models play a very important role in generating datasets. High-quality datasets containing a wide variety of samples are very important for training a model. However, producing such a dataset in the real-world is both expensive and time-consuming. Stable diffusion can produce new data to fill this gap by imitating real-world data.

Figure 2.6 depicts where data in pixel space (x) is encoded into a latent space (z) via an encoder ($\mathcal{E}$), followed by a denoising process to progressively remove noise through a U-Net. The model integrates cross-attention mechanisms to condition the generative process on semantic maps, text, representations, or images, enabling controlled generation during the diffusion and denoising steps.

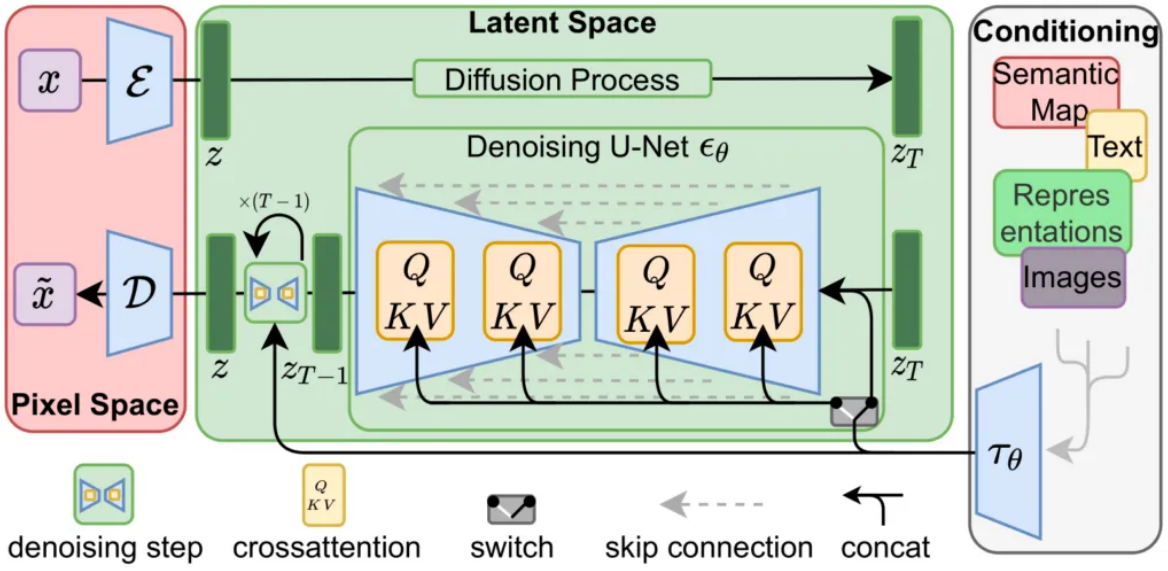

Figure 2.6 This diagram illustrates the architecture of a diffusion-based generative model, the diagram is taken from [7].

2.6. LORA

LORA(Low-rank adaptation) allows fine-tuning of large language and visual models using memory and computational time efficiently. Fine-tuning means customizing or adapting the pre-trained model for a specific task so that it can produce good, close results to the input. First mentioned in this work [50].

Large models have millions of weights and when training these models, normally millions of parameters have to be relearned, which is undesirable both in terms of time and computational resources used. LORA proposed the Low-Rank adaptation method to solve this difficulty. LORA aims to keep the number of parameters to be trained in a generative model as small as possible while at the same time not losing performance.

LORA can work on specific layers, applying only to certain parts of the model. For instance, in the scenario where we want to customize the stable diffusion model in the hand drawing area, the model must be re-trained.

In this case, either the entire model can be trained for a long time on hardware with very high computation power, or the model can be specialized in this area without memory overhead

on machines with faster and more limited computation power thanks to LORA. For LORA training, firstly a pre-trained model is needed. This model already has general information, but it is not specialized for this task. Then all the weights of the model are kept frozen, i.e. the weight information of the model is kept unchanged. Only the weights that we want to customize are changed by converting them into low-rank matrices. Accordingly, the weights in some special layers of the model mostly aim to represent the attention layers with low-rank matrices utilizing low-rank decomposition.

$$\mathbf{W}_{\text{adapt}} = \mathbf{W} + \mathbf{AB} \tag{1}$$

In this equation:

- $\mathbf{W}_{\text{adapt}}$ is the adapted weight matrix.
- $\mathbf{W}$ is the original weight matrix.
- $\mathbf{A}$ and $\mathbf{B}$ are low-rank matrices that are learned during the adaptation process.

$$\min_{\mathbf{A},\mathbf{B}} \mathcal{L}(\mathbf{X}, \mathbf{Y}; \mathbf{W} + \mathbf{AB}) \tag{2}$$

Where:

- $\mathcal{L}$ is the loss function,
- $\mathbf{X}$ is the input data,
- $\mathbf{Y}$ is the target output,
- $\mathbf{W}$ is the original weight matrix,
- $\mathbf{A} \in \mathbb{R}^{m \times r}$ is a low-rank matrix,
- $\mathbf{B} \in \mathbb{R}^{r \times n}$ is another low-rank matrix.

As a result of these low-dimensional matrices, less memory space is occupied and fewer calculations are performed.

Self-attention layers in transformer-based models require large matrix multiplications.

One of the hyper-parameters chosen in LORA is the size of the low rank. Small r values produce faster and relatively lighter models, but if this r value is chosen large, LORA produces more detailed results while the computational overhead is high. Only low-rank matrices are updated in the back-propagation process.

Compared to adapter methods, LORA does not require additional parameters to be trained, since it does not change the original structure of the model compared to adapter methods.

Algorithm 1 LoRA Training Algorithm

Input: Pre-trained model M, training dataset D, rank r, learning rate η, number of epochs E
Output: Adapted model M'
Initialize low-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$, where d is the dimension of the original weight matrix
Freeze the original weights of the model M
for each epoch $e = 1$ to E **do**
  for each batch B in dataset D **do**
    Forward pass: Compute output O using M and low-rank matrices A and B
    Compute loss L based on the output O and ground truth labels
    Backward pass: Compute gradients ∇L with respect to A and B
    Update low-rank matrices:
      $A \leftarrow A - \eta \nabla_A L$
      $B \leftarrow B - \eta \nabla_B L$
  end for
end for
Construct adapted model M' using original weights and low-rank matrices
return M'

2.7. Object Detection

Object detection is the process of both labeling and locating objects in an image. The objects in the image are enclosed in boxes called bounding boxes. From satellite imagery, security cameras, and autonomous vehicles to civilian and military applications, it appears almost

everywhere. Object detection has developed considerably, especially due to developments in the field of deep learning. While classical methods use algorithms such as HOG [51] and SVM [52], deep learning has led to the development of artificial neural networks such as CNN.

YOLO architecture is mainly based on the working mechanism that first divides an image into grids, each grid is responsible for finding the object within it. Here, finding the object means finding the class to which the object belongs and the x-y coordinates of the bounding box. YOLO is based on two architectures: backbone and neck. YOLO uses the backbone as a feature extractor, the given input image is processed and output as a feature map. YOLOv8 [8] uses deep CNN networks as a backbone, the backbone is usually based on the Darknet-53 architecture, while YOLOv8 [8] uses depthwise separable convolution networks such as efficient MobileNet, thus ensuring the number of parameters and computational efficiency. With a Depthwise separable network, instead of applying convolution to the entire input volume, the convolution process is applied to each input channel independently, and then pointwise convolution is applied. YOLOv8 [8] also uses residual connections as the backbone, inspired by ResNet [53], which helps to avoid the vanishing gradient problem. YOLOv8 [8] as Neck allows the multiscale feature fusion of different feature maps. They use it to offer a better feature fusion with FPN and PAN. The most significant change in YOLOv8 [8]is the introduction of anchor-free detection, which has led to significant improvements in the detection of small objects. When the development of YOLO models is reviewed. The first version of YOLO architecture, YOLOv1 [19], was introduced by Joseph Redmon and Ali Farhadi in 2015. Unlike traditional (R-CNN, faster R-CNN [54]), which aims to perform fast object detection using a single CNN, YOLO does not classify each region separately. It operates by looking at the whole image once. While YOLOv1 [19] is reasonably fast compared to other models of its time, it is weak in detecting small objects. The model known in the literature as Improved YOLO in 2016 was introduced to improve the previous version. In this context, anchor boxes were started to be used based on faster R-CNN [54]. Faster and more precise predictions by utilizing anchor boxes for specific object sizes. YOLOv2 [55] also avoids overfitting by using batch normalization. It [56] is

faster and more effective compared to previous versions due to being built on the Darknet-53 backbone. For the first time, multi-scale prediction, i.e. the detection of objects of different sizes at the same time, was also achieved in this model. Developed by Alexey [57] and Hong Yuan in 2020. The mish activation function it uses is an important difference compared to other previous models, and it uses spatial pyramid spp to detect objects of different scales. Despite not being an official YOLO version, it was developed by Ultralytics [58] based on Pytorch. It was the most popular among the models released up to that time with its easy use thanks to Apisi, offering models for different needs in different sizes. It was designed by Bochrovsky [59] and his team to improve on previous models. A dynamic anchor box is an important feature. YOLOv8 [8] provides significant advances in speed and accuracy with its anchor-free approach that improves the accuracy of the model while providing lower computation costs.

As can be seen in Figure 2.7, the YOLOv8 [8] architecture consists of several parts.

Backbone: YOLOv8 [8] uses a CSPNet-like network as a backbone network that extracts features from the input image

Neck: It creates a rich pyramid of features by combining features from different layers of the backbone, usually PANet or FPN, allowing objects of various sizes to be detected.

Head: This is the part of the model where the actual detection takes place. Bounding boxes, classes, and object scores are calculated in this part.

Loss Function: YOLOv8 [8] aims to improve overall performance by utilizing the custom loss function.

Inference: In a single pass, the model image is processed to make it fit for real-time applications.

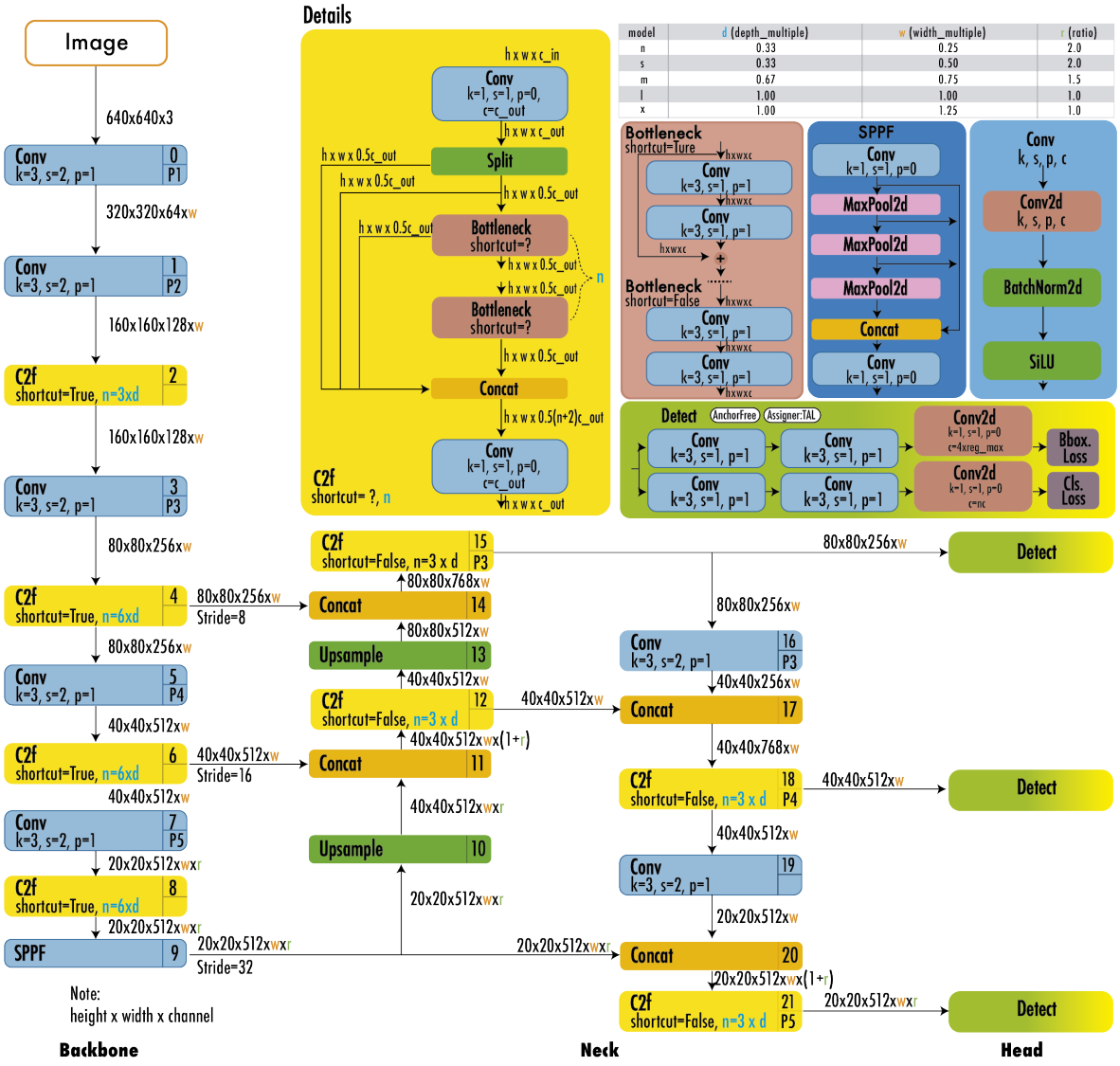

Figure 2.7 This diagram represents the architecture of YOLOv8 [8], showcasing its convolutional layers (Conv), C2f modules, Bottlenecks, Spatial Pyramid Pooling-Fast, and detection head for object detection across different scales, the diagram is taken from [9].

3. RELATED WORK

Since there is no thermal dataset containing corner cases and no work on corner case detection on such a dataset, this section will focus on corner case studies in the visible domain. There is some work in the visible domain to create and detect corner case scenarios using generative language models. After discussing these studies, some of the most widely used thermal infrared datasets in the field of autonomous driving are mentioned.

3.1. Corner Case Creation with Stable Diffusion

3.1.1. DriveDreamer: Towards Real-World-driven World Models for Autonomous Driving

In Ref. [10] argues that the autonomous datasets available in the real-world are not as complex as the driving scenarios encountered in the real-world. For this purpose, many different driving video generation processes were performed by using stable diffusion in many different scenes. Figure 3.1 shows some examples generated in the dataset.

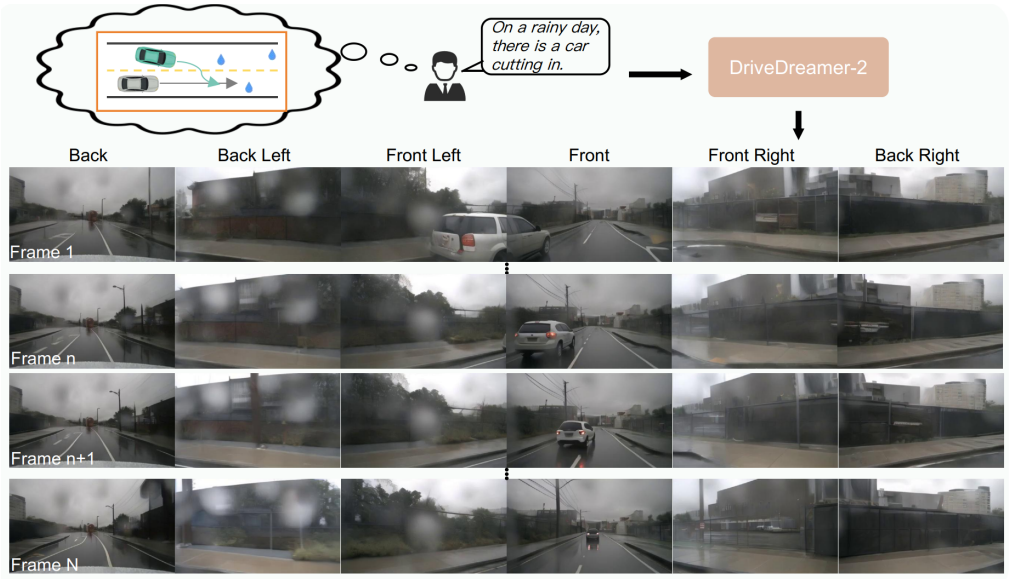

Figure 3.1 Example image from DriveDreamer, the diagram is taken from [10].

3.1.2. WEDGE: A multi-weather autonomous driving dataset built from generative vision-language models

Ref. [11] is a synthetically generated dataset in the visible domain to improve autonomous driving in very extreme weather conditions. Figure 3.2 shows some examples generated in the dataset.

Figure 3.2 Example images from the WEDGE dataset, the diagram is taken from [11].

3.1.3. Time to Shine: Fine-Tuning Object Detection Models with Synthetic Adverse Weather Images

Time to Shine [60], another study that aims to create a dataset in very challenging conditions, aims to create extreme weather conditions for autonomous driving such as rain, snow, and fog by utilizing Midjourney [61].

3.2. Corner Case Detection on Visible Domain

3.2.1. Pixel-wise Anomaly Detection in Complex Driving Scenes

For safe autonomous driving, authors [12] performed pixel-wise anomaly detection at the granular level. Current semantic segmentation methods have some limitations in anomaly detection in complex driving scenarios. In this study, the authors present a framework by improving the uncertainty map with re-synthesis methods.

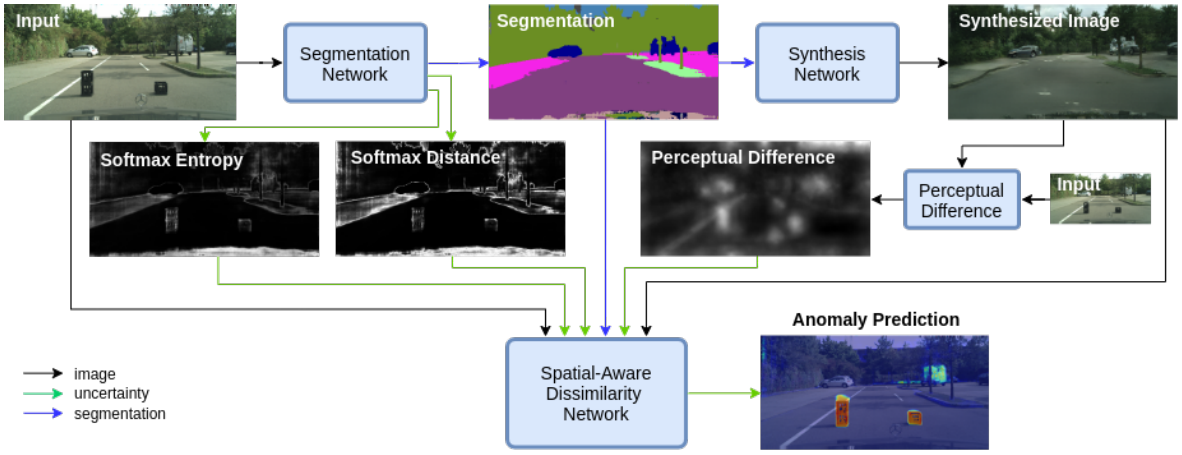

Figure 3.3 Anomaly Segmentation Framework, the diagram is taken from [12].

3.2.2. Conditioning Latent-Space Clusters for Real-World Anomaly Classification

In this paper, the authors tried to find the difference between normal and abnormal states by clustering similar points in latent space.

The novelty presented in this work [13] is the proposal of a framework that provides normal and abnormality by conditioning the latent space of the VAE. In addition to this method, which is very effective in small-scale anomalies, it is aimed to improve anomaly detection without deteriorating the image reconstruction quality with the discrepancy map. The authors tested their method on different datasets such as Cityscapes [62] and Fishyscapes [63].

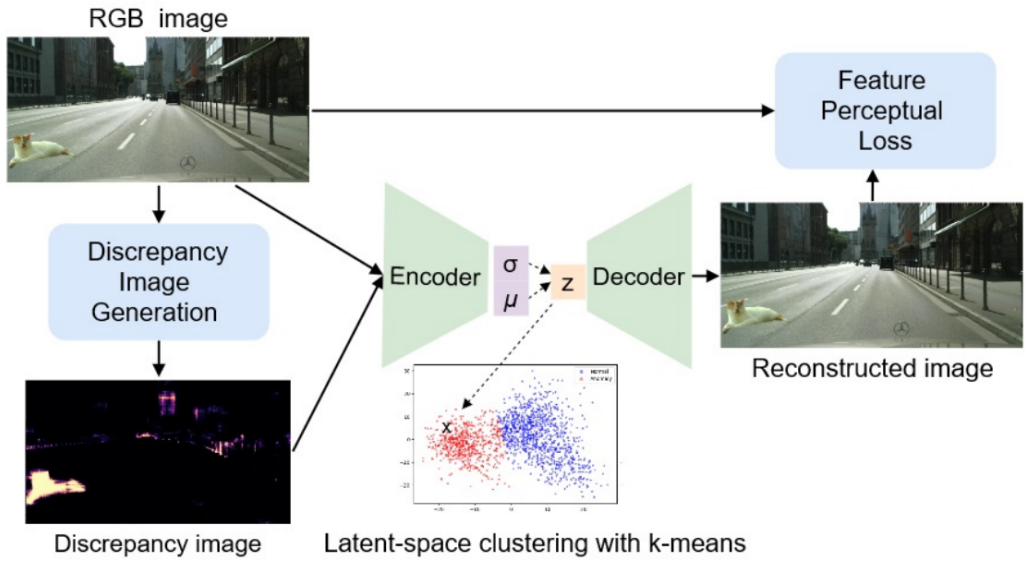

Figure 3.4 VAE-based method is proposed in for real-world anomaly classification, the diagram is taken from [13].

3.2.3. HazardNet: Road Debris Detection by Augmentation of Synthetic Models

The debris on the roads is very dangerous for autonomous driving and traditional methods fail to handle it, so they [14] synthetically created realistic road debris scenarios in a simulation environment. They then tested the CNN-based model they developed both in the real-world and in their datasets.

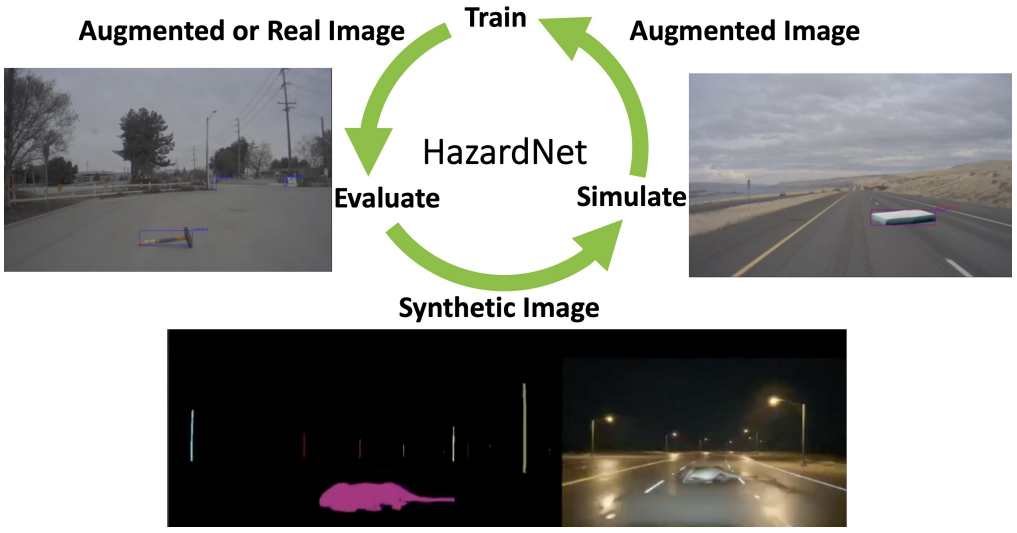

Figure 3.5 HazardNet, the diagram is taken from [14].

3.3. Thermal datasets for autonomous driving

3.3.1. KAIST Multispectral Pedestrian Detection Benchmark Dataset

The KAIST [15] dataset is very comprehensive for algorithms to be developed and tested in the field of autonomous driving, containing approximately 95000 images in 640x480 size, including many different scenes. The dataset, which offers multi-modal data to researchers working in the field of autonomous driving, especially in pedestrian detection, is publicly available in an annotated form.

Figure 3.6 A sample infrared image from the KAIST [15] dataset.

3.3.2. FLIR ADAS

FLIR ADAS [16] dataset is one of the most used datasets by researchers in computer vision studies in the field of autonomous driving. It contains high-resolution thermal and RGB images at different times of the day, both during the day and the night. The dataset contains many different classes such as pedestrian, vehicle, and cyclist in these images.

Figure 3.7 A sample infrared image from the FLIR ADAS [16] dataset.

3.3.3. SODA

The SODA [17] dataset is specially prepared for the development of object detection algorithms in the IR domain. It contains high-resolution IR images taken at night and daytime. It is especially important for semantic segmentation studies in the field of thermal imagery. Consists of indoor and outdoor objects such as people, buildings, trees, roads, poles, grass, doors, tables, chairs, cars, bicycles, lamps, monitors, traffic cones, trash cans, animals, fences, sky, and rivers. The dataset contains a total of 2168 annotated images, including 20 different classes of semantic region labels.

Figure 3.8 A sample infrared image from the SODA [17] dataset.

4. PROPOSED METHOD

This part of the thesis covers the creation of corner case scenarios and the detection of these corner cases. In the first part, the generation of corner case scenarios is discussed. In the second part, the detection model is trained by using the generated dataset which contains corner cases. By using this detection model, the detection of corner case scenarios both in real life and on the new dataset created has been performed with high prediction scores.

The image generation phase of this thesis can be examined in four sub-headings. Firstly, we will mention the step of determining the objects that create a corner case. In this step, the objects that cause the corner case scenario are selected considering their rate of causing traffic accidents. According to the AAA report [64], more than 200,000 crashes per year involve debris on the roadway as a contributing factor. Therefore, in this thesis, because road obstacles cause so many accidents, 4 different classes of objects were selected deer, dog, trash bin, and traffic cone. The selected objects are from publicly available sources and images from the SODA [17] dataset were used. The traffic cone and trash bin objects in the SODA [17] dataset are not present in this dataset in a format that creates a corner case. However, the fact that they can be found in different variants at different distances for LORA training makes the images with these objects suitable for LORA modeling. Another novel aspect of this work is that the characteristics of an object in one dataset can be learned and transferred to another dataset. In other words, the LORA model of a traffic cone in the SODA [17] dataset can learn important features such as thermal characteristics and shape and generate this object in a newly masked area in another dataset with an inpainting process. In this way, it was possible to generate corner-case scenarios with an object whose real-world characteristics were learned.

Insurance Information Institute reports [65] that 1.5 million deer-vehicle collisions occur each year, resulting in 150 deaths and over $1 billion in vehicle damage. Deer primarily pose a risk due to their sudden appearance on roads, especially in rural and forested areas. In conditions such as fog or night with low visibility, deer are either not detected or detected too

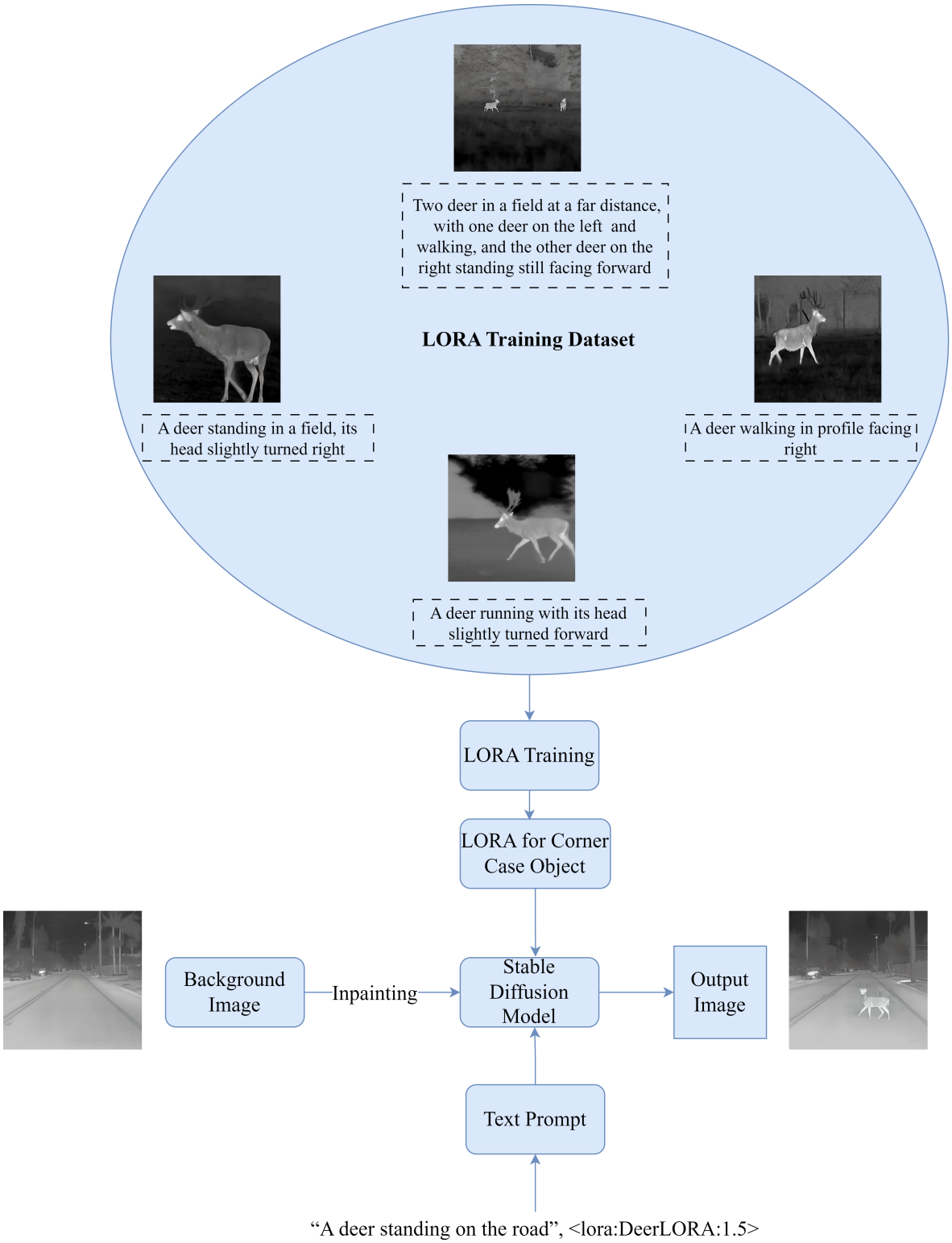

Figure 4.1 GISD workflow.

late by vehicles. Serious accidents also occur, especially as a result of a very fast collision. In contrast to visible cameras, infrared cameras, on the other hand, operate successfully even in conditions of very low visibility, enabling detection. However, since there are no corner cases such as deer in an autonomous driving dataset, the detection of these creatures with algorithms running on this camera is a subject that has not been studied. Deer can be detected by thermal cameras from very long distances, even in extreme weather conditions, before they even get in front of the vehicle. For this reason, we created corner case scenarios by including deer class in our dataset from very different distances in different poses and even in different variants, for example, female and male.

As the second object, the dog class is included in this thesis. They pose a threat to the safety of autonomous driving due to their unpredictable behavior and frequent presence in traffic areas. Other reasons for the selection of the dog class are their sudden emergence into traffic, chasing moving vehicles, and their unpredictable and unpredictable behavior, unlike other animals. Many sensors on autonomous vehicles can recognize objects in their environment, but these objects are usually static. Dogs, on the other hand, are difficult for algorithms running on these sensors to detect because they create dynamic situations that autonomous vehicles have not seen before with behaviors that do not fit a pattern. In addition, although autonomous vehicles have sensors such as radar and LIDAR, it is difficult for these sensors to detect dogs in real time due to their relatively small and fast movement. In addition, if limited visibility is added to these conditions, the detection of this class is very challenging. Infrared cameras, on the other hand, allow the detection of dogs that are partially visible or behind an object, even in very limited visibility conditions, regardless of the weather conditions at night. Thus, corner case scenarios such as a dog that may get on the road in advance or a dog waiting behind a tree are no longer dangerous. In our dataset, we have created many such scenarios using stable diffusion for the dog class.

Another object in our dataset that creates a corner case is a traffic cone. Traffic cones are difficult to recognize by autonomous vehicles due to their size and shape. The reason for this is that algorithms working on sensors such as LIDAR are generally developed for the detection of objects such as larger vehicles and pedestrians. In addition, in difficult weather

conditions or low light conditions, visible cameras also have difficulty detecting traffic cones. This causes the traffic cone to be misinterpreted in a complex environment when it indicates an event. For example, if a traffic cone, which is placed to indicate that there is an accident ahead, indicating that the road is closed or divided, is not properly interpreted, it poses a great hazard to traffic. Unlike other objects in traffic, they can dynamically change their location, which is another risk for autonomous vehicles. Dynamic displacement is the intentional movement of an object (e.g. moving a pothole to another part of the road) and the unintentional movement of a traffic cone by external factors such as wind. A robust model that can recognize traffic cones in such situations plays an important role in improving the safety of autonomous driving. Therefore, in our dataset, for the first time in the literature, we have created corner case scenarios with thermal traffic cones of different sizes, numbers, and locations.

Another object that appears harmless in traffic but is very dangerous for autonomous vehicles are trash bins. They can be found in a wide variety of different forms, from those for very different household waste to large-sized ones for very large industrial waste. Trash bins are often difficult to detect due to their obscured appearance, as they are often found behind a parked vehicle or tree. As in the case of traffic cones, a trash can that suddenly slides on the autonomous vehicle from a position such as slipping, wind, etc. can cause the autonomous vehicle to brake suddenly, change direction, and cause very serious consequences. If the autonomous vehicle cannot detect the correct position of the trash bin, a collision will result, so the fourth and last object that we consider important to have in our dataset is the trash bin.

The second step in creating the dataset is the background images where we will place the corner case objects. The reason for choosing the background from a real infrared autonomous dataset is to provide a dataset that is very close to the real world without the models trained with datasets created in the synthetic environment called the simulation gap being affected by the decrease in their performance in the real world. We minimized this loss by using a real infrared autonomous dataset as the background image. The background images used in this context were selected from the FLIR ADAS [16] dataset. We chose the FLIR ADAS [16] dataset because it is very popular in the field of autonomous driving, contains many different

scenes, and contains images at different times of the day and night. However, this dataset does not include corner case scenarios. 1000 images were randomly selected from the FLIR ADAS [16] dataset. We randomly selected 1000 images for the generation of corner case scenarios in the FLIR ADAS [16] dataset. The selected images should not be consecutive frames, because consecutive frames reduce the scene diversity.

Stable diffusion models generally do not have infrared domain knowledge because these models are usually trained with visible images and as a result, if an infrared image is generated using these models, the thermal radial distribution creates inaccurate images. There are 2 ways to generate corner case scenarios by teaching the infrared domain information to the stable diffusion model. The first of these is to train the model with very large thermal datasets for long periods on high-resource hardware, and the second is to train an existing model with a fine-tuning method such as LORA with captioned images in short periods with limited hardware. To train LORA, it is first necessary to collect sample images for the classes identified in Step 1. The images should be taken from different angles and at different distances from the object as much as possible so that the LORA model class does not over-train and can generate a more generic approach when learning.

After collecting the images to create LORA in step 3, the caption *.txt* file describing each image should be collected in the same directory as the image. After a series of parameters such as image repeats, number of epochs, learning rate, and trigger keyword are selected appropriately, the training process is performed on the pre-trained model and LORA is created.

As a final step for image generation, our method leveraged the inpainting. Inpainting is the process of modifying or filling a selected area of the image as specified in the text prompt. The created mask can be used for different purposes such as adding details, deleting objects, and adding new objects by applying to the specified region of the image. To perform this process, the area where the stable diffusion model will apply noise, i.e. the mask, must first be selected.

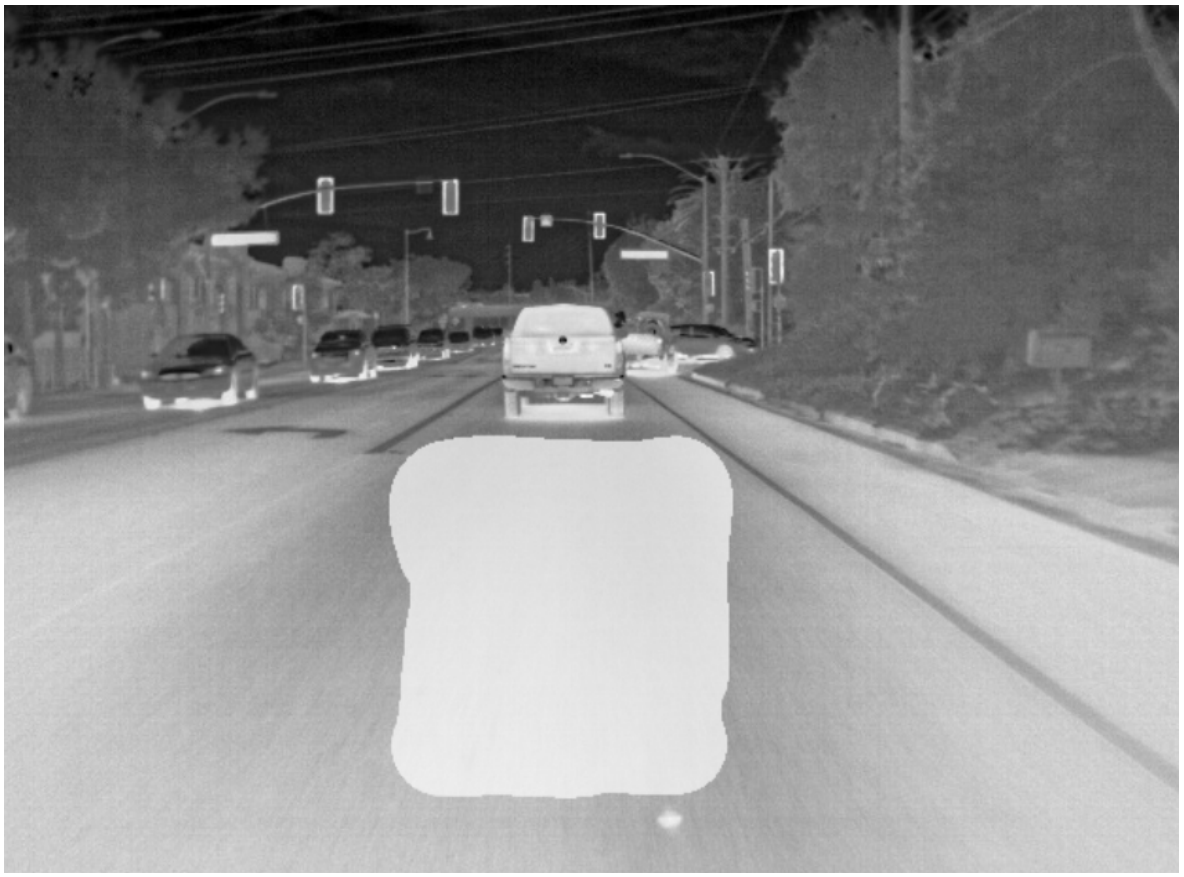

Figure 4.2 The process of masking where the object will be generated during inpainting with stable diffusion.

Once the mask is applied, the stable diffusion model iteratively adds noise to the masked area and tries to generate the content of the input specified in the text prompt in the masked area in a way that preserves the consistency of the rest of the image.

"A deer standing on the road", <lora:DeerLORA:1.5>"

"Three traffic cones are placed in a line along the road", <lora:TrafficConeLORA:1.5>"

"A dog sitting on the next to the sidewalk", <lora:DogLORA:1.5>"

"A trash bin on the road", <lora:TrashBinLORA:1.5>"

A thermal infrared autonomous driving dataset containing corner cases has not been created before, nor has a model been developed to detect these corner cases. In this thesis, for the first time in the literature, we trained a model for corner case detection on an autonomous

dataset. By training the YOLOv8 detection model based on the YOLO architecture on the dataset we created, we performed the evaluation process both on the test dataset and on real-world images. Before proceeding with the training, the thermal infrared autonomous driving dataset must be made suitable for training. The first of is to make the folder structure suitable for YOLO training. For each image, a .txt file with the same name as the image in the respective folder(train, val, test) should contain information about the object as follows:

$$\text{class} \quad x_{\text{center}} \quad y_{\text{center}} \quad \text{width} \quad \text{height}$$

- **class**:
  - This is an integer representing the class of the object detected. The class index corresponds to the order of the class names in the `label.txt` file, starting from 0. For example, if "person" is the first class in `label.txt`, it is represented by 0.
- $x_{\textbf{center}}$:
  - This parameter represents the x-coordinate of the center of the bounding box around the detected object. It is normalized by the width of the image, meaning it ranges from 0 to 1. A value of 0.5 would indicate that the center of the bounding box is in the middle of the image.
- $y_{\textbf{center}}$:
  - Similar to x_{center}, this parameter represents the y-coordinate of the center of the bounding box. It is also normalized by the height of the image, ranging from 0 to 1. A value of 0.5 would indicate that the center of the bounding box is vertically centered in the image.
- **width**:

  - This parameter indicates the width of the bounding box around the detected object. It is normalized by the width of the image, so it also ranges from 0 to 1. A width of 0.1 means the bounding box is 10% of the image's width.

- **height**:

  - This parameter represents the height of the bounding box. Like width, it is normalized by the height of the image, ranging from 0 to 1. A height of 0.1 means the bounding box is 10% of the image's height.

Before training the model, we labeled the images in the dataset. We performed the labeling process using the LabelImg [66] tool. Labeling is annotated to show which class the object in the image belongs to. After the corner case, objects in the images in the dataset were labeled with the classes they are related to, they were divided into 3 different groups train validation, and test. The created dataset has 215 images in the train part, 15 images in the validation part, and 10 images in the test part. While making this separation, it was ensured that there were an equal number of pictures from each class. The reason for this is that there is a risk that the model may over-train by seeing more images than one class. Afterward, a variety of different sizes of YOLO and different batch sizes were trained. The environmental parameters used for training are shown in Table 4.1.

Model Names	**YOLOv8 Nano/Small/Medium**
CPU	Intel(R) Xeon(R) CPU @ 2.20GHz
GPU	Tesla T4 TPU
Memory	16 GB
Learning Rate	0.00125
Batch Size	16
Epoch	50/75/100

Table 4.1 Environmental parameters.

5. EXPERIMENTAL RESULTS

In this part of the thesis, the results of image generation are discussed and a detection model trained with this dataset is tested both on the test part of the GISD dataset and on images taken from the real world.

To obtain corner case scenarios in a thermal autonomous dataset with stable diffusion, LORA and inpainting were used to generate 1000 different images for deer, dog, trash bin, and traffic cone classes, 250 images from each class in 512x512 size. The objects that create the corner case have many different variants of size, location, and action type in the data set.

Normally, since the datasets created in the synthetic environment do not use real-world images during the generation process, a loss called a simulation gap occurs in the models using these datasets compared to their real-world performance. However, in our thesis, since we used background images from a dataset containing real-world images, and since real images were used while creating LORA models, we minimized the loss due to the simulation gap. The classes in the dataset are equally distributed.

Figure 5.1 depicts the generation of 4 different deer corner case scenes for the same background using DeerLORA. The generated LORA model can augment the deer object in the scene in different poses and sizes.

Figure 5.2 shows 4 different corner case scenes created for deer, dog, traffic cone, and trash bin objects.

Figure 5.3 shows the corner case scenario by augmenting the object to the scene with the text prompt *Deer at standing on the road,<lora:DeerLORA:1.5>* given to the stable diffusion model of DeerLORA trained with images obtained from open source sources.

DeerLORA:1.5 where 1.5 is the scaling factor of the LORA model. The model aims to generate a final output using the LORA parameters at a weight of 1.5. The higher the value, the greater the impact of the LORA model on the original model.

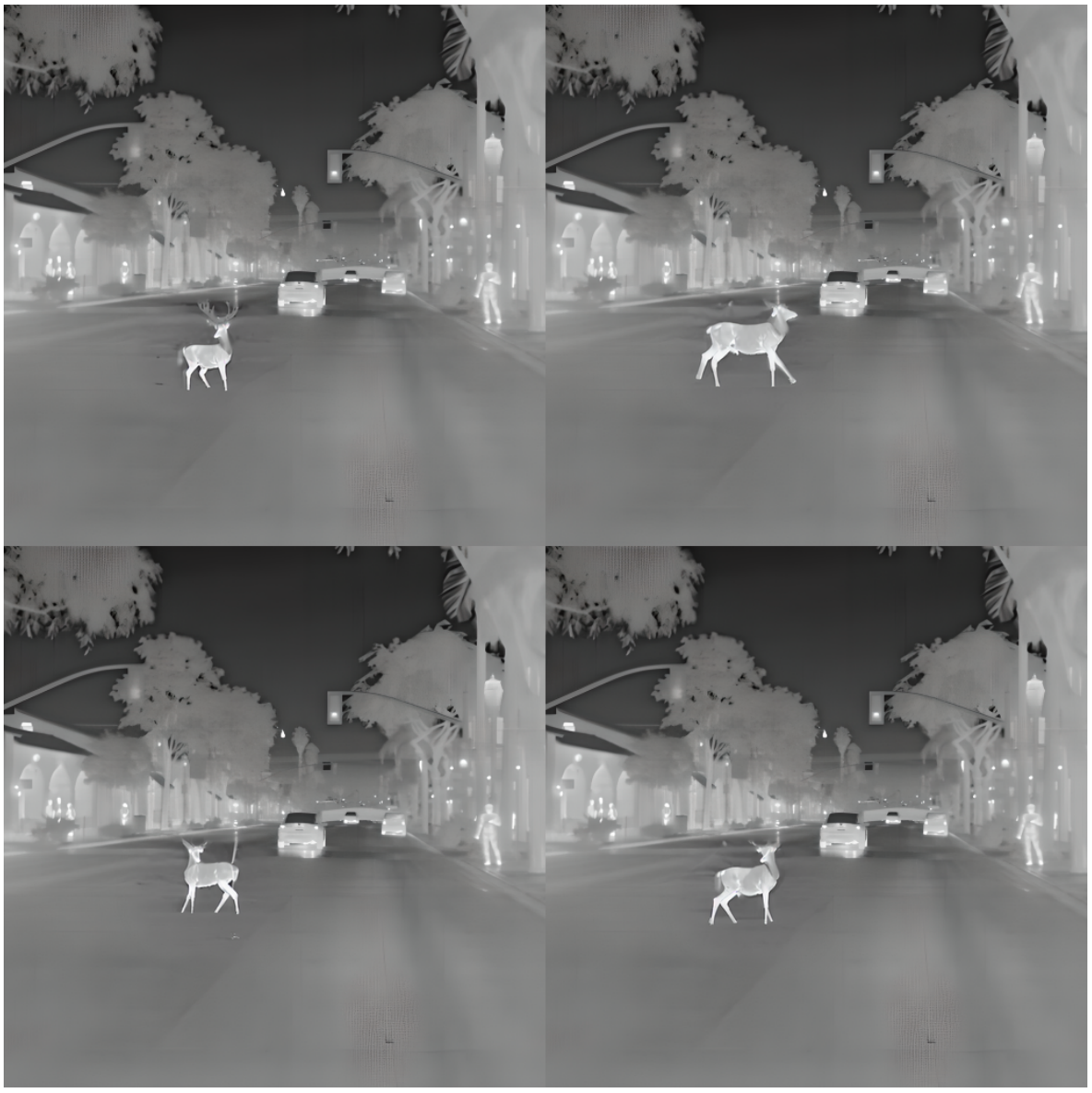

Figure 5.1 This image represents the augmentation of deer LORA with different poses, and sizes on a single background with the text prompt: ”Deer at standing on the road,<lora:DeerLORA:1.5>”.

The LORA models created for each object were augmented to different scenes by applying masks to the corner case objects with the inpainting process to the relevant scenes with the text prompts entered. Figure 5.4 shows the augmented version of the dog object in the scene.

Figure 5.2 This image represents scenes with different distances, sizes, and numbers produced from 4 different classes.

Figure 5.5 shows that the trash bin object is augmented to the scene to create a scene with a corner case. One value point to notice here is that the stable diffusion model generates the object in a way that preserves the consistency of the scene while LORA augments the object to the scene. The LORA model augments the object to the scene during the inpainting process and teaches the stable diffusion model the thermal characteristics of the object to be generated in a way that is consistent with the rest of the scene.

Figure 5.3 Generation of deer with stable diffusion.

As can be seen in Figure 5.6, the corner case scene was obtained by augmenting the traffic cone objects to the scene. Since the number of objects given in the text prompt specifies the number of objects to be augmented to the scene, corner case scenes with a wide number of objects could be generated.

Another point that we have seen as a result of our work is that the LORA model not only learns the thermal characteristics of objects but also how multiple objects are positioned

Figure 5.4 Generation of dog with stable diffusion.

relative to each other. This allows the traffic cone objects to be placed in the scene using this information as in the real-world images it was trained on.

Figure 5.7 shows the performance of the YOLOv8 model trained with the GISD dataset. Detection was performed on real-world images of deer on the road or crossing the road. The detection model correctly classified the deer class that created a corner case and was able to detect it with a high prediction score.

Figure 5.5 Generation of trash bin object with stable diffusion.

Detection experiments were performed with different-size models of the YOLOv8 architecture. Improvements in detection results were observed when the model size was changed from YOLOv8 Nano to YOLOv8 Small. However, a decrease in detection results was observed when moving from the YOLOv8 Small model to the YOLOv8 Medium model. According to our analysis, the reason for this decrease in performance is that the size of the dataset is insufficient for the size of the YOLOv8 Medium model. Since YOLOv8 Small model size is the most optimal model for the dataset size, YOLOv8 small provided the best

Figure 5.6 Generation of traffic cone object with stable diffusion.

results on the GISD dataset among the other 3 models. Figure 5.8 shows the detection results of the YOLOv8 Small model on the test part of the GISD dataset.

Table 5.1 shows that YOLOv8 Small model performed very well on the GISD test dataset with very high precision (P), recall (R), and mean average precision (mAP) scores.

Figure 5.7 This image represents the detection results of the model trained with the generated dataset on real-world corner case images.

Class	Instances	Precision (P)	Recall (R)	mAP@50	mAP@50-95
all	57	0.99	0.982	0.995	0.809
deer	11	1	0.932	0.995	0.849
dog	10	0.98	1	0.995	0.818
traffic cone	24	1	0.998	0.995	0.75
trash bin	12	0.979	1	0.995	0.82

Table 5.1 Model performance metrics for different classes on the test dataset.

Figure 5.9 contains four different graphs about the trained YOLOv8 Small model metrics such as F1-score, precision, accuracy, and confidence matrix. The F1-Confidence graph shows that the model produces a highly accurate prediction at a prediction of 0.99 when the threshold is at least 0.651. The Precision-Confidence curve shows that the model makes highly reliable predictions for all classes at a threshold of 0.864.

The mAP50 metric begins at approximately 0.95 and maintains this high level consistently throughout the training process, showing only minor variations can be seen in Figure 5.10. This demonstrates that the model is excelling in both precision and recall at a 50% IoU threshold, achieving almost perfect accuracy. The mAP50-95 metric begins at a lower value around 0.68, and gradually increases over the epochs, eventually stabilizing between 0.82 and 0.85 towards the end of the training.

Figure 5.8 This image represents the results of the predictions on the test images.

We observed that the model tends to overfit as we increase the number of epochs in our tests on the dataset. This result can be seen in the accuracy chart in Figure 5.11. Therefore, the best detection results are in our tests with 50 epochs. After 50 epochs, the model begins to memorize the training data and fails in the detection of new instances.

As can be seen in the confusion matrix Figure 5.12, the model classified all objects in the test dataset as true positive except one deer.

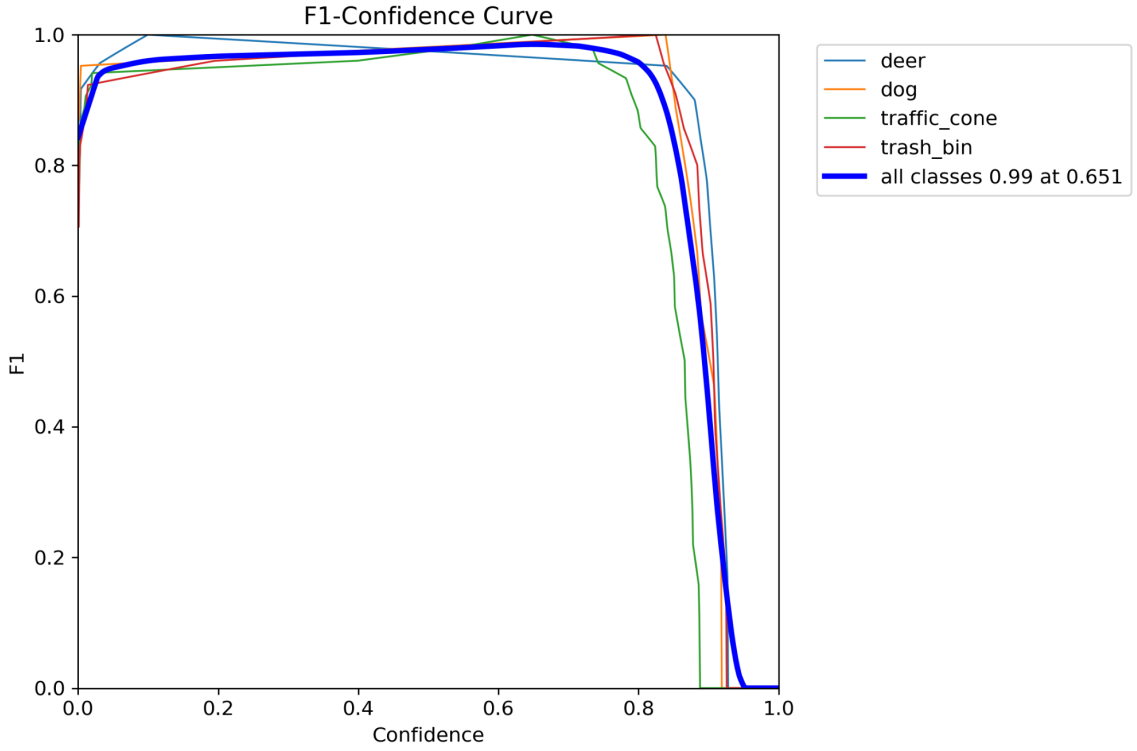

Figure 5.9 This figure presents F1 score of the detection model.

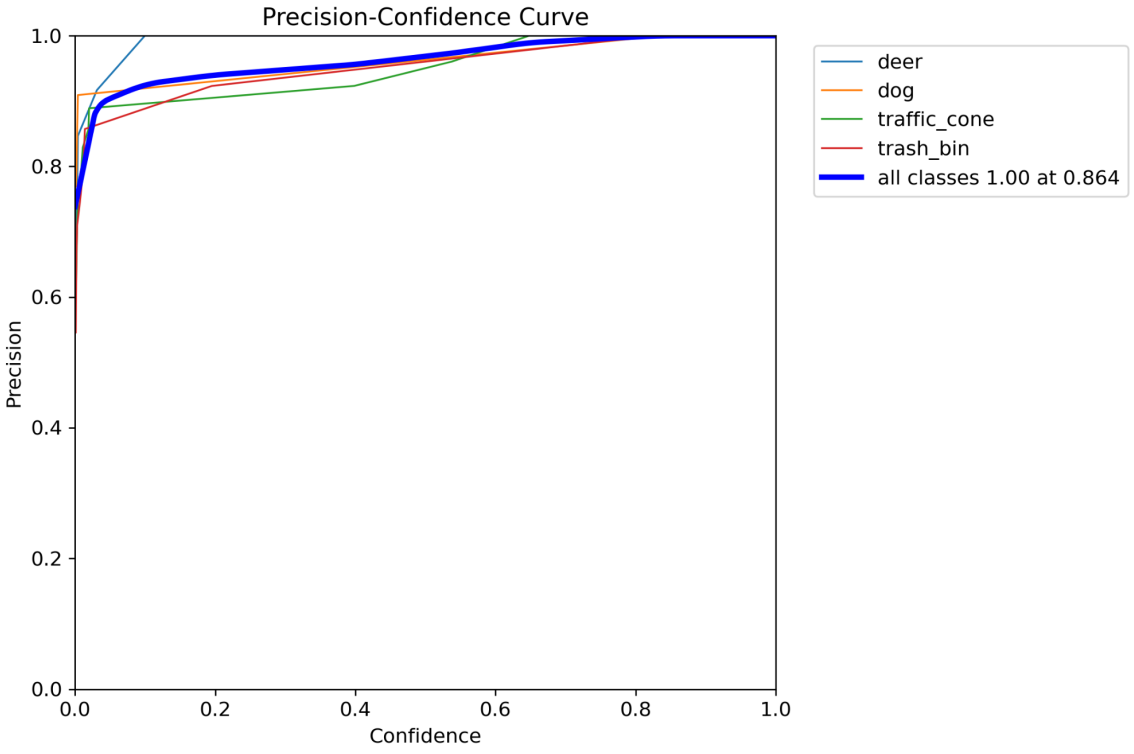

Figure 5.10 This figure presents, the precision performance of the detection model.

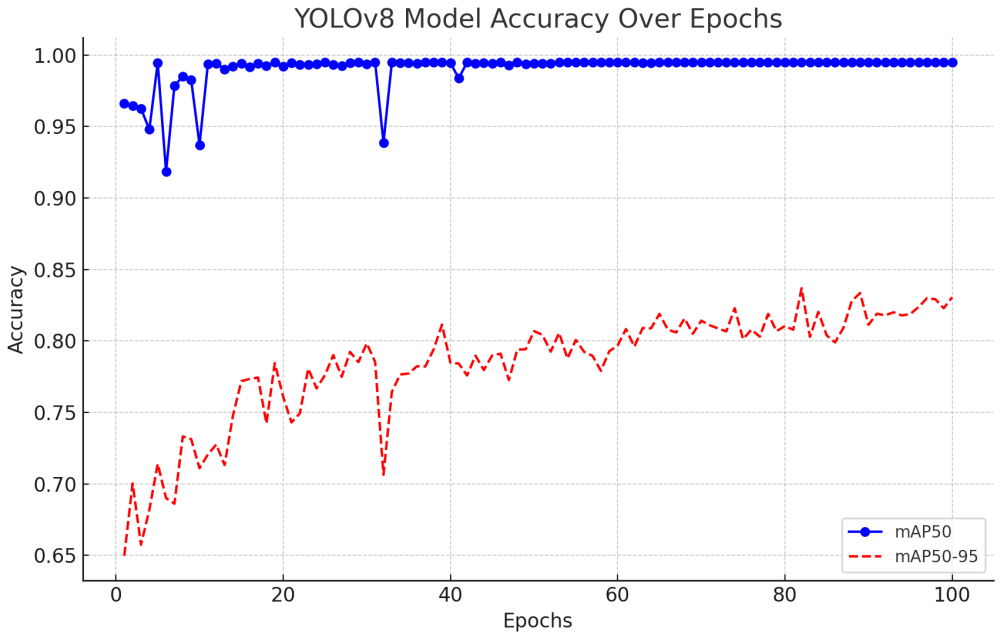

Figure 5.11 This figure presents model accuracy of the detection model.

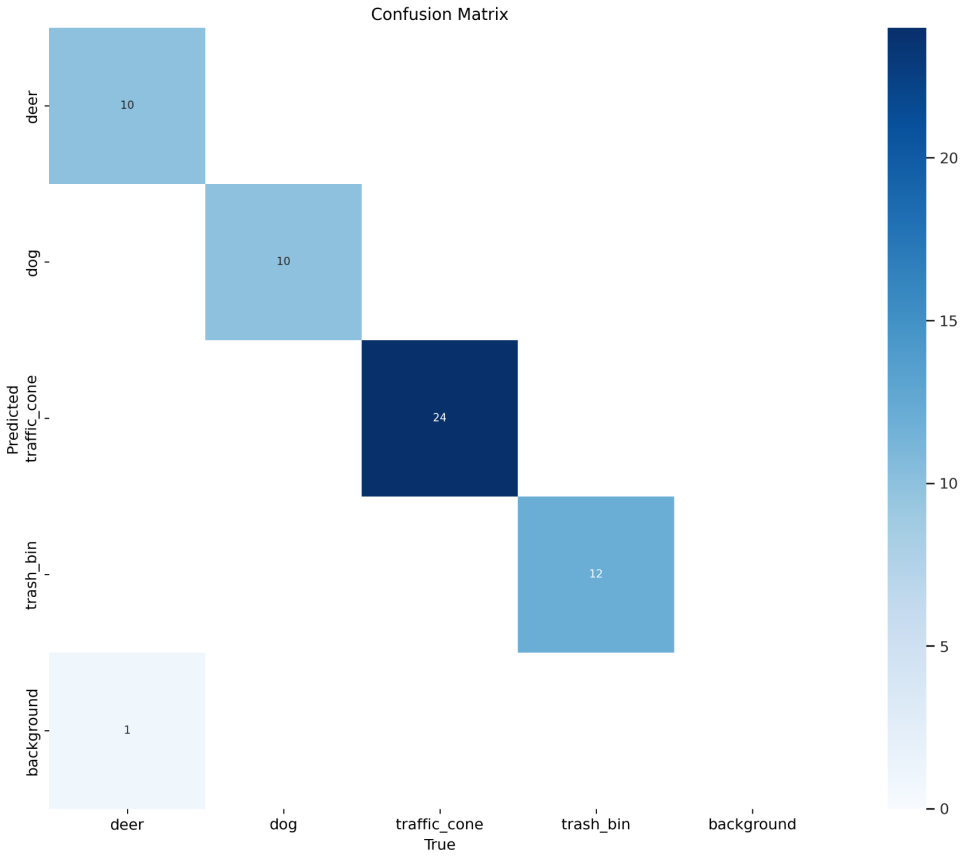

Figure 5.12 This figure presents confusion matrix metrics of the detection model.

6. DISCUSSION

Generating corner cases in the infrared domain by stable diffusion is not an easy task. In this section, we will talk about the challenges faced at different stages of this process and how these challenges were met.

First of all, a number of difficulties were encountered during the training of the LORA model in order to teach the thermal characteristics of the objects that would create a corner case for the stable diffusion model.

When creating LORA models, it is important to ensure that the object used is of very different variations. To explain with an example, while training LORA for the deer object, we ensured that the deer images collected from publicly available sources had many different variants. For this, it is important to have images of different sizes, performing different actions, and taken at different distances in the LORA train dataset. If these are not considered, the LORA model, which will fine-tune the stable diffusion model, might import images that are very similar to the images used to create this model, meaning that the LORA model is overtrained with fewer images, which leads to less scene diversity.

Another difficulty faced during the LORA training phase is that the relevant train dataset must be defined in detail with a text file pair. This is achieved through a detailed scene description, such as the object's action, its location, etc. If this condition is not met, the LORA model will make assumptions about the object in the scene and will only weigh it based on the relevant text caption described, which will lead to incorrect results. Finally, when training LORA, it is important to make sure that the training process is performed with enough images. There should not be too many of the same image, nor too few images with too many different states of the same object. In our experiments, we observed that at least 15 images at very different distances and variants were sufficient to properly represent the thermal characteristics.

Another issue encountered is that in images generated with image-to-image and LORA without inpainting, the relevant corner case objects may be generated in very irrelevant places

in the scene. For example, when we want to generate a corner case with the image-to-image method by prompting: "*Deer at standing on the road,<lora:DeerLORA:1.5>*" directly without applying inpaint, the dog placed in the scene can sometimes collide with another object. The reason for this is the same reason why we train the LORA model. This is because the stable diffusion model does not know enough about the infrared domain, where stable diffusion can get the correct thermal characteristics of a dog through LORA and even learn the *standing* action from the LORA model. The main problem here is that the characteristic of a road in the thermal domain does not know what a stable diffusion looks like. Even if this information is transferred when training with LORA, the stable diffusion model often fails to perform a smooth fitting process when augmenting the thermal characteristic of the scene. Even if this information is transferred when training with LORA, the stable diffusion model often fails to perform a smooth embedding process when augmenting the thermal characterized object into the scene.

Another challenge in this thesis is that manual captioning of the images in the dataset is time-consuming and challenging. Each generated object in the dataset needs to be captioned and given to the model for the detection task. This process can become a very time-consuming task when the number of images in the dataset increases. It is important that the detection model correctly detects the bounding box of the object with a high percentage of the detection model since the targeted study is to address a life-critical issue by creating corner case scenarios in the infrared domain in autonomous driving. Therefore, the bounding boxes were placed very carefully when manually annotating the objects.

7. CONCLUSION

This thesis addresses the corner case for the autonomous driving in the infrared domain. This issue has not been studied before in the literature and is a critical issue for driving safety in autonomous vehicles. Therefore, we created a dataset in four different classes, including deer, dog, trash bin, and traffic cone, that jeopardized safety in autonomous driving and made it available to future academic studies in this field.

In this thesis, we aimed to generate the scenes by teaching the thermal characteristics of the objects that could create a corner case to the stable diffusion model through LORA models. For this purpose, the characteristics of corner case objects were transferred to the stable diffusion model through LORA models, and corner case scenes were created using the inpainting process.

The dataset consists of a total of 1000 infrared images, 250 from each class, with a size of 512x512 containing corner cases. The objects in the dataset are available in many different variants. In terms of size, location, and action taken, the dataset contains images of very different varieties. In addition, the dataset we created has a minimal simulation gap. Since the LORA models that transfer thermal characteristic information to the stable diffusion model are trained with real-world images, the simulation gap is minimized in our generated dataset.

LORA models were used to transfer knowledge of how corner case objects appear in the thermal infrared domain to the stable diffusion model. This facilitated the augmentation of these objects in the inpainted area based on text prompts during image generation. Another benefit of fine-tuning the generative model with LORA is that a limited amount of data can be transferred to the generative model with limited hardware. If one wanted to transfer the thermal characteristics of the objects to the stable diffusion model by training the entire model from scratch, this would require millions of images a very high computing resource, and a long time of training. The image generated at the end of this process would not be guaranteed to accurately represent the object's thermal characteristics. Unlike retraining

the whole model, LORA fine-tunes stable diffusion models by updating only through the addition of low-rank matrices, rather than retraining all the weights. This approach enables efficient fine-tuning of stable diffusion models with minimal training data, time, and computing resources.

In this respect, LORA models can be trained in future studies with limited data for the target domain and then a new dataset can be created with this learned characteristic. LORA models can be very important in situations where data collection is limited and dangerous and difficult, space exploration, nuclear disaster zones, deep-sea exploration are some of the areas that can be studied in this field in the future. It is very challenging to collect data in these areas, and even if data is collected, it may be in limited amounts. In the studies to be carried out in these areas, it is possible to achieve data diversity by including LORA models in the dataset generation process.

We also trained a detection model using the dataset created in this thesis and tested the model on the test set. The model produced with high prediction scores in our dataset's test part and real-world images containing corner cases. The corner case objects in our dataset are generated with the correct characteristics as in the real world. One verifying feature for our claim is that, the detection model trained using our dataset was able to detect real-world corner case objects with high accuracy scores.

The LORA models produced for 4 different objects, a dataset of 1000 infrared images including corner cases, and the YOLOv8 Small model trained with the dataset can be accessed from the following link for academic studies in this field: https://github.com/mert-yigit/GISD

In future studies, the manual labeling process using the mask during inpainting process can be made more algorithmic. Especially when the dataset size increases, the time spent on labeling increases considerably. Instead, after the corner case object is added, segmenting it and obtaining a bounding box that will surround the object during the image generation process will speed up the process. In addition, LORA models can be trained for different objects and the variety of objects that create corner cases can be increased.

REFERENCES

[1] Sabir Rangwala. Thermal cameras gain acceptance for adas and autonomous cars. *Forbes*, **2022**.

[2] Waymo. Waymo: Self-driving technology, **2023**. Accessed: 2023-10-01.

[3] Yahoo News. Police dashcam shows deer slamming into two oncoming cars, **2017**. Accessed: 2023-10-01.

[4] Raptor Photonics. Vswir cameras, **2023**. Accessed: 2023-10-01.

[5] Lintao Han, Yuchen Zhao, Hengyi Lv, Yisa Zhang, Hailong Liu, Guoling Bi, and Qing Han. Enhancing remote sensing image super-resolution with efficient hybrid conditional diffusion model. *Remote Sensing*, 15(13):3452, **2023**.

[6] Byungjoon Kim and Yongduek Seo. A study of pattern defect data augmentation with image generation model. *Journal of the Korea Computer Graphics Society*, 29(3):79–84, **2023**.

[7] Muhammad Danyal Malik and Danish Humair. Evaluating the feasibility of using generative models to generate chest x-ray data. *arXiv preprint arXiv:2305.18927*, **2023**.

[8] Ultralytics. Yolov8: Real-time object detection and image segmentation. `https://github.com/ultralytics/ultralytics`, **2023**.

[9] Xu Zhao. Deep learning based visual perception and decision-making technology for autonomous vehicles. **2023**.

[10] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, **2023**.

[11] Aboli Marathe, Deva Ramanan, Rahee Walambe, and Ketan Kotecha. Wedge: A multi-weather autonomous driving dataset built from generative vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3317–3326. **2023**.

[12] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16918–16927. **2021**.

[13] Daniel Bogdoll, Svetlana Pavlitska, Simon Klaus, and J Marius Zöllner. Conditioning latent-space clusters for real-world anomaly classification. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1620–1625. IEEE, **2023**.

[14] Tae Eun Choe, Jane Wu, Xiaolin Lin, Karen Kwon, and Minwoo Park. Hazardnet: Road debris detection by augmentation of synthetic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 161–171. **2023**.

[15] Yukyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyounghwan An, and In So Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, **2018**.

[16] FLIR. Free flir thermal dataset for algorithm training, **2022**. Available at: https://www.flir.com/oem/adas/adas-dataset-form/.

[17] Chenglong Li, Wei Xia, Yan Yan, Bin Luo, and Jin Tang. Segmenting objects in day and night: Edge-conditioned cnn for thermal image semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3069–3082, **2020**.

[18] Darsh Parekh, Nishi Poddar, Aakash Rajpurkar, Manisha Chahal, Neeraj Kumar, Gyanendra Prasad Joshi, and Woong Cho. A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, 11(14), **2022**. ISSN 2079-9292. doi:10.3390/electronics11142162.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE, **2016**.

[20] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587. IEEE, **2014**.

[21] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182. **2017**.

[22] François Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1827–1836. **2017**.

[23] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3569–3577. **2018**.

[24] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927. **2018**.

[25] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting road obstacles by erasing them. *IEEE transactions on pattern analysis and machine intelligence*, **2023**.

[26] Yimeng Li and Jana Košecká. Uncertainty aware proposal segmentation for unknown object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 241–250. **2022**.

[27] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1037–1045. **2015**.

[28] Daniel Olmeda, Cristiano Premebida, Urbano Nunes, José María Armingol, and Arturo de la Escalera. Pedestrian detection in far infrared images. *Integrated Computer-Aided Engineering*, 20(4):347–360, **2013**.

[29] Zhiwei Deng, Afshin Vahdat, Hexiang Hu, and Greg Mori. Structure inference machines: Recurrent neural networks for analyzing relations in group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4772–4781. **2017**.

[30] Jie Dong and Yong Xiao. Pedestrian detection in infrared images based on convolutional neural networks. In *Proceedings of the 37th Chinese Control Conference*, pages 5213–5218. IEEE, **2018**.

[31] Shaohua Guan, Jian Wang, Wei Deng, and Rui Xu. Pedestrian detection in thermal infrared images using a convolutional neural network. *Infrared Physics & Technology*, 97:174–181, **2019**.

[32] Discover Magazine. The driverless car era began more than 90 years ago. **2023**. Accessed: 2023-10-01.

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313. **1989**.

[34] Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. In *The 2005 DARPA grand challenge*, pages 1–43. Springer, **2006**.

[35] Waymo. Waymo - the world's most experienced driver. `https://www.waymo.com/`, **2022**.

[36] J Breitenstein, JA Termöhlen, D Lipinski, and T Fingscheidt. Corner cases for visual perception in automated driving: Some guidance on detection approaches. arxiv 2021. *arXiv preprint arXiv:2102.05897*.

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, **2013**.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, **2014**.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, **2017**.

[40] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, **1986**.

[41] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. **2019**.

[42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, **2019**.

[43] Jonathan Shen, Ruoming Pang, Chengyi Yang, Yuxuan Zhang, Wei Chai, and et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, **2018**.

[44] Yan Kong, Juntang Xu, Yujia Zhang, and et al. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33. **2020**.

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, and et al. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, **2021**.

[46] Alec Radford, Jongyoon Kim, Chris Hallacy, and et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, **2021**.

[47] M. Frid-Adar, A. Ganin, and et al. Gans for medical image synthesis: A review. In *Proceedings of the 2018 IEEE International Conference on Image Processing (ICIP)*, pages 1–5. **2018**.

[48] Rafael Gomez-Bombarelli, R. Nussinov, and et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, **2018**.

[49] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, **2015**.

[50] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, **2021**.

[51] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, **2005**.

[52] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, **1995**.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. **2016**.

[54] Shaoqing Ren, Kaiming He, Ross Girshick, and Piotr Dollár. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28. **2015**.

[55] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271. **2017**.

[56] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, **2018**.

[57] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, **2020**.

[58] Glenn Jocher et al. Yolov5. *GitHub repository*, **2020**.

[59] Chien-Yao Wang, I-Huang Yeh, Chih-Yuan Wu, Po-Hsiang Chen, and Jeng-Shyang Hsieh. Yolov6: A single-stage object detection model for real-time applications. *arXiv preprint arXiv:2207.08430*, **2022**.

[60] Thomas Rothmeier, Werner Huber, and Alois C Knoll. Time to shine: Fine-tuning object detection models with synthetic adverse weather images. In *Proceedings*

of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 4447–4456. **2024**.

[61] Midjourney. `https://www.midjourney.com/`.

[62] Marius Cordts, Mohamed Omran, Sebastian Ramos, Nikolai Rehfeld, Markus Enzweiler, Ramin Zabih, Thomas Schultz, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223. IEEE, **2016**.

[63] Dan Hendrycks, Steven Basart, Mantas Mu, Saurav Kadavath, Fabio Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Maki Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. Fishyscapes: Lost and found. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3989–3999. **2021**.

[64] AAA Foundation for Traffic Safety. Prevalence of motor vehicle crashes involving road debris in the united states, 2011-2014, **2016**.

[65] Insurance Information Instute. Facts + statistics: Deer vehicle collisions, **2024**. Accessed: 2024-01-04.

[66] Tzutalin. Labelimg: A graphical image annotation tool. `https://github.com/tzutalin/labelImg`, **2015**. Accessed: 2023-10-01.