Computer Vision for Atmospheric Turbulence: Generation, Restoration, and its Applications

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering by Research

by

Shyam Nandan Rai
20172088
shyam.nandan@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
December 2020

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled **" Computer Vision for Atmospheric Turbulence: Generation, Restoration,and its Applications"** by **Shyam Nandan Rai**, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. C.V. Jawahar
Date	Adviser: Prof. Vineeth N Balasubramanian
Date	Adviser: Dr. Anbumani Subramanian

To Papa

Acknowledgments

I would like to express my sincere gratitude to my adviser, Prof. C.V. Jawahar, who guided me through the path of becoming ”a research student” from ”a student” . He taught me the importance of commitment and perseverance in research, which would be of great help in my future academic journey. Next, I would like to thank my co-advisers Prof. Vineeth N Balasubramanian and Dr. Anbumani Subramanian, for all the suggestions and pieces of advice, which greatly helped in improving the quality of work. I owe a special thanks to Prof. Anand Mishra, who helped me start my research career.

I am really fortunate to have friend Ashutosh Mishra who has been a constant source of invaluable support and motivation, which made my MS journey a lot easier. I am grateful to my CVIT seniors Praveen sir, Aditya sir, Thrupthi mam, Ankit sir, Anil sir and Ajoy sir for giving insightful reviews and sharing their knowledge and experience. I would also like to thank my MS buddies Vamshi, Siddhant, Chris, Ashish, and Santoshini. I also thank Kishan, Ashish, Kranthi, Chandrakant, Vishal, Aakash, Vanditha, and Nikunj for all the fun we had and the beautiful memories we made.

I acknowledge the support and great love of my family, that kept me going on, and this work would not have been possible without their help. I am thankful to my father and brother, who always helped me make tough decisions.

Abstract

Real-world images often suffer from variation in weather conditions such as rain, fog, snow, and temperature. These variations in the atmosphere adversely affect the performance of computer vision models in real-world scenarios. This problem can be bypassed by collecting and annotating images for each weather. However, collecting and annotating images in such conditions is an extremely tedious task, which is time-consuming as well as expensive. So, in this work, we address the forementioned problems. Among all the weather conditions, we focus on the distortions in the image caused by high temperature, also known as atmospheric turbulence. These distortions introduce geometrical deformation around the boundaries of an object in an image which causes a vision algorithm to perform poorly and pose a major concern. Hence, in this thesis, we address the problem of artificially generating atmospheric turbulence and restoring the images from it.

In the first part of our work, we attempt to model atmospheric turbulence. Since such models are critical to extending computer vision solutions developed in the laboratory to real-world use cases. And, simulating atmospheric turbulence by using statistical models or by computer graphics is often computationally expensive. To overcome this problem, we train a generative adversarial network(GAN) which outputs an atmospheric turbulent image by utilizing less computational resources than traditional methods. We propose a novel loss function to efficiently learn the atmospheric turbulence at the finer level. Experiments show that by using the proposed loss function, our network outperforms the existing state-of-the-art image to image translation network in turbulent image generation.

In the second part of the thesis, we address the ill-posed problem of restoring images degraded due to atmospheric turbulence. We propose a deep adversarial network to recover the images which are distorted due to atmospheric turbulence and show the applicability of restored images in several tasks. Unlike previous methods, our approach neither uses any prior knowledge about atmospheric turbulence conditions at inference time nor requires the fusion of multiple images to get a single restored image. To train our models, we synthesized turbulent images by following a series of efficient 2D operations. Thereafter, using our trained models we run inference on real and synthesized turbulent images. Our final restoration models DT-GAN+ and DTD-GAN+ qualitatively and quantitatively outperforms the general state-of-the-art image-to-image translation models. The improved performance of our model is due to the use of optimized residual structures along with channel attention and sub-pixel mechanism which exploits the information between the channels and removes atmospheric turbulence at the finer

level. We also perform extensive experiments on restored images by utilizing them for downstream tasks such as classification, pose estimation, semantic keypoint estimation, and depth estimation.

In the third part of our work, we study the problem of the semantic segmentation model in adapting to hot climate cities. This issue can be circumvented by collecting and annotating images in such weather conditions and training segmentation models on those images. But, the task of semantically annotating images for every environment is painstaking and expensive. Hence, we propose a framework that improves the performance of semantic segmentation models without explicitly creating an annotated dataset for such adverse weather variations. Our framework consists of two parts, a restoration network to remove the geometrical distortions caused by hot weather and an adaptive segmentation network that is trained on an additional loss to adapt to the statistics of the ground-truth segmentation map. We train our framework on the Cityscapes dataset, which showed a total IoU gain of 12.707 over standard segmentation models.

In the last part of our work, we improve the performance of our joint restoration and segmentation network via a feedback mechanism. In, the previous approach the restoration network does not learn directly from the errors of the segmentation network. In other words, the restoration network is not task aware. Hence, we propose a semantic feedback learning approach, which improves the task of semantic segmentation giving a feedback response into the restoration network. This response works as an attend and fix mechanism by focusing on those areas of an image where restoration needs improvement. Also, we proposed loss functions: Iterative Focal Loss (iFL) and Class-Balanced Iterative Focal Loss (CB-iFL), which are specifically designed to improve the performance of the feedback network. These losses focus more on those samples that are continuously miss-classified over successive iterations. Our approach gives a gain of 17.41 mIoU over the standard segmentation model, including the additional gain of 1.9 mIoU with CB-iFL on the Cityscapes dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Figure 1.1: Real world images in different weather conditions. *[Image Credits: Google]*

Weather conditions dramatically change when we move from one geographic location to another. These changes occur due to the variation in atmospheric parameters such as temperature, humidity, and precipitation. Standard datasets like ImageNet used for training vision models usually have images taken in a clean environment that do not capture these real-world variabilities. As a result, computer vision models trained on such datasets give poor generalization results. For instance, consider an object detection model trained on a dataset that contains images captured in a clean environment. Now, we deploy the trained model in a foggy climate. Then, the test image given to the model will have blurriness injected by the fog, which might result in object detection failure.

Solving problems caused by variations in weather conditions could improve real-world computer vision systems. It has immense potential applications in long-range video surveillance, defense systems, and drone imaging. These systems can capture images of an object at a distance measured in the order of several kilometers. The ability to view through bad weather will enable us to track and detect such

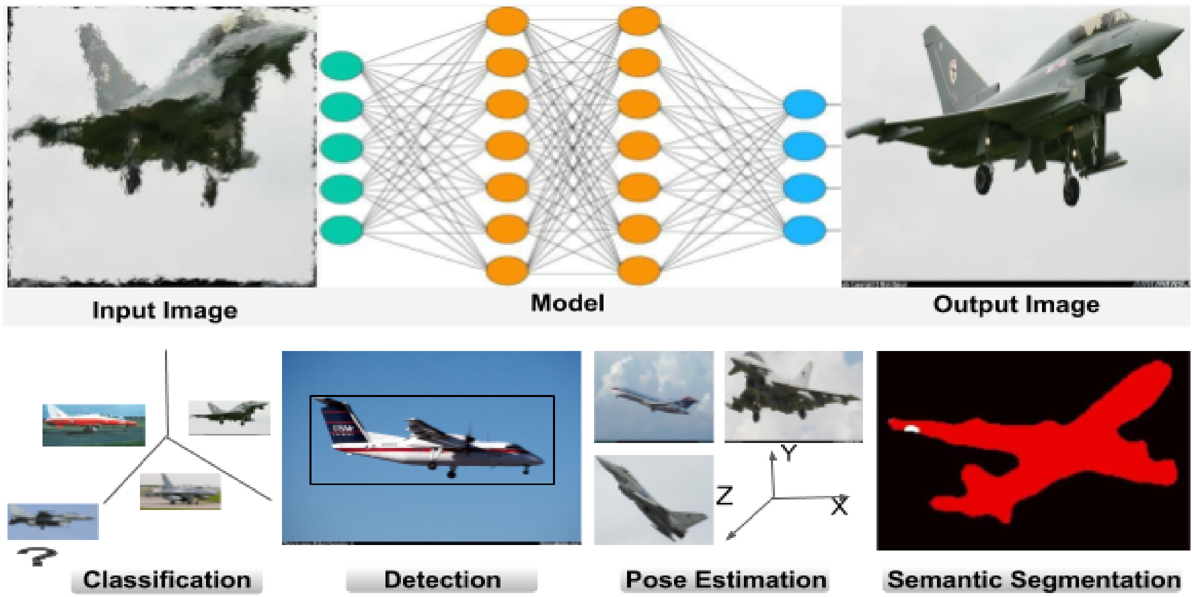

Figure 1.2: Symbolic illustration of restoration model and its application in various spectrum of problems. *[Image Credits: Google]*

objects accurately. Trains running late due to dense fog or rain is a huge problem in countries like India. Deploying vision systems that can provide additional supervision while driving could be a possible approach to address this problem. Lastly, improving Advanced Driver Assistance Systems(ADAS) during turbulent weather shows vast promise. ADAS uses Light Detection and Ranging (LIDAR) and cameras to create maps of a vehicle's surroundings. However, during bad weather, this information can be interrupted and thereby adversely impact the maps created and the safety level of a self-driving car.

Among all-weather conditions, in this thesis, we choose to address the problems of image distortion that occurs in hot weather conditions. We begin our work by generating distortions caused by a hot climate and injecting them into an image. Then, we moved towards the problem of restoring image deteriorated by high temperature. Further, we used the restored images to show improvement in vision tasks such as recognition, segmentation, etc.

1.1 Motivation

Slight perturbations in atmospheric conditions can trigger significant changes in the image's geometrical and perceptual information. Capturing images in areas with extremely high temperatures such as desert introduces such geometric distortions, which is also termed as atmospheric temperature. We will interchangeably use hot temperature and atmospheric turbulence further in the thesis.

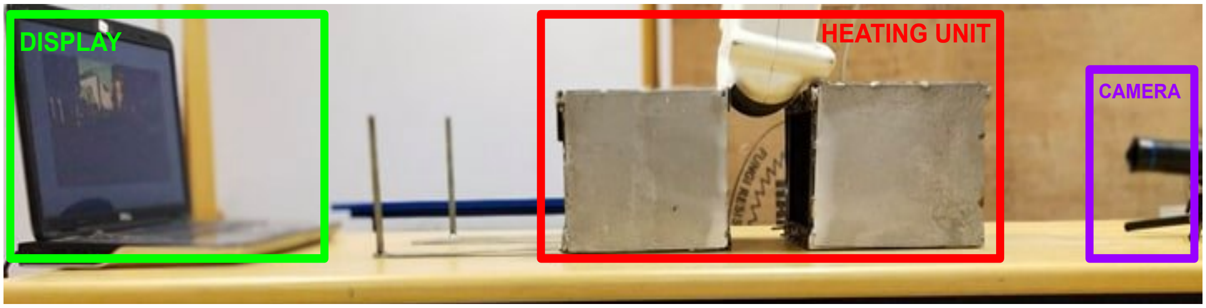

Figure 1.3: Experimental setup for generating turbulent images.

Geometrical distortions caused by high temperature disturbs the overall orientation of an object present in the image. These distortions lead to a change in the image features. As a result, when a computer vision model trained on standard dataset is deployed in such weather conditions gives poor results. Hence, we need to have a restoration model to remove the geometrical distortions from an image. Adding a restoration model can help in improving a spectrum of tasks, as shown in Figure 1.2.

A data-centric approach to address the problem is to train a restoration network on those images that are collected in hot weather conditions. But, collecting and annotating data in such harsh conditions is a costly and time-consuming task. Hence, we use simulation-based methods to generate images intended for hot weather conditions and bypass the cumbersome task of data collection and annotation.

1.2 Problem Context

It is essential to consider that there are many problems related to restoring images in adverse weather conditions such as rain and fog, but we realized there hadn't been much prior work done on restoring images in a hot weather. Hence, we move our focus on a spectrum of problems related to atmospheric turbulence.

Restoring images in atmospheric turbulence is an ill-posed inverse problem. Hence, this problem poses a challenge for us to find the exact solution. Thus, we use a deep network to approximate results that are closest to the exact answer. On the other hand, we also tried to generate atmospheric turbulent images using an experimental setup shown in Figure 1.3. There were other factors involved, like - the illumination of the display, the lighting condition of the environment, and regulating the temperature that needed focus, which made us drop the idea of using an experimental setup. Then, we switched to the physics-based simulation method to generate a turbulent image.

We validated the efficacy of our restoration method using traditional methods like- Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), etc. But, these evaluation methods do not ensure the performance of our model in real-world applications. Hence, we restore the turbulent images and then pass those restored images into a task-specific network like- pose estimation, classification, or semantic segmentation to get performance evaluation of our restoration network. Further, we focus on improving semantic segmentation in a turbulent environment as semantic segmentation has its extensive use in autonomous navigation systems.

1.3 Background

1.3.1 Restoring images in atmospheric turbulence

The initial methods of removing atmospheric turbulence from the images can be broadly categorized as a) Adaptive Optics b) Lucky Imaging. Adaptive optics requires expensive and huge instruments for removing atmospheric turbulence. These methods were mainly used for astronomical applications [56, 4]. On the other hand, Lucky Imaging algorithms take a set of short exposed distorted images from which lucky regions are selected and then fused to get the final image. Some of the preliminary work in lucky imaging [13] uses probabilistic analysis to restore the degraded images. Various methods [1, 88] have been proposed for enhancing images and videos which have been degraded due to atmospheric turbulence using lucky imaging. In image enhancement, these methods use a multi-frame image reconstruction approach to correct geometric distortion and reduce the blur introduced by atmospheric turbulence. For video enhancement, it is performed by fusing a stream of randomly-distorted images. Statistical methods and Fourier analysis techniques [26, 82] were also attempted to remove atmospheric turbulence.

Recent methods for recovering images from atmospheric turbulence rely more on machine learning. Zhu *et al.*[89] proposed a method that first registers each frame to suppress geometric distortion through B-spline based on nonrigid registration. Next, a temporal regression process is carried out to produce an image from the registered frames, which can be viewed as being convolved with space invariant near-diffraction-limited blur. Finally, a blind deconvolution algorithm is implemented to deblur the fused image and generate the final output. But, the method suffered from several limitations; for example, the method uses the temporal mean to calculate the reference image, which results in poor registration. Xie *et al.*[80] proposed an improved method that first constructs a high-quality reference image from the set of observed frames using low-rank matrix decomposition. Further, to improve the registered sequence, the reference image is iteratively optimized using a variational framework containing a new spatiotemporal regularization.

1.3.2 Dataset Simulation

Rendering turbulent images using computer graphics utilizes high computational power and time. To overcome this computational burden, an efficient way of rendering turbulent images was derived by Schwartzman *et al.*[65]. This method bypasses 3D numerical calculations and relies on a closed-form model for creating a computationally inexpensive and faster way of simulating turbulent fields. The virtual camera parameters used in imaging process are: focal distance $= 300mm$ with lens diameter $\approx$

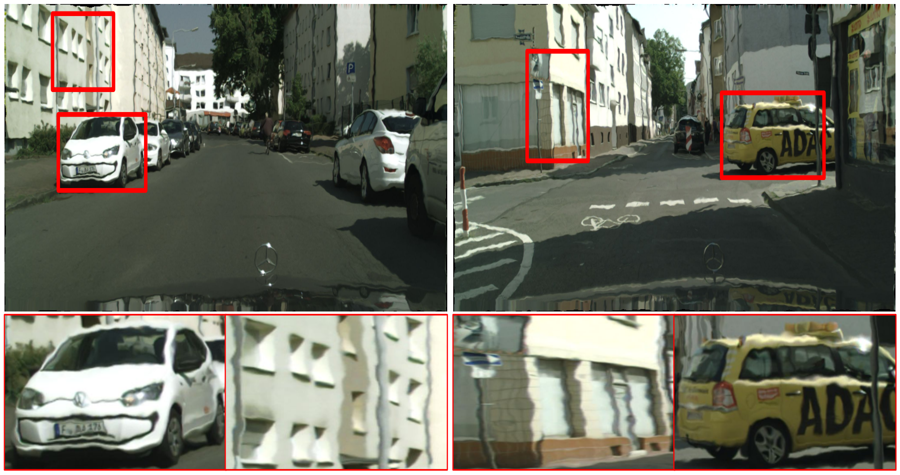

Figure 1.4: Examples of simulated turbulent images [65]. Image patches in red boxes show the geometrical distortions caused by atmospheric turbulence.

$5.357cm$ and pixel size of $4e-3mm$. The virtual camera is placed at an elevation of $4m$ with object distance of $2km$. Structure constant C_n^2 is an important variable for measuring atmospheric turbulence. Higher the value of C_n^2 stronger the turbulence. $C_n^2 = 0$ indicates a medium free of turbulence. The value for structure constant C_n^2 which expresses the turbulent strength is $3e-13m^{-2/3}$. The light traveling from an object point is assumed to be having a spherical wavefront with a wavelength of $550nm$. Figure 1.4 shows the simulated atmospheric turbulent images.

1.4 Contribution

The major contributions of our thesis are in three folds:

A. Generation:

1. We attempt to model atmospheric turbulence via a generative adversarial network. Our method outputs an atmospheric turbulent image by utilizing less computational resources than traditional methods.
2. We propose a novel loss function to learn the atmospheric turbulence at the finer level efficiently.

B. Restoration:

1. We propose a deep adversarial network to recover the images distorted due to atmospheric turbulence and show the applicability of restored images in several tasks.
2. Unlike previous methods, our approach neither uses any prior knowledge about atmospheric turbulence conditions at inference time nor requires the fusion of multiple images to get a single restored image.

3. We find that our restored images outperform turbulent images in downstream tasks by a significant margin qualitatively and quantitatively, showing the applicability of the restoration model in real-world problems.

C. Application:

1. We propose a framework that improves the performance of semantic segmentation models without explicitly creating an annotated dataset for turbulent weather.
2. Our framework consists of a restoration network that removes the geometrical distortions caused by hot weather and an adaptive segmentation network that is trained on an additional loss to adapt to the statistics of the ground-truth segmentation map.
3. We train our framework on the Cityscapes dataset, which resulted in a total IoU gain of 12.707 over standard segmentation models.
4. We improve the performance of our joint restoration and segmentation network via a feedback mechanism. The semantic feedback learning approach improves the task of semantic segmentation, giving a feedback response into the restoration network.
5. Also, we proposed an additional two loss functions: Iterative Focal Loss (iFL) and Class-Balanced Iterative Focal Loss (CB-iFL), which are specifically designed to improve the performance of the feedback network. These losses focus more on those samples that are continuously miss-classified over successive iterations.
6. Our approach gives 17.41 mIoU over the standard segmentation model, including the additional gain of 1.9 mIoU with CB-iFL on the Cityscapes dataset.

1.5 Thesis Organisation

We now give a brief description of the structure of the thesis. In Chapter 2, we propose a generative model that models the atmospheric turbulence, which utilizes less computational resources than traditional methods. In Chapter 3, we address the ill-posed problem of restoring images degraded due to atmospheric turbulence by proposing a generative adversarial network and applying the restored images to improve computer vision tasks. Further, in Chapter 4, we propose a framework that enhances the semantic segmentation model in hot weather conditions. We also introduced a Channel Attentive Multi-Scale Residual Block (CA-MSRB) block that was particularly designed for removing geometrical distortions caused by atmospheric turbulence. Chapter 5 further improves our restoration network by proposing a feedback framework where the restoration network learns directly from the segmentation network's errors. In Conclusion, we summarise our finding along with the possible future directions.

Chapter 2

Learning To Generate Atmospheric Turbulent Images

In this chapter, we train a generative adversarial network, which outputs an atmospheric turbulent image by utilizing less computational resources than traditional methods. We propose a novel loss function to learn the atmospheric turbulence at the finer level efficiently. Experiments show that by using the proposed loss function, our network outperforms the existing state-of-the-art image to image translation network in turbulent image generation. We also perform extensive ablation studies on the loss function to demonstrate the improvement in the perceptual quality of turbulent images.

2.1 Introduction

The performance of computer vision algorithms drastically decreases when deployed in changing weather conditions [49, 54]. Especially in applications like autonomous navigation and aerial imaging, the atmospheric condition adversely affects the performance of the underlying vision algorithm. Computer vision models adapting to changing weather could be solved by collecting data for all the weather conditions and training vision algorithms on them. But, collecting data for each weather condition would require a considerable cost and time. Hence, we propose a deep learning-based approach that models hot weather conditions among all weather conditions. We also show that our method's computational time to generate hot weather images was less than the traditional methods. The phenomena of geometrical distortion caused by hot weather is termed as atmospheric turbulence.

The primary cause of atmospheric turbulence is the heterogeneous nature of the atmosphere between the camera and the object. The heterogeneity in the medium is caused by the time-space varying in temperature, air pressure, wind speed, and humidity, resulting in the introduction of geometrical distortion and a decrease in perceptual information of an imaged object. Several methods have been proposed to model the nature of atmospheric turbulence and render images with the help of these models. Turbulence modeling methods can be divided into two major approaches: Ray tracing in computer graphics [19, 66] and image distortion simulation [34, 65]. Initial approaches in computer graphics either used curved ray tracing [66] or solved physically-based differential equations [19, 60] to estimate the parameters of atmospheric turbulence based on real turbulent images, which help in describing the trajectory of

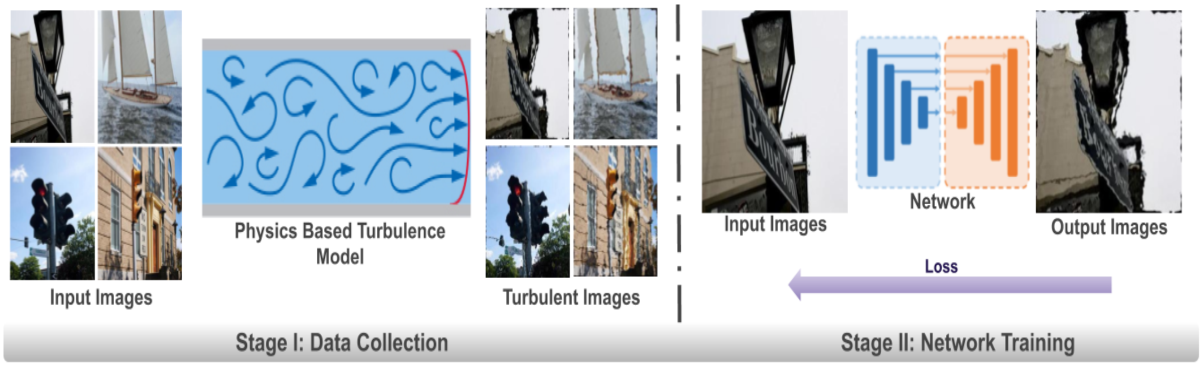

Figure 2.1: Illustration of our overall pipeline to generate a turbulent image. *Stage I*: In this stage, we create a turbulent dataset using [65] method to train our network. *Stage II*: Training of our generative adversarial network using the created turbulent dataset. After training, turbulent images can be generated by simply passing the corresponding non-turbulent image through the network.

light. Another approach to model atmospheric turbulence is to model the turbulence and distort images using those models statistically. Earlier methods used light propagation through multiple phase screens [34] for this purpose. Later methods model turbulence as a simple Gaussian function [89] or derived a physics-based method [65] by which turbulent fields were generated efficiently. However, these methods are computationally expensive, which requires a large amount of computational time to generate a large dataset. Recently, Deep learning [36] has become a powerful framework to generate complex non-linear data such as images, speech, and videos. In particular, Generative Adversarial Networks(GANs) are widely used for the generation of images as it directly learns the empirical data distribution from the data samples. Application of GANs include images-to-image translation tasks such as super-resolution [37], style transfer [2], and synthetic data generation [72].

Leveraging the advantages of GANs, we train a deep adversarial network that takes an input image and gives out the corresponding turbulent image. While training, our GAN tries to learn the natural image distribution of the turbulent image. GANs provides the flexibility of sampling infinite samples from the learned distribution by simply feed-forwarding input samples into the network. Hence after training, using our trained GAN, we can generate large datasets of turbulent images at inference, which would take few milliseconds to generate a turbulent image on an average GPU enabled device. Hence, we solve the problem of high computational issues encountered with traditional methods by using GAN. Additionally, we introduce a novel loss function specifically intended to improve the quality of output turbulent images. To train the adversarial network, we require a substantial number of turbulent and non-turbulent image pairs. Since there was no public dataset available for training, we build a dataset by generating turbulent fields using [65], method. Then we randomly placed the turbulent field onto the non-turbulent images to generate the corresponding turbulent image. Figure 2.1 illustrates the overall pipeline of our approach for generating turbulent images. We quantitatively and qualitatively show that

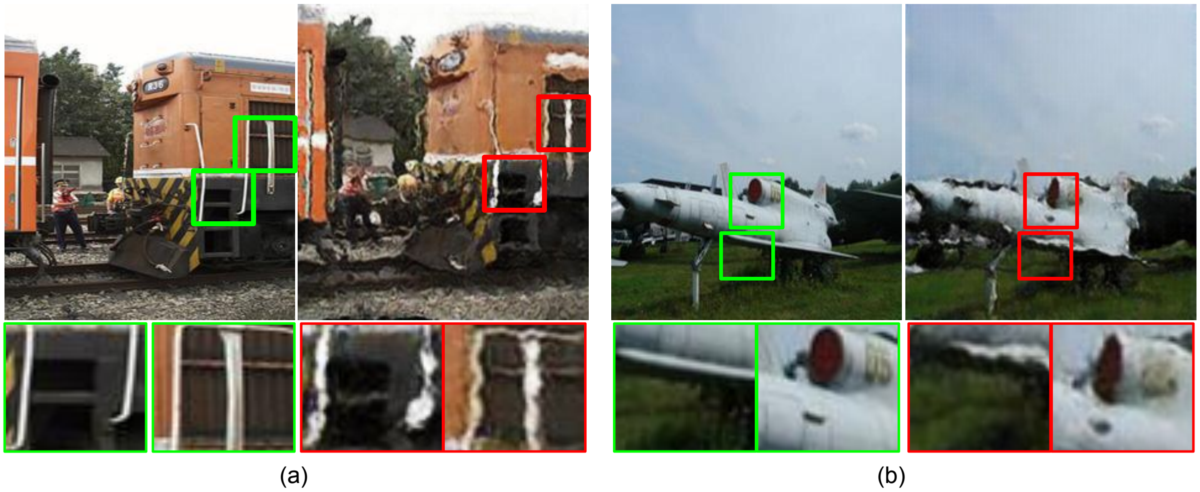

Figure 2.2: (a) and (b) are the samples turbulent images generated by our network. Red boxes show the magnified patches of the input image and green boxes shows the turbulent patches.

the turbulent images generated by our network are close to natural turbulent images. Figure 2.2 shows the sample turbulent image generated by our model. These generated turbulent images can be further used as turbulent datasets in building restoration networks. It also increases the accuracy of various computer vision algorithms such as classification and semantic segmentation in a turbulent environment.

2.2 Proposed Model

Generative models are widely used for generating samples from a data distribution. GAN is one of the models among all the generative models. Consider, two differential function generator G_α and D_β where α and β are learnable parameters. The generator creates fake samples that are intended to be closer to the training data distribution. The discriminator is a classifier where it examines whether the given samples belong to training data distribution or not. The generator tries to generate fake samples by adjusting its α to fool the discriminator. On the other hand, the discriminator learns to discriminate between real and fake samples. In this way, the generator and discriminator learn their parameters α and β. This framework can be viewed as a mini-max game where the generator tries to minimize the probability of its samples being fake, whereas discriminator tries to maximize it. Formally, we assume G_α and D_β to be a deep neural network.

In our problem setting, we train a generative adversarial network that aims to estimate a turbulent image $\hat{I}^{TR}$ from a non-turbulent image I. Here, I^{TR} is the turbulent image corresponding to a non-turbulent image I. Our goal is to train a generator G_α with learnable parameter α, which can generate turbulent images by minimizing loss function and can fool the discriminator D_β. Here, D_β is a classi-

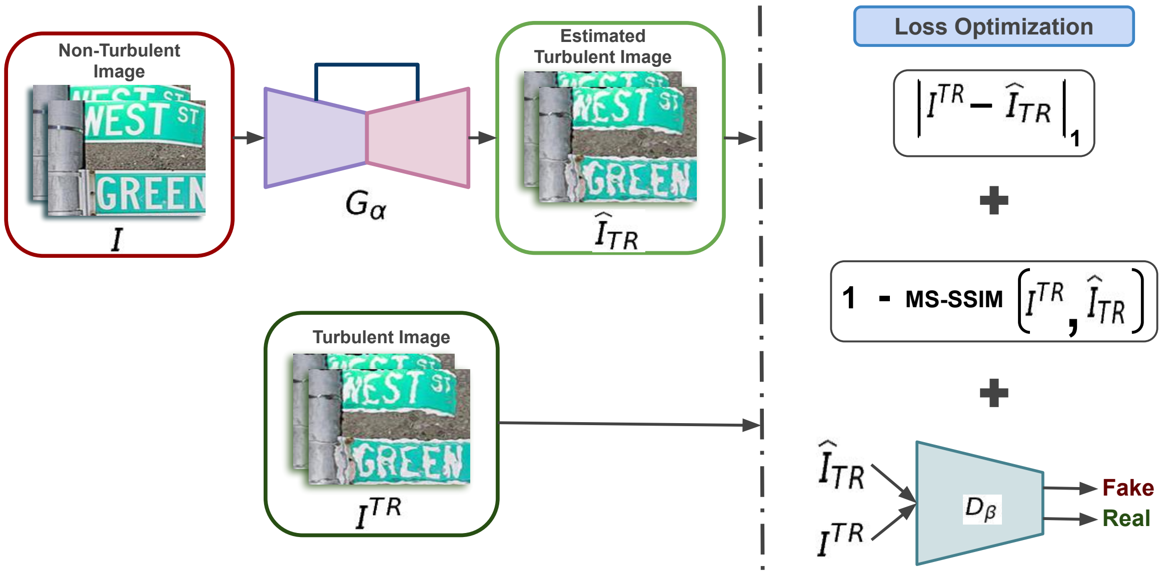

Figure 2.3: Overview of our proposed generative adversarial network. While training the network, a non-turbulent image I is passed through generator G_α with learnable parameter α which gives the estimated turbulent image $\hat{I}^{TR}$. I^{TR} and $\hat{I}^{TR}$ are passed through loss optimization block to minimize the L1 loss, MS-SSIM loss and adversarial loss between them.

fication network which classifies between estimated non-turbulent turbulent and non-turbulent images with learnable parameter β. Figure 2.3 shows the overall outline of our proposed network architecture. In the subsequent subsections, we describe the architecture of our generative adversarial network along with the new loss function, which is particularly formulated to improve the quality of the turbulent image generation process.

2.2.1 Network Architecture

The generative adversarial network is divided into two networks: generator and discriminator. Our generator mainly follows the architecture of U-Net [64]. The architecture of our generator is divided into two paths: contracting path and expanding path. The contracting path downsamples the input image into the feature space. It consists of four contracting blocks, where, each block contains two 4×4 convolutional layer of stride 2 and padding 1. Each convolutional layer is followed by a Leaky Rectified Linear activation unit(LeakyReLU) layer and a batch normalization layer. After, each contracting block, we double the depth of the feature maps. For each contracting block, we have a corresponding expanding block in the expanding path. An expanding block contains two 4×4 transpose convolutional layer of stride 2 and padding 1. Each transpose convolutional layer is followed by a ReLU layer and a batch normalization layer. Each expanding block concatenates the cropped feature from its contracting block. The depth of the feature map decreases to half after each contracting block. The activation function of the outer most convolution layer in the expanding path is tanh as our input image pixel value ranges from -1 to 1. The contracting path and expanding path are joined by a bottleneck. The bottleneck consists of a convolution layer with LeakyReLU as activation and a transpose convolution layer with ReLu as activation. The discriminator follows the standard architecture of [55] which is used in most of the GANs architecture.

2.2.2 Loss Functions

Least absolute deviations (L1 loss) is widely used as a loss function for many deep networks. However, the minimization of L1 loss for generation tasks results in blurry output images lacking in high-frequency details. Hence, we add adversarial loss to our generator encourages it to produce images that lie in natural image manifold with sharp textures. We use LS-GAN [46] over vanilla GAN for better stability and faster convergence. We also use Multi-Scale Structural Similarity (MS-SSIM) loss [85] to improve the generation of turbulence at the finer level. The final loss function of our generator network is a weighted linear combination of L1 loss, MS-SSIM loss and adversarial loss which is:

$$L_{generator} = \lambda_1 L_1 + \lambda_2 L_{Adverserial} + \lambda_3 L_{MS-SSIM} \tag{2.1}$$

$$L_{generator}=\lambda_1|I^{TR}-G_\alpha(I)|_1+\lambda_2[D_\beta(G_\alpha(I))-1]^2+\lambda_3(1-MS\text{-}SSIM(I^{TR},G_\alpha(I))) \tag{2.2}$$

where, λ_1, λ_2 and λ_3 are the hyper-parameter of the generator loss function. Similarly, the loss of our discriminator network would be:

$$L_{discriminator} = D_\beta(G_\alpha(I))^2 + (D_\beta(I^{TR}) - 1)^2 \tag{2.3}$$

2.2.3 Training Description

We train our network end-to-end by following the training methodology from Mao *et al.* [47] and Ledig *et al.* [37]. The dataset consists of 100,000 of turbulent and non-turbulent image pairs for training and the validation is performed on ImageNet validation dataset. Figure 2.4 shows sample turbulent images. To optimize our network, we use Adam [31] optimizer with β_1 = 0.5 and β_2 = 0.999 for computing running average of gradient and its squares. The value of λ_1, λ_2, and λ_3 were found to be 100, 1, and 5, respectively. We first train our network for 10,000 iterations with a learning rate of $1e-4$ and then decrease the learning rate to $5e-5$ for another 15,000 iterations with a batch size of 16. For each iteration, the discriminator and the generator are updated only once.

2.3 Experimentation

2.3.1 Evaluation Metrics

To measure the structural and perceptual similarity between the estimated turbulent image and ground-truth turbulent image, we use Peak-Signal-To-Noise-Ratio (PSNR), Structural Similarity [76] (SSIM)

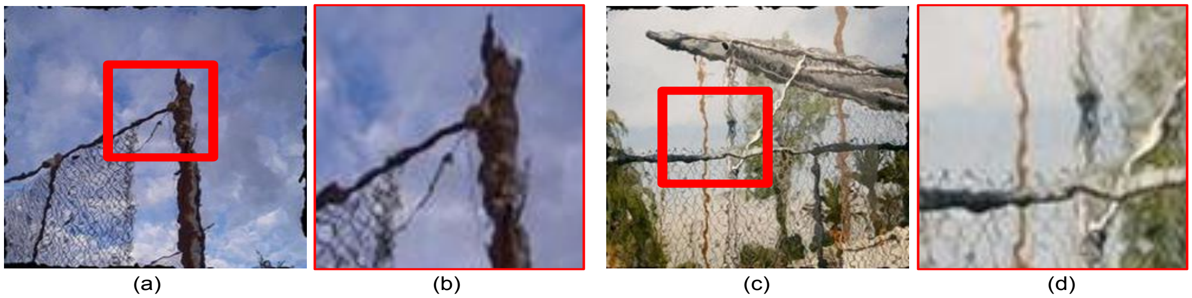

Figure 2.4: (a) and (c) show the sample turbulent images of the training dataset. (b) and (d) shows the enlarged patches in red boxes.

Methods	**PSNR**	**SSIM**	**MSE**
Encoder-Decoder (L1 loss) [3]	12.1040	0.1071	461.8092
Encoder-Decoder (MSE loss) [3]	14.9994	0.2423	333.6431
CycleGAN [87]	14.0471	0.2803	402.6070
Pix2Pix [28]	17.8422	0.4412	254.0445
Encoder-Decoder with skip connections (L1-loss) [47]	15.4503	0.2905	320.6347
Encoder-Decoder with skip connections (MSE loss) [47]	15.5124	0.2925	318.4140
Ours	**18.2321**	**0.5211**	**234.9368**

Table 2.1: Quantitative results of various turbulence models. Our method outperforms baseline methods by using the proposed loss function.

and Mean Squared Error (MSE). These evaluation metrics are applied to various generated turbulent images and ground-truth turbulent images to give a qualitative comparison.

2.3.2 Results

We compare our results of final turbulence generation model with the image-to-image translation networks: Pix2Pix [28], CycleGAN [87], Encoder-Decoder [3], and Encoder-Decoder with skip connections [47]. Our final turbulence generation model shows significant quantitative improvement over the other aforementioned methods as shown in Table 2.1. In Table 2.1, we observe that methods with MSE loss as a loss function give better results on evaluation metrics over the L1 loss. The quantitative performance of Pix2Pix is better than CycleGAN as Pix2Pix learns from paired images whereas CycleGAN learns from unpaired images, which makes the generation of turbulent images difficult. Qualitative results of turbulent image generation on ImageNet images by various methods are shown in Figure 2.5. In Figure 2.5, we infer that CycleGAN and Pix2Pix struggle to generate larger geometrical distortion compared to our method. Moreover, Pix2Pix suffers from color artifacts which could be seen in Figure 2.5 (column e)(iii)(iv). Encoder-Decoder and Encoder-Decoder with skip connections failed to generate turbulent images with sharp details. Although encoder-decoder suffers from the checkerboard effect

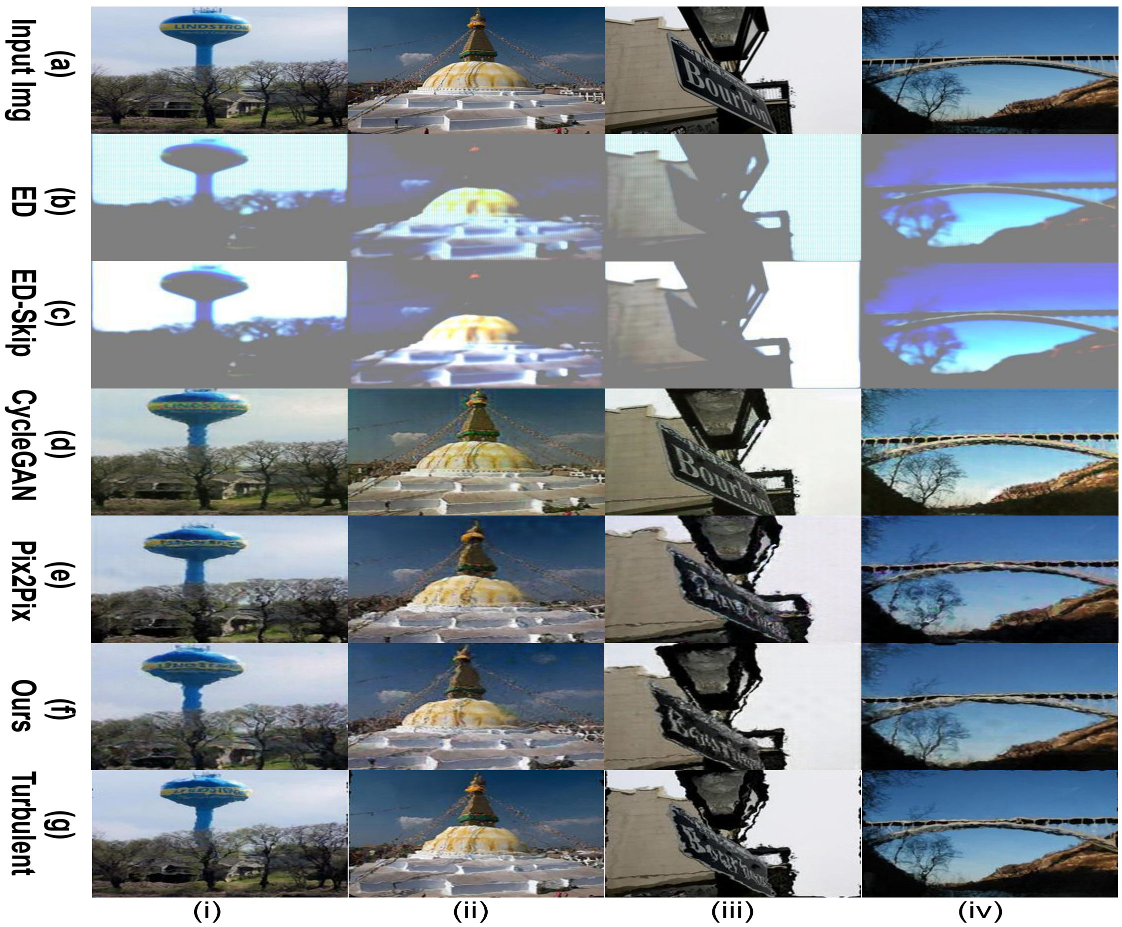

Figure 2.5: Qualitative results of various turbulence models. Row a: Input Img is the non-turbulent input image to the network. Row b: ED is the Encoder-Decoder network output. Row c: ED-Skip is the Encoder-Decoder with a skip-connections network output. Row d: CycleGAN network output. Row e: Pix2Pix network output. Row f: Our network output. Row g: Turbulent is the Ground-truth turbulent image. *(Best view when zoomed)*

which is eliminated by skip connections. Table 2.2 shows the comparison between the computational time taken by Schwartzman *et al.* [65] and our method to generate atmospheric turbulent images. We can observe from the table that our approach requires a small fraction of computational time to generate turbulent images.

2.3.3 Ablation Study

We perform an ablation study on our loss function to show its advantages qualitatively and quantitatively. The ablative study is performed by training our generator architecture individually on L1 loss, MSE loss, L1 loss + Adversarial Loss, and L1 loss + Adversarial loss + MS-SSIM loss. We use L1 loss instead of MSE loss as it leads to sharper images. All the networks were trained for 10,000 iterations

Number of Images	**1**	**100**	**500**	**1000**
Schwartzman *et al.* [65]	184.367	197.357	280.472	389.559
Ours	0.018	0.979	5.243	10.646

Table 2.2: Computational resource time comparison between the turbulent image generation methods on different number of turbulence images to be generated. (All time values reported in seconds.)

Loss Function	**PSNR**	**SSIM**	**MSE**
L1 loss	16.7413	0.3523	293.5724
MSE loss	17.0215	0.3915	273.3178
L1 loss + Adversarial loss	17.5243	0.4215	259.2911
L1 loss + MS-SSIM loss + Adversarial Loss	**17.6515**	**0.4402**	**256.5432**

Table 2.3: Ablation study results of our loss function.

with the learning rate of $1e-4$ and batch size of 16. From Table 2.3, we observe that adding MS-SSIM loss into the total loss leads to higher PSNR and SSIM which implies that it improves the structural as well as perceptual information of the generated image. Figure 2.6 shows the qualitative results of the ablation where we can observe in Figure 2.6(row 5) the output generated by our proposed method looks more realistic. Although in Figure 2.6(row 4) the output looks more promising when zoomed, it looks more like a uniform artifact.

2.4 Summary

We proposed a turbulent image generation model by training a deep adversarial network. Unlike traditional turbulent image generation methods that rely on a statistical model, our approach uses data to learn the parameters of the generative adversarial network, which transforms a non-turbulent image into a turbulent image. We proposed a novel loss function that encourages the learning of finer turbulence fields. To support our claim, we performed extensive ablation studies on the loss function.

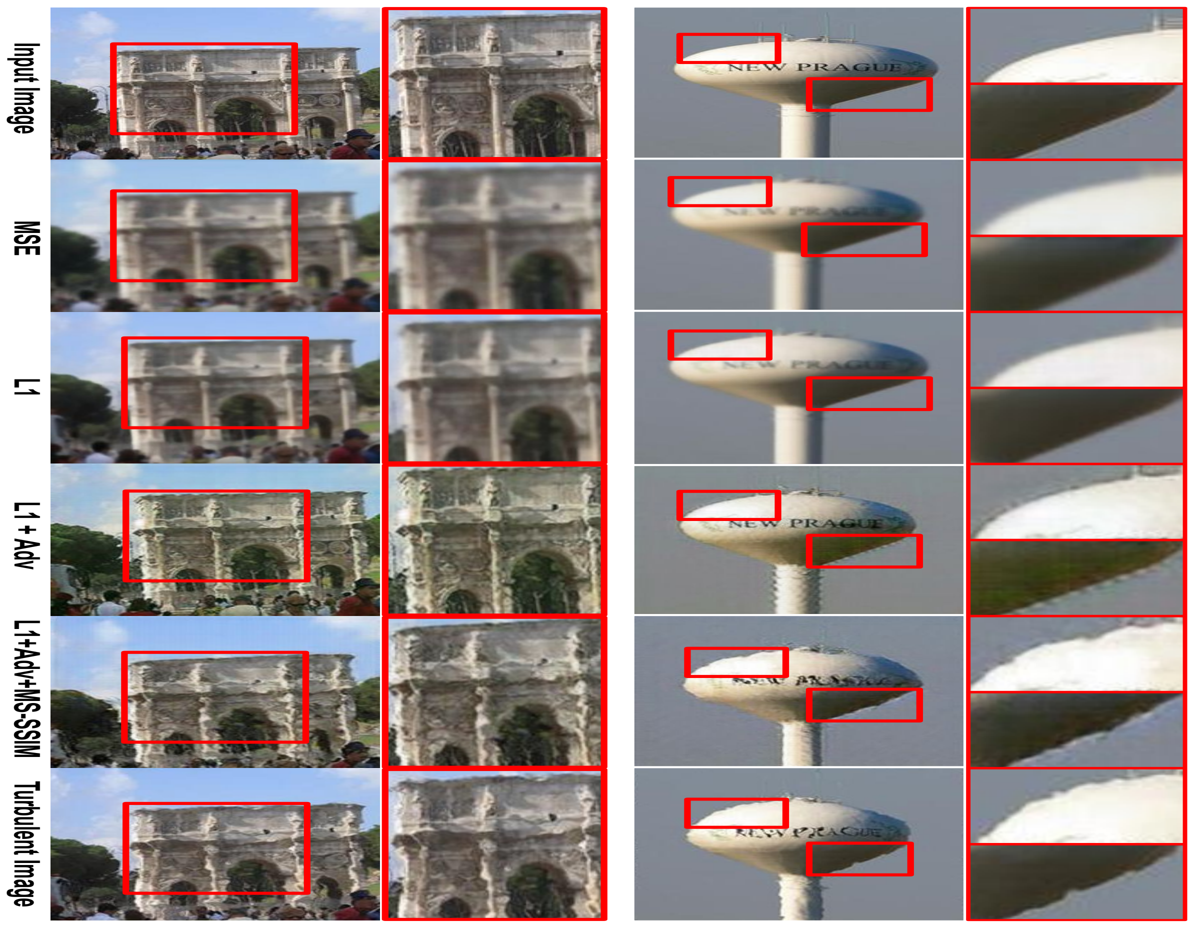

Figure 2.6: Qualitative results of the ablative study on the various loss function. Row 1: Input image to the network. Row 2: MSE loss output. Row 3: L1 loss output. Row 4: L1 + Adv is the output of the combined output of L1 loss and adversarial loss. Row 5: L1 + Adv + MS-SSIM is the output of the combined output of L1 loss, MS-SSIM, and adversarial loss. Row 6: Ground-truth turbulent image. Red boxes show the zoomed image patch. *(Best viewed when zoomed)*

Chapter 3

Removing Turbulence via Deep Adversarial Learning

In this chapter, we address the problem of restoring images degraded due to atmospheric turbulence. We propose a deep adversarial network to recover the images distorted due to atmospheric turbulence and show the applicability of restored images in several tasks. To train our models, we synthesized turbulent images by following a series of efficient 2D operations. After that, using our trained models, we run inference on real and synthesized turbulent images. Our final restoration models De-Turbulence Generative Adversarial Networks (DT-GAN+) and De-Turbulence Dense Generative Adversarial Networks (DTD-GAN+) qualitatively and quantitatively outperforms the general state-of-the-art image-to-image translation models. We also perform extensive experiments on restored images by utilizing them for downstream tasks such as classification, pose estimation, semantic keypoint estimation, and depth estimation.

3.1 Introduction

Imaging through atmospheric turbulence has been a well-researched subject over the past few decades. Preliminary works in this topic were focused on astronomical applications [11]. Image restoration from atmospheric turbulence has tremendous potential applications in long-range video surveillance, defense systems, and drone imaging. These systems capture images of an object at a distance measured in the order of several kilometers. Slight perturbations in atmospheric conditions can trigger significant changes in the image's geometrical and perceptual information. Optical setups used to remove turbulence from images require huge investments and a high maintenance cost. Leveraging the recent developments in deep learning, we attempt to solve the problem of restoring images under air turbulence, as shown in Figure 3.1.

The phenomenon of turbulence [62] occurs when there is a slight perturbation in air pressure, temperature, and gaseous levels in the atmosphere. Near the earth's surface, temperature change acts as a major component in atmospheric turbulence. A slight change in temperature can cause random fluctuations in the refractive index along the optical path resulting in the perceptual degradation of the imaged object. Also, deep atmospheric turbulence introduces geometrical distortion [35] similar to the shearing

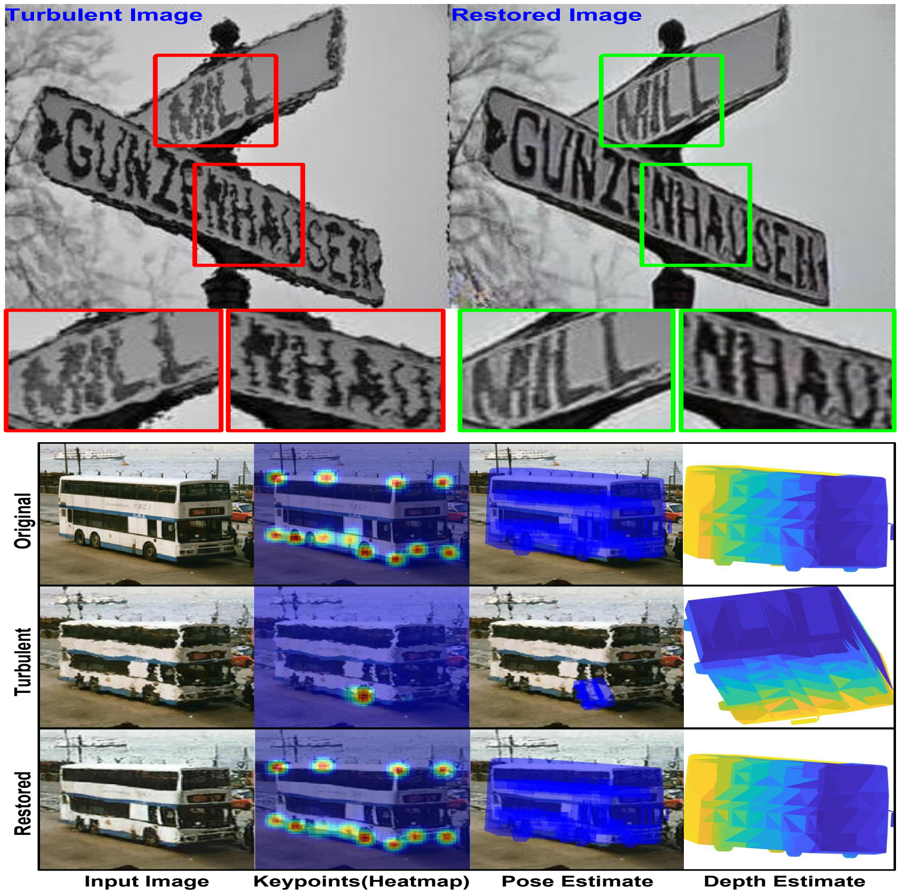

Figure 3.1: Top: An example of image restoration by the network along with magnified image patches of the turbulent image in red color and restored image in green color. Bottom: Improvement in qualitative results of restored image over the turbulent image in several computer vision tasks. **(Best viewed when zoomed)**

effect, which is unevenly distributed into different parts of the image. These effects are commonly observed in the sandy desert and rocket exhaust. Mathematically, atmospheric turbulence in an image can be modeled[21] as:

$$T = HI + \epsilon \tag{3.1}$$

T is the turbulent image, I is the original image, H is a transformation matrix, and ϵ includes other noises like camera sensor noise. From equation 3.1, we can infer that the problem is an ill-posed inverse problem because the value of H and ϵ needs to be solved using a single equation. Therefore, instead of finding the exact solution, we approximate results closest to the exact solution.

Neural networks have proven to have sufficient capabilities to approximate [23, 22, 52] complex non-linear functions. The recent success of deep neural networks have drastically improved the performance for image classification [33, 69, 20], object detection [59, 58] and object segmentation [43].These

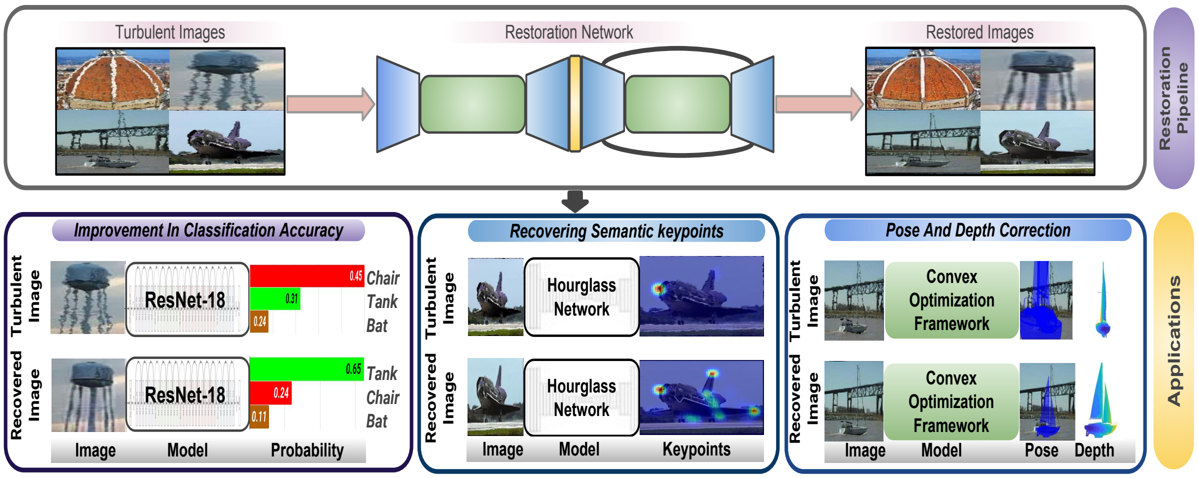

Figure 3.2: Conceptual illustration of the overall methodology. Top: Restoration pipeline for the turbulent images. Bottom left to bottom right: Application of restored images indicating improvements in tasks such as classification, keypoints detection, pose, and depth estimation.

networks are employed for generating natural images using Deep generative networks, such as Generative Adversarial Network [16, 55], VAE [32, 17], PixelRNN [51] and PixelCNN [74] which involves mapping from a highly non-linear manifold to another. Among the generative networks, Generative Adversarial Networks are more successful than other models due to their photo realistic output quality and applicability in computer vision tasks such as image super-resolution [37], image in-painting [81] and image-to-image translation [28]. Apart from generating realistic images, deep neural network architectures are also capable of manipulating geometric structures. These architectures have been used in gaze manipulation [15], image matching [61], image registration [9] and spatial transformation [29] of an image. Hence, utilizing the recent advancements in deep learning, it becomes possible to learn a mapping from a turbulent image to an original image.

A deep network requires a large amount of turbulent and original image pairs to learn the mapping. However, there is no such large scale dataset available to train the network. Moreover, rendering such images using computer graphics requires a high computational cost. To overcome this problem, we follow a systematic way [65] to inject atmospheric turbulence into images by a series of efficient 2D operations.

Our proposed model is an optimized architecture of Li *et al.* [39], achieved by analyzing the performance of the network architecture with various residual blocks along with a modified training strategy, which experimentally showed better results. We also add channel attention and sub-pixel mechanism to utilize the information between the channels and learn smaller warps to remove atmospheric turbulence at the finer level. The restoration model performance is investigated over images having diverse levels of simulated atmospheric turbulence. Our final model DTD-GAN+ quantitatively and qualitatively outperforms the state-of-the-art general image-to-image translation methods. We also used our final model to restore images with natural atmospheric turbulence, which were curated from YouTube. These restored images had impressive image quality, which demonstrates the generalization ability of our model.

In the subsequent sections, we also explain the use of restored images in vision applications such as semantic keypoints detection, depth estimation, classification, and pose estimation, which outperform results on turbulent images. Additionally, we demonstrate the performance of restored images among datasets with various number of classes. Figure 3.2 illustrates the overall methodology. Non-turbulent and original images terms are used interchangeably to represent the ground-truth images.

3.2 Method

In this section, we describe the proposed model for restoring images degraded due to atmospheric turbulence. Our proposed deep adversarial network removes the geometric distortions introduced by atmospheric turbulence and restores the perceptual information lost in an image. In the following subsections, we describe the generator's architecture of the adversarial network along with the objective function and improved training methodology.

3.2.1 Channel Attention Blocks

In this subsection, we describe the various architectures of residual structures used in the generator of the deep adversarial network.

- **Residual Blocks**: Residual networks perform extremely well on various computer vision tasks as they efficiently carry forward information deep into the network with the help of residual connections. However, these networks do not focus on the channel relationships within the residual block. Using the channel attention mechanism of [24], we introduce the channel attention in our residual architecture, Figure 3.4(d), which adaptively recalibrates the responses of channel relationships by modeling the inter-channel dependencies between the channels. Also, instead of using Batch Normalization(BN) [27] or Instance Normalization(IN) [73], we used Group Normalization(GN) [78] as it has lower training error on a smaller batch size. In Figure 3.4, we compare the architectural building blocks of each network from He *et al.* [20], Li *et al.* [39], Baseline1, and DT-GAN.

- **Dense Blocks**: Recently, densenet [25] has become the state-of-the-art network by giving the best performance on computer vision tasks like - image classification, etc. We use residual dense block [84] to construct our Baseline2 by adding the ReLU layer shown in Figure 3.3(b). Adding the ReLU layer increases the ability of dense residual blocks to learn non-linearity efficiently. We also add channel attention mechanism to Baseline2 similar to residual blocks, as shown in Figure

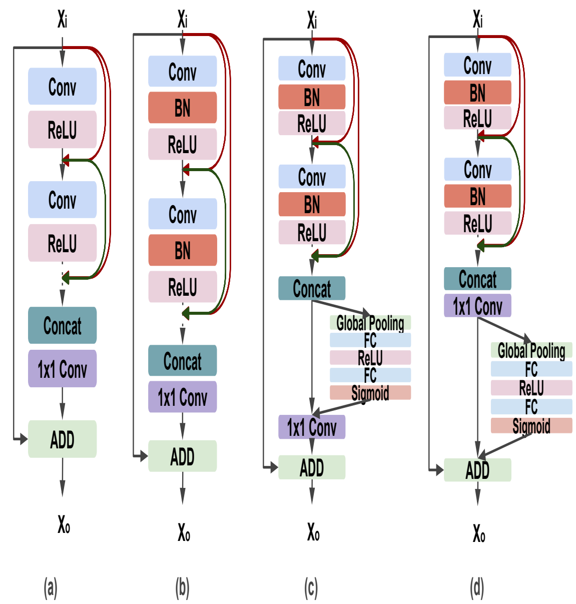

Figure 3.3: Architectural comparison of varying dense residual blocks: (a) Zhang *et al.* [84], (b) Baseline2 (ours), (c) DTD-GAN(pre) (ours), and (d) DTD-GAN (ours).

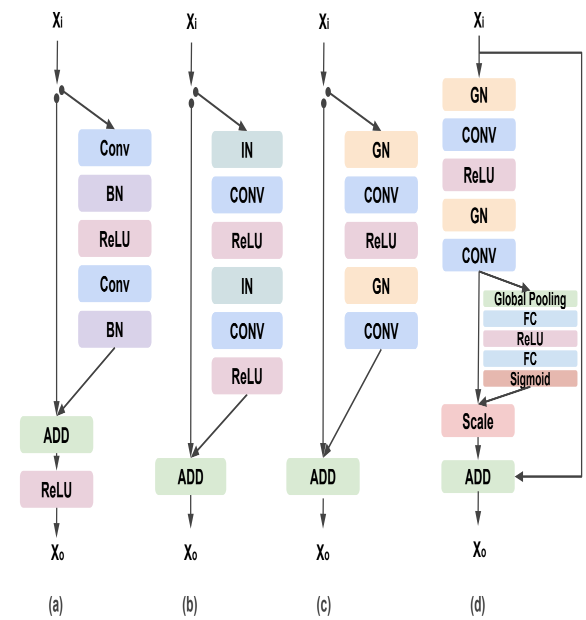

Figure 3.4: Architectural comparison of varying residual blocks: (a) He *et al.* [20], (b) Li *et al.* [39], (c) Baseline1 (ours), and (d) DT-GAN (ours).

3.3(c)&(d). We add the channel attention mechanism either before or after the concatenation to empirically explore our network's performance. In Figure 3.3, we compare the architectural blocks of each network from Zhang *et al.* [84], Baseline2, DTD-GAN(Pre), and DTD-GAN.

3.2.2 Network Architecture

Figure 3.5 shows the generator architecture of our adversarial network, which is further divided into two sub-networks, namely, WarpNet and ColorNet. WarpNet removes the geometrical distortion introduced by atmospheric turbulence by learning a warping field, which is then applied to the input image resulting in a warped image as output. The WarpNet output is blurry and lacks high-frequency details since one-to-one mapping is not ensured in forward mapping. So, ColorNet is appended after the WarpNet, which restores the perceptual details of an image lost due to forward warping. The detailed network architecture of WarpNet and ColorNet, along with the refinement in residual structures, are discussed below.

WarpNet consists of the Input Convolutional layer, Upsampling Block, 8 Residual Blocks, Downsampling Block, and Output Convolutional layer. The Input Convolutional layer projects input image to feature space, which is then passed to Downsampling Blocks. The Downsampling Block consists of 3 convolutional layers, 3 ReLU layers, and 3 Group Normalization layers. Each Convolutional layer comprises of filter size of 4×4 with the stride of 2 and padding 1. The depth of the output channel doubles

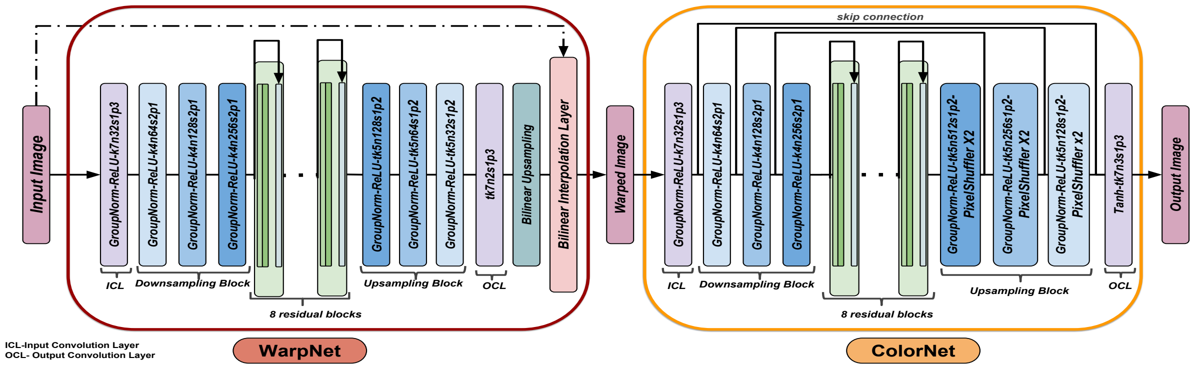

Figure 3.5: The general architecture of the generator for adversarial training. The generator architecture of DT-GAN and DTD-GAN are formed by using the modified residual blocks of Figure 3.4(d) and 3.3(d), respectively. In Figure, k is the kernel size, n is the number of feature maps, and s is the stride in each convolutional layer with p as padding.

after each convolution layer. The output of the Downsampling Block is then passed through residual blocks. The variation in residual blocks is discussed later. The output of residual blocks is then upsampled using Upsampling Block. The structure of the Upsampling Block is similar to the Downsampling Block. It uses a deconvolutional layer with a bi-linear strategy for upsampling the feature map. The depth of the output channel decreases from 256 to 32 output channels across the deconvolution layers in Upsampling Block. At last, we use the output Convolutional layer to get the final warping field. In our final restoration model, to enable the network to learn smaller geometric distortion, the last warping field of WarpNet is bi-linearly upsampled twice, as shown in Figure 3.5.

The architecture of ColorNet is slightly different from WarpNet, as ColorNet has skip-connections [64], which help in recovering information that is lost during downsampling. In our final restoration model, the Upsampling Block of ColorNet replaces bi-linear interpolation with an efficient sub-pixel convolutional [68] layer enabling the network to learn complex upsampling filters.

We construct Baseline1 using our proposed residual structure in generator architecture, as shown in Figure 3.4(c). The structure of our residual block is similar to Figure 3.4(b), but our residual block has group normalization, and the last ReLU layer is omitted out. In our final restoration model **DT-GAN** of Baseline1, we use the residual structure as depicted in Figure 3.4(d) with sub-pixel layers added to generator architecture as shown in Figure 3.5. The dense residual architecture for Baseline2 is constructed by adding a batch normalization layer to the residual dense block of the generator, as shown in Figure 3.3(b). The final model of Baseline2 is **DTD-GAN**, which has a channel attention mechanism post concatenation layer as shown in Figure 3.3(d). Later in the experiment section, we empirically show that the post concatenation model is better than the pre concatenation model. We also adopt a self-ensemble strategy [71] to maximize the performance of our model. The self-ensemble models are denoted by adding $'+'$ as postfix to the model name, which denotes the use of a self-ensemble strategy in that model.

3.2.3 Objective Functions

The network's objective is to estimate the original image I from an image of ID, which is degraded due to atmospheric turbulence. In order to achieve the same, we train the generator consisting of WarpNet (W_θ^r) parameterised by learnable parameters (θ) and ColorNet (C_ϕ^o) parameterised by learnable parameters (ϕ). While training the network, we tend to minimize the loss between the original image with generator output and WarpNet output.

The optimization of loss between WarpNet output and original image acts as additional supervision to the network, which is trained by minimizing the sum of perceptual loss [30] and content loss. The content loss is formulated by taking the pixel-wise L1 distance between the original image and reconstructed image from WarpNet. We use a perceptual loss to measure the perceptual similarity between images by taking the L1 distance between feature representation of Conv4_3 of VGG16 [69] of the original and restored image by WarpNet. The overall loss function for WarpNet can be described as:

$$l_{WarpNet} = \sum_{i=1}^{N}\sum_{j=1}^{M}|I_{i,j} - I^{\omega}_{i,j}| + \sum_{x=1}^{w_{4,3}}\sum_{y=1}^{h_{4,3}}|\psi_{4,3}(I)_{x,y} - \psi_{4,3}(I^{\omega})_{x,y}| \tag{3.2}$$

where, $\psi_{4,3}$ is the feature map and $w_{4,3}$, $h_{4,3}$ are the dimensions of the feature map of VGG16 at Conv4_3 layer output. I and I^{ω} are the original and restored image from the WarpNet receptively. The generator is trained by minimizing the loss between the generator output and the original image. The loss function consists of a linear combination of content loss, perceptual loss and adversarial loss. The intuition behind adding adversarial loss into the loss function is to produce images that lie in natural image manifold. The architecture of discriminator is the same as used for training DCGAN [55]. Instead of training discriminator with the sigmoid cross entropy loss function, we adopted least square loss function [46] which resulted in more stable training and high-quality image generation. The loss function for the generator and discriminator can be formulated as:

$$l_{generator} = \lambda_2[D_\gamma(C_\phi^o(I^{\omega}))-1]^2+\sum_{i=1}^{N}\sum_{j=1}^{M}|I_{i,j}-C_\phi^o(I^{\omega})_{i,j}|+\lambda_1\sum_{x=1}^{w_{4,3}}\sum_{y=1}^{h_{4,3}}|\psi_{4,3}(I)_{x,y}-\psi_{4,3}(C_\phi^o(I^{\omega}))_{x,y}| \tag{3.3}$$

$$l_{discriminator} = [D_\gamma(C_\phi^o(I^{\omega}))^2 + (D_\gamma(I) - 1)^2] \tag{3.4}$$

where, D_γ represents the discriminator with γ as the trainable parameter and the values of λ_1 and λ_2 were empirically found during training the network.

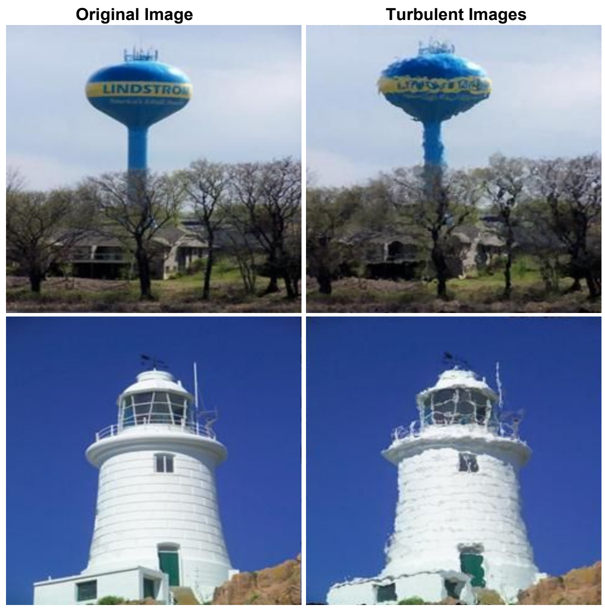

Figure 3.6: Sample images from the generated turbulent image dataset via simulation. **(Best viewed when zoomed)**

3.3 Experimentation

3.3.1 Dataset

The simulated dataset consists of 400,000 of turbulent and original image pairs which are selected from 1000 classes of ImageNet for training the restoration network. For validation of the network, all the images of the ImageNet validation dataset are used for creating a validation dataset consisting of 5995 pairs of turbulent and original images. Figure 3.6 shows a few examples of synthesized samples. To test our restoration model on natural turbulent images, we constructed a small dataset of images with natural atmospheric turbulence which were curated from YouTube videos. The YouTube videos containing atmospheric turbulence were either taken in deep turbulent environments or objects were recorded at a distance of several km by long-range video cameras.

The application of the restoration model is verified on different computer vision tasks, trained on various datasets. The restored images for each task are evaluated on three levels of datasets: (a) General (multiple categories) (b) Categorical (single category) (c) Single object instance (single class instance within a category) to show the generality of our restoration model. For image classification, a subset of 100,000 ImageNet images is selected for training and evaluated on 5000 images for the general classification. In the case of categorical classification, the FGCV [45] dataset is used consisting of 6667 training images of aircraft belonging to 70 categories and validated on 3334 images. For evaluating a single instance, we randomly choose spitfire class from FGCV datasets as one class and randomly collect

	Pix2Pix [28]	CycleGAN [87]	Li *et al.* [39]	Baseline1	DT-GAN	**DT-GAN+**	Baseline2	DTD-GAN*(pre)*	DTD-GAN	**DTD-GAN+**
PSNR	17.4403	18.8531	20.4092	21.1302	21.4483	**21.8652**	21.6801	21.7324	21.8960	**22.1382**
SSIM	0.4445	0.5392	0.6508	0.6936	0.7076	**0.7145**	0.7127	0.7134	0.7226	**0.7316**
MSE	1.4373	1.4549	0.3527	0.3272	0.3168	**0.3010**	0.3033	0.3019	0.2964	**0.2876**

Table 3.1: Quantitative analysis of various restoration architecture. Bold text shows the best performing models consisting of residual block and dense residual block, respectively.

other aircraft images for another class. The training set is comprised of 200 training images and 130 validation images. We use PASCAL3D+ [79] dataset for keypoint detection, pose, and depth estimation with the same distribution of training and validation set of images in all the tasks. The dataset consists of 2D images containing objects belonging to 12 categories that have semantic keypoint annotation, and each keypoint is mapped to a 3D CAD model giving pose and depth of the object in 3D space. For the general case, 15,271 training images are used, and the trained model is validated on 5991 images. We randomly choose the airplane category of the PASCAL3D+ dataset, which consists of 1244 training images and 566 validation images. For evaluating improvement in a single instance, all the images from the propeller sub-class of airplane class are selected.

3.3.2 Training and Implementation Details

The restoration network is trained on 400 thousand pairs of turbulent and original images. For data augmentation, each image is randomly flipped horizontally and resized, followed by cropping. Also, each batch is normalized with a mean of 0.5 and a standard deviation of 0.5. The restoration network is jointly trained end-to-end for 6 epochs with an initial learning rate of $1e-4$ for 3 epochs and the rest learning rate $5e-5$ with batch size 16. To optimize the network, we use Adam [31] optimizer with β_1=0.5 and β_2=0.999 for computing running averages of gradient and its squares. The values of λ_1 and λ_2 were empirically found to be 0.5 and 0.2, respectively. For each iteration, the discriminator and the generator are updated only once. The additional flow of gradients with respect to WarpNet output acts as additional supervision to the network resulting in faster training and higher quality of output image. It took 24 hours to train the entire network on a single Nvidia GeForce GTX 1080 Ti GPU.

3.3.3 Image Quality Metrics

Image quality is an important measure to evaluate the perceptual and structural information present in an image. To measure the quality of the restored images, we use PSNR, SSIM and MSE. These image quality metrics are applied to the restored images and non-turbulent images to show a quantitative comparison of various restoration models.

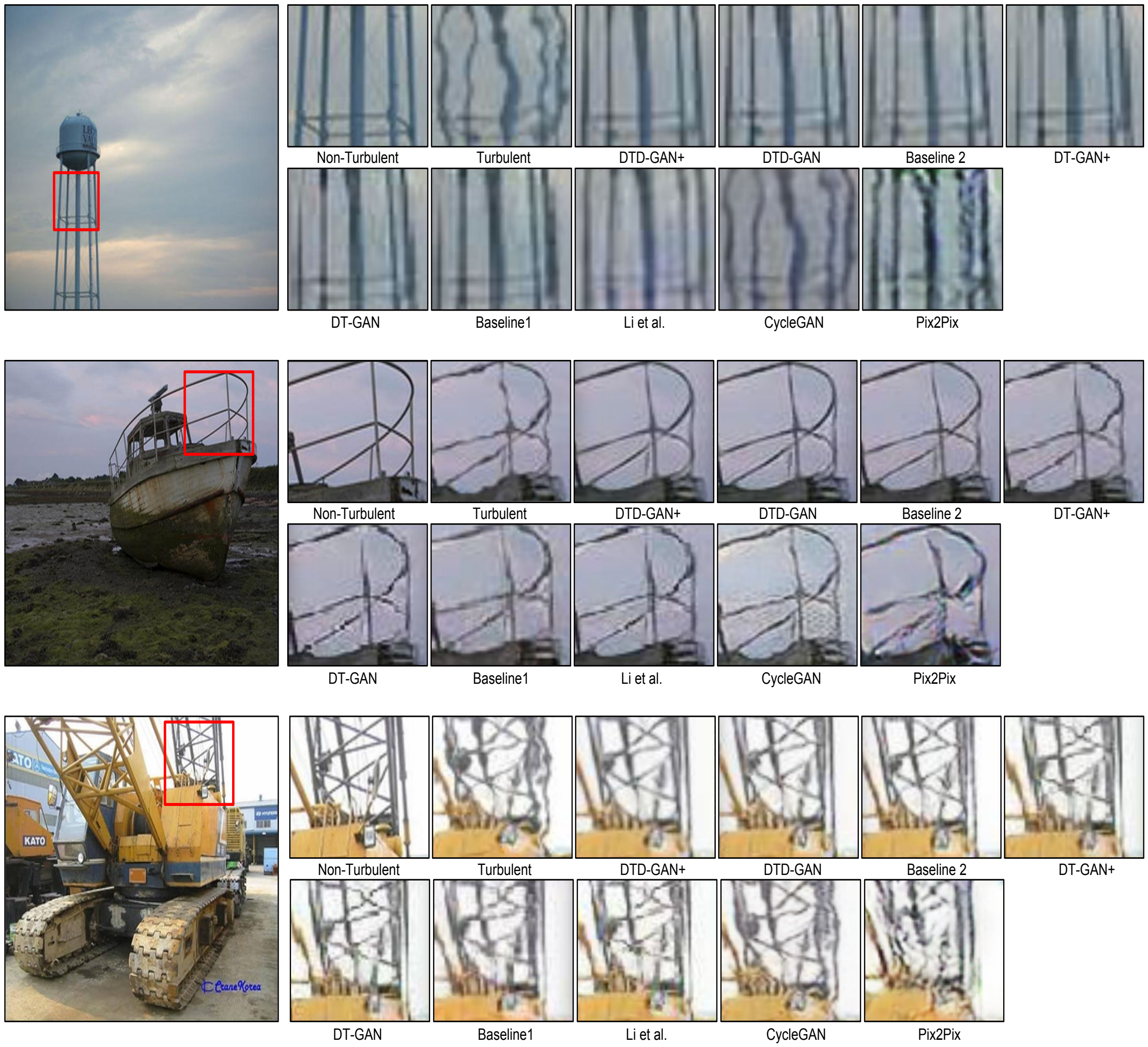

Figure 3.7: Qualitative comparison of our methods with other works in the task of image restoration.

3.3.4 Restoration Results

We compare our final restoration models(DT-GAN+, DTD-GAN+) with state-of-the-art image-to-image translation methods: Pix2Pix [28] and CycleGAN [87] along with Li *et al.* [39] which is the best performing model for removing water turbulence in images. The quality of the restored images is quantitatively analyzed using image quality metrics. Our final model DTD-GAN shows significant quantitative improvement over the other state-of-the-art-method as shown in Table 3.1. The results are further improved when the self-ensemble version of our final model i.e DTD-GAN+ is used.

Qualitative restoration results of different restoration models on ImageNet images are visualized in Figure 3.7 from which we can infer that CycleGAN, Pix2Pix struggle to remove the geometric distortion introduced by atmospheric turbulence. In contrast, our DTD-GAN+ model not only removes the

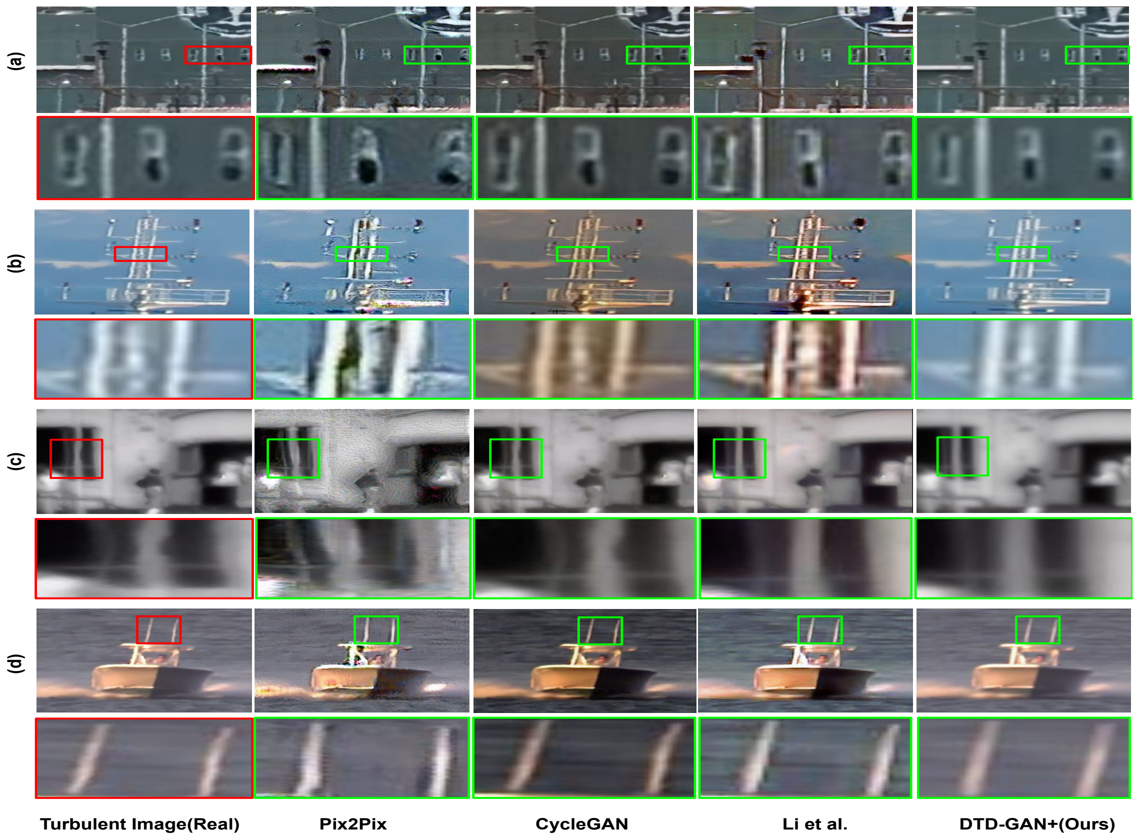

Figure 3.8: Shows the restoration of natural turbulent images curated from Youtube. Red and green box contains the turbulent and restored image patch, respectively. Column 1: Natural turbulent image curated from YouTube. Column 2: Restored images from Pix2Pix. The restored images highly suffer from image artifacts and color disparity. Column 3: Restoration by CycleGAN. The restored image struggles to produce the true color as shown in (b, c). Column 4: Natural turbulent images restored by Li et al. The restored images have partially elevated the problem of geometrical distortion but suffers from color artifacts shown in (b, d). Column 5: Restored images by DTD-GAN+. The restored images remove the geometrical distortion at the finer level shown in (a, b) and produce true color output observed in (b, c, d). **(Best viewed when zoomed)**

geometric distortion but also alleviates the artifacts. Another observation can be drawn from Figure 3.7. Our DT-GAN struggles to remove finer distortion, which is drastically reduced by DTD-GAN.

Figure 3.8 shows the restoration of images having natural atmospheric turbulence. We can observe Pix2Pix and CycleGAN is unable to remove geometrical distortion caused due to atmospheric turbulence. Moreover, CycleGAN struggles to regain the textural colors of the image. Li *et al.* [39] removes the geometrical distortions at courser image level, but, it fails to remove the geometric distortion in Figure 3.8 (a, b) and color artifacts in Figure 3.8 (b, d) at finer image level. Our best restoration model DTD-GAN+ can remove the geometric distortion caused by atmospheric turbulence and not suffer from image artifacts at the finer image level. However, we find the results of Li *et al.* [39] to be more percep-

Figure 3.9: Row 1 shows turbulent image and Row 2 shows its corresponding restored image over a range of structure constant. Row 3 display the turbulent image and Row 4 shows its corresponding restored image over a range of object distance. **(Best viewed when zoomed)**

tually pleasant in Figure 3.8 (c) than our restored images, but, they fail to maintain the color consistency when observed at the granular level.

We also perform an ablative investigation on Baseline1 and Baseline2 showing the effects of adding WarpNet and ColorNet. From Table 3.2, we find that, if the baseline consists of only either WarpNet or ColorNet, then the restoration performance of these models is inferior compared to baselines consisting of both WarpNet and ColorNet. This is because of ColorNet alone struggles to remove the geometric distortion caused by atmospheric turbulence whereas WarpNet finds it difficult to retain the information lost due to forward warping. Another inference can also be drawn from Table 3.2, the restoration performance of WarpNet is better than ColorNet as WarpNet restores the structural information of an image.

	Baseline1			Baseline2		
WarpNet	✗	✓	✓	✗	✓	✓
ColorNet	✓	✗	✓	✓	✗	✓
PSNR	20.943	21.075	**21.130**	21.730	21.817	**21.896**
SSIM	0.6761	0.6854	**0.6936**	0.7051	0.7127	**0.7226**
MSE	0.3321	0.3292	**0.3272**	0.2984	0.2977	**0.2964**

Table 3.2: Ablation investigation of WarpNet and ColorNet. We observe in Baseline1 and Baseline2, the joint model consisting of WarpNet and ColorNet performs the best which is indicated in bold.

	Baseline1				Baseline2			
Channel Attention	✗	✗	✓	✓	✗	✗	✓	✓
Sub-Pixel Mechanism	✗	✓	✗	✓	✗	✓	✗	✓
PSNR	21.130	21.293	21.405	**21.448**	21.681	21.730	21.840	**21.896**
SSIM	0.6936	0.7014	0.7057	**0.7076**	0.7156	0.7206	0.7217	**0.7226**
MSE	0.3272	0.3217	0.3173	**0.3168**	0.3033	0.2997	0.2989	**0.2964**

Table 3.3: Effects in the performance of the baselines by adding channel attention and sub-pixel mechanisms to the baselines. Bold entries indicate the best performing model achieved by attaching both.

3.3.5 Effects of Channel-Attention and Sub-Pixel Mechanism

In this subsection, we show the improvement in restoration obtained by adding channel attention and sub-pixel mechanism in our Baseline1 and Baseline2. From Table 3.3, we observe that there is a significant decrease in the performance of Baseline1 and Baseline2 without the addition of these modules. This is because of the channel attention and sub-pixel mechanisms that improve the learning of inter-channel dependency and remove atmospheric turbulence at the finer level. In Table 3.3, we show the contribution of an individual module in efficiently improving the performance of baselines. We observe a more noticeable improvement in Baseline1 than Baseline2 because dense residual blocks already carry forward the lower-level information, which reduces the combined effect of channel-attention and sub-pixel mechanism.

3.3.6 Analysis of Varying Distance and Structure Constant on Restoration

The major cause in atmospheric changes that introduces turbulence into the images are: a) Distance from an object, b) refractive fluctuation of an environment is characterized by structure constant C_n^2. In Figure 3.9(Row 1), we can see that our model is able to recover images at various strength of C_n^2 ranging from $3e-13m^{-2/3}$ to $1e-12m^{-2/3}$ at 2km object distance. A high value of C_n^2 is $8e-13m^{-2/3}$, which is usual in summer daytime can be easily restored by our model, as shown in Figure 3.9(Row 1).

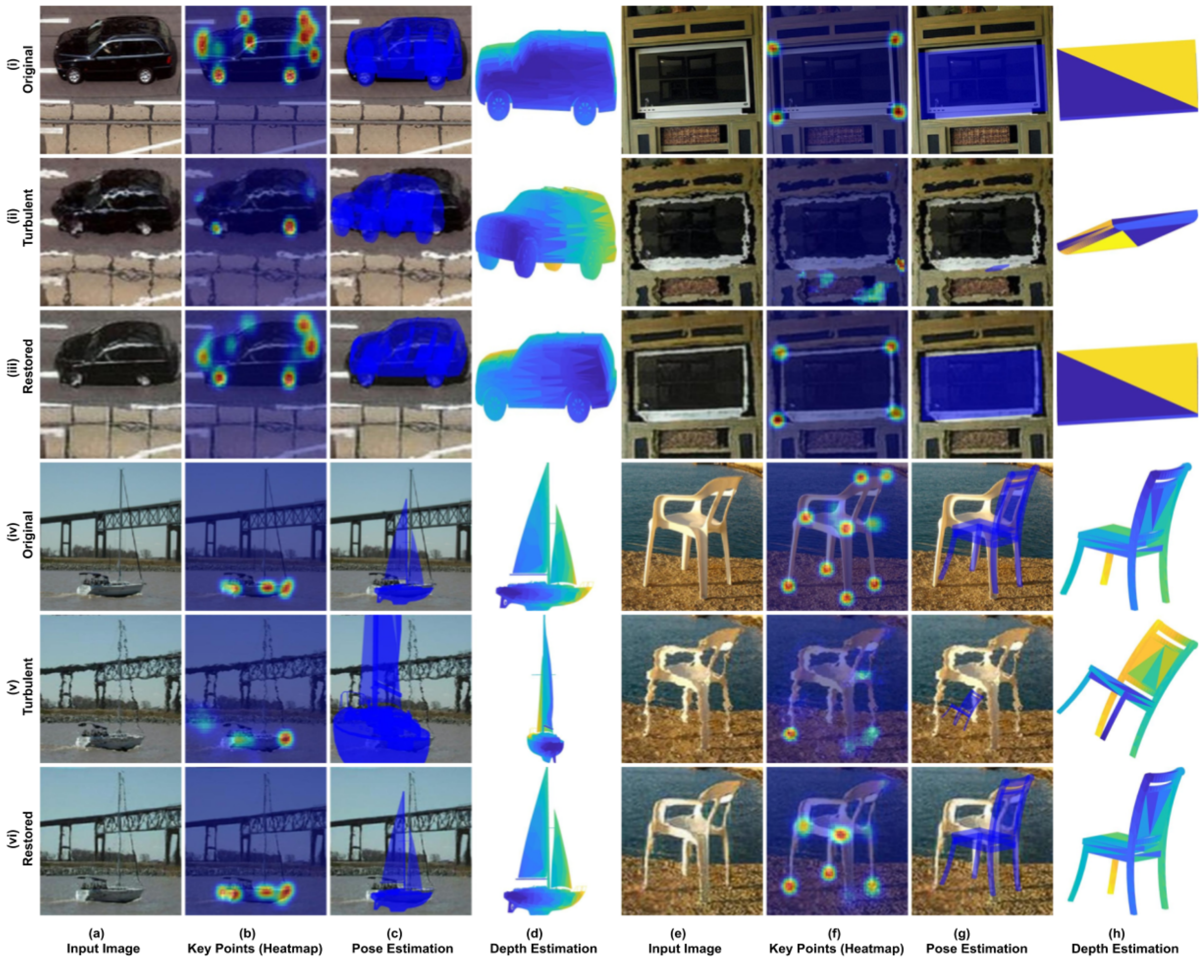

Figure 3.10: **Columns wise:** (a) and (e) are the input images to the network. (b) and (f) is the output heatmaps of the hourglass network overlapped on the input image. (c) and (g) is the CAD models projected on the input image. (d) and (h) is the depth visualization of CAD models. **Row wise:** (i) and (iv) are the results of original images. (ii) and (v) are the results of the turbulent images. (iii) and (vi) are the results of the restored images by DTD-GAN+. **(Best viewed when zoomed)**

Another factor that controls the atmospheric turbulence is the distance, analyzed in Figure 3.9(Row 2), showing that our restoration model can recover images at a distance ranging from $2-10$Km keeping the C_n^2 to be $3e-13m^{-2/3}$. We can also see in Figure 3.9(Row 2) that our model struggles to restore turbulent images at a distance of 10Km as the image degradation due to atmospheric turbulence is very high.

3.4 Application

This section shows the improvement in image classification, keypoint detection, pose, and depth estimation tasks using restored images. The restoration model's performance in improving a task is

	Original	Turbulent	Pix2Pix [28]	CycleGAN [87]	Li*etal.* [39]	BaseL1	DT-GAN	DT-GAN+	BaseL2	DTD-GAN	DTD-GAN+
Classification Accuracy											
ResNet-18											
	Original (%)	Turbulent (%)	Pix2Pix [28] (%)	CycleGAN [87] (%)	Li*etal.* [39] (%)	BaseL1 (%)	DT-GAN (%)	DT-GAN+ (%)	BaseL2 (%)	DTD-GAN (%)	DTD-GAN+ (%)
General	52.68	35.02	23.46	35.12	36.88	44.58	44.96	**45.66**	45.11	45.46	**45.81**
Categorical	67.53	27.48	26.74	28.03	32.52	39.75	40.91	**42.78**	40.39	41.58	**42.94**
Instance	91.53	77.69	76.15	79.23	80.77	83.08	84.62	**87.69**	85.38	86.92	**88.46**
View Point Estimation Error											
Convex Optimization Framework											
	Original (*degrees*)	Turbulent (*degrees*)	Pix2Pix [28] (*degrees*)	CycleGAN [87] (*degrees*)	Li*etal.* [39] (*degrees*)	BaseL1 (*degrees*)	DT-GAN (*degrees*)	DT-GAN+ (*degrees*)	BaseL2 (*degrees*)	DTD-GAN (*degrees*)	DTD-GAN+ (*degrees*)
General	11.231	25.351	29.032	28.931	25.457	25.035	23.573	**23.201**	22.734	22.470	**22.392**
Categorical	9.671	16.365	22.233	21.011	16.715	15.801	14.792	**14.519**	15.598	14.717	**14.223**
Instance	11.445	27.124	31.391	29.732	25.567	23.541	18.082	**17.814**	20.122	17.668	**16.962**
Keypoints Estimation Error											
Stacked Hourglass Network											
	Original (*relative error*)	Turbulent (*relative error*)	Pix2Pix [28] (*relative error*)	CycleGAN [87] (*relative error*)	Li*etal.* [39] (*relative error*)	BaseL1 (*relative error*)	DT-GAN (*relative error*)	DT-GAN+ (*relative error*)	BaseL2 (*relative error*)	DTD-GAN (*relative error*)	DTD-GAN+ (*relative error*)
General	0.000	341.249	366.657	352.945	338.374	324.316	323.075	**321.157**	323.624	322.168	**319.894**
Categorical	0.000	282.992	321.212	309.814	261.586	250.732	245.653	**240.454**	246.189	242.166	**239.284**
Instance	0.000	311.657	342.543	322.674	308.890	291.728	270.969	**264.787**	281.664	262.594	**256.561**

Table 3.4: Quantitative results on original, turbulent and restored images in downstream computer vision tasks which are validated on datasets consisting of different number of classes. **Top**: Classification accuracy of various image sets on trained ResNet-18. **Middle**: View point estimation error calculated by convex optimization framework capturing the deviation between the predicted and groundtruth pose in various datasets. **Bottom**: Keypoints estimation error obtained by using the heatmaps of stacked hourglass network on different image sets. BaseL1 and BaseL2 are abbreviated form of Baseline1 and Baseline2.

evaluated by training a task-specific model with non-turbulent images and validating the performance of the model by turbulent, restored, and original image sets.

For image classification task, we train a ResNet-18 [20] model. From Table 3.4, we can observe that across all the validation datasets, there is a more than 10% increase in classification accuracy for the restored images by DTD-GAN+ over turbulent images. Restored images by Pix2Pix suffered from images artifacts which results in lower classification accuracy than turbulent images.

Stacked Hourglass Network [50] is trained for correctly estimating semantic keypoints in an image degraded due to atmospheric turbulence. The evaluation process is performed by taking the absolute square difference of output heatmaps containing probable keypoints area generated by non-turbulent and turbulent or restored images, relative to the heatmaps predicted by non-turbulent images. In Figure 3.10(column 2 and column 6), we can observe that the restored images by DTD-GAN+ can detect all the semantic keypoints left undetected by the network for turbulent images. Quantitatively, there is more than a 5% decrease in the relative error for the restored images by DTD-GAN+ compared to turbulent images, as shown in Table 3.4. Restored images by CycleGAN and Pix2Pix have higher keypoint estimation errors than turbulent images as they are unable to remove geometrical distortion from the turbulent images completely. Moreover, the restored images suffer from image artifacts during reconstruction.

We use the methodology of Pavlakos *et al.* [53] to get the pose and depth of an object in a 2D image. In detail, this approach uses semantic keypoints locations on the 3D CAD model along with the keypoints of a 2D image predicted by hourglass network and then employing these correspondences in a convex optimization framework which estimates the shape and pose of an object in the 2D image. For measuring the view estimation error between the poses, geodesic distance is used between an estimated pose, P1, and ground truth pose, P2, which is as follows:

$$\delta(P_1, P_2) = \frac{||\log(P_1^T P_2)||_F}{\sqrt{2}} \tag{3.5}$$

where, F denotes the Frobenius norm.

Quantitatively, results shown in Table 3.4 indicate that across all the datasets, there is more than 11% decrease in median viewpoint estimation error on restored images over turbulent images. However, the restored images by Pix2Pix and CycleGAN suffer from image artifacts that cause difficulties in identifying semantic keypoints, resulting in higher viewpoint estimation error than turbulent images. In Figure 3.10(column 3-4 and column 7-8), we can visualize that pose and depth estimation of restored images by DTD-GAN+ are close to non-turbulent images. Inferring from all the above experiments, we show qualitatively and quantitatively the restoration model's application in improving computer vision tasks.

3.5 Summary

In this chapter, we propose a deep adversarial network to remove atmospheric turbulence. Compared to the traditional restoration methods, our model neither utilizes any prior knowledge about the turbulence field, nor does it combines multiple images to generate the restored image at test time. Our proposed restoration architecture achieves state-of-the-art performance by surpassing the general image-to-image translation models. To train our model, we simulate a large dataset and show its generalization by testing the trained model on simulated images over a range of atmospheric turbulence parameters and on images having natural atmospheric turbulence. Extensive experiments are also conducted for various tasks such as image classification, semantic keypoints detection, pose and depth estimation which shows quantitatively and qualitatively the applicability of our restoration model in improving computer vision tasks. Additionally, the experiments are conducted on datasets from a thousand classes to a single instance to show that the restoration model can be generalized for any size of the dataset.

Chapter 4

Improving Segmentation in Atmospheric Turbulence

In this chapter, we focus on the problem of the semantic segmentation model in adapting to hot climate cities. We propose a framework that improves the performance of semantic segmentation models without explicitly creating an annotated dataset for such adverse weather variations. The framework consists of two parts, a restoration network to remove the geometrical distortions caused by hot weather and an adaptive segmentation network trained on an additional loss to adapt to the statistics of the ground-truth segmentation map. Our framework showed a total IoU gain of 12.707 on the Cityscapes dataset over standard segmentation models.

4.1 Introduction

In computer vision literature, understanding the semantics of the scene is achieved by semantic segmentation. Formally, we define semantic segmentation as a method of classifying each pixel into its object category. Cityscapes [7] is one of the widely used datasets for training semantic segmentation models in autonomous navigation-based settings. The images of Cityscapes have been captured from road scenes from different Germany cities, which have relatively colder and clear weather. Now, for instance, if we train a semantic segmentation model on the Cityscapes dataset and deploy it in places having extremely hot weather conditions such as Dubai, then the trained model finds hard to keep its optimal performance and give poor segmentation results, as shown in Figure 4.1(c) (even though the roads may look similar). This phenomenon happens due to the geometrical distortion caused by hot weather into the input image shown in Figure 4.1(a). Hence, it becomes necessary for us to ponder upon the problem of adapting semantic segmentation models in such weather variations due to the change in geographical location. This condition, especially variations caused by hot weather, is also referred to as atmospheric turbulence [89] as it affects the atmospheric parameters such as the refractive index between an object and a camera. The problem of semantic segmentation model to generalize for hot weather can be bypassed by collected data, especially in such weather conditions, and training a model on the collected images. However, collecting and annotating images for such atmospheric conditions is extremely tedious, which is time-consuming and very expensive.

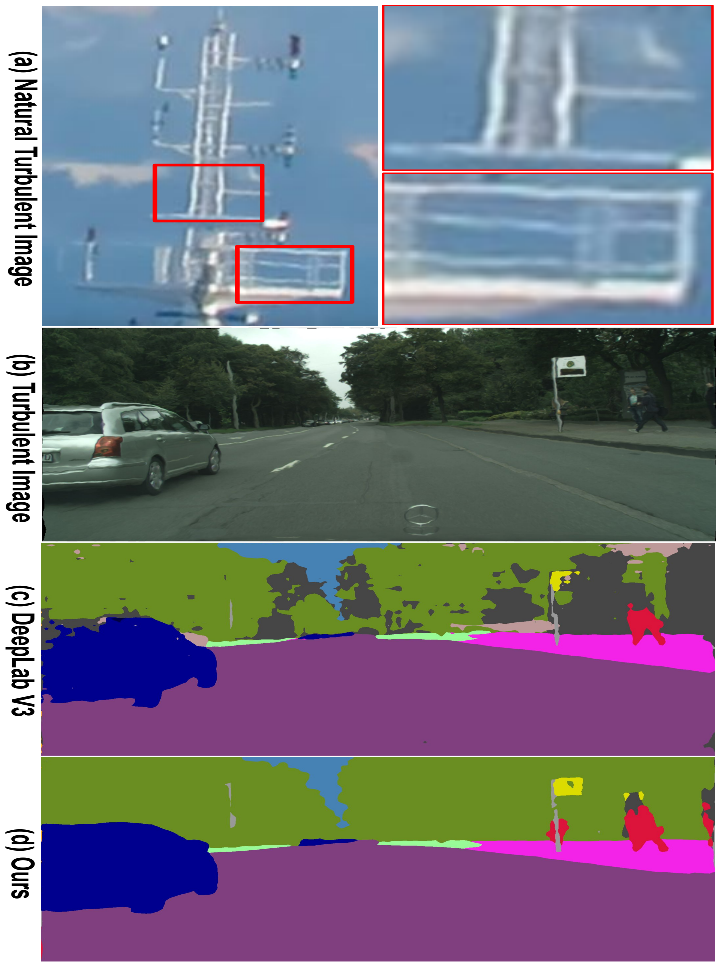

Figure 4.1: (a) Natural atmospheric turbulent images curated from the internet. The zoomed patches show the geometrical distortion caused by atmospheric turbulence. (b) A simulated atmospheric turbulent image of the Cityscapes dataset. (c)-(d) Performance of DeepLabV3 [6] and our proposed method on the turbulent image.

In this chapter, we propose a solution to improve semantic segmentation model performance in hot weather without explicitly creating an annotated dataset. The proposed framework consists of two networks: Restoration network and Segmentation network. The restoration network is specifically intended to minimize the geometrical distortion caused by atmospheric turbulence in an image. We could have used existing machine learning methods [56, 80] for restoring images from atmospheric turbulence. But, these methods suffer from two significant limitations: (i) none of the methods works for single image restoration, and (ii) considerable variation in an atmospheric parameter cannot be handled by these methods. To overcome these issues, we trained our restoration model on a large scale dataset, having images with varying atmospheric parameters for better generalization. At inference time, the trained restoration network can perform single image restoration. The architecture of our restoration network is adapted from the widely used image-to-image translation network [28]. Additionally, we introduce

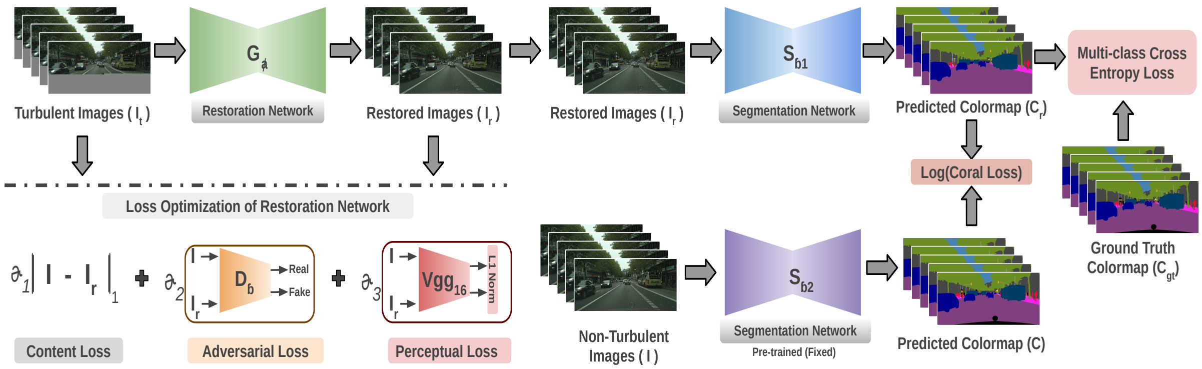

Figure 4.2: **Overview of our framework:** Our restoration network G_θ takes the input turbulent images I_t and gives the restored images I_r. To train the parameters of restoration network θ, a linear combination of losses is minimized between the restored and non-turbulent images. The restored images are further fed into the segmentation network $S_{\beta 1}$, which predicts the segmentation colormap C_r. The parameters of segmentation network $\beta 1$ is trained by calculating multi-class cross-entropy loss between C_r and ground-truth segmentation colormap C_{gt}. Additionally, we take the logarithm of CORAL loss between C_r and the predicted segmentation colormap C, which acts as an additional loss to train $\beta 1$. The segmentation colormap C is obtained by passing non-turbulent images I into pre-trained segmentation network $S_{\beta 2}$, with fixed parameters $\beta 2$.

Channel Attentive Multi-Scale Residual Block (CA-MSRB), which learns local multi-scale features and the inter-dependencies between residual channels using an attention mechanism.

The restored images obtained from the restoration network are passed to the segmentation network. The segmentation network consists of a DeepLabV3 [6] model, which is trained on multi-class cross-entropy loss between segmentation colormap of the restored image and ground-truth segmentation colormap. To make our semantic segmentation model more adaptive to the turbulent environment, we additionally use CORAL loss [70] between the reconstructed image segmentation colormap and the non-turbulent image segmentation colormap got from pre-trained DeepLabV3 model. By using the additional loss, there is further improvement in segmentation results, and the domain gap between restored and non-turbulent segmentation colormap reduces. Our method shows significant improvement in segmentation results on the Cityscapes dataset, particularly for small classes (Figure 4.1) like poles, person, and, rider which are essential and valuable classes in autonomous navigation.

4.2 Our Approach

In this section, we describe the formalization of our proposed restoration model, followed by an improved segmentation model. The formalization of our restoration framework begins with a dataset consisting of turbulent images $I_t = \{I_t^i : i = 1...n\}$ and their corresponding non-turbulent images $I = \{I^i : i = 1...n\}$, where $I^i \in \mathsf{R}^{N\times M}$ and n is the total number of samples in the dataset. Then,

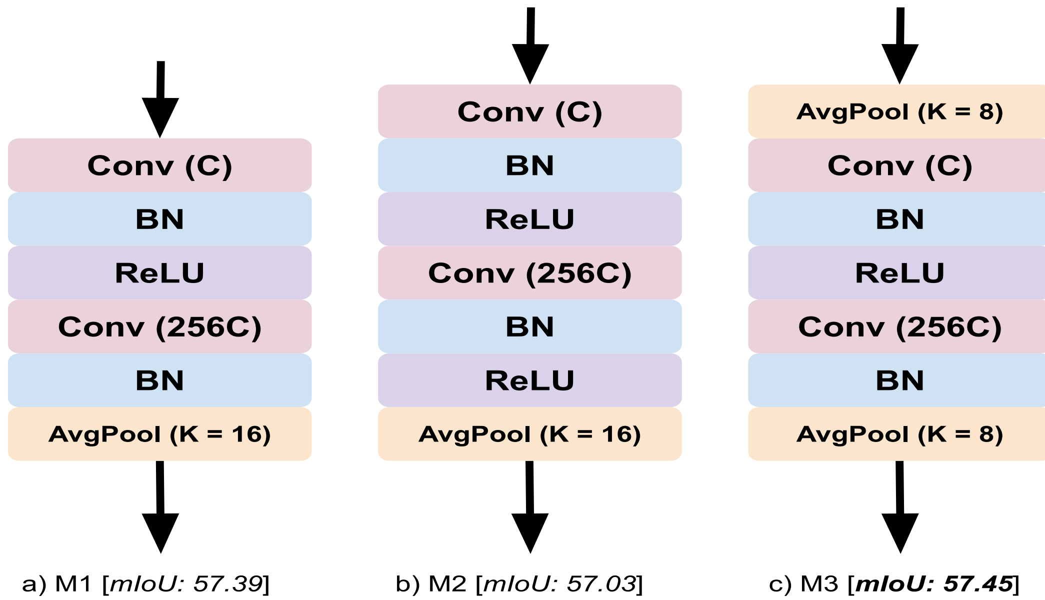

Figure 5.9: Shows various feedback architecture and its mIoU used in feedback framework. Conv. and C represent a convolutional layer with C output channels. AvgPool and K is an average pooling layer with kernel size K. BN stands for Batch normalization.

segmentation probability prediction $abs(I_{SP}^{i_{t-1}} - I_{SP}^{i_{t-2}})$ gives better mIoU over concatenation. Also, we can observe individually $I_R^{i_{t-1}}$ or $I_{SP}^{i_{t-1}} - I_{SP}^{i_{t-2}}$ do not give adequate input to the feedback network. We propose 3 feedback architectures: $M1$, $M2$, and $M3$, which takes the feedback input and adds its output in the restoration network. Among all the models, $M3$ performed the best because it gradually averages the feature by adding average pooling layers at the head and tail of the feedback network instead of adding only an average pooling at the tail of the network showed in $M1$ and $M2$. We also observe that adding a ReLU layer before average pooling in $M2$ reduces the performance of the network as it increases sparsity in the output due to which sufficient feedback information is not passed.

5.4 Summary

In this work, we demonstrated semantic feedback learning ability to improve semantic segmentation models in hot weather conditions. We propose a feedback framework that consists of a restoration network, a segmentation network, and a feedback network. The feedback network gives a feedback response to the restoration network, which attends those areas and fixes regions that need to be restored. We further boosted the performance of our model by new loss functions: iFL and CB-iFL. We proved the effectiveness of our proposed feedback model and losses through extensive experiments and ablation studies. Our work unlocks doors for further potential application of the feedback mechanism in other weather conditions such as snow, rain, and fog.

Chapter 6

Conclusions and Future Directions

In this thesis, we proposed methods designed to improve the performance of computer vision models in hot weather conditions, also known as atmospheric turbulence. We began our dissertation with an inverse problem of generating atmospheric turbulence. Unlike the traditional turbulent image generation methods that rely on a statistical model, we used a data-driven approach to learn the generative network's parameters that transform a non-turbulent image into a turbulent image.

Next, we proposed a deep adversarial network to remove atmospheric turbulence. Compared to the traditional restoration methods, our model neither utilizes any prior knowledge about the turbulence field nor combines multiple images to generate the restored image at test time. Our proposed restoration architecture achieves state-of-the-art performance by surpassing the general image-to-image translation models. We use the restored images in various tasks such as image classification, semantic keypoints detection, pose, and depth estimation to see the performance gain over the turbulent image.

Further, we specifically chose to improve semantic segmentation in hot weather because of its broad applicability in autonomous navigation systems. Our restoration framework circumvents the painstaking tasks of semantic annotation on turbulent images. The proposed framework worked in two stages. In the first stage, we proposed a restoration network specifically intended to remove geometrical distortion from turbulent images. Additionally, to improve our restoration network's performance, we introduced a CA-MSRB block, which learns local residuals at multi-scale and interdependencies between the residual channels. In the next stage, the restored images were passed through an adaptive semantic segmentation model to give segmentation colormap. The segmentation results showed by our framework gave a significant improvement in small classes such as poles, person, and rider, which are more important and valuable classes for autonomous applications. Further, we proposed a feedback framework that consists of a restoration network, a segmentation network, and a feedback network. The feedback network gives a feedback response to the restoration network, which attends those areas and fixes regions that needed restoration. We further boosted the performance of our model by new loss functions: iFL and CB-iFL.

Our work unlocks doors for further potential application of the proposed frameworks in other weather conditions such as snow, rain, and fog. Mainly, the feedback mechanism's application has not been applied in the contexts designed for improving computer vision model performance in other weather conditions. Another future direction for the current work is to consider restoring images in all weather conditions as a holistic problem, which could be addressed by multi-source domain adaptation kind of settings.

Related Publications

Conference:

1. **Shyam Nandan Rai**, Vineeth N Balasubramanian, Anbumani Subramanian, C. V. Jawahar. **Spatial Feedback Learning to Improve Semantic Segmentation in Hot Weather.** *British Machine Vision Conference (BMVC), 2020*

2. **Shyam Nandan Rai**, Vineeth N Balasubramanian, Anbumani Subramanian, C. V. Jawahar. **Munich to Dubai: How far is it for Semantic Segmentation?.** *In The IEEE Winter Conference on Applications of Computer Vision (WACV), 2020*

3. **Shyam Nandan Rai**, C. V. Jawahar. **Learning To Generate Atmospheric Turbulent Images.** *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2019*

Journal:

1. **Shyam Nandan Rai**, C. V. Jawahar. **Removing Atmospheric Turbulence via Deep Adversarial Learning.** *IEEE Transactions on Image Processing (TIP) [Under Review]*

Bibliography

[1] M. Aubailly, M. A. Vorontsov, G. W. Carhart, and M. T. Valley. Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach. In *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation III*, volume 7463, page 74630C. International Society for Optics and Photonics, 2009.

[2] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[4] M. Campbell. Atmospheric turbulence and its influence on adaptive optics. 2009.

[5] B.-k. Chen, C. Gong, and J. Yang. Importance-aware semantic segmentation for autonomous driving system. In *IJCAI*, 2017.

[6] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[8] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

[9] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum. End-to-end unsupervised deformable image registration with a convolutional neural network. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 204–212. Springer, 2017.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

[11] B. L. Ellerbroek. First-order performance evaluation of adaptive-optics systems for atmospheric-turbulence compensation in extended-field-of-view astronomical telescopes. *JOSA A*, 11(2):783–805, 1994.

[12] D. L. Fried. Limiting resolution looking down through the atmosphere. *JOSA*, 56(10):1380–1384, 1966.

[13] D. L. Fried. Probability of getting a lucky short-exposure image through turbulence. *JOSA*, 68(12):1651–1658, 1978.

[14] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017.

[15] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *European Conference on Computer Vision*, pages 311–326. Springer, 2016.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[17] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *JMLR*, 2012.

[19] D. Gutierrez, F. J. Seron, A. Munoz, and O. Anson. Simulation of atmospheric phenomena. *Computers & Graphics*, 30(6):994–1010, 2006.

[20] 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*, pages 770–778, 2016.

[21] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 607–614. IEEE, 2010.

[22] K. Hornik, M. Stinchcombe, and White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.

[23] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[24] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[26] R. Hufnagel. Restoration of atmospherically degraded images: Woods hole summer study, 1966.

[27] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[29] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[30] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[31] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[34] R. Lane, A. Glindemann, J. Dainty, et al. Simulation of a kolmogorov phase screen. *Waves in random media*, 2(3):209–224, 1992.

[35] C. P. Lau, Y. H. Lai, and L. M. Lui. Restoration of atmospheric turbulence-distorted images via rpca and quasiconformal maps. *arXiv preprint arXiv:1704.03140*, 2017.

[36] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[37] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE/CVPR conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[38] J. Li, F. Fang, K. Mei, and G. Zhang. Multi-scale residual network for image super-resolution. In *ECCV*, 2018.

[39] Z. Li, Z. Murez, D. Kriegman, R. Ramamoorthi, and M. Chandraker. Learning to see through turbulent water. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 512–520. IEEE, 2018.

[40] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. In *CVPR*, 2019.

[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[42] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE TIP*, 2018.

[43] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[44] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[45] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[46] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[47] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[48] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[49] S. K. Nayar and S. G. Narasimhan. Vision in bad weather. In *ICCV*. IEEE, 1999.

[50] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[51] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[52] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[53] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2011–2018. IEEE, 2017.

[54] K. Purohit, S. Mandal, and A. N. Rajagopalan. Multilevel weighted enhancement for underwater image dehazing. *J. Opt. Soc. Am. A*, 36(6):1098–1108, Jun 2019.

[55] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[56] R. Ragazzoni, E. Marchetti, and G. Valente. Adaptive-optics corrections available for the whole sky. *Nature*, 403:54–56, 2000.

[57] S. N. Rai, V. N. Balasubramanian, A. Subramanian, and C. Jawahar. Munich to dubai: How far is it for semantic segmentation? In *WACV*, 2020.

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

[59] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1137–1149, 2017.

[60] K. Riley, D. S. Ebert, M. Kraus, J. Tessendorf, and C. D. Hansen. Efficient rendering of atmospheric phenomena. *Rendering Techniques*, 4:374–386, 2004.

[61] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017.

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

[63] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *Transactions on Intelligent Transportation Systems*, 2017.

[64] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[65] 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)*, pages 1–13. IEEE, 2017.

[66] F. J. Seron, D. Gutierrez, G. Gutiérrez, and E. Cerezo. Implementation of a method of curved ray tracing for inhomogeneous atmospheres. *Computers & Graphics*, 29(1):95–108, 2005.

[67] F. Shama, R. Mechrez, A. Shoshan, and L. Zelnik-Manor. Adversarial feedback loop. In *ICCV*, 2019.

[68] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[69] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[70] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[71] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016.

[72] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, and V. Chari. Learning to generate synthetic data via compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 461–470, 2019.

[73] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016.

[74] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.

[75] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*. IEEE, 2019.

[76] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[77] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. IEEE, 2003.

[78] Y. Wu and K. He. Group normalization. In *ECCV*, 2018.

[79] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 75–82. IEEE, 2014.

[80] Y. Xie, W. Zhang, D. Tao, W. Hu, Y. Qu, and H. Wang. Distortion-driven turbulence effect removal using variational model. *CoRR*, abs/1401.4221, 2014.

[81] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*.

[82] Y. Yitzhaky, I. Dror, and N. S. Kopeika. Restoration of atmospherically blurred images according to weather-predicted atmospheric modulation transfer functions. *Optical Engineering*, 36(11):3064–3073, 1997.

[83] A. R. Zamir, T.-L. Wu, L. Sun, W. B. Shen, B. E. Shi, J. Malik, and S. Savarese. Feedback networks. In *CVPR*, 2017.

[84] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018.

[85] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2016.

[86] Y. Zheng, X. Yu, M. Liu, and S. Zhang. Residual multiscale based single image deraining. In *BMVC*, 2019.

[87] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *ICCV*, 2017.

[88] X. Zhu and P. Milanfar. Image reconstruction from videos distorted by atmospheric turbulence. In *Visual Information Processing and Communication*, volume 7543, page 75430S. International Society for Optics and Photonics, 2010.

[89] X. Zhu and P. Milanfar. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):157–170, 2013.