SIMULATING AND AUGMENTING TURBULENT THERMAL IMAGES FOR DEEP OBJECT DETECTION MODELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ENGIN UZUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MULTIMEDIA INFORMATICS

SEPTEMBER 2024

SIMULATING AND AUGMENTING TURBULENT THERMAL IMAGES FOR DEEP OBJECT DETECTION MODELS

submitted by **ENGIN UZUN** in partial fulfillment of the requirements for the degree of **Master of Science in Modeling and Simulation Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, **Graduate School of Informatics**

Assoc. Prof. Dr. Elif Sürer
Head of Department, **Modeling and Simulation**

Assoc. Prof. Dr. Erdem Akagündüz
Supervisor, **Modeling and Simulation**

Examining Committee Members:

Prof. Dr. Alptekin Temizel
Modeling and Simulation Department, METU

Assoc. Prof. Dr. Erdem Akagündüz
Modeling and Simulation Department, METU

Assoc. Prof. Dr. Fatih Nar
Computer Engineering Department, Ankara Yıldırım Beyazıt University

Date: 03.09.2024

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname:	**Engin Uzun**
Signature :	

ABSTRACT

SIMULATING AND AUGMENTING TURBULENT THERMAL IMAGES FOR DEEP OBJECT DETECTION MODELS

Uzun, Engin
M.S., Department of Modeling and Simulation
Supervisor: Assoc. Prof. Dr. Erdem Akagündüz

September 2024, 48 pages

Atmospheric turbulence, caused by factors such as temperature, wind speed, and humidity, leads to random fluctuations in the atmosphere's refractive index. This phenomenon degrades the image quality of long-range observation systems through geometric distortions and spatial-temporal varying blur. Turbulence can affect various imaging spectra, including visible and thermal bands. This thesis addresses the challenge of atmospheric turbulence in thermal imagery and its impact on object detection models. To tackle this challenge, we propose a data augmentation method that enhances the performance of object detectors by utilizing turbulent images with varying severity levels as training data. We generate training samples using a geometric turbulence simulator and use Geometric, Zernike-based, and P2S-based simulators to create the turbulent test sets, confirming the effectiveness of our augmentation method across different types of simulated turbulence. Our results demonstrate that this data augmentation approach significantly improves performance for both turbulent and non-turbulent thermal test images.

Keywords: atmospheric turbulence, data augmentation, object detection, thermal imagery

ÖZ

DERİN NESNE TESPİT MODELLERİ İÇİN TÜRBÜLANSLI TERMAL GÖRÜNTÜLERİN SİMÜLASYONU VE ARTIRILMASI

Uzun, Engin
Yüksek Lisans, Modelleme ve Simülasyon Bölümü
Tez Yöneticisi: Doç. Dr. Erdem Akagündüz

Eylül 2024, 48 sayfa

Atmosferik türbülans, sıcaklık, rüzgar hızı ve nem gibi faktörlerden kaynaklanır ve atmosferin kırılma indeksinde rastgele dalgalanmalara yol açar. Bu fenomen, uzun menzilli gözlem sistemlerinin görüntü kalitesini geometrik bozulmalar ve mekansal-zamansal değişen bulanıklık ile düşürür. Türbülans, görünür ve termal bantlar da dahil olmak üzere çeşitli görüntüleme spektrumlarını etkileyebilir. Bu tez, termal görüntülerdeki atmosferik türbülans sorununu ve bunun nesne tespit modellerine etkisini ele almaktadır. Bu zorluğun üstesinden gelmek için, değişen şiddet seviyelerine sahip türbülanslı görüntüleri eğitim verisi olarak kullanarak nesne modellerinin performansını artıran bir veri artırma yöntemi öneriyoruz. Geometrik bir türbülans simülatörü kullanarak eğitim örnekleri üretiyor ve Geometrik, Zernike tabanlı ve P2S tabanlı simülatörleri kullanarak türbülanslı test setlerini oluşturuyoruz. Bu sayede, artırma yöntemimizin farklı türlerdeki simüle edilmiş türbülanslar arasında etkinliğini doğruluyoruz. Sonuçlarımız, bu veri artırma yaklaşımının hem türbülanslı hem de türbülanssız termal test görüntüleri için performansı önemli ölçüde artırdığını göstermektedir.

Anahtar Kelimeler: atmosferik türbülans, veri artırma, nesne tespiti, termal görüntüleme

To my wife Melis, for all of her love and support.

ACKNOWLEDGMENTS

First and foremost, to my advisor, Erdem Akagündüz, for your invaluable guidance, insightful feedback, and unwavering support. Your mentorship has been instrumental in shaping the direction and success of this work.

To my family, whose love, patience, and encouragement have been my constant source of strength. Your support has been the foundation upon which I have built this achievement.

To my friend and research teammate, Ahmet Anıl Dursun, for your collaboration, shared insights, and camaraderie. Your support has been crucial in overcoming the challenges we faced together.

To my team leader at Aselsan Inc., Erkan Okuyan, for your leadership, encouragement, and belief in my work. Your support within the professional sphere has been vital in the practical application and impact of this research.

I would also like to extend my sincere thanks to my Thesis Committee Members, Prof. Dr. Alptekin Temizel and Assoc. Prof. Dr. Fatih Nar, for their valuable time, thoughtful feedback, and commitment to evaluating my work. Your expertise and insights have greatly enriched this thesis and its outcomes.

I also extend my heartfelt thanks to all my friends and colleagues, whose names I cannot list individually, but whose contributions, support, and encouragement have been deeply appreciated. Your friendship and assistance have made this journey both rewarding and fulfilling.

Thank you all for being a part of this important chapter in my life.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADAS	Advanced Driver-Assistance Systems
AP	Average Precision
APS	Average Precision for Small objects
APM	Average Precision for Medium objects
APL	Average Precision for Large objects
BCE	Binary Cross-Entropy
CIoU	Complete Intersection over Union
COCO	Common Objects in Context
CSPNeXt	Cross-Stage Partial Networks
CVPR	Conference on Computer Vision and Pattern Recognition
DFL	Distributional Focal Loss
DINO	DEtection TRansformers with Improved DeNoising Anchor Boxes
FCOS	Fully Convolutional One-Stage Object Detection
FLIR	Forward Looking Infrared
FLOPs	Floating Point Operations
GFL	Generalized Focal Loss
HSV	Hue, Saturation, Value
IACS	IoU-aware Classification Score
IoU	Intersection over Union
IR	Infrared
MMYOLO	OpenMMLab YOLO series toolbox and benchmark
P2S	Phase-to-Space

PCA	Principal Component Analysis
PSF	Point Spread Function
RGB	Red, Green, Blue
RTMDet	Real-Time Multi-Task Detector
SOTA	State of the Art
TOOD	Task-aligned One-stage Object Detection
T-Head	Task-aligned Head
TAL	Task Alignment Learning
TAP	Task-Aligned Predictor
VFNet	VarifocalNet
YOLO	You Only Look Once
YOLOR	You Only Learn One Representation
YOLOv8	You Only Look Once version 8

CHAPTER 1

INTRODUCTION

The increasing importance of thermal imagery in various pivotal applications such as surveillance, defense, and environmental monitoring necessitates robust and accurate object detection systems. Atmospheric distortions, including scattering, absorption, and refraction, significantly impact the quality of thermal imagery by altering the path and intensity of the incoming light. Among these distortions, atmospheric turbulence, driven by temperature changes, humidity, and wind, is particularly challenging as it disrupts light transmission through the atmosphere. Atmospheric turbulence leads to spatial and temporal variations in the air's refractive index, significantly impacting the propagation of light and introducing distortions that are particularly evident in long-range imaging systems across various electromagnetic spectra [4]. Turbulence in long-range imaging systems is often treated as isoplanatic, with spatially uniform distortions typically mitigated through adaptive optics [5, 6]. However, in ground-to-ground imaging systems, the turbulence becomes anisoplanatic, resulting in non-uniform, non-rigid distortions that blur images and present substantial challenges for computer vision algorithms [7, 8]. These challenges are exacerbated in the infrared (IR) spectrum, where the refractive index structure parameter, ${C_n}^2$, and path length can greatly influence image quality [9]. Although the visible spectrum is more adversely affected by turbulence due to shorter wavelengths, significant effects are also observed in the IR spectrum, complicating tasks such as thermal imagery object detection.

Thermal imagery object detection faces unique challenges due to the limited scale of IR image sets and the distinct nature of IR data. Despite advancements in domain adaptation techniques using supervised and adversarial-based unsupervised learning [10, 11, 12], the issue of atmospheric turbulence and its effect on thermal images remains underexplored. Optical systems affected by atmospheric turbulence experience noise, non-rigid distortions, and blurry images as light rays are bent and scattered due to changes in the refractive index of the air. These effects result in fluctuations in the air's temperature and density, causing temporal variations in the electromagnetic radiation received by optical systems [13].

Although several IR image and video sets are publicly available [14], to the best of our knowledge, there is no public IR image set that includes turbulence effects, for object detection or any other computer vision tasks. This is mainly because creating a set consisting of both turbulent and non-turbulent images of a given scene is a very expensive and difficult task. A feasible option is to create synthetic turbulence on existing image sets, using mathematical models of turbulence. The existing models in the literature [15, 16, 17] generally represent the dynamics of atmospheric turbulence in 3D coordinates and project synthetic 3D scenes to 2D images. However, creating a large corpus of images with different 3D scene settings is also a very difficult task to achieve. Furthermore, in the thermal domain, creating 3D models with realistic material properties is problematic as well. Another approach can be working on existing thermal object detection datasets that additionally include 3D objects models. However, this case would require a reconstruction of the 3D scene with depth information, which is not available for almost any image set.

1.1 Problem Definition

Atmospheric turbulence significantly impacts optical systems, causing noise, non-rigid distortions, and blurry images. This occurs because turbulence changes the refractive index of the air, causing light rays to bend and scatter in various directions, resulting in blurring and geometric distortions that degrade image quality. Moreover, turbulence causes fluctuations in air temperature and density, leading to temporal variations in the electromagnetic radiation received by optical systems. For detailed explanations about the effects of turbulence on optical imaging systems, readers may refer to [13].

Various techniques are being studied to counteract the effects of turbulence on computer vision tasks [18, 19, 20, 21, 22]. Regardless of the problem, achieving generalization capability and robustness in challenging computer vision tasks, such as detection in turbulent images, occluded images, and adversarial samples, typically requires high-quality data.

This thesis specifically focuses on thermal-adapted deep object detection models, which are specialized frameworks designed to detect and localize objects using thermal imaging data. By adapting deep learning techniques to thermal imagery, these models offer enhanced detection capabilities in diverse real-world scenarios. While there has been a recent increase in research on object detection using thermal imagery [23, 24, 25, 26, 27], there are limited studies on the atmospheric effects on thermal object detection systems [21, 1, 28, 29, 30].

To mitigate the effects of atmospheric turbulence, various algorithms [31, 32, 33, 34] aim to reduce or eliminate these effects on images. However, estimating and correcting turbulence distortion is computationally intensive and may not be feasible in real-time due to limited resources. An alternative approach, explored in this thesis, is to train models with turbulence-augmented samples, thereby avoiding additional computational costs.

Employing turbulent image augmentation techniques enhances the accuracy and robustness of baseline models in handling degradation effects induced by turbulence [1, 35]. Augmentation methods involving blurring and geometric distortions improve the generalization capability of detection models, even when test images are undistorted.

1.2 Research Questions

In examining the impact of atmospheric turbulence on thermal imagery and the potential solutions through data augmentation, several key research questions arise:

- How does atmospheric turbulence affect the performance of state-of-the-art object detection models in thermal imagery?
- What data augmentation strategies can be developed to improve object detection performance under turbulent conditions?

- How do different levels of turbulence severity impact the robustness and accuracy of thermal-adapted object detection models?
- Can synthetic turbulence models be effectively used to simulate real-world atmospheric conditions for training object detection systems?

1.3 Objectives of the Thesis

This thesis aims to address the challenges posed by atmospheric turbulence in thermal imagery through a series of targeted objectives:

- To analyze the effects of atmospheric turbulence on the performance of object detection models in thermal imagery.
- To develop and evaluate data augmentation strategies that incorporate varying levels of synthetic turbulence to enhance model robustness.
- To compare the effectiveness of different turbulence simulation models in generating training data for object detection.
- To propose a comprehensive framework for improving object detection accuracy and reliability in turbulent thermal imagery.

1.4 Contributions of the Study

In this thesis, we conduct a thorough analysis of turbulent image augmentation techniques for thermal band images in the context of object detection. Our primary goal is to identify the most effective augmentation method for improving accuracy. To achieve this, we generate turbulent images using three different atmospheric turbulence simulation methods [1, 2, 3]. The state-of-the-art real-time detection models evaluated in our study include VFNet [36], TOOD [37], YOLOR [38], RTMDet [39], DINO [40], and YOLOv8 [41].

Previous research has mainly focused on generating synthetic atmospheric turbulence in the visible spectrum, utilizing methods like generative adversarial networks (GANs) [42], neural network-based geometric models [43], and physics-based models [44, 45]. Despite some recent studies investigating atmospheric turbulence effects on long-range observation systems [46], there has been a lack of solutions for turbulence-related issues in infrared (IR) computer vision problems. To address this gap, we apply a geometric model to introduce turbulence distortions in IR images from the "FLIR ADAS v2" dataset [47]. Both the generated turbulent images and the original non-turbulent images are used to improve the performance of deep learning-based object detectors. Our experiments involve benchmarking six different state-of-the-art deep object detectors, pretrained and fine-tuned for the thermal domain. Additionally, we propose a data augmentation method to enhance detection performance for both turbulent and non-turbulent thermal test images.

1.5 Organization of the Thesis

This thesis is organized into several chapters, starting with an Introduction (Chapter 1) that outlines the research problem, objectives, and significance, followed by a Literature Review (Chapter 2) that contextualizes the study within existing research about the data augmentation methods. The Turbulent Image Generation (Chapter 3) chapter explains the simulators and the process of simulating turbulence effects on thermal images, while the Experimental Setup (Chapter 4) chapter describes the datasets, object detection models, and experimental procedures employed. The Results (Chapter 5) chapter presents the findings, and the Conclusion (Chapter 6) summarizes the key findings and offers recommendations for future research.

CHAPTER 2

DATA AUGMENTATION LITERATURE

In recent years, the field of deep learning has experienced significant advancements, driving a growing demand for more labeled data. One notable development is the rise of vision transformer architectures, which have enabled attention-based algorithms to dominate various vision benchmarks. These benchmarks include image classification [48, 49, 50], object detection [40, 51, 50], and image generation [52, 53]. However, these powerful algorithms come with a substantial appetite for labeled data. To address this challenge, contemporary deep learning research has increasingly turned to self-supervised [54] and semi-supervised learning [55, 56] methods. These approaches aim to reduce the dependency on large labeled datasets by utilizing unlabeled data or combining a small amount of labeled data with a larger pool of unlabeled data. Despite the advances in these learning strategies, the overall demand for data to train deep neural networks continues to grow. Data augmentation remains one of the most efficient and effective techniques to amplify the amount of both labeled and unlabeled data. By artificially increasing the diversity of training data, augmentation helps mitigate issues such as small-scale datasets, class imbalance, and domain shift. This technique is crucial for enhancing the robustness and generalization capabilities of deep learning models, ensuring they perform well across a wide range of scenarios.

When augmenting data for a learning model, it is crucial to understand two fundamental dimensions: the method used to generate the augmented data and the strategy used to train the model with this augmented data. Mingle et al. [57] propose a novel augmentation taxonomy that integrates both data generation methods and training strategies. Essentially, they categorize augmentation into three groups: model-free, model-based, and optimization-based. The first two categories cover data generation methods, while the third pertains to augmentation strategies. Our augmentation classification, partially inspired by their taxonomy, is depicted in Figure 1. Our approach to augmentation diverges from that of [57] by first splitting the process into its two main dimensions: data generation methods and augmentation strategies. We categorize data generation methods into model-based and model-free. As the name suggests, model-free data augmentation refers to techniques that increase the diversity of training data without the use of a model[1]. Model-free data augmentation can be broadly divided into two subcategories: augmentation using a single sample or a fusion of multiple data samples. Model-free single-sample augmentation involves diversifying the dataset by applying various basic data processing techniques, such as rotation, scaling, and flipping [58, 59]. Previous work, such as [60], demonstrates that combining these techniques and merging their outputs with the original sample can lead to further improvements. These techniques are applied randomly, enabling the creation of a

[1] Although many methods, such as flipping or rotating a signal, can be considered to involve a mathematical model, they are termed model-free due to their simplicity.

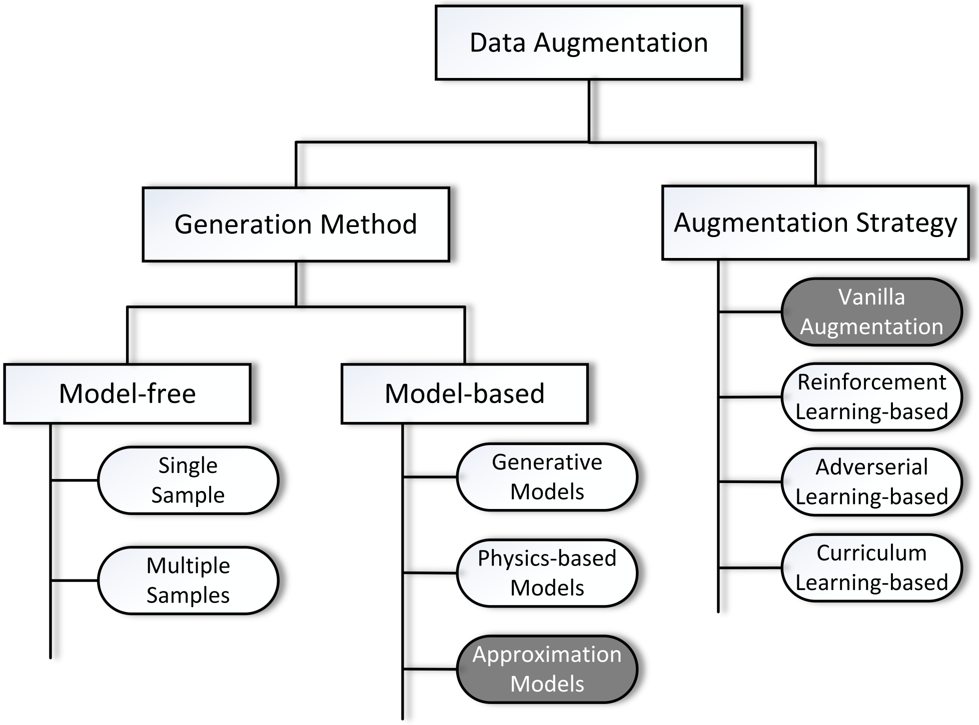


Figure 1: Taxonomy of Data Augmentation Methods and Strategies. This diagram categorizes the data augmentation techniques into generation methods and augmentation strategies. The generation methods are divided into model-free (single sample and multiple samples) and model-based (generative models, physics-based models, and approximation models). The augmentation strategies include vanilla augmentation, reinforcement learning-based, adversarial learning-based, and curriculum learning-based approaches.

diverse set of augmented samples from a single original source. Despite their simplicity, model-free augmentation techniques are widely used for various computer vision tasks [61, 62, 63, 64, 65, 66, 67].

Model-free multiple-sample augmentation methods aim to expand the training dataset distribution by combining more than one sample. The most common approach to implementing these techniques involves fusing two samples. The pioneering methods "Mix-up" [68] and "Cut-Mix" [69] have inspired numerous subsequent augmentation techniques, such as [70, 71, 72, 73, 74]. Additionally, various algorithms use more than two samples for augmentation, including "RICAP" [75] and mosaic augmentation [76]. Some augmentation methods, like YOLOv4's mosaic augmentation [76], use more than two images, significantly reducing the need for large mini-batch sizes according to the authors. Straightforward multiple-sample augmentation methods include Pairing Samples [77], which averages two images pixel-wise. In Mixup [68], the authors not only fuse samples but also labels. BC Learning [78], originally proposed for sound recognition, was adapted for image classification by the same group [79], treating images similarly to waveforms. While fusing multiple images may not visually make sense to humans, it is effective for machines. The aforementioned methods fuse images linearly, whereas CutMix [69] spatially fuses images and linearly fuses labels based on the mixing ratio. Recent works [71, 80, 81, 82] have followed the principles of CutMix and Mixup to tackle various challenges. There are also instance-level augmentation approaches for multiple samples that require segmentation masks of object instances. "Cut, Paste and Learn" [83] involves a two-stage process: cutting (cropping object instances) and pasting (fusing the cropped instances onto random background images). Similarly, [84] uses geometric and semantic information to reduce blending artifacts. These instance-level approaches are designed for object detection tasks. Different from these works, "Simple Copy-Paste" [85] aims to enhance instance segmentation performance of strong baseline models. The copy-paste concept is also applied to videos for multiple object tracking in [86]. The effectiveness of using multiple images for augmentation is demonstrated across various computer vision tasks, such as image classification [69], object detection [76], and instance segmentation [85].

Model-based data augmentation involves techniques that utilize complex mathematical models to generate new samples. These models can be categorized into three types: generative models, physics-based models, and approximation models. Generative model-based augmentation methods primarily use techniques such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and neural style transfer algorithms. According to [57], generative model-based approaches can be further classified into three categories: unconditional, label-conditional, and sample-conditional. Unconditional sample generation creates new samples from a given input noise [87, 88] (also known as the latent code in GAN literature), producing samples that resemble the training data distribution. Label-conditional sample generation, on the other hand, involves conditioning the generator and discriminator models on a label [89, 90, 91, 92], facilitating augmentation for supervised learning models. Sample-conditional generation, the final category, conditions the sample generation process on other existing samples. This can lead to either label-preserving [93, 94, 95, 96] or label-changing generation processes [97, 98, 99, 100, 101]. Label-preserving approaches are particularly effective for addressing domain shift problems [96], while augmentation with translated images can enhance the generalization ability of the model by incorporating transferred styles [102]. Label-changing approaches are useful for tackling class imbalance issues, where the label might be completely altered [103, 97] or fused as proposed in [68] to eliminate style or content bias [104].

Physics-based data augmentation involves utilizing physical laws and principles to generate synthetic samples that accurately reflect the underlying physics of the system creating the sample space. Depending on the model, the process to be physically simulated may vary, such as simulating different lighting conditions, adding noise or blur, or applying realistic transformations caused by sensors or the capturing environment. These physical transformations aim to represent phenomena that make the augmented samples as realistic as originally captured images. For example, to simulate atmospheric turbulence effects on an image, physics-based models use the principles of fluid mechanics governing turbulent atmospheric flows. Accurate simulations of turbulence effects on 2D images typically require detailed 3D environmental information, such as temperature distribution and wind velocity [105, 44]. Although these models can provide highly detailed and precise simulations, they demand significant computational resources to solve the complex fluid mechanics equations [2]. In medical imaging, physics-based augmentation methods are popular for enhancing generalization capability for new patient data [106] or increasing MRI image denoising [107]. Yaman et al. [108] proposed a self-supervision-based image reconstruction method that works without fully-sampled data used as training samples. Another example is VORTEX [109], which improves supervised baseline performance in MRI augmentation by generating physics-driven augmented samples, increasing robustness to SNR ratio changes and motion corruption for accelerated MRI reconstruction. Moreover, Wei et al. [110] proposed a physics-based illumination augmentation scheme for generating realistic facial images under different directional illumination conditions. Lou et al. [111] developed an image augmentation method to address domain shifts caused by sensor characteristic variations, demonstrating its effectiveness in both sound and image domain adaptations. Miles et al. [112] introduced a 3D occluded target augmentation method to improve target detection performance in 3D radar imagery. Deraining is another challenging problem in computer vision. Various methods have been proposed to generate rainy images based on the physical nature of rain [113, 114, 115, 116, 117, 118, 119]. These rainy image generation techniques can be used for data augmentation to enhance the robustness and generalization capability of deep learning models against rainy weather conditions. Similarly, atmospheric turbulence can significantly affect image quality. Physics-based approaches have been developed to generate turbulent images [2, 120, 3, 44] and to mitigate turbulence effects [121]. For instance, Yuxin et al. [122] proposed an augmentation method considering governing equations, observable perception, and physical phenomena of seismic full-waveform inversion (SWI) to improve seismic image quality and enhance small CO2 leak detection performance. In contrast to physics-based data augmentation methods, approximation models are less computationally intensive and often do not require detailed system models. These methods use simplified or empirical models to approximate the generative process. For instance, in simulating atmospheric turbulence, approximation models often bypass the need for detailed 3D environmental data, thus reducing computational demands. Some of these models utilize statistical data collected from various atmospheric conditions to simulate atmospheric effects on samples [123, 124], while others apply physically motivated approximations to simplify the calculation of turbulence effects on images. These simplified models aim to produce turbulent images by modeling spatial-temporal geometric distortions at the pixel level [3, 1]. However, the distinction between physics-based and approximation models can be blurred in practice. Some physics-based models may incorporate simplified system models to reduce computational costs, while some approximation models may still integrate physical principles or empirical data. This overlap illustrates that, regardless of the generation domain, elements of both approaches can be present in a single model to balance accuracy and computational efficiency.

In contrast to the classification proposed by [57], we consider the augmentation strategy as a distinct dimension within our taxonomy. These strategies employ data generation methods detailed on the left side of our taxonomy. The simplest way to incorporate augmented data into a training model is to treat the augmented samples the same as the original ones, a method we call "Vanilla Augmentation," reflecting its widespread use in deep learning literature. Beyond this conventional approach, we identify three key directions in augmentation strategies: curriculum-based, reinforcement learning-based, and adversarial learning-based methods. Unlike vanilla augmentation, these methods aim to optimize the augmentation policy. For instance, curriculum learning-based augmentation refines datasets according to curriculum learning principles, ensuring a gradual and targeted enhancement process [125]. Reinforcement learning-based data augmentation strategies seek to optimize the augmentation process based on the model's learning performance [126]. This approach dynamically generates augmented samples that are most informative to the model, enhancing its learning process. The augmentation process is modeled as a Markov Decision Process (MDP), where the state represents the current training dataset, actions are the augmentation operations, and the reward signal indicates improvements in model performance. The goal is to find an optimal policy that balances exploration and exploitation to generate the most informative augmented samples [127]. Although reinforcement learning-based augmentation can lead to better performance compared to vanilla augmentation, it is more complex to implement and requires more computational resources. Adversarial-based data augmentation involves generating samples similar to the original ones but designed to fool the model into making incorrect predictions. This method generates new samples by seeking small transformations of the original samples that yield maximum loss [128]. The aim is to improve the model's robustness by exposing it to samples that are close to the original data but can cause the model to fail. Training the model on these adversarial samples helps it become more resistant to similar attacks [129, 130, 131]. Adversarial-based data augmentation can be used alongside other methods and is especially useful in applications where the data distribution is complex and non-linear, such as computer vision and speech recognition. However, it can also be computationally expensive and may not always lead to significant performance improvements.

CHAPTER 3

TURBULENT IMAGE GENERATION

Simulating turbulence effects on sensory signals is a critical challenge due to the complex and dynamic nature of turbulence, which can significantly degrade the quality and accuracy of sensory data. In surveillance, turbulence can distort thermal and visual images, complicating the identification and tracking of objects [132]. In navigation, especially for aerial and maritime applications, turbulence can affect the reliability of sensor data, leading to potential safety risks [133]. Addressing these challenges requires sophisticated simulation techniques to accurately model and mitigate the impact of turbulence on sensory signals. Various methodologies exist for simulating turbulence effects, each with its own strengths and limitations [7]. Common models include optical flow-based methods, statistical turbulence models, approximation, and physical simulation techniques. Optical flow-based methods estimate the apparent motion of objects within an image sequence, which can be used to simulate the distortions caused by turbulence. Statistical models use mathematical functions to describe the statistical properties of turbulence, such as the power spectral density of the refractive index fluctuations. Physical simulation techniques, on the other hand, directly model the physical processes underlying turbulence, such as fluid dynamics and heat transfer. In this thesis, we utilize three specific *approximation models* for generating turbulent images: a geometrical turbulence simulator (Section 3.1), a Zernike-based simulator (Section 3.2), and a phase-to-space simulator (Section 3.3). These models were chosen for their ability to represent different aspects of turbulence effects on thermal images. For the Zernike-based and phase-to-space simulators, it is essential to use specific camera-related parameters to produce physically accurate turbulent images. In order to accomplish this, we use the parameters of the FLIR Tao v2 640 x 512 13mm 45°HFoV - LWIR Thermal Imaging Camera since the "FLIR-ADASv2" dataset is collected by using this optical system. The related parameters are listed in Table 2 and Table 3. A detailed explanation of the "FLIR-ADASv2" dataset and the augmented images are presented in Section 4.1.

3.1 Geometrical Turbulence Simulator

Approximation models [1, 35, 3] mimic the effect of atmospheric turbulence by combining blurring and random distortions. Our geometric simulator [1], which operates in real-time, leverages a geometrical turbulence approach. This model employs a Gaussian kernel for blurring and uses image warping to introduce random distortions. For a detailed description of the physical approximations underlying the geometric model, refer to Section 3.1.1. The model described in Equation 1 is used to generate turbulent images from non-turbulent ones.

$$F_n(x,y) = D((G_B(x,y) \circledast I_n(x,y)), d_n^u(x,y), d_n^v(x,y)) \tag{1}$$

In Equation 1, $F_n(x, y)$ is the source image, D is the warping function, $\circledast$ is the convolution operation and $G_B(x, y)$ is a Gaussian kernel with variance σ_B^2, responsible for blurring. Note that warping is applied to both horizontal and vertical directions using the random distortion fields, $d_n^u(x, y)$ and $d_n^v(x, y)$, respectively, which are defined as:

$$d_n^u(x, y) = \gamma * (G_D(x, y) \circledast v_n^u(x, y)) \tag{2}$$

$$d_n^v(x, y) = \gamma * (G_D(x, y) \circledast v_n^u(x, y)) \tag{3}$$

where γ is the amplitude of the random distortion and $G_D(x, y)$ is the Gaussian kernel with variance σ_D^2. $v_n^u(x, y)$ and $v_n^v(x, y)$ are random vectors with zero-mean, unit-variance normal distributions. Convolution operation with $G_D(x, y)$ provides spatial correlation of the random distortions over the image. σ_D^2 is used to adjust the strength of the spatial correlation while γ is the amplitude of the distortions in the model. σ_B^2 is used to adjust the amount of blurring. Table 1 denotes the parameter values used for turbulent image generation.

Table 1: Simulation parameters for the proposed geometric simulator.

Distortion amplitude (γ)	[25, 50, 100, 150]
Spatial correlation strength (σ_D^2)	5
Blurring ratio (σ_B^2)	0.5

3.1.1 Physical Approximation

In order to relate the proposed geometric turbulence model parameters with real-world turbulence conditions, in this section, we formulate the relation between the magnitude of the pixel shift (distortion vectors) and the model parameters, namely γ and σ_D^2. Note that we do not utilize σ_B^2 for this approximation since it is unrelated to the distortion process. Let $\boldsymbol{z(x, y)}$ be the random distortion vector over an arbitrary image region,

$$\boldsymbol{z(x, y)} = [d_n^u(x, y), d_n^v(x, y)]^T \tag{4}$$

$\boldsymbol{z(x, y)}$ is a bivariate Gaussian distribution, where u and v components are independent of each other and are zero-mean. Variance of u and v components depends on the values of both γ^2 and σ_D^2. In Equations 2 and 3, the vertical and horizontal distortions are derived from zero-mean, unit-variance Gaussian distributions. In the proposed model, these random variables, $v_n^u(x, y)$ and $v_n^v(x, y)$, are firstly filtered spatially by a 2D Gaussian kernel with σ_D^2 variance, then multiplied with the turbulence severity level, γ. Spatial filtering operation on $d_n^u(x, y)$ and $d_n^v(x, y)$ corresponds to the weighted linear combination of zero-mean unit-variance uncorrelated Gaussian random variables, where the weights are the parameters in $G_D(x, y)$. The equivalent variance of the spatially filtered random distortions is the sum of the squared kernel parameters. For a sufficiently large kernel, this summation is equivalent to integrating the corresponding squared continuous bivariate Gaussian function in 2D space. Then, γ is simply a multiplier over the variance. The resultant variance of $d_n^u(x, y)$ and $d_n^v(x, y)$, σ_z^2, can be expressed as,

$$\sigma_z^2 = \gamma^2 \int\int G(p, \mu_p, \Sigma_p)^2 dp \tag{5}$$

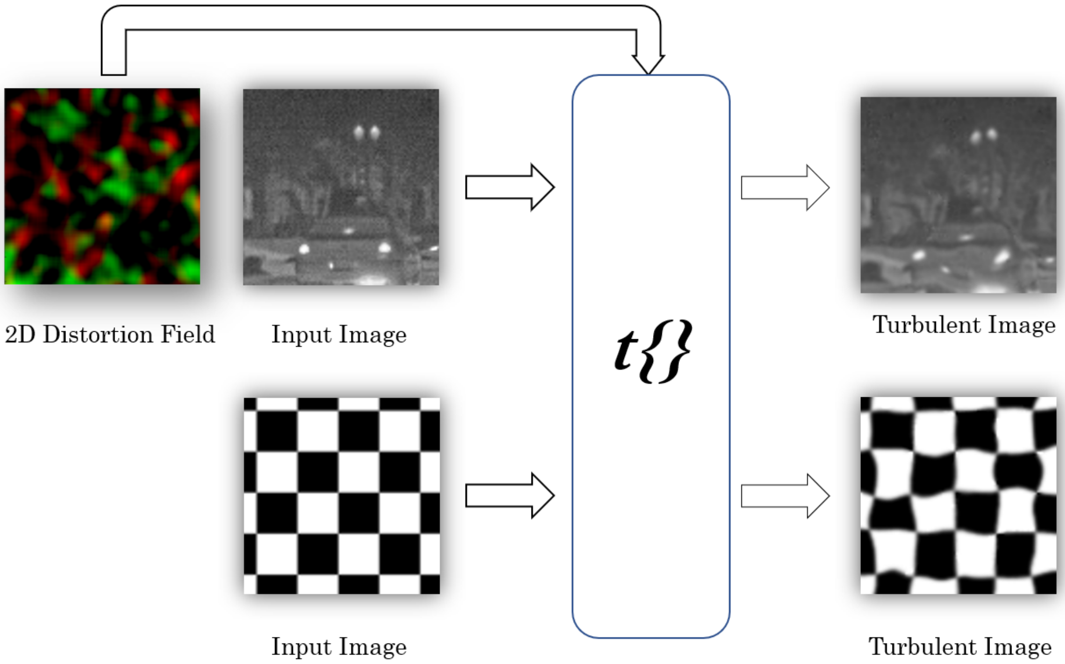


Figure 2: The turbulence generation processes on an IR and a chessboard images are illustrated. $t\{\}$ is the turbulence operation and the vertical ($d_n^u(x,y)$ in red) and the horizontal ($d_n^v(x,y)$ in green) elements of the 2D distortion field are depicted in false color.

where $\boldsymbol{\mu_p}$ is the mean and $\boldsymbol{\Sigma_p}$ is the covariance matrix of the bivariate Gaussian kernel. For our case, the Gaussian kernel $G_D(x,y)$ is assumed to be zero-mean and have constant diagonal covariance matrix $\boldsymbol{\Sigma_p}$. Given these conditions, in explicit form 5 reduces to,

$$\sigma_z^2 = \gamma^2 \int\int \left(\frac{1}{2\pi\sigma_D^2} e^{-\left(\frac{p_0^2+p_1^2}{2\sigma_D^2}\right)} \right)^2 dp_0 dp_1 \tag{6}$$

The resultant expression for σ_z^2 becomes,

$$\sigma_z^2 = \frac{\gamma^2}{4\sigma_D^2\pi} \tag{7}$$

Let $l(x,y)$ be the Euclidean norm of the random distortion vector $\boldsymbol{z(x,y)}$, which is given as,

$$l(x,y) = \sqrt{d_n^u(x,y)^2 + d_n^v(x,y)^2} \tag{8}$$

$l(x,y)$ is also a random variable, and because it is the Euclidean norm of a vector sampled from a normal distribution with zero-mean and constant diagonal covariance matrix, $l(x,y)$ is expected to have a Rayleigh distribution, which can be written as,

$$f_{l(x,y)}(l(x,y)) = \frac{l(x,y)}{\sigma_z^2} e^{-l(x,y)^2(2\sigma_z^2)} \tag{9}$$

where $f_{l(x,y)}(l(x,y))$ is the probability density function of random variable $l(x,y)$. Mean of the $l(x,y)$, namely μ_l, is the measure of how much the pixels are distorted on an image for the given parameters γ and σ_D^2, and given as,

$$\mu_l = \sigma_z \sqrt{\frac{\pi}{2}} = \frac{\gamma}{2\sqrt{2}\sigma_D} \tag{10}$$

Note that μ_l is a measure in pixel dimensions. In order to relate the distortions in the pixel dimensions to the real-world measurements, a pin-hole camera model can be utilized using internal parameters of the thermal camera used in collecting the FLIR ADAS v2 dataset. Relation between pixel distortions and the corresponding real-world distortion for both horizontal and vertical axes, t_s^h and t_s^v respectively, can be written as,

$$t_s^h = \frac{2\mu_l d_s tan(\alpha_h/2)}{W} = \frac{\gamma d_s tan(\alpha_h/2)}{\sqrt{2}\sigma_D W} \tag{11}$$

$$t_s^v = \frac{2\mu_l d_s tan(\alpha_v/2)}{H} = \frac{\gamma d_s tan(\alpha_v/2)}{\sqrt{2}\sigma_D H} \tag{12}$$

where d_s is the depth of the scene region, α_h and α_v are the horizontal and the vertical field of views, respectively, W is the horizontal pixel count, and H is the vertical pixel count of the camera. In [47], horizontal and vertical fields of views of the thermal camera are given as $45°$ and $37°$, and the image resolution is given as 640x512. Then, 11 and 12 reduces to,

$$t_s^h = 0.0004576 \frac{\gamma d_s}{\sigma_D} \tag{13}$$

$$t_s^v = 0.0004621 \frac{\gamma d_s}{\sigma_D} \tag{14}$$

3.2 Zernike-based Turbulence Simulator

The second model we employed in our experiments is a novel approximation-based approach as detailed by Chimitt and Chan [2]. The authors aimed to strike a satisfactory balance between precision and computational complexity for simulating turbulence effects. This model utilizes a propagation-free simulation technique to sample spatially correlated Zernike coefficients. Zernike polynomials, denoted as $\mathrm{Z}_m^n(\rho,\theta)$, are a set of orthogonal functions defined over a circular aperture, commonly used to represent aberrations in optical systems [134]. These polynomials and their coefficients are expressed mathematically as follows:

$$W(\rho,\theta) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} \mathrm{c}_n^m \mathrm{Z}_n^m(\rho,\theta) \tag{15}$$

Here, C_m^n represents the Zernike coefficients, which are the weights associated with each Zernike polynomial in the wavefront aberration. In their work, Chimitt and Chan employed these Zernike coefficients to simulate atmospheric turbulence effects such as tilt and blur. The terms C_1^1 and C_{-1}^1 specifically correspond to horizontal and vertical tilts, respectively, introducing geometric distortions across the input image. Higher-order coefficients are associated with the blur effect.

The simulation parameters utilized in our experiments are summarized in Table 2. These parameters were chosen to closely mimic the physical characteristics of the imaging system.

Table 2: Simulation parameters used for Zernike-based Turbulence Simulator.

Image Dimensions	[640, 640] pixels
Aperture Diameter	29 mm
Wavelength	10.5 μm
Refractive Index Structure Parameter (C_n^2)	1×10^{-15}
Focal Length	13 mm
Propagation Length	2000 m

The approach hinges on the statistical properties of Zernike coefficients, derived from the equivalence between the angle-of-arrival correlation and the multi-aperture correlation. By utilizing a covariance matrix to define these correlations, the model compresses the wave propagation problem into a more computationally efficient sampling problem. The Zernike coefficients are drawn according to this covariance matrix, ensuring that the spatial correlations inherent in real turbulence effects are accurately represented. The model's ability to decouple tilts from higher-order aberrations leverages the fact that tilts occupy a majority of the turbulent energy, as shown [135, 134]. This decoupling allows for a more efficient simulation, as it separates the computationally intensive task of describing distortions per pixel into manageable components.

3.3 Phase-to-Space Simulator

The third method we utilize for generating turbulent images is called the Phase-to-Space (P2S) Transform [3]. Table 3 denotes the simulation parameters used for this technique in our experiments. The P2S Transform builds upon the second method [2] by reformulating the spatially varying convolution as a set of spatially invariant convolutions by leveraging basis functions. The source image x and the pupil image y are related through the linear operator H, which consists of spatially varying point spread functions (PSFs). To make the problem computationally feasible, each PSF h_n at pixel n is expressed as a linear combination of basis functions $\{\phi_m\}$ and coefficients $\{\beta_{m,n}\}$. This transformation turns the computationally expensive spatially varying convolutions into a set of spatially invariant convolutions, significantly reducing the computational cost if the number of basis functions M is much smaller than the number of pixels N. The basis functions are derived using a process that generates zero-mean Gaussian vectors with a covariance matrix determined by the turbulence parameters, such as the aperture diameter D and the Fried parameter r_0, based on Fried's statistical model [135]. Principal component analysis (PCA) is then employed to extract the spatial basis functions from a dataset of

PSFs representing various turbulence levels [2]. Once the basis functions are obtained, they are combined with the tilts and processed through a phase-to-space (P2S) transform, which is implemented via a shallow neural network. This network converts the Zernike coefficients of the phase to the coefficients of the basis functions in the spatial domain, eliminating the need to directly compute the PSFs at every pixel.

Table 3: Simulation parameters used for Phase-to-Space Simulator.

Image Dimensions	[640, 640] pixels
Aperture Diameter	29 mm
Wavelength	10.5 μm
Fried parameter	0.0145
Focal length	13 mm
Propagation Length	4000 m

CHAPTER 4

EXPERIMENTAL SETUP

To evaluate the impact of data augmentation on turbulent images, we focused on six state-of-the-art object detection models. These models were specifically chosen for their leading performance in various benchmarks. A brief description of each model is provided in Section 4.2. To train the models, we utilize the open-source frameworks MMDetection [136] and MMYOLO [137] except for YOLOR detector. DINO [40], RTMDet [39], and YOLOv8 [41] were trained using four NVIDIA Quadro RTX 8000 GPUs, while VfNet [36], TOOD [37], and YOLOR [38] were trained using four NVIDIA GeForce RTX 2080Ti GPUs. During the training of all models, the backbone weights were kept frozen. Table 4 provides the key specifications of the six selected models, detailing batch size, number of epochs, augmentation techniques used, backbone architecture, and their Average Precision (*AP*) scores on the COCO dataset.

Table 4: The configuration and runtime settings of the training processes, along with the published AP results on the COCO dataset.

	Batch Size	Epochs	Augmentations	Backbone	AP on COCO[1]
RTMDet-x	2	300	Random Flip Random Resize Random Crop Mosaic MixUp	CSPNeXt with P6 architecture	52.8
DINO-4scale	2	36	Random Resize Random Flip Random Crop	ResNet50	50.1
YOLOv8-x	2	500	Mosaic MixUp Random Flip Albumentations	YOLOv8CSPDarknet	52.7
YOLOR-P6	2	100	Mosaic MixUp HSV Augmentation Random Affine Transoformation	CSPDarkNet53	52.6
TOOD	2	100	Random Flip Random resize	ResNet50	44.5
VFNet	2	100	Random Flip Random Crop CutOut	ResNet50	48.0

[1] The *AP* scores in the table are sourced from the MMDetection [136] and MMYOLO [137] framework benchmark documentation, except for YOLOR-P6 [38], which sourced from its official publication.

4.1 The Dataset

The experiments in this thesis utilize the publicly available FLIR-ADAS v2 image set [47] to evaluate the impact of augmented turbulent samples on the performance of state-of-the-art object detection models adapted for thermal imagery. To this end, three different turbulence simulators were employed to augment the images, as described in Chapter 3. Examples of original and augmented turbulent images are displayed in Figure 3. FLIR-ADAS v2 is a medium-scale dataset annotated for object detection in both thermal and visible bands, comprising 26,442 annotated frames from videos and still images with 15 different object classes. The dataset includes 9,711 thermal and 9,233 RGB still images. For this study, we focused on the 9,711 thermal images captured with a Teledyne FLIR Tau 2, each with a resolution of 640×512 pixels. Our experiments specifically target the *car* and *person* classes due to the limited number of samples available for domain adaptation in other classes.

The original FLIR-ADAS v2 thermal still image set is split into 90% for training and 10% for validation. Following previous studies using this dataset [138, 139], we use the original 10% validation set as our test set and create a new validation set for fine-tuning purposes by taking 10% of the training set from the original FLIR-ADAS v2 thermal still image set.

To construct the turbulent augmentation for training, we use the approximation-based simulator [1] as previously described. We generate different versions of this turbulent training set with γ levels of 25, 50, 100, and 150. These various turbulent training sets are referred to as the "turbulent augmentation sets" and are used for training in experiments with turbulent image augmentation.

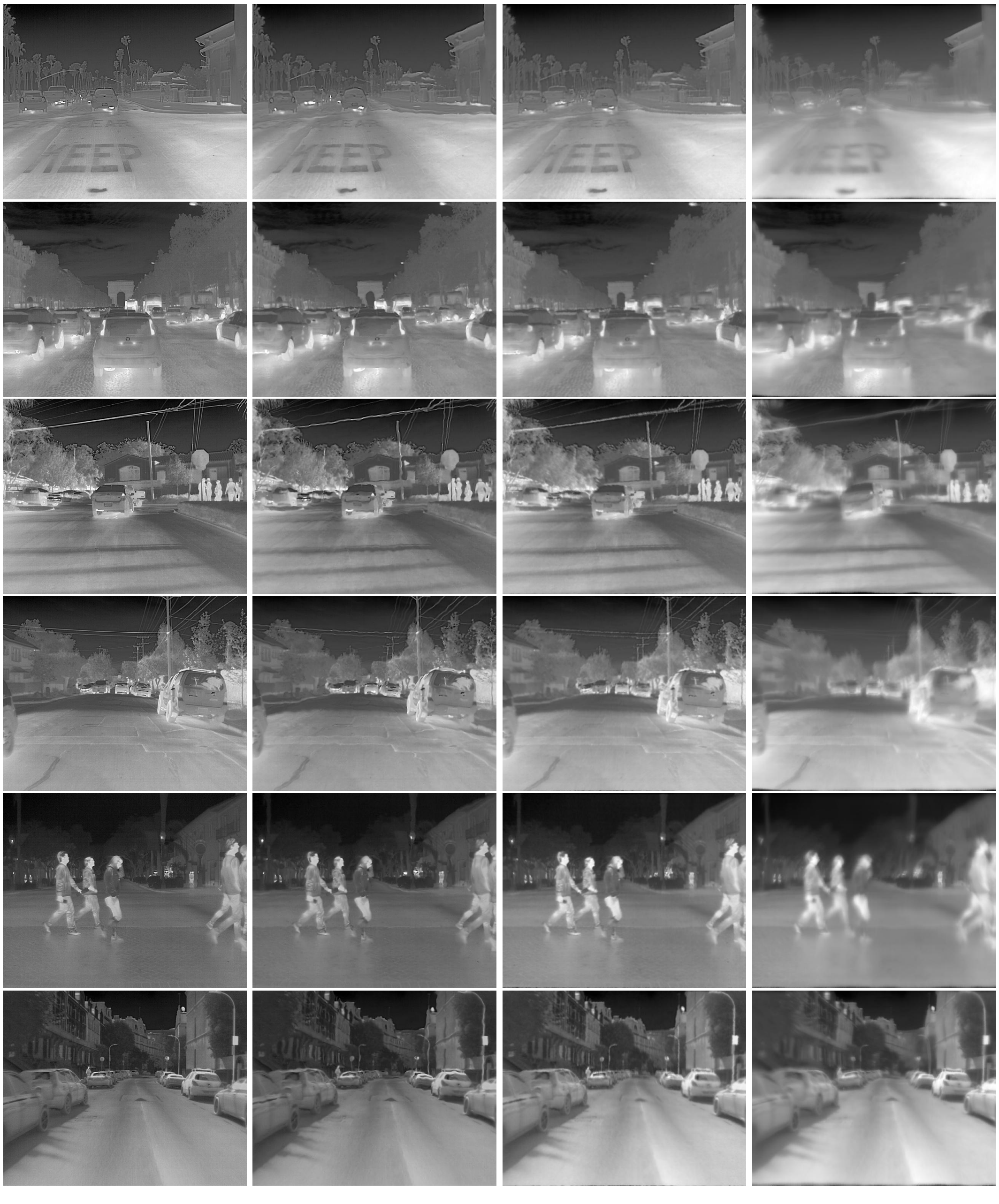
For the test sets, we employ all three turbulence simulators: geometric, Zernike-based, and P2S. Consequently, we obtain four distinct test sets: the original test set and the turbulent sets generated by the three simulators.

4.2 Object Detection Models

In recent years, there have been remarkable advancements in deep learning-based object detection models, with innovations focusing on enhancing accuracy, efficiency, and versatility. For a survey on the subject, readers may refer to [140, 141]. For our experiments, we selected six state-of-the-art models, DINO [40], RTMDet [39], YOLOv8 [41], VfNet [36], TOOD [37], and YOLOR [38].

4.2.1 DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection

DINO [40] is an enhanced object detection model built upon the DETR [142] framework. It introduces several innovations to improve performance and efficiency, including contrastive denoising training, mixed query selection, and a look-forward twice scheme for box prediction. Contrastive denoising training helps the model by adding both positive and negative samples of the same ground truth to stabilize training and reduce duplicate outputs. Mixed query selection enhances the initialization of anchor boxes as positional queries, while the look forward twice scheme refines box predictions by utilizing gradients from later layers to optimize parameters of early layers.



(a) Original Image (b) Geometric Sim. (c) Zernike Sim. (d) P2S Sim.

Figure 3: Original images (input) 3a alongside the outputs of three turbulence simulators: geometric simulator with $\gamma = 100$ 3b, Zernike-based simulator 3c, and P2S Simulator 3d.

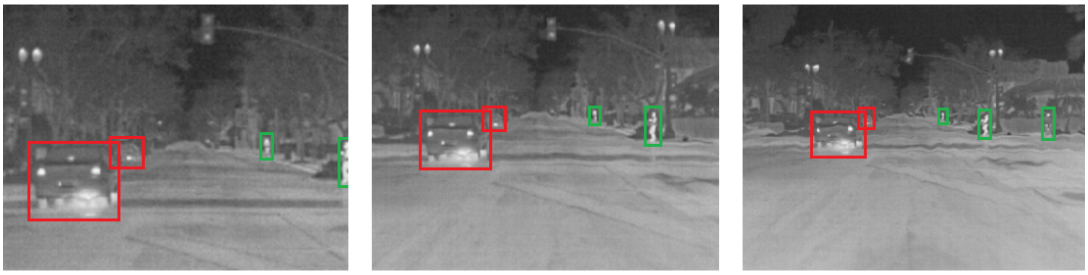


Figure 4: The leftmost image represents a patch from a sample image of the FLIR ADAS v2 dataset. The middle and rightmost images are patches from the same image, created using the proposed geometric turbulence model with γ values of 50 and 100, respectively. Annotations for the *person* and *car* classes are indicated by colored rectangles.

The improvements DINO brings are validated through extensive experiments on the COCO dataset, showing superior performance and faster convergence compared to previous DETR-like models. With a ResNet-50 backbone and multi-scale features, DINO achieves 49.4 AP in 12 epochs and 51.3 AP in 24 epochs, significantly outperforming its predecessors. Additionally, DINO demonstrates excellent scalability, achieving state-of-the-art results on COCO val2017 and test-dev benchmarks when pre-trained on the Objects365 dataset with a SwinL backbone, reducing the model size and pre-training data requirements while delivering better performance.

DINO's architecture comprises a backbone, a multi-layer Transformer encoder, a multi-layer Transformer decoder, and multiple prediction heads. It adopts deformable attention for computational efficiency and includes a denoising branch for improved stability during training. The model's query formulation links it closely with classical anchor-based detectors, while its innovative techniques, such as contrastive denoising and look forward twice, ensure high-quality anchor selection and accurate box prediction. This positions DINO as a leading end-to-end object detection model, setting new benchmarks in the field.

4.2.2 RTMDet: An Empirical Study of Designing Real-Time Object Detectors

RTMDet [39] presents an innovative macro architecture comprising a backbone, neck, and head, specifically optimized for real-time object detection. The backbone, inspired by CSPDarkNet, integrates large-kernel depth-wise convolutions to enhance context capture and improve model accuracy. This approach increases the effective receptive fields, allowing the model to understand the global context of images better. The neck, which uses the same building blocks as the backbone, is designed to balance computational load, thereby optimizing the speed-accuracy trade-off.

A key feature of RTMDet is its dynamic soft label assignment strategy, which improves the quality of matching predictions with ground truth boxes and reduces label assignment noise. This strategy enhances the discrimination in the cost matrix, leading to more precise object detection. Additionally, RTMDet employs advanced data augmentation techniques, such as cached Mosaic and MixUp, to enrich the training data and improve model robustness. The two-stage training strategy starts with strong augmentations and transitions to weaker ones, balancing augmentation strength and improving final model accuracy. These innovations make RTMDet highly effective for various tasks, including real-time object detection, instance segmentation, and rotated object detection.

The architecture's efficiency is further optimized by reducing the number of blocks in each backbone stage while enlarging the block width to maintain capacity and enhance parallelization. This approach ensures fast inference without sacrificing accuracy. RTMDet also incorporates a shared detection head with separate Batch Normalization layers for different feature scales, reducing parameters while maintaining performance. The training strategy leverages a combination of data augmentations, optimization techniques, and pre-training on general object detection datasets to further boost performance.

4.2.3 YOLOv8

YOLOv8 [41] is a groundbreaking advancement in object detection, boasting numerous enhancements over its predecessors. Key features include mosaic data augmentation, anchor-free detection, the C2f module, a decoupled head, and an innovative loss function. Mosaic data augmentation, which provides the model with better context information by mixing four images, is employed until the last ten training epochs, improving the model's final performance. This technique, inherited from YOLOv4, ensures that the model adapts better to varied contexts throughout most of the training while focusing on accuracy refinement towards the end.

Anchor-free detection marks a significant shift from predefined anchor boxes, which often slow down learning on custom datasets. By predicting objects' mid-points directly, YOLOv8 reduces the number of bounding box predictions, thus accelerating the Non-max Suppression (NMS) process. The C2f module, which replaces the C3 module in the backbone, concatenates outputs from all bottleneck modules, as opposed to only the last one. This change enhances the gradient flow and speeds up the training process, reducing computational costs.

A notable architectural enhancement in YOLOv8 is the decoupled head, which separates the classification and regression tasks. This division improves the overall model performance by allowing more specialized processing. However, this separation can lead to misalignment, where the model might localize one object while classifying another. To address this, YOLOv8 introduces a task alignment score, combining the classification score with the Intersection over Union (IoU) score to determine positive and negative samples. This alignment ensures that the model accurately matches predicted boxes with ground truth. The modified loss function in YOLOv8 further refines this process. By utilizing Binary Cross-Entropy (BCE) for classification loss and Complete IoU (CIoU) along with Distributional Focal Loss (DFL) for regression loss, the model achieves a more nuanced understanding of object boundaries. CIoU accounts for the spatial relationship between the predicted and actual boxes, considering factors like center point and aspect ratio. Meanwhile, DFL focuses on improving the predictions of samples that the model tends to misclassify, particularly false negatives, ensuring that bounding box boundaries are optimized effectively. This combination of advanced features and innovative techniques makes YOLOv8 a powerful and versatile tool in the field of object detection.

4.2.4 VarifocalNet (VfNet)

VFNet[36] is a dense object detector based on the FCOS+ATSS architecture, designed for high detection performance by accurately ranking candidate detections [143], [144]. A key component of

VFNet is the IoU-aware Classification Score (IACS), which merges object presence confidence and localization accuracy into a single score. This model includes three new components: Varifocal Loss, a star-shaped bounding box feature representation, and a bounding box refinement module. The star-shaped representation captures the geometry and context of bounding boxes using features at nine fixed sampling points, while the refinement module improves localization accuracy by learning residuals for bounding box adjustments.

VFNet employs Varifocal Loss, inspired by focal loss [145], to train the model for IACS prediction. Unlike focal loss, Varifocal Loss asymmetrically weights positive and negative examples, down-weighting only negative examples to address class imbalance while up-weighting high-quality positive examples to focus on precise detections. The training process also incorporates a dynamic label assignment strategy and advanced data augmentations like Mosaic and MixUp, which improve the robustness and accuracy of the model. The bounding box refinement step further enhances object localization by learning scaling factors for initial bounding box adjustments, ensuring the refined boxes are closer to the ground truth.

4.2.5 Task-aligned One-stage Object Detection (TOOD)

TOOD [37] is designed to address the common issue in one-stage object detectors where the two sub-tasks—object classification and localization—often suffer from spatial misalignment due to their independent processing. TOOD introduces a novel Task-aligned Head (T-Head) and a Task Alignment Learning (TAL) strategy to explicitly align these tasks during training. The T-Head balances learning task-interactive and task-specific features through a task-aligned predictor, enhancing the collaboration between classification and localization tasks. This is achieved by computing task-interactive features and making predictions via the Task-Aligned Predictor (TAP), which adjusts spatial distributions of predictions based on alignment signals from TAL.

The Task Alignment Learning (TAL) component further refines the model by dynamically aligning the optimal anchors for both tasks using a designed sample assignment scheme and a task-aligned loss. TAL assigns training samples based on a metric that evaluates the degree of task-alignment, ensuring that the most aligned anchors contribute to the training process. This metric is a combination of the classification score and IoU, allowing the network to focus on high-quality anchors that are well-aligned for both tasks. The task-aligned loss further unifies the anchors for classification and localization during training, resulting in a bounding box with the highest classification score and the most precise localization being preserved during inference.

Extensive experiments on the MS-COCO dataset demonstrate the effectiveness of TOOD. It achieves a remarkable 51.1 AP in single-model single-scale testing, significantly surpassing recent one-stage detectors like ATSS, GFL, and PAA, with fewer parameters and FLOPs. Qualitative results show TOOD's superiority in aligning classification and localization tasks, resulting in more accurate and consistent object detection. The design of T-Head and TAL collectively ensures high-quality predictions and state-of-the-art performance, making TOOD a robust solution for real-time object detection challenges

4.2.6 You Only Learn One Representation (YOLOR)

YOLOR[38] is a unified network designed to perform multiple tasks simultaneously, leveraging both explicit and implicit knowledge. The primary innovation of YOLOR lies in its ability to integrate these two types of knowledge to create a general representation that can be effectively utilized across various tasks. This approach mimics the human brain's ability to learn from both conscious and subconscious experiences.

The YOLOR model incorporates several key components to enhance its performance. It introduces kernel space alignment, prediction refinement, and multi-task learning into the network architecture. Kernel space alignment ensures that features from different tasks are harmonized, improving the network's overall effectiveness. Prediction refinement helps in fine-tuning the model's outputs, leading to more accurate results. Multi-task learning allows YOLOR to handle diverse tasks, such as object detection, instance segmentation, and image classification, within a single framework.

The network architecture of YOLOR is based on compressive sensing and deep learning principles. By combining these techniques, YOLOR constructs a unified network that can generate a general representation suitable for multiple tasks. This unified network is achieved with minimal additional computational cost, making it highly efficient. YOLOR's innovative use of implicit knowledge enables the model to capture the physical meaning of different tasks, leading to significant improvements in performance. The effectiveness of YOLOR is demonstrated through extensive experiments, showing its superior capability in handling various computer vision tasks with enhanced accuracy and efficiency.

CHAPTER 5

RESULTS

This chapter presents the experimental findings. Six state-of-the-art models were evaluated using both turbulent and original images, focusing on their object detection performance under challenging atmospheric conditions. The experiments were divided into two main sections. The first section 5.1, "Impact of Augmentation over Varying Gamma Levels," analyzes the effect of atmospheric distortion levels on deep object detectors and the improvements achieved with turbulent thermal image augmentation. The second section 5.2, "Impact of Augmentation over Different Turbulence Simulators," examines the impact of different atmospheric simulators and the enhancements provided by turbulent thermal image augmentation.

5.1 Impact of Augmentation over Varying Gamma Levels

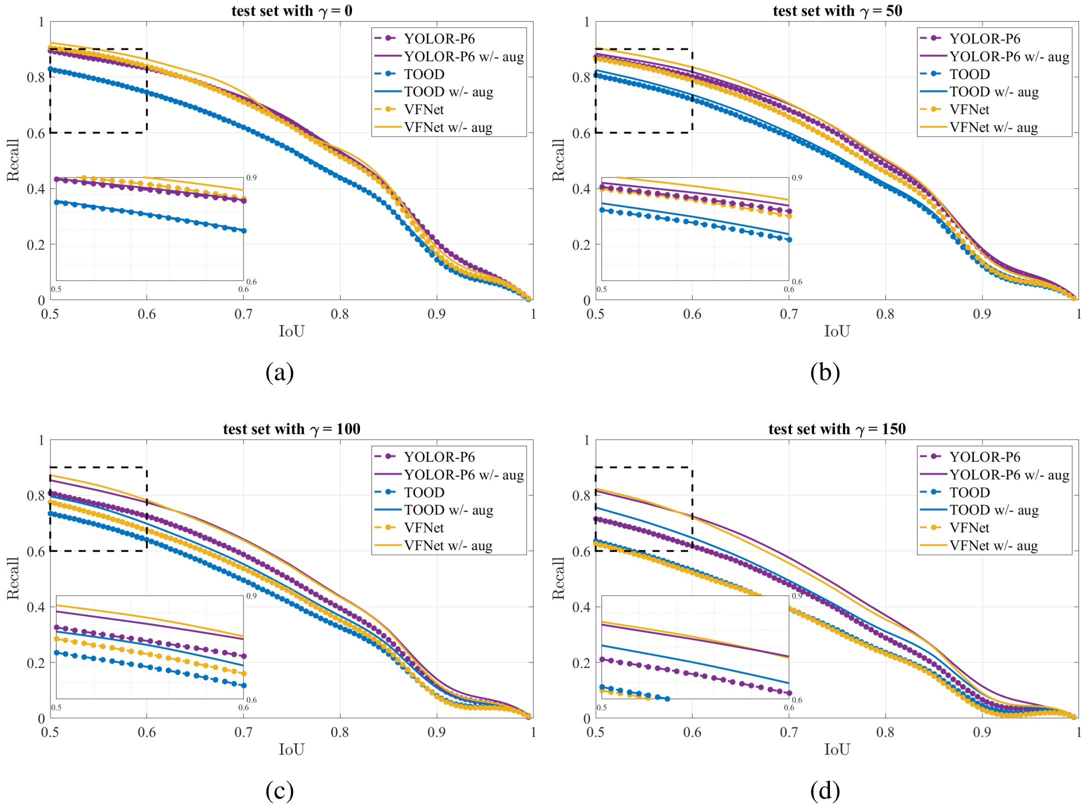
The results of all the experiments are presented as average precision (AP) values in Table 5. Each row in this table represents a separate experiment. The first column, labeled *Aug*, indicates whether turbulent data was augmented during training (w/) or not (w/o). The next two columns state the *Model* and the *Backbone* used. The *AP* column shows the mean average precision calculated over the two categories, while AP_{50} and AP_{75} represent the cases when IoU is set to 0.5 and 0.75, respectively. The last three columns display the AP values for the test subsets, where the test samples are categorized by their pixel size. AP_S is the performance score for the subset that includes test objects smaller than 32×32 pixels, whereas AP_L is calculated for test objects larger than 96×96 pixels. AP_M represents the performance for test objects sized in between.

An in-depth analysis of Table 5 reveals several key implications. Firstly, augmenting turbulent images with varying γ values during training consistently improves detection performance across all models and turbulence levels. This finding underscores the generalizability of the proposed augmentation for any thermal adaptation or model training experiment. Another vital observation is that as the turbulence level of the test set increases, the positive impact of the augmentation also increases, indicating the robustness of this approach.

Examining the individual behavior of the models, distinct characteristics emerge. While VfNet shows slightly better performance for test sets with lower turbulence levels, YOLOR generally becomes the top performer as turbulence levels rise. However, the performance difference between VfNet and YOLOR is negligible. Conversely, TOOD consistently performs worse than the other two models, aligning with its relative performance on RGB before thermal adaptation. The RGB performances are presented in Table 4 A consistent pattern across all models is that, despite their varying performance

Table 5: Mean Average Precision results obtained for different experiments with different models, with or without turbulent image augmentation for varying levels of turbulence gain γ.

Aug	**Model**	**Backbone**	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
test set without turbulence								
w/o	VfNet	ResNet -50	52.4	81.9	54.8	41.1	73.4	76.8
w/	VfNet	ResNet -50	54.7	83.8	57.1	43.7	75.2	79.9
w/o	TOOD	ResNet -50	46.1	74.6	47.3	33.1	71.1	76.3
w/	TOOD	ResNet -50	46.8	75.4	47.9	33.7	71.3	77.8
w/o	YOLOR	YOLOR-P6	53.1	80.5	56.2	40.6	77.3	82.9
w/	YOLOR	YOLOR-P6	53.8	81.0	57.2	41.4	77.5	83.3
test set with turbulence $\gamma = 25$								
w/o	VfNet	ResNet -50	51.0	80.4	53.1	39.2	73	77.0
w/	VfNet	ResNet -50	53.9	83.1	56.1	42.7	74.7	79.4
w/o	TOOD	ResNet -50	45.3	74.0	45.7	31.9	70.9	76.8
w/	TOOD	ResNet -50	46.2	75.0	46.7	33.2	71.1	77.0
w/o	YOLOR	YOLOR-P6	52.3	79.9	54.7	39.5	77.1	83.3
w/	YOLOR	YOLOR-P6	53.1	80.5	56.0	40.6	77.2	83.8
test set with turbulence $\gamma = 50$								
w/o	VfNet	ResNet -50	47.5	77.0	48.3	35.1	70.8	76.2
w/	VfNet	ResNet -50	51.8	81.4	53.5	39.8	73.9	79.8
w/o	TOOD	ResNet -50	43.4	71.9	44.0	29.7	69.7	77.0
w/	TOOD	ResNet -50	44.9	73.9	45.1	31.5	69.8	77.2
w/o	YOLOR	YOLOR-P6	50.0	77.8	51.8	36.5	75.7	82.8
w/	YOLOR	YOLOR-P6	51.4	79.4	53.7	38.5	76.2	83.7
test set with turbulence $\gamma = 100$								
w/o	VfNet	ResNet -50	36.7	64.4	36.1	23.4	61.3	72.5
w/	VfNet	ResNet -50	45.6	76.0	45.1	32.3	69.9	78.1
w/o	TOOD	ResNet -50	35.7	62.8	34.4	21.3	62.8	73.3
w/	TOOD	ResNet -50	40.4	69.7	39.0	26.4	66.2	75.3
w/o	YOLOR	YOLOR-P6	41.4	69.1	40.9	26.4	69.7	82.4
w/	YOLOR	YOLOR-P6	45.9	74.5	46.3	31.8	72.5	82.6
test set with turbulence $\gamma = 150$								
w/o	VfNet	ResNet -50	23.0	43.8	21.0	12.1	43.7	61.1
w/	VfNet	ResNet -50	38.9	68.6	36.5	25.0	64.2	76.2
w/o	TOOD	ResNet -50	26.0	49.3	24.1	13.0	49.4	68.1
w/	TOOD	ResNet -50	35.4	63.7	33.2	20.9	61.6	74.1
w/o	YOLOR	YOLOR-P6	31.1	55.8	29.9	16.2	58.2	78.6
w/	YOLOR	YOLOR-P6	39.7	67.9	38.4	24.5	67.4	81.3



(a) (b) (c) (d)

Figure 5: "IoU vs Recall" curves for all models, with and without turbulent data augmentation, for selected γ levels.

without turbulent data augmentation, our proposed augmentation improves their performance. This is evident in the experiments with the highest turbulence levels, where VfNet's performance becomes comparable to YOLOR's, with a significant increase in mean AP from 23.0 to 38.9.

The AP_S values in Table 5 indicate that detection performance for small objects significantly drops at high turbulence levels. This is expected, as objects farther from the camera not only appear smaller but are also more affected by turbulence. However, Table 5 clearly demonstrates that the proposed augmentation is most effective for small objects at higher turbulence levels. It is important to note that before adaptation, the benchmarked models were trained with various augmentation strategies, which did not mitigate high-level turbulence effects. For medium and large objects, the positive impact of the augmentation is minimal at low turbulence levels, yet it is never diminishing. The AP_M and AP_L values are consistently higher for experiments utilizing turbulent image augmentation.

To evaluate the localization success of each experiment, AP_{50} and AP_{75} columns are presented in Table 5. Additionally, "IoU vs Recall" graphs are given in Figure 5. In this figure, for different levels of γ, recall curves with respect to varying IoU are depicted, for all models, with or without turbulent data augmentation. Both the AP_{75} values in Table 5 and recall curves in Figure 5 show that YOLOR model provides superior localization. In addition, we see from Figures 5a, 5b, 5c and 5d that as the turbulence levels increase, the proposed augmentation becomes significantly more effective for any level of localization (i.e., IoU).

5.2 Impact of Augmentation over Different Turbulence Simulators

In this section, we present the results of our experiments for the second section. The evaluation metrics are based on the COCO (Common Objects in Context) metrics [146], similar to the first section. The evaluated scores of the metrics are presented as Tables 6, 7, 8, 9. The tables have identical layout. The first column indicates whether turbulent image augmentation was applied during training (w/ for "with" and w/o for "without"). The second column lists the model names. The next six columns present various Average Precision (AP) metrics: AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L. These (AP) values are calculated by determining the area under the precision-recall curve for IoU thresholds ranging from 0.50 to 0.95. Similarly, mean AP values are computed specifically at IoU thresholds of 0.50 and 0.75 by averaging the AP values from the precision-recall curve at these thresholds. Each row represents a specific experiment, showing the performance metrics of the corresponding model under the given condition. Horizontal lines separate different sections of the table, providing clarity in distinguishing between experiments conducted with different models. Tables 6, 7, 8, and 9 respectively present the results obtained using the original (i.e. clean) test dataset, geometric turbulence simulator, Zernike-based turbulence simulator, and P2S turbulence simulator applied to the test dataset.

Table 6: Mean Average Precision results for different experiments using each detection model, with or without turbulent image augmentation during training at varying levels of turbulence gain γ. These experiments use only the original test dataset.

Clean Test Set.

Aug	**Model**	***AP***	***AP$_{50}$***	***AP$_{75}$***	***AP$_S$***	***AP$_M$***	***AP$_L$***
w/o	RTMDet-x	56.8	84.3	60.5	45.3	78.3	81.8
w/	RTMDet-x	58.2	85.2	63.1	47.2	79.1	82.3
w/o	DINO-4scale	56.3	85.9	59.9	45.7	75.5	81.1
w/	DINO-4scale	57.8	87.7	61.8	47.5	77.1	81.8
w/o	YOLOv8-x	57.8	85.0	62.4	47.0	78.9	80.0
w/	YOLOv8-x	58.4	85.6	62.9	47.6	79.2	81.5

Table 7: Mean Average Precision results for different experiments using each detection model, trained with and without the turbulent augmentation set, and for varying levels of turbulence gain γ. In this experiment, the test set is constructed using the geometric simulator [1].

Test set with turbulence γ = 100

Aug	**Model**	***AP***	***AP$_{50}$***	***AP$_{75}$***	***AP$_S$***	***AP$_M$***	***AP$_L$***
w/o	RTMDet-x	41.8	69.9	41.9	27.8	67.7	78.2
w/	RTMDet-x	47.1	76.4	47.9	33.3	72.7	81.7
w/o	DINO-4scale	35.9	65.9	34.1	23.0	59.4	75.2
w/	DINO-4scale	44.6	77.2	43.9	32.0	67.7	78.5
w/o	YOLOv8-x	43.1	72.5	42.8	29.5	68.8	77.4
w/	YOLOv8-x	47.8	77.6	48.6	34.6	73.1	79.1

Table 8: Mean Average Precision results for different experiments using each detection model, trained with and without the turbulent augmentation set, and for varying levels of turbulence gain γ. In this experiment, the test set is constructed using the Zernike-based simulator [2].

Test set Zernike-method.

Aug	Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	RTMDet-x	36.2	63.5	34.9	20.7	65.9	76.0
w/	RTMDet-x	40.8	67.0	37.2	24.8	67.7	79.9
w/o	DINO-4scale	31.7	60.3	28.7	17.4	57.2	72.0
w/	DINO-4scale	34.9	64.3	32.1	20.3	61.7	75.1
w/o	YOLOv8-x	36.6	64.9	35.1	21.5	66.5	77.3
w/	YOLOv8-x	41.2	68.2	37.5	24.9	68.8	79.7

Table 9: Mean Average Precision results for different experiments using each detection model, trained with and without the turbulent augmentation set, and for varying levels of turbulence gain γ. In this experiment, the test set is constructed using the P2S-based simulator[3].

Test set with turbulence based-on P2S-method

Aug	Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
w/o	RTMDet-x	19.7	38.3	17.5	6.4	42.2	67.5
w/	RTMDet-x	24.5	49.1	21.2	9.6	49.8	74.5
w/o	DINO-4scale	17.1	35.9	14.4	5.3	37.0	64.7
w/	DINO-4scale	19.5	42.5	16.6	7.5	39.6	66.9
w/o	YOLOv8-x	23.0	46.7	19.5	7.6	47.7	69.2
w/	YOLOv8-x	24.8	49.9	21.4	9.8	49.9	73.0

The experimental results reported in Tables 6, 7, 8, and 9 provide valuable insights into the effectiveness of turbulence image augmentation for enhancing object detection performance under challenging atmospheric turbulence conditions. The following are key observations based on these results.

Firstly, as shown in Table 6, using turbulence image augmentation during training improves detection performance across all evaluated models, even when tested on the clean test set. This enhancement indicates that augmenting with turbulent samples helps the models generalize better and become more robust to various image degradations caused by atmospheric turbulence.

Furthermore, the results presented in Tables 7, 8, and 9 clearly demonstrate the critical importance of turbulence augmentation when dealing with images degraded by atmospheric turbulence. For all three turbulence simulators used, models trained with turbulence augmentation consistently outperform those trained without it. This pattern highlights the necessity of incorporating turbulence-specific augmentations to enhance detection accuracy in real-world scenarios affected by atmospheric distortions.

We observe a noticeable performance improvement with turbulence augmentation for smaller objects (see AP_S in tables) compared to larger objects (see AP_L in tables). This observation aligns with the expected impact of atmospheric turbulence, as clearly illustrated in Fig 3. Atmospheric turbulence tends to have a more significant disruptive effect on the visibility and detection of smaller objects due to their limited spatial area and lower contrast.

While all evaluated models benefit from turbulence augmentation, the degree of improvement varies across different architectures. For example, the RTMDet-x and YOLOv8-x models exhibit more significant performance enhancements compared to DINO-4scale, as evidenced by the larger differences in AP scores between their augmented and non-augmented versions shown in the Tables 7, 8, and 9.

Table 9 illustrates the most pronounced performance degradation compared to the other two simulators. The P2S-simulated tests exhibit more significant performance drops. This observation, along with Figure 3, suggests that the P2S simulator may produce more challenging turbulent images due to the harsh parameter settings employed in this study, as detailed in Table 3.

CHAPTER 6

CONCLUSIONS

This study explores the effectiveness of turbulence image augmentation for enhancing thermal-adapted object detection models under atmospheric turbulence. By employing three distinct turbulence simulators (geometric, Zernike-based, and P2S-based) and evaluating on corresponding turbulent test sets, the research demonstrates significant improvements in detection accuracy and robustness. These augmentations benefit all evaluated models (VfNet, TOOD, YOLOR-P6, RTMDet-x, DINO-4scale, and YOLOv8-x), although the degree of improvement varies, indicating that the specific design and characteristics of each model influence the augmentation's effectiveness.

Notably, turbulence augmentation proves particularly effective in improving the detection of smaller objects, which are more affected by atmospheric turbulence due to their limited spatial area and lower contrast. This finding underscores the importance of addressing the challenges posed by atmospheric turbulence for the detection of small-scale objects. Additionally, augmenting turbulent images of different severity levels always increases detection performance across all models, especially at high turbulence levels, bringing different model performances to a similar satisfactory level.

In summary, this research provides valuable insights for developing robust computer vision systems capable of operating in challenging atmospheric conditions. The findings underscore the importance of turbulence-specific augmentations to improve detection accuracy and robustness, contributing to the advancement of reliable computer vision solutions for real-world applications affected by atmospheric turbulence.

6.1 Future Directions

The study primarily focuses on thermal-adapted object detection, but the insights gained are likely applicable to other computer vision tasks affected by atmospheric turbulence, such as visible-light object detection, tracking, or segmentation. Future research could explore the generalization of these findings to different domains and applications.

The geometric distortions caused by atmospheric turbulence pose unique challenges that conventional augmentation methods, like affine transformations, cannot adequately simulate, especially for long-range vision problems and small objects. The proposed turbulence image augmentation strategy shows promise for becoming a standard practice for both visible and thermal vision systems utilizing deep learning models. Expanding this approach to other deep learning-based solutions for various IR vision problems could be highly beneficial. Additionally, employing sophisticated augmentation strategies like curriculum learning or reinforcement learning, could further optimize the augmentation process. Moreover, while the geometric turbulence model used in this study is computation-friendly and practical for online learning systems, incorporating physics-based models and collecting data from calibrated scenes would yield more realistic turbulent images, thereby enhancing the proposed augmentation strategy's impact.

REFERENCES

[1] E. Uzun, A. A. Dursun, and E. Akagündüz, “Augmentation of atmospheric turbulence effects on thermal adapted object detection models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 241–248, 2022.

[2] N. Chimitt and S. H. Chan, “Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated zernike coefficients,” *Optical Engineering*, vol. 59, no. 8, pp. 083101–083101, 2020.

[3] Z. Mao, N. Chimitt, and S. H. Chan, “Accelerating atmospheric turbulence simulation via learned phase-to-space transform,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14759–14768, 2021.

[4] J. R. Schott, *Remote sensing: the image chain approach.* Oxford University Press, 2007.

[5] J. W. Hardy, *Adaptive optics for astronomical telescopes*, vol. 16. Oxford University Press, USA, 1998.

[6] R. K. Tyson and B. W. Frazier, *Principles of adaptive optics.* CRC press, 2022.

[7] M. C. Roggemann and B. M. Welsh, *Imaging through turbulence.* CRC press, 2018.

[8] M. A. Hoffmire, R. C. Hardie, M. A. Rucci, R. Van Hook, and B. K. Karch, “Deep learning for anisoplanatic optical turbulence mitigation in long-range imaging,” *Optical Engineering*, vol. 60, no. 3, pp. 033103–033103, 2021.

[9] G. Holst, “Electro-optical imaging system performance,” 01 2000.

[10] C. Herrmann, M. Ruf, and J. Beyerer, “CNN-based thermal infrared person detection by domain adaptation,” in *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything* (M. C. Dudzik and J. C. Ricklin, eds.), vol. 10643, pp. 38 – 43, International Society for Optics and Photonics, SPIE, 2018.

[11] F. Munir, S. Azam, and M. Jeon, “Sstn: Self-supervised domain adaptation thermal object detection for autonomous driving,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 206–213, 2021.

[12] I. B. Akkaya, F. Altinel, and U. Halici, “Self-training guided adversarial domain adaptation for thermal imagery,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4322–4331, June 2021.

[13] M. C. Roggemann, B. M. Welsh, and B. R. Hunt, *Imaging through turbulence.* CRC press, 1996.

[14] K. I. Danaci and E. Akagunduz, “A survey on infrared image and video sets,” *arXiv*, vol. abs/2203.08581, 2022.

[15] E. Repasi and R. Weiss, "Computer simulation of image degradations by atmospheric turbulence for horizontal views," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII* (G. C. Holst and K. A. Krapels, eds.), vol. 8014, pp. 279 – 287, International Society for Optics and Photonics, SPIE, 2011.

[16] I. Ihrke, G. Ziegler, A. Tevs, C. Theobalt, M. Magnor, and H.-P. Seidel, "Eikonal rendering: Efficient light transport in refractive objects," *ACM Trans. on Graphics (Siggraph'07)*, p. to appear, Aug. 2007.

[17] D. Gutierrez, F. J. Serón, A. Muñoz, and O. Anson, "Simulation of atmospheric phenomena," *Comput. Graph.*, vol. 30, pp. 994–1010, 2006.

[18] R. Yasarla and V. M. Patel, "Learning to restore images degraded by atmospheric turbulence using uncertainty," in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1694–1698, IEEE, 2021.

[19] Y. Lou, S. H. Kang, S. Soatto, and A. L. Bertozzi, "Video stabilization of atmospheric turbulence distortion," *Inverse Probl. Imaging*, vol. 7, no. 3, pp. 839–861, 2013.

[20] Z. Mao, A. Jaiswal, Z. Wang, and S. H. Chan, "Single frame atmospheric turbulence mitigation: A benchmark study and a new physics-inspired transformer model," in *European Conference on Computer Vision*, pp. 430–446, Springer, 2022.

[21] O. Oreifej, X. Li, and M. Shah, "Simultaneous video stabilization and moving object detection in turbulence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 2, pp. 450–462, 2012.

[22] E. Chen, O. Haik, and Y. Yitzhaky, "Detecting and tracking moving objects in long-distance imaging through turbulent medium," *Applied optics*, vol. 53, no. 6, pp. 1181–1190, 2014.

[23] X. Wu, D. Hong, and J. Chanussot, "Uiu-net: U-net in u-net for infrared small object detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 364–376, 2022.

[24] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "Yolo-firi: Improved yolov5 for infrared image object detection," *IEEE access*, vol. 9, pp. 141861–141875, 2021.

[25] X. Dai, X. Yuan, and X. Wei, "Tirnet: Object detection in thermal infrared images for autonomous driving," *Applied Intelligence*, vol. 51, pp. 1244–1261, 2021.

[26] C. Jiang, H. Ren, X. Ye, J. Zhu, H. Zeng, Y. Nan, M. Sun, X. Ren, and H. Huo, "Object detection from uav thermal infrared images and videos using yolo models," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102912, 2022.

[27] Y. Chen, L. Li, X. Liu, and X. Su, "A multi-task framework for infrared small target detection and segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–9, 2022.

[28] F. Erlenbusch, C. Merkt, B. de Oliveira, A. Gatter, F. Schwenker, U. Klauck, and M. Teutsch, "Thermal infrared single image dehazing and blind image quality assessment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 459–469, 2023.

[29] C. Li, H. Zhou, Y. Liu, C. Yang, Y. Xie, Z. Li, and L. Zhu, "Detection-friendly dehazing: Object detection in real-world hazy scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[30] X. Xu, P. Yang, H. Xian, and Y. Liu, "Robust moving objects detection in long-distance imaging through turbulent medium," *Infrared Physics & Technology*, vol. 100, pp. 87–98, 2019.

[31] J. M. Patel, D. Israni, and C. Bhatt, "The comprehensive art of atmospheric turbulence mitigation methodologies for visible and infrared sequences," in *Advances in Information Communication Technology and Computing: Proceedings of AICTC 2021*, pp. 145–153, Springer, 2022.

[32] R. Nieuwenhuizen, J. Dijk, and K. Schutte, "Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects," *EURASIP journal on image and video processing*, vol. 2019, pp. 1–22, 2019.

[33] N. G. Nair, K. Mei, and V. M. Patel, "At-ddpm: Restoring faces degraded by atmospheric turbulence using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3434–3443, 2023.

[34] T. Jain, M. Lubien, and J. Gilles, "Evaluation of neural network algorithms for atmospheric turbulence mitigation," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI*, vol. 12122, pp. 223–236, SPIE, 2022.

[35] W. H. Chak, C. P. Lau, and L. M. Lui, "Subsampled turbulence removal network," *arXiv preprint arXiv:1807.04418*, 2018.

[36] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "Varifocalnet: An iou-aware dense object detector," *arXiv preprint arXiv:2008.13367*, 2020.

[37] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "Tood: Task-aligned one-stage object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3490–3499, IEEE Computer Society, 2021.

[38] C.-Y. Wang, I.-H. Yeh, and H. Liao, "You only learn one representation: Unified network for multiple tasks," *ArXiv*, vol. abs/2105.04206, 2021.

[39] C. Lyu, W. Zhang, H. Huang, Y. Zhou, Y. Wang, Y. Liu, S. Zhang, and K. Chen, "Rtmdet: An empirical study of designing real-time object detectors," *arXiv preprint arXiv:2212.07784*, 2022.

[40] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022.

[41] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023.

[42] S. Rai and C. Jawahar, *Learning to Generate Atmospheric Turbulent Images*, pp. 261–271. 11 2020.

[43] N. Chimitt and S. H. Chan, "Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients," *Optical Engineering*, vol. 59, no. 8, pp. 1 – 26, 2020.

[44] A. Schwartzman, M. Alterman, R. Zamir, and Y. Y. Schechner, "Turbulence-induced 2d correlated image distortion," in *2017 IEEE International Conference on Computational Photography (ICCP)*, pp. 1–13, IEEE, 2017.

[45] R. C. Hardie and D. A. LeMaster, "On the simulation and mitigation of anisoplanatic optical turbulence for long range imaging," in *Long-Range Imaging II* (E. J. Kelmelis, ed.), vol. 10204, pp. 40 – 59, International Society for Optics and Photonics, SPIE, 2017.

[46] R. L. Espinola, K. R. Leonard, K. A. Byrd, and G. Potvin, "Atmospheric turbulence and sensor system effects on biometric algorithm performance," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXVI* (G. C. Holst and K. A. Krapels, eds.), vol. 9452, pp. 120 – 129, International Society for Optics and Photonics, SPIE, 2015.

[47] FLIR, "Free flir thermal dataset for algorithm training," *https://www.flir.com/oem/adas/adas-dataset-form/*.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[49] Y. Wei, H. Hu, Z. Xie, Z. Zhang, Y. Cao, J. Bao, D. Chen, and B. Guo, "Contrastive learning rivals masked image modeling in fine-tuning via feature distillation," *arXiv preprint arXiv:2205.14141*, 2022.

[50] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, *et al.*, "Image as a foreign language: Beit pretraining for all vision and vision-language tasks," *arXiv preprint arXiv:2208.10442*, 2022.

[51] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," *arXiv preprint arXiv:2203.16527*, 2022.

[52] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[53] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv: 2205.11487*, 2022.

[54] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[55] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1476–1485, 2019.

[56] X. Yang, Z. Song, I. King, and Z. Xu, "A survey on deep semi-supervised learning," *arXiv preprint arXiv:2103.00550*, 2021.

[57] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *arXiv preprint arXiv:2205.01491*, 2022.

[58] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[59] J. Nalepa, M. Marcinkiewicz, and M. Kawulok, “Data augmentation for brain-tumor segmentation: a review,” *Frontiers in computational neuroscience*, vol. 13, p. 83, 2019.

[60] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “Augmix: A simple data processing method to improve robustness and uncertainty,” *arXiv preprint arXiv:1912.02781*, 2019.

[61] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[62] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, “Scaled-yolov4: Scaling cross stage partial network,” in *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 13029–13038, 2021.

[63] W. Ma, Y. Wu, F. Cen, and G. Wang, “Mdfn: Multi-scale deep feature learning network for object detection,” *Pattern Recognition*, vol. 100, p. 107149, 2020.

[64] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “You only learn one representation: Unified network for multiple tasks,” *arXiv preprint arXiv:2105.04206*, 2021.

[65] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[66] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning*, pp. 10347–10357, PMLR, 2021.

[67] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[68] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.

[69] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, “Cutmix: Regularization strategy to train strong classifiers with localizable features,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

[70] D. Walawalkar, Z. Shen, Z. Liu, and M. Savvides, “Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification,” *arXiv preprint arXiv:2003.13048*, 2020.

[71] J.-H. Kim, W. Choo, and H. O. Song, “Puzzle mix: Exploiting saliency and local statistics for optimal mixup,” in *International Conference on Machine Learning*, pp. 5275–5285, PMLR, 2020.

[72] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, “Manifold mixup: Better representations by interpolating hidden states,” in *International conference on machine learning*, pp. 6438–6447, PMLR, 2019.

[73] A. Uddin, M. Monira, W. Shin, T. Chung, S.-H. Bae, *et al.*, “Saliencymix: A saliency guided data augmentation strategy for better regularization,” *arXiv preprint arXiv:2006.01791*, 2020.

[74] S. Huang, X. Wang, and D. Tao, “Snapmix: Semantically proportional mixing for augmenting fine-grained data,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 1628–1636, 2021.

[75] R. Takahashi, T. Matsubara, and K. Uehara, “Ricap: Random image cropping and patching data augmentation for deep cnns,” in *Asian conference on machine learning*, pp. 786–798, PMLR, 2018.

[76] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.

[77] H. Inoue, “Data augmentation by pairing samples for images classification,” *arXiv preprint arXiv:1801.02929*, 2018.

[78] Y. Tokozume, Y. Ushiku, and T. Harada, “Learning from between-class examples for deep sound recognition,” *arXiv preprint arXiv:1711.10282*, 2017.

[79] Y. Tokozume, Y. Ushiku, and T. Harada, “Between-class learning for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5486–5494, 2018.

[80] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, “Co-mixup: Saliency guided joint mixup with supermodular diversity,” *arXiv preprint arXiv:2102.03065*, 2021.

[81] A. Dabouei, S. Soleymani, F. Taherkhani, and N. M. Nasrabadi, “Supermix: Supervising the mixing data augmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13794–13803, 2021.

[82] K. Baek, D. Bang, and H. Shim, “Gridmix: Strong regularization through local context mapping,” *Pattern Recognition*, vol. 109, p. 107594, 2021.

[83] D. Dwibedi, I. Misra, and M. Hebert, “Cut, paste and learn: Surprisingly easy synthesis for instance detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1301–1310, 2017.

[84] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, “Synthesizing training data for object detection in indoor scenes,” *arXiv preprint arXiv:1702.07836*, 2017.

[85] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, “Simple copy-paste is a strong data augmentation method for instance segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2918–2928, 2021.

[86] Z. Xu, A. Meng, Z. Shi, W. Yang, Z. Chen, and L. Huang, “Continuous copy-paste for one-stage multi-object tracking and segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15323–15332, 2021.

[87] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “Synthetic data augmentation using gan for improved liver lesion classification,” in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 289–293, IEEE, 2018.

[88] A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood, "Chest x-ray generation and data augmentation for cardiovascular abnormality classification," in *Medical imaging 2018: Image processing*, vol. 10574, pp. 415–420, SPIE, 2018.

[89] H. Yang and Y. Zhou, "Ida-gan: a novel imbalanced data augmentation gan," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8299–8305, IEEE, 2021.

[90] A. Ali-Gombe and E. Elyan, "Mfc-gan: class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, 2019.

[91] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[92] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Systems with applications*, vol. 91, pp. 464–471, 2018.

[93] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "Auggan: Cross domain adaptation with gan-based data augmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 718–731, 2018.

[94] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, and H. T. Campus, "Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants.," in *BMVC*, p. 324, 2018.

[95] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[96] R. Li, W. Cao, Q. Jiao, S. Wu, and H.-S. Wong, "Simplified unsupervised image translation for semantic segmentation adaptation," *Pattern Recognition*, vol. 105, p. 107343, 2020.

[97] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 349–360, Springer, 2018.

[98] Z. Zheng, Z. Yu, Y. Wu, H. Zheng, B. Zheng, and M. Lee, "Generative adversarial network with multi-branch discriminator for imbalanced cross-species image-to-image translation," *Neural Networks*, vol. 141, pp. 355–371, 2021.

[99] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, A. Kumar, R. Feris, R. Giryes, and A. Bronstein, "Delta-encoder: an effective sample synthesis method for few-shot object recognition," *Advances in neural information processing systems*, vol. 31, 2018.

[100] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[101] M. Hong, J. Choi, and G. Kim, "Stylemix: Separating content and style for enhanced data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14862–14870, 2021.

[102] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

[103] M. Xu, S. Yoon, A. Fuentes, J. Yang, and D. S. Park, “Style-consistent image translation: A novel data augmentation paradigm to improve plant disease recognition.,” *Frontiers in Plant Science*, vol. 12, pp. 773142–773142, 2021.

[104] Y. Li, Q. Yu, M. Tan, J. Mei, P. Tang, W. Shen, A. Yuille, and C. Xie, “Shape-texture debiased neural network training,” *arXiv preprint arXiv:2010.05981*, 2020.

[105] J. P. Bos and M. C. Roggemann, “Technique for simulating anisoplanatic image formation over long horizontal paths,” *Optical Engineering*, vol. 51, no. 10, pp. 101704–101704, 2012.

[106] Z. Fabian, R. Heckel, and M. Soltanolkotabi, “Data augmentation for deep learning based accelerated mri reconstruction with limited data,” in *International Conference on Machine Learning*, pp. 3057–3067, PMLR, 2021.

[107] D. Eckert, S. Vesal, L. Ritschl, S. Kappler, and A. Maier, “Deep learning-based denoising of mammographic images using physics-driven data augmentation,” in *Bildverarbeitung für die Medizin 2020: Algorithmen–Systeme–Anwendungen. Proceedings des Workshops vom 15. bis 17. März 2020 in Berlin*, pp. 94–100, Springer, 2020.

[108] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, “Self-supervised physics-based deep learning mri reconstruction without fully-sampled data,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pp. 921–925, IEEE, 2020.

[109] A. D. Desai, B. Gunel, B. M. Ozturkler, H. Beg, S. Vasanawala, B. A. Hargreaves, C. Ré, J. M. Pauly, and A. S. Chaudhari, “Vortex: Physics-driven data augmentations using consistency training for robust accelerated mri reconstruction,” *arXiv preprint arXiv:2111.02549*, 2021.

[110] W. Liu, C. Wang, and C. Ding, “Physics-based appearance and illumination estimation from a single face image,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, 2022.

[111] W. Luo, Z. Yan, Q. Song, and R. Tan, “Physics-directed data augmentation for deep model transfer to specific sensor,” *ACM Transactions on Sensor Networks*, vol. 19, no. 1, pp. 1–30, 2022.

[112] M. Crosskey, P. Wang, R. Sakaguchi, and K. D. Morton Jr, “Physics-based data augmentation for high frequency 3d radar systems,” in *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIII*, vol. 10628, pp. 430–440, SPIE, 2018.

[113] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, “Single image deraining: From model-based to data-driven and beyond,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4059–4077, 2020.

[114] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1357–1366, 2017.

[115] W. Yang, J. Liu, and J. Feng, “Frame-consistent recurrent video deraining with dual-level flow,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1661–1670, 2019.

[116] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proceedings of the IEEE international conference on computer vision*, pp. 3397–3405, 2015.

[117] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2736–2744, 2016.

[118] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3233–3242, 2018.

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

[120] K. Mei and V. M. Patel, "Ltt-gan: Looking through turbulence by inverting gans," *IEEE Journal of Selected Topics in Signal Processing*, 2023.

[121] N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2398–2408, 2013.

[122] Y. Yang, X. Zhang, Q. Guan, and Y. Lin, "Making invisible visible: Data-driven seismic inversion with spatio-temporally constrained data augmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2022.

[123] B. Fishbain, L. P. Yaroslavsky, and I. A. Ideses, "Real-time stabilization of long range observation system turbulent video," *Journal of Real-Time Image Processing*, vol. 2, pp. 11–22, 2007.

[124] X. Zhu and P. Milanfar, "Removing atmospheric turbulence via space-invariant deconvolution," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 157–170, 2012.

[125] J. Choi and Y. Kim, "Colorful cutout: Enhancing image data augmentation with curriculum learning," *arXiv preprint arXiv:2403.20012*, 2024.

[126] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

[127] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proceedings of the IEEE international conference on computer vision*, pp. 2488–2496, 2015.

[128] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3688–3692, Ieee, 2016.

[129] A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," *Advances in neural information processing systems*, vol. 30, 2017.

[130] X. Zhang, Q. Wang, J. Zhang, and Z. Zhong, "Adversarial autoaugment," *arXiv preprint arXiv:1912.11188*, 2019.

[131] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, “Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2226–2234, 2018.

[132] Y. Yitzhaky, E. Chen, and O. Haik, “Surveillance in long-distance turbulence-degraded videos,” in *Electro-Optical Remote Sensing, Photonic Technologies, and Applications VII; and Military Applications in Hyperspectral Imaging and High Spatial Resolution Sensing*, vol. 8897, pp. 26–31, SPIE, 2013.

[133] G. Stroe and I.-C. Andrei, “Analysis regarding the effects of atmospheric turbulence on aircraft dynamics,” *INCAS Bulletin*, vol. 8, no. 2, p. 123, 2016.

[134] R. J. Noll, “Zernike polynomials and atmospheric turbulence,” *JOsA*, vol. 66, no. 3, pp. 207–211, 1976.

[135] D. L. Fried, “Optical resolution through a randomly inhomogeneous medium for very long and very short exposures,” *Journal of the Optical Society of America*, vol. 56, no. 10, pp. 1372–1379, 1966.

[136] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, *et al.*, “Mmdetection: Open mmlab detection toolbox and benchmark,” *arXiv preprint arXiv:1906.07155*, 2019.

[137] M. Contributors, “MMYOLO: OpenMMLab YOLO series toolbox and benchmark.” `https://github.com/open-mmlab/mmyolo`, 2022.

[138] F. Munir, S. Azam, and M. Jeon, “Sstn: Self-supervised domain adaptation thermal object detection for autonomous driving,” in *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 206–213, IEEE, 2021.

[139] M. A. Farooq, P. Corcoran, C. Rotariu, and W. Shariff, “Object detection in thermal spectrum for advanced driver-assistance systems (adas),” *IEEE Access*, vol. 9, pp. 156465–156481, 2021.

[140] X. Wang, H. Li, X. Yue, and L. Meng, “A comprehensive survey on object detection yolo,” *Proceedings http://ceur-ws. org ISSN*, vol. 1613, p. 0073, 2023.

[141] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *Proceedings of the IEEE*, 2023.

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

[143] Z. Tian, C. Shen, H. Chen, and T. He, “Fcos: Fully convolutional one-stage object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627–9636, 2019.

[144] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, “Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9759–9768, 2020.

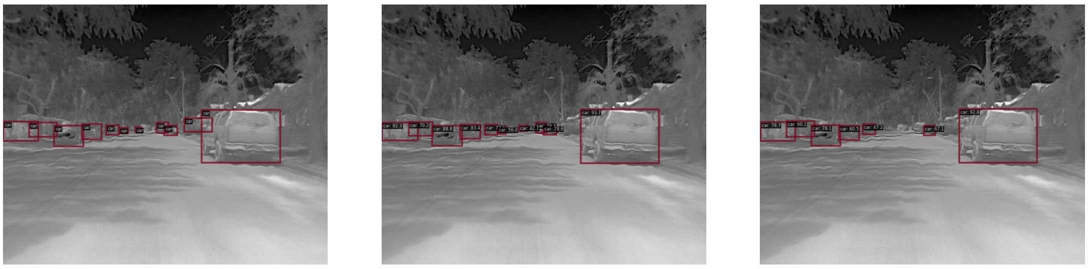
[145] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[146] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

APPENDIX A

ADDITIONAL VISUAL RESULTS

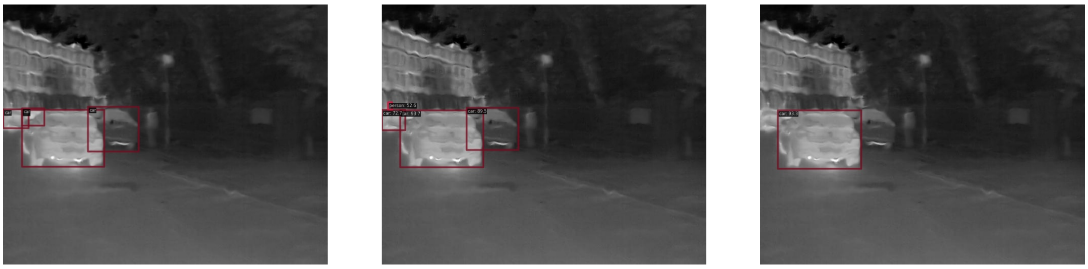
The following figures are presented to provide additional insights and visual support for the findings discussed in the main chapters. These illustrations serve to clarify and expand upon key aspects of the research, offering detailed results. Figure 6 presents the visual results from experiments using YOLOv8 on turbulent images generated by the Geometric Simulator, with the γ value set to 100. Figure 6a shows the ground truth image generated with the Geometric Simulator at $\gamma = 100$, while



(a) Ground Truth (b) YOLOv8-aug (c) YOLOv8-w/o aug

Figure 6: YOLOv8 models (augmented and non-augmented) result over random selected turbulent test image.

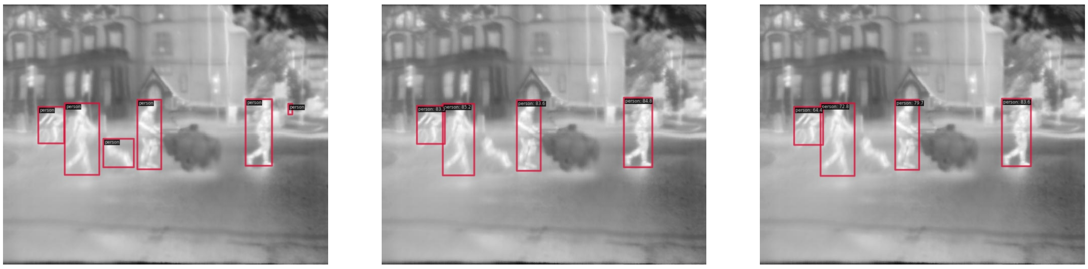
Figures 6b and 6c present the detection results obtained using the YOLOv8 model trained with both clean and turbulent augmented images, and only clean images, respectively.



(a) Ground Truth (b) DINO-aug (c) DINO- w/o aug

Figure 7: DINO models (augmented and non-augmented) result over random selected turbulent test image.

Figure 7 presents another visual result, similar to Figure 6. Figure 7a is ground truth image generated with the Geometric Simulator at $\gamma = 150$, while Figures 7b and 7c present the detection results obtained using the DINO model trained with both clean and turbulent augmented images, and only clean images, respectively.



(a) Ground Truth

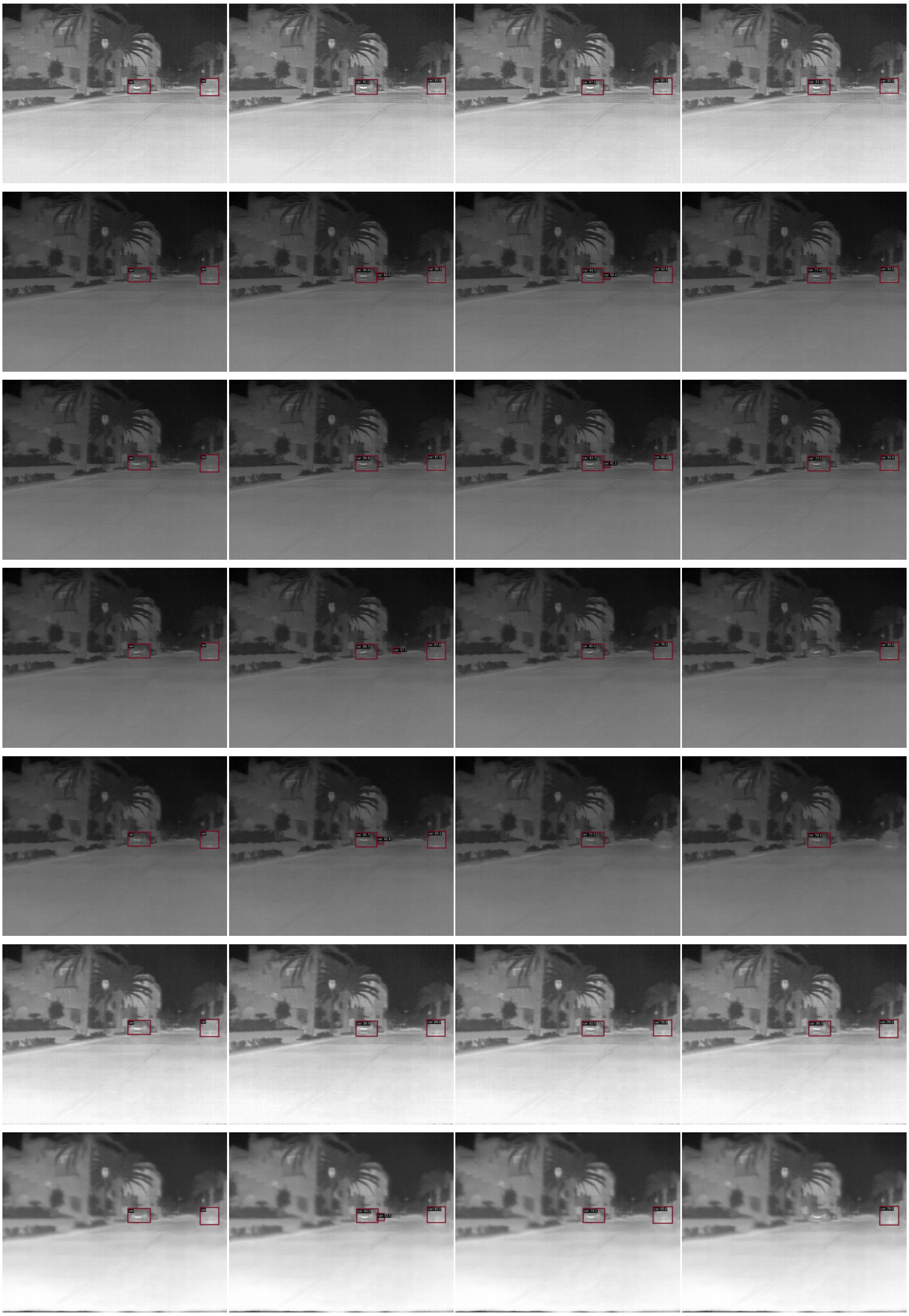
(b) RTMDet-aug

(c) RTMDet-w/o aug

Figure 8: RTMDet models (augmented and non-augmented) result over random selected turbulent test image.

Figure 8 is ground truth image generated with P2S(Phase-to-Space) Simulator at given parameters in Table 3, while Figures 8b and 8c present the detection results obtained using the DINO model trained with both clean and turbulent augmented images, and only clean images, respectively.

Figures 9 and 10 present comprehensive visual results. In both figures, the first column displays the ground truth images, which consist of a clean image, followed by images generated by the Geometric Simulator at $\gamma = 25$, $\gamma = 50$, $\gamma = 100$, and $\gamma = 150$, as well as images from the Zernike Simulator and the P2S Simulator, respectively, from top to bottom. The second, third, and fourth columns of Figure 9 represent the detection inference results from YOLOv8, RTMDet, and DINO models trained with only clean images, respectively. In contrast, Figure 10 presents the corresponding detection inference results from the same models trained with both clean and turbulent augmented images.



Ground Truth YOLOv8 RTMDet DINO

Figure 9: The results were obtained from models without turbulence augmentation over all test sets.

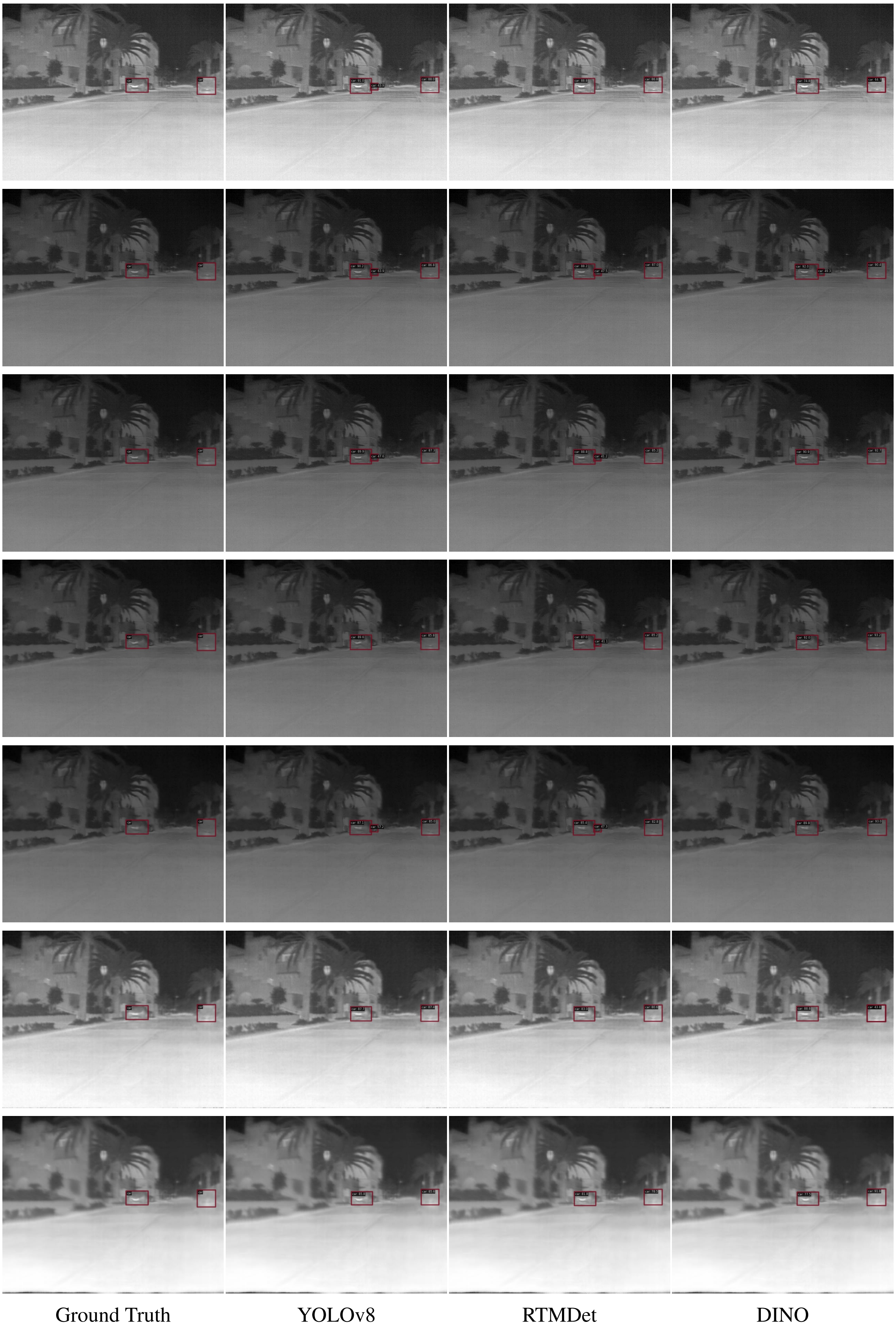


Figure 10: The results were obtained from models with turbulence augmentation over all test sets.