Mojgan Madadikhaljan

Single-Image Dehazing
On Aerial Imagery Using
Convolutional Neural Networks

Supervisor: Prof. Dr.-Ing. Uwe Sörgel
Examiner: Prof. Dr.-Ing. Uwe Sörgel
Master’s Thesis

Single-Image Dehazing On Aeria Imagery Using Convolutional Neural Networks

Author: Mojgan Madadikhaljan
Examiner: Prof. Dr.-Ing. Uwe Sörgel
Main advisor: Prof. Dr.-Ing. Uwe Sörgel
First assistant advisor: Seyedmajid Azimi
Second assistant advisor: Dr. Reza Bahmanyar
Submission Date: October 31st, 2019
I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

October 31st, 2019

Mojgan Madadikhaljan
Acknowledgments

I would like to express my sincere gratitude to my advisor Prof. Dr.-Ing. Uwe Sörgel at the University of Stuttgart that motivated me to work enthusiastically and willingly on this topic.

I am also grateful to my advisors at Deutsches Zentrum für Luft und Raumfahrt (DLR), specifically Prof. Dr. -Ing. Peter Reinartz for his kind hospitality and support during my master thesis.

I also want to denote my deepest appreciation to my supervisors, Dr. Reza Bahmanyar and Majid Azimi for their continuous support, immense knowledge, encouragement and insightful comments. I would not be able to find my way without your great guidance. Last but not least, I would like to state how thankful I am to my family Nasrin Ghoreishi, Reza Madadi and Maryam Madadi for all of their effort during my whole study.
“Zwei Dinge sind unendlich, das Universum und die menschliche Dummheit, aber bei dem Universum bin ich mir noch nicht ganz sicher.”

-Albert Einstein
Abstract

Aerial images are widely used in various applications such as land-use planning, environmental studies, sustainable development projects, smart cities and traffic management systems. The atmospheric conditions such as hazy weather at the flight time can affect the resolution and quality of the aerial image and consequently influence the performance of the aforementioned tasks. Haze contains floating particles in the air which can result in image quality degradation and visibility reduction in airborne data. Haze removal task has several applications in image enhancement and can improve the performance of automatic image analysis systems, namely object detection, and segmentation. In this thesis, we propose a well-performing method to dehaze aerial images using Convolutional Neural Network (CNN).

Despite rich haze removal literature in the ground-level imagery, there is a lack of methods specifically designed for aerial imagery. Considering the fact that there is a characteristic difference between the aerial imagery domain and ground one, we may not obtain the best dehazing results by applying a ground-level imagery dehazing method with no appropriate modifications on aerial images, so, a domain adaption is needed which is proved by our experiments. Investigating through different state-of-the-art ground imagery dehazing methods, All-in-One Dehazing Network (AOD-Net) is chosen as the baseline for adaption. Its outperforming results in ground-level imagery as well as its straightforward structure, flexible for further adaptation, motivate us to utilize this network in our experiments.

The domain transfer is done by training the network on aerial imagery. To do so, an aerial hazy image dataset is needed. To the best of our knowledge, there is no publicly available hazy aerial image dataset available and therefore, we create a new synthetically-hazed aerial image dataset in both homogeneous and inhomogeneous versions using different assumptions and computational strategies. For the homogeneous case, we assume to have the same ground height for all image pixels, while in the inhomogeneous case, a random Digital Elevation Model (DEM) is created and used in the hazy image generation procedure.

After training the network on the generated dataset, we test our model on natural as well as the synthetically-hazed aerial images. Both qualitative and quantitative results of the adapted network show an improvement in dehazing results. We show that the adapted
AOD-Net on our aerial image test set increases Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSim) by 2.2 dB and 9%, respectively.

Despite the improvement, the color suppression problem is still seen in the adapted network. This effect can also be realized from the Kullback-Leibler Divergence (KLD) of the color histograms between dehazed and ground truth images. Hence, we further modify the network by making it keep the Mean Squared Error (MSE) between the dehazed image and the ground truth one as minimum as possible, not only in the time domain but also in the frequency domain. The results show an improvement of the dehazed results by 2.24 dB in PSNR and 16.6% in SSim.
# Contents

Acknowledgements .......................... vii

Abstract ................................... ix

1 Introduction and Background Theory ............................................. 1
   1.1 Related Works ........................................................................... 6
      1.1.1 Prior-Based Methods .............................................................. 7
      1.1.2 Neural Network-Based Methods .............................................. 7
   1.2 Haze ....................................................................................... 8
      1.2.1 Atmospheric Scattering Model ............................................. 8
   1.3 Introduction To Convolutional Neural Networks ........................... 10

2 Synthetically Hazed Aerial Image Dataset Generation .................... 15
   2.1 Synthetically-Homogeneously-Hazed Aerial Images .................... 18
      2.1.1 Depth Image Generation ...................................................... 18
      2.1.2 Homogeneously-Hazed Images ............................................. 20
   2.2 Synthetically-Inhomogeneously-Hazed Aerial Images ................ 22

3 Methodology ............................................................................. 27
   3.1 Introduction to AOD-Net ........................................................... 29
   3.2 Adapting AOD-Net on aerial imagery ........................................ 31

4 Experiments and Results ............................................................. 35
   4.1 Evaluation Indices ................................................................. 37
   4.2 Experiments Of Network Adaption On Aerial Imagery ................. 38

5 Conclusion and Future Works ....................................................... 49

Glossary ..................................................................................... 53

Bibliography ............................................................................... 53
Chapter 1

Introduction and Background Theory
Aerial imagery has a wide variety of applications in surveying, sustainable development, city planning, change detection, cartography, traffic management systems, etc. The quality of an aerial image plays an important role in the results of the aforementioned task. Hence, new technologies such as modern cameras with higher spatial and radiometric resolutions are employed to acquire high-quality images. However, atmospheric conditions such as hazy weather may negatively affect the quality of images and decrease its visibility, transparency, and clearness which should be removed by image enhancement techniques.

In this project, we propose a technique to remove haze from the aerial images.

Haze in the atmosphere is defined as tiny floating particles that originated from smoke, dust, volcanic ashes, foliage exudation, combustion products, humidity, etc. The haze particles have the size varying from 0.01 to 10 µm [22]. The existence of the haze in the air causes the visibility to be reduced, which is also the case when taking images in hazy conditions. For example, when utilizing aerial images for different applications, haze could be a reason to decrease the performance of automatic image analysis systems such as image recognition, object detection, segmentation and tracking [18, 4, 3]. Hazy images and their dehazing strategies can be divided into four main types:

- Indoor: The image is captured inside an enclosed area such as a room. Assuming to have a homogeneous haze condition, the depth of the pixels i.e. the distance of the point to the camera determines the dense of the haze at that point (see fig. 1.1a).

- Outdoor: The camera is stationed on the ground while capturing the image from outside of an enclosed area such as roads and buildings. The amount of haze at each pixel is also depth-dependent in the case of homogeneous haze in the area (see fig. 1.1b).

- Aerial: The camera is stationed on an airplane (or helicopter) and collects pictures while moving. The pictures can be either nadir looking or tilted. The haze is seen mostly homogeneous. The homogeneity of haze in the aerial domain will be discussed further in section 1.2 (see fig. 1.1c).

- Optical satellite: The images are obtained from a sensor on the satellite and often the haze appears in some small parts of the image (see fig. 1.1d).

Since the characteristics of the aforementioned imagery domains are different, the appearance and physical parameters of haze will be different as well, which requires developing and employing appropriate procedures to remove haze and enhance the visibility in these imageries. There are many techniques available to dehaze a single input hazy image and reconstruct a haze-free one in the ground [18, 14, 19, 7, 29, 30, 5, 31] and satellite imagery domains [21, 24, 6, 20], whereas the techniques focused on dehazing of aerial images are a few.

To develop a well-performing dehazing method for aerial imagery, we studied the existing indoor/outdoor dehazing algorithms and applied them to aerial images. Then we select the best performing one for a further adaption to aerial image dehazing problem.
existing single-image dehazing methods are either based on prior knowledge [14, 31, 5] or Convolutional Neural Networks (CNNs) [7, 19, 18, 29, 30].

While prior-based dehazing strategies are limited to the cases, where their assumptions are valid, CNN-based dehazing algorithms have the potential to be trained and consequently adapted onto varying applications. This is the first motivation for us to narrow down our search into CNN-based methods. Among the studied CNN-based state-of-the-art methods, AOD-Net [18] outperformed the others, thus, we have chosen this dehazing network for our experiments. AOD-Net is a shallow network with a straightforward structure containing five convolutional layers. It also reformulates the physical haze model. In contrary to some dehazing methods [14, 31, 5] which first estimate the physical haze parameters and then recover the haze-free image, AOD-Net has no intermediate parameter and it directly outputs the dehazed image from the input. Further details will be discussed in section 3.1.
Same as typical deep learning methods, AOD-Net needs a large-enough dataset to be trained on. To pursue this purpose, in the original work [18], NYU depth V2 [27] dataset was used to provide synthetically hazed images with their ground truth for training the network. In order to apply AOD-Net to aerial imagery, one could either use the pre-trained model or train the model on hazy aerial image datasets. Due to the diverse characteristics of the aerial and ground imagery domains, a model trained on one domain may not be directly transferable to the other domain. This is also verified through our experiments. Therefore, for our model adaption, we consider training AOD-Net on a hazy aerial image dataset. To the best of our knowledge, there is no publicly available dehazing aerial image dataset so far. Thus, as the first contribution to this thesis, we generate a new dataset consisting of synthetically hazed aerial images together with their haze-free versions.

It is not practically possible to have a naturally hazed aerial image dataset. The reason is the fact that, if a dataset is to be created, there should be images from the same scene in the same imaging conditions in both hazy and haze-free versions. But this is not possible for aerial images, because the atmospheric conditions of the different parts of the whole imaging area could be different. For example, if a city is to be imaged from an airplane, there can be varying haze density or haze sources all over the city. Thus, the captured images of a city block would not be all hazy or haze-free, and consequently, we would not have a set of images with natural haze and their corresponding clear image with one flight. In other words, for the haze-free images, there is no hazy equivalent and for the hazy images, there is no haze-free image as ground truth. Having a second acquisition to fulfill the required data needs too much of time and computation to find the desired condition and is also economically demanding. Therefore, we put our focus on generating a synthetically hazed aerial image dataset.

Our aerial haze dataset consists of haze-free aerial images acquired by Deutsches Zentrum fr Luft und Raumfahrt (DLR) in 2018 in the framework of VABENE++ project using the 3K camera system [17] mounted on a helicopter flying over the city of Munich, Germany.

As a prerequisite for the synthetic hazy image creation procedure, the depth of each pixel (the euclidean distance of the corresponding pixel to the object point on the ground) should be computed and a depth map should be generated. To this end, based on the camera information, the position of the camera is recovered. Then using the co-linearity equation, the light rays are reconstructed at the time of image acquisition and the position of the object is found. Using the generated depth map, the homogeneous-hazy images are created based on the atmospheric scattering equation which will be explained in section 1.2. We also propose a method to create inhomogeneous-hazy images based on random but smooth Digital Elevation Model (DEM). The data generation process will be explained in detail in chapter 2.

As the second contribution to this thesis, we adapt AOD-Net to the aerial image dehazing problem. For this purpose, we train AOD-Net on the created aerial haze dataset. In order to evaluate the trained models, we split the dataset into train and test sets. The performance of the dehazing models is then evaluated using two popular indices in this area.
of research namely, Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSim) (discussed in section 4.1). The images from the test set are dehazed using the trained model and their PSNR and SSim are compared to the result of the trained AOD-Net on ground imagery, similar to the original work [18].

Despite the significant improvement of the dehazing performance by training the model on our aerial data set, there is a remaining color histogram suppression effect. This is a mutual artifact when dealing with dehazing algorithms [23]. In order to overcome this effect, we further modify the network to balance the color histograms of the dehazed images. As the last contribution to this thesis, we set the network to minimize the Mean Squared Error (MSE) between dehazed and haze-free image not only in the time domain but also in the frequency domain. For color histogram comparison and quantitatively show that the proposed network works reasonably better, we use Kullback-Leibler Divergence (KLD), as a distribution comparison measure. The final results show that the new proposed network performs more stable in resisting the color distribution of the hazy image with respect to the ground truth image.

We discuss some of the state-of-the-art dehazing techniques and the haze formation and formulation on an image later in this chapter. We also provide a short introduction to the basics of CNNs. In chapter 2, we go through all the progress of generating synthetically hazed aerial image dataset step by step. Next in chapter 3, we clarify the theoretical aspects of this project. The detailed experiment and corresponding results will be described in chapter 4, and finally, we conclude the research and propose potential future work in chapter 5.

1.1 Related Works

Dehazing is an important step as a pre-processing in different image enhancement challenges and image analysis tasks such as classification, specifically when dealing with outdoor and aerial images. Therefore, there are lots of single image dehazing algorithms developed so far, which can be divided into two main groups:

- Prior-based [14, 31, 5],
- CNN-based [7, 19, 18, 29, 30].

The prior-based methods use assumptions that are often based on the statistics of the corresponding image types. These statistics and assumptions help to find a way back to recover the haze-free image from its hazy versions.

On the other hand, since NNs burst into the dehazing branch, lots of NN-based algorithms are developed and outperformed the prior-based methods. In these dehazing methods, synthetically generated hazy images together with their hazy versions are used to train the network. We will give a short overview of the most breakthrough methods in section 1.1.2.
1.1 Related Works

1.1.1 Prior-Based Methods

Inspired by the dark-object subtraction technique, Dark Channel Prior (DCP) [14] dehazing technique benefits from the statistics of the outdoor scenes, in which at least in one color channel there exist local pixel patches with very low values (called ‘dark pixels’). The reason is that the objects in the outdoor scenes often contain shadows, colorful objects and dark objects which all lead to having low intensity in at least one color channel. Taking this prior as a starting point, the DCP dehazing method finds the dark channel, subtracts the contribution of the haze from it and recovers the haze-free image.

The color attenuation prior is also used to restore the haze-free image from an input hazy image [31]. It assumes that the depth of the scene is positively correlated to the density of the haze. The authors aim to find a linear model for the scene depth of the hazy image. The parameters of the linear model are learned via supervised learning. As soon as the connection between the hazy image and its depth is provided, using the assumption of positive correlation of the pixel depth and the density of haze on it, the haze-free image can be recovered.

In [5], a non-local single image dehazing method is employed. They assume that when transferring a haze-free image to RGB color space, the image will be mapped into a limited number of small color clusters, namely a few hundred. The pixels in the same cluster are mainly non-local and distributed all over the image. They also have different depths relative to the pixels in the same clusters. It means that when haze appears in the scene, the transmission coefficient of the pixels in the same cluster will be dissimilar as well. in the presence of haze, each color cluster turns to form a line, which is called haze lines. in other words, all the pixels in a hazy image can be modeled by several lines in RGB space. Using the haze lines and therefore the transmission of each pixel leads to reform the haze-free image.

1.1.2 Neural Network-Based Methods

As mentioned earlier, CNNs also spread over the image enhancement algorithms which dehazing is one of them.

In [30], the dehazing network aims to learn a non-linear function between hazy image and haze-free one. It consists of an encoder-decoder structure. The encoder consists of dense blocks [13] where each layer is connected to all the layers in front, and one residual block [11] where the networks learn the residual functions concerning the input. The encoder is designed in a way that the information flow is maximized and it is highly stable to the vanishing gradient problem. Later in the decoder module, a dense residual block performs. The output of the decoder is the dehazed image which is then compared to the ground truth image using MSE loss function and an additional perceptual loss function. The perceptual loss function forces the network to minimize the perceptual distance of the dehazed and ground truth image.

Recently, Conditional Generative Adversarial Network (cGAN) is employed in single-
image dehazing challenges as well. The authors in [19] used a network architecture with a generator to create haze-free images and a discriminator to identify whether the image is real or not. They also add a perceptual regularization based on the VGG network’s feature-maps and applied L1 regularized gradient prior in order to avoid artifacts and color distortions in the dehazed images.

A proposed method based on cGAN [7] uses two generators and two discriminators in the network to add or remove haze to the images. This method further takes advantage of cycle-consistency and perceptual losses to refine texture information and improve the final results.

The authors in [30] proposed a perceptual pyramid deep network for image dehazing, where the network architecture benefits from dense and residual blocks. It is composed of an encoder to map the image into a latent feature space and a decoder to transfer the features back from the latent space and generate the haze-free image.

Not only the deep neural networks but also the shallow neural networks with quite simple structures have shown promising results in the image dehazing scenarios. For example, the widely-used All-in-One Dehazing Network (AOD-Net) [18] achieves very high accuracy by its simple network design and a reformed mathematical formulation of the haze equation.

1.2 Haze

The tiny pieces of dust, smoke or vapor which float in the air are called haze. Even in the clearest sky and atmosphere, there will be some amount of haze. An increase in the haze density reduces air transparency, which limits the object visibility (see fig. 1.2).

When an image is captured in a hazy condition, it would not be as sharp and clear as in the haze-free condition. This happens because, during their travel from the light source, the light beams hit the haze particles and get scattered. Therefore, not only the true scene radiance but a superposition of true scene radiance and scattered light in the air will be captured by the camera. See fig. 1.3. The atmospheric scattering model explains the physical model of a hazy image formation.

1.2.1 Atmospheric Scattering Model

In order to reconstruct the haze formation process in an image, the atmospheric scattering model is defined as:

\[ I_{x,y} = J_{x,y} t_{x,y} + A(1 - t_{x,y}), \]  \hspace{1cm} (1.1)
1.2 Haze

Figure 1.2: Sample hazy images.

where

- \( I \) = Hazy image,
- \( J \) = True scene radiance,
- \( t \) = Medium transmission,
- \( A \) = Global atmospheric light,
- \( x, y \) = Pixel location.

Equation (1.1) shows that the true scene radiance is attenuated while traveling through the air to reach the camera. The illumination from the atmosphere affects the traveling light beam by adding the airlight \((A)\) term. In the homogeneous atmospheres, the medium transmission term can be calculated by

\[
t_{x,y} = e^{-\beta d_{x,y}},
\]

(1.2)

where

- \( \beta \) = Scattering coefficient of the atmosphere
- \( d_{x,y} \) = Depth of each pixel.

Most of the single-image dehazing algorithms [18, 19, 7, 30, 23] use this model to reconstruct haze-free image by estimating the transmission map, airlight or combination of them.

---

1.3 Introduction To Convolutional Neural Networks

Artificial Intelligence (AI) plays an important role in breakthrough technologies such as image recognition, image classification, object detection. The main idea in AI is to train machines on how to make decisions based on a considerable number of training samples.

Artificial Neural Networks (NNs) are one of the crucial levers to the AI technology, which have been inspired by biological NNs [10]. Artificial NNs comprised several “neurons” (or nodes), where each neuron can perform a simple task. More precisely, each neuron receives one or more signals as real numbers, processes them and transmits the output further to its adjacent neurons and in this way, the input information is distributed through the network in a forward fashion (forward pass or forward propagation). In the end, the output of the network is compared to the ground truth data and the loss of the network is computed. The contribution (weight) of each neuron is then refined through backward propagation in such a way to minimize the loss.

In image processing challenges such as ImageNet classification [25], CNNs are widely used. In CNNs, additional to the neural layers, there exist convolution layers as well. These convolutional layers act like moving window filters on the image to extract image features on different scales. Since 2012, when Krizhevsky et al. [16] won the ImageNet classification competition, CNNs have been expanded further and various network architectures have been utilized to overcome computer vision challenges. Image dehazing is one of the many tasks in which CNNs have been employed.

A CNN consists of several convolutional and neural (hidden) layers where the weights are initialized. The input image is fed into the system and the output results are put into a loss function. The difference between the estimated output and ground truth data is defined as the loss of the system. To optimize the weights, the gradient of the loss is calculated and using the chain rule, the weights are changed in a way that the loss is minimized. To understand how CNNs works, we provide a short explanation about its basics. See fig. 1.4 for more details.
1.3 Introduction To Convolutional Neural Networks

Convolution layers: the convolution layers are widely used in image processing applications. They perform like a moving window filter on the image and extract the features on different scales. The weights or coefficients of the window filters are learned during backward propagation.

Pooling layers: the pooling process is an optional intermediate process between convolutional layers which reduces the spatial size of data through forward propagation. It acts like an empty window filter that moves over the image and chooses either the maximum or average of the values which are inside the window. In fig. 1.5 an example of a max-pooling layer is illustrated.

**Neurons or nodes:** the neural layers consist of some neurons. Each neuron (or node) is a simple basic element of the network which applies weight and bias to all received input signals and sums them up. The result is inserted into a non-linear function called *activation function*:

\[ y = f(\sum wx_i + b), \]  

(1.3)
where \( f \) = Activation function, \\
\( w \) = Weights of the neuron, \\
\( x_i \) = Input signal, \\
\( b \) = Bias, \\
\( y \) = Output signal.

The fired neuron, sends its output signal to the neurons in the front layer. The input signal is processed in each layer and distributed to the front. This process is called forward propagation.

**Loss function**: A loss function is defined to quantitatively interpret the distance of the estimated result to the ground truth one. The definition of the network loss is application dependent, for instance in classification purposes, the network loss will be the difference between the estimated class of the input and the true class. Besides the classification tasks, a network may aim to enhance an input image. In this case, the loss of the network will be defined as the difference between recovered (or enhanced) image and its ground truth. We shortly refer to the most famous loss functions proposed so far:

- **Hinge or Support Vector Machines (SVM) loss**: the Hing or SVM loss is mainly used in classification tasks. To utilize this loss, the last layer of the network is assumed to be the scores \( s_j \) indicating the belonging chance of the input image to each class category. The score of the true class is subtracted from all other scores with a safety margin (default to be 1). It will be considered at the final loss calculation summation in the case it is larger than 0. The final total loss is the result of the summation (see eq. (1.4)).

\[
L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1), \tag{1.4}
\]

where \( L_i \) = Loss value for input \( i \),
\( y_i \) = The true class label (integer),
\( j \) = All incorrect class score indices,
\( s \) = Score.

- **Softmax or Cross-Entropy loss**: It is also a classification loss function. The output scores of the network which indicates the chance of belongings to each class for the input, will be first exponentiated. They will then divided by the the of all exponentiated scores to follow a probabilistic form and have the summation equal to 1 (the results are \( p_i \) for each class). At this stage, to calculate the loss for each input it is enough to compute the negative of logarithm of the true class score:

\[
L_i = -\log p_i, \tag{1.5}
\]
where \( L_i \) = Loss value for input i, 
\( p_i \) = The scores after being exponentiated and divided to summation.

- **Mean Squared Error (MSE):** This loss can be used when directly comparing a ground truth or desired image with an output image. It is the summation of the squared difference of the pixel values:

\[
L_i = \frac{1}{n} \sum \sum (I_{x,y} - G_{x,y})^2, \tag{1.6}
\]

where \( L_i \) = Loss value for input i, 
\( I_{x,y} \) = Output image in the \((x, y)\) pixel location, 
\( G_{x,y} \) = Ground truth image in the \((x, y)\) pixel location, 
\( n \) = Total number of pixels.

**Activation function:** Several defined non-linear functions, whose main task is to avoid the linear process of neurons to be unified in mathematical computations and to set the firing rate of each neuron. We will refer to some very famous activation functions:

- **Step function:** Suppresses all the input values smaller than zero, to be zero. Pushes all the input values larger or equal to zero, to be one.

\[
f(x) = \begin{cases} 
0, & x < 0 \\
1, & x \geq 0. 
\end{cases} \tag{1.7}
\]

- **Sigmoid:** Mapping all the input values in between 0 to 1. It has a S-shaped curve and has a changing gradient in the range close to zero.

\[
f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}. \tag{1.8}
\]

- **Hyperbolic tangent:** From the family of hyperbolic functions which are the analogue form of trigonometric functions. It has a similar shape to Sigmoid function, though maps the input values between -1 to 1, and has steeper change with respect to Sigmoid function.

\[
f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{1.9}
\]

- **Rectified Linear Unit (ReLU):** One of the most commonly used activation functions whose value is zero, for the inputs smaller than zero and maps the same input value for the inputs larger than 0.

\[
f(x) = \begin{cases} 
0, & x < 0 \\
x, & x \geq 0. 
\end{cases} \tag{1.10}
\]
The aforementioned activation functions are the most common ones, widely used in CNNs.

**Forward and backward propagation:** The computation of the network output based on the inputs, weights and activation functions is called *Forward pass*. The initialized weights are then changed to minimize the loss through *back propagation*.

**Batch processing:** The network is set to process not only one input image (or signal) but a batch of several inputs.

**Over and Underfitting problem:** In some cases, while training the network, the network may focus too much on fitting the weights and results on the training data, therefore it does not work well on the test set. This is called the overfitting problem which can happen due to insufficient or unbalanced data. Inversely, underfitting is the situation where the trained network performs badly on the training data. The problem of overfitting can be solved to some extent using regularization.

**Weight initialization:** In order to perform a forward pass from the first epoch, the weights should be initialized by, for instance, Xhavier weight initialization method [8].

**Regularization:** To avoid the extreme match of the network weights to the training data, a regularization function can be added to the loss by weight. It creates a penalty and can be chosen in such a way to keep an optional feature balanced.

**Dense blocks:** A dense block is a block of convolutional layers where all of the layers are connected to every layer in front to maximize information flow.

**Residual blocks:** A residual block is a block where the network learns the residual of the function concerning input. These blocks help to have larger information flow and avoid vanishing gradient problem.

So far, we have discussed very basic elements of CNN architecture that we need in this thesis (refer to [10] for more information).
Chapter 2

Synthetically-Hazed Aerial Image Dataset Generation
As discussed in chapter 1, there are many different dehazing methods from which the CNN-based ones need datasets to be trained on. Even though there exist several hazy image datasets publicly available, creating new datasets is still an active area of research [23]. The reason is the fact that different imagery domains have varying characteristics and consequently, the behavior of haze in the images diverges in each domain (see fig. 1.1). This is even valid for smaller groups of images. For instance, in the indoor case, the content of the image differs when taking images from the kitchen and bedroom. To the best of our knowledge, there is no hazy aerial image dataset available and all of the accessible datasets, include either indoor or outdoor images.

One may think of a possible aerial image dehazing using networks trained on the ground image datasets. Our experiments show that it works to some degree. However, due to the basic differences between these imageries, a domain adaption could lead to better results which are verified by our experiments. The objects in the ground imagery are relatively closer to the camera compared to the aerial case, resulting in a higher spatial resolution of the ground image than the aerial one. Besides, due to the wider field of view of the aerial images, the scene coverage in aerial imagery is often larger than the ground images.

Blurring effect is a common problem in aerial imagery as the cameras are moving most of the time on a flying platform, whereas in the ground imagery, blurring effect can be avoided. When flying over a city, the weather or atmospheric conditions may differ in different parts of the city, especially when flying over mega-cities. To be more clear, in the large cities like Tehran, it often happens that in the cold seasons, it snows quite heavily in the north part of the city while the south part remains rainy. This varying weather conditions is specific to aerial imagery and will not be seen in ground imagery case.

Furthermore, due to the difference in the viewing angles, the objects could look totally different in the two imageries. Additionally, the difference between the ratio of height (or depth) differences of the objects in the scene and their distance to the camera in the two imagery domains plays a key role in forming various haze conditions on each image type. For instance, in flights with 1000m altitude, the height difference between buildings and roads in the image would be around 5 meters on average in the cities with not much of skyscrapers. However, when taking a photo from a scene in a room, the ratio of height (or depth) differences and the distance to the camera is close to 1:1 (see fig. 2.1).

Taking into account all these deviations of the two imagery domains, we can conclude that an adapted network on aerial imagery may work better on aerial images. The adaption of the networks needs training data. In most of the cases [18, 29, 23, 20, 7, 14, 19, 30, 5], in order to create synthetically hazed images in the homogeneous atmosphere, the atmospheric scattering model is used (see section 1.2.1). As mentioned in section 1.2.1, in this model, depth is the key requirement to create hazy images (see eq. (1.2)). In indoor cases, such as the NYU depth V2 dataset, the depth of the scene has been captured using Microsoft depth cameras called Kinect [27, 1]. In outdoor cases such as the O-Haze dataset [2], the same scene is captured in normal (haze-free) and hazy conditions, where the haze in the image was produced using professional haze machines.

In the aerial imagery scenario, it is not practically possible to utilize haze machines and
obviously, having multiple flights in desired atmospheric conditions is not cost-efficient. Therefore, for our experiments, we create synthetically hazed aerial images both with homogeneous and in-homogeneous appearances, using the atmospheric scattering model. The crucial step is to provide the corresponding depth map of each image with the same size and same resolution. The depth map of homogeneous and inhomogeneous hazy images are being calculated differently. In the next sections, we will explain our assumptions and methodology to create synthetic-homogeneous and inhomogeneous hazy aerial images.

2.1 Synthetically-Homogeneously-Hazed Aerial Images

2.1.1 Depth Image Generation

As shown in eq. (1.2), the key to generate hazy images is the depth map. Assuming that the depth map is provided, given $\beta$ value, the transmission map ($t$) for each pixel can be calculated. Then, we put the transmission values into eq. (1.2) and for different values of $\lambda$, we construct different levels of hazy images.

In order to have the depth for each pixel, we need the 3D world coordinates of each pixel $(X, Y, Z)$ on the ground. Then, the depth or Euclidean distance of the object point to the principal point of the camera can be calculated using

$$d_{x,y} = \sqrt{(X_{x,y} - X_c)^2 + (Y_{x,y} - Y_c)^2 + (Z_{x,y} - Z_c)^2},$$

(2.1)

where $d_{x,y} =$ Depth of each pixel,

$(X_{x,y}, Y_{x,y}, Z_{x,y}) =$ 3D world coordinates of each pixel,

$(X_c, Y_c, Z_c) =$ 3D world coordinates of principal point.
eq. (2.1) shows that to have the depth of each pixel, it is necessary to have 3D coordinates of each pixel together with the 3D coordinate of the principal point of the camera in the same coordinate system, namely world coordinate system. There are two strategies to obtain the 3D position of each pixel on the ground:

1. The first one is to use DEM of the corresponding image. DEM is a geo-referenced raster data of the same size and resolution to the image containing ground height of each pixel. As it is geo-referenced, we have $X$ and $Y$ coordinates of the pixels as well as $Z$ which is stored in the pixel values of the DEM data.
2. As the second option, we can use the camera information such as focal length, flight height, and rotation angles. Together with the ground height of each pixel ($Z$) to calculate $X$ and $Y$ for each pixel (using well-known Collinearity Equation).

In our case, the DEM of the whole city is available, but we face the problem of matching the position and resolution of each image with its corresponding area on the whole DEM data. Hence, we need to crop each area of the DEM to cover each image location and match the spatial resolution of the DEM data with that of the image. Therefore, we use the second strategy to provide a depth map. As can be seen from fig. 2.1, the flight height is significantly larger than the height differences of objects e.g. buildings on the ground. Consequently, when computing the Euclidean distance, there is no significant difference in the depth of the neighboring objects. Thus, we assume that all the objects on the ground to have the same height namely, the average height of the region.

According to fig. 2.1 we have:

$$\Delta H \ll d(x).$$

Therefore,

$$H_i = Z_{avgr}, \forall H_i,$$

where $H_i =$ Ground height of the image pixels,

$Z_{avgr.} =$ The average ground height of the region.

Considering the ground height of all pixels in image similar to each other, i.e., the average height of the region, $X$ and $Y$ ground coordinates of the pixels can be computed by

$$X_{x,y} = X_c + (Z_{x,y} - Z_c) \frac{r_{11}(x - x_c) + r_{21}(y - y_c) - r_{31}c}{r_{13}(x - x_c) + r_{23}(y - y_c) - r_{33}c},$$

$$Y_{x,y} = Y_c + (Z_{x,y} - Z_c) \frac{r_{12}(x - x_c) + r_{22}(y - y_c) - r_{32}c}{r_{13}(x - x_c) + r_{23}(y - y_c) - r_{33}c},$$

where $(x_c, y_c) =$ pixel coordinates of principal point,
\( r_{11}, r_{12}, \ldots, r_{33} = \) Rotations of the camera at the capture time,
\( c = \) Focal length of the camera,
\( Z_{x,y} = Z_{\text{avg}}, \forall x, y \) in the image.

Since all camera parameters are available for the images, by taking the average altitude of the region as the height value for all pixels, we can calculate the world coordinates of each pixel and use them in eq. (2.1) to obtain the depth map.

### 2.1.2 Homogeneously-Hazed Images

Now by having the corresponding depth maps, we can insert different values for \( \beta \) and \( A \) in eq. (1.1) and create different hazy images. Since the depth in our images (around 1000 m) is very different from that of ground imagery, we set different values as compared to the ground imagery scenarios for the medium transmission and global atmospheric light. In our case, we use \( \beta = \{0.0005, 0.001, 0.0015, \ldots, 0.002\} \) and \( A = \{230, 240, 250\} \).

By assuming the atmosphere to be homogeneous and the heights to be similar, the haze on the images appear homogeneously. This condition holds in most of the images collected from different flight campaigns. This is valid for the cases where there is no pollution source around because when this is not the case. For example, when there is a factory in the area, the smoke coming out of the smokestack of the factory causes the density of the haze or pollution to be greater in some parts; therefore, the assumption of homogeneous atmosphere or haze does not hold anymore. This applies also for the flights over mountains and valleys, which there might be haze on the valley, but on top of the mountain is clear, making the haze not homogeneous. In fig. 2.2, a haze-free image on the top, together with nine hazy versions of it is depicted.

The training images in the aerial hazy image dataset contain hazy images for a single image in 9 different levels. In our experiments, 140 images are chosen as training and 17 images as test sets. Flight height is around 1600 m above the sea level, where the average ground height is 600 m. The viewpoints are almost nadir-looking and the images consist of urban, rural, and some forest areas. The camera in this flight is a hyperspectral camera with 30 cm ground spatial resolution and 88 mm focal length.
2.1 Synthetically-Homogeneously-Hazed Aerial Images

Figure 2.2: Synthetically and homogeneously hazed aerial images. The image on top is the ground truth image which is assumed to be haze-free. From the second row, left to right the haze density increases in the images.
2.2 Synthetically-Inhomogeneously-Hazed Aerial Images

In the previous section, we mentioned that the height differences of the objects in the scene are relatively small in comparison to their distance to the camera which allows us to assume the same height for all objects. Taking this condition as a start point, the assumptions and computations are used in order to insert homogeneous haze on our images. However, in the real world, haze is not always homogeneous. When there is a polluting source such as flowing smoke in the area close by a factory or when there is humidity in the area with a varying density over the scene, haze could inhomogeneously influence images. As can be seen from the right image in fig. 2.3, the haze density does not only depend on the depth of each pixel but also depends on the density of particles through the light beam received by the camera. Another case where inhomogeneous haze can be seen in the image is where the pictures are taken from a mountain and valley. Sometimes in these areas, there is a moving haze mass (mist or cloud) which is covering only one part of the image while the other parts are clear as shown in the left image of the fig. 2.3.

Since we are using the atmospheric scattering model (see eq. (1.1)) to add haze on our images and this formulation is modeling the haze formation with the precondition of homogeneous atmosphere, it is not possible to change the atmospheric conditions to be inhomogeneous. Therefore, we need to consider a different strategy to put an inhomogeneous haze on the aerial images. In practice, there are two parameters which can define the density of haze through the light beam received by the camera:

- The atmospheric condition through the distance traveled by the light beam.
- The depth of the point (the distance traveled by the light beam to reach the camera).

---

2.2 Synthetically-Inhomogeneously-Hazed Aerial Images

Figure 2.4: Random DEM generation

Raster with zero values

Random numbers inserted in intervals

Cubic spline interpolation for grid data

Generated random digital elevation model
Since the first condition is fixed in eq. (1.1), we replace the depth of the points from the real values with values driven from randomly generated DEMs. The reason to use randomness in our depth maps is that the haze can appear anywhere on the image with a varying density. In the case of using a traceable pattern in our inhomogeneous hazy data generation process, it may influence the network to overfit to our data. Useful to mention that the generated random depths should be as smooth as a terrain topography to avoid the noisy appearance of the haze on the image.

In order to create random DEM, an empty raster grid with the same size of the source aerial images ($m \times n$) is taken, where in our case, $m = 3744$ and $n = 5616$. Its pixel values are initialized all to be zero in the first step (see fig. 2.4). As the second step, the four corner pixels of the empty image takes random values ($r_{11}$, $r_{1n}$, $r_{m1}$, $r_{mn}$). Also in the determined interval distances $i$, random numbers are used to put into pixel values ($r_{12}$, $r_{13}$, etc.). The values in between are then computed using cubic spline interpolation [15].

As can be seen from the fig. 2.5, some data points are taken as given values which are illustrated in red dots called knots. The values within consecutive data points are computed using cubic spline interpolation. This interpolation strategy fits a 3rd order polynomial between each neighboring point pair as:

$$F(x) = \begin{cases} 
a_1x^3 + b_1x^2 + c_1x + d_1, & x \in I \\
a_2x^3 + b_2x^2 + c_2x + d_2, & x \in II \\
a_3x^3 + b_3x^2 + c_3x + d_3, & x \in III \\
a_4x^3 + b_4x^2 + c_4x + d_4, & x \in IV 
\end{cases}$$

(2.6)
The coefficients of these functions, are found using constraints as below:

- The internal knots should pass through both of the functions of their neighboring intervals.
- First and Last functions should pass through the start and end knots.
- The first and second derivatives of two neighboring functions should be equal at internal knots.
- The first derivatives of two neighboring functions should be equal at internal knots.
- The third derivative at the first point should be equal to zero.

We have applied the cubic interpolation algorithm to our data and then quantized them into gray values. Despite the smoothness of the interpolation results and since naturally there are no sharp edges or boundaries for haze in the air, we have applied a Gaussian filter to our random DEM as well to avoid line like patterns in our depth results. Same as the process explained in section 2.1.1, the interpolated DEM is then inserted into the collinearity equation (eq. (2.4) and eq. (2.5)) to obtain the position of each pixel in the world coordinate system and then the depth maps are generated using eq. (2.1). Some examples of randomly generated depth maps are shown in fig. 2.6.

The intervals of random number assignment and the Gaussian filter parameters are chosen empirically to have natural-looking hazy images. The Gaussian filter size is 400 and the intervals set to be 1000.

The same 140 source images are used for the inhomogeneous data generation. Each image has one corresponding random depth map and is hazed in 9 levels. As the random depths have a similar range to the depths in the homogeneous case, the hyper-parameters \((A\) and \(B\)) remain the same in this process as well. In fig. 2.6, some examples of inhomogeneously hazed images are illustrated.
Figure 2.6: Synthetic-inhomogeneous hazy images. Images in the left column are ground truth images and assumed to be haze-free. The random depths and hazed images are shown in the second and third column respectively.
Chapter 3

Methodology
3.1 Introduction to AOD-Net

In this chapter, we will give an outline of the methodology we employ to adapt a ground imagery-based dehazing network to the aerial imagery domain. The chapter also describes the various setups we took to come up with our proposed network.

As already mentioned in chapter 1, we choose AOD-Net as benchmark dehazing network for our experiments. This network composed of some convolutional layers together with several skip connections that support the information flow through the network. The simple structure of the network allows us to modify it easily. AOD-Net also follows the physical model of the haze formation in estimating the dehazed image from a single hazy input image. Therefore, it is well suited to the criteria of our generated data. We explore the network structure and its original training process in section 3.1. Later, we discuss the theoretical aspects of the experiments we implement in order to adapt the network from ground imagery to the aerial domain in section 3.2 and track our path from the experiments, results, and analysis, all the way down to the newly developed technique.

3.1 Introduction to AOD-Net

AOD-Net is a lightweight CNN-based method to dehaze single hazy images [18]. It is based on reformulating the equation of the atmospheric scattering model (see eq. (1.1)). In contrary to the prior-based dehazing algorithms [14, 5, 31] that estimate transmission matrix \( t_{x,y} \) and atmospheric light \( A \) in the first step from which the dehazed image will be recovered, AOD-Net reformulates eq. (1.1) to unify two aforementioned parameters and directly estimates the dehazed image. For this purpose, a new parameter \( K_{x,y} \) is defined as:

\[
K_{x,y} = \frac{1}{t_{x,y}} \left( I_{x,y} - A + (A - b) \right) \left( I_{x,y} - 1 \right)
\]

(3.1)

where

- \( I = \) Received radiance of a hazy image,
- \( b = \) Constant bias equal to 1,
- \( t = \) Medium transmission,
- \( A = \) Global atmospheric light,
- \( x, y = \) Pixel location.

The parameter \( K_{x,y} \) contains both \( t_{x,y} \) and \( A \) parameters. The core idea is to minimize the estimation error of the two parameters at the same time so that the network does not adapt itself only to one parameter during training. The dehazed image \( J_{x,y} \) is then restored using:

\[
J_{x,y} = K_{x,y} I_{x,y} - K_{x,y} + b
\]

(3.2)

In the implementation phase, the network consists of two modules. First, there is a CNN block estimating the \( K_{x,y} \) value, and then revisit the haze-free image from the estimated \( K_{x,y} \) using eq. (3.2). The network structure is illustrated in fig. 3.1.
The $K_{x,y}$ parameter is estimated as shown in fig. 3.1. This block consists of five convolutional layers and has varying filter sizes to create a multi-scale feature extraction form. The input hazy image goes through a $1 \times 1$ convolution layer, resulting in the first feature map. The first feature map is convolved with a $3 \times 3$ convolution layer and forms the second feature map. The first and second feature maps are then concatenated and inserted into a $5 \times 5$ convolution layer ending up to the third feature layer, which later is concatenated with the second feature layer and placed into a $7 \times 7$ convolution layer, creating the fourth feature layer. Finally, all four feature maps are concatenated and fed into the last $3 \times 3$ convolution layer. The output of the last operation will be the estimated $K_{x,y}$ matrix which then will be inserted in eq. (3.2) and provide the estimated haze-free image.

To train the dehazing network, there should be hazy images from the scene as well as the haze-free or ground truth images from the same scene under the same conditions (illuminations, light, camera positions, etc.) available. As discussed in section 1.2.1, the depth of each pixel is a prerequisite to generate synthetic hazy images. There are several image datasets available that provide images from different indoor scenes and the corresponding depth image which is mostly produced using laser scanners, stereo-matching techniques or haze generator machines. The authors of AOD-Net have used the well-
3.2 Adapting AOD-Net on aerial imagery

In the way to find an effective and well-performing dehazing algorithm on aerial imagery, we first dehazed a number of natural and synthetic hazy images with AOD-Net trained on NYU depth V2 dataset. For some images, the haze was not completely removed while for others over-enhancement effects were observed.

In order to improve the results, several new trainings are conducted on the aerial image dataset generated in chapter 2. During the new set of trainings, we try different setups with varying hyper-parameters such as image size, batch size, size of the training set while keeping the main configuration namely the loss, learning rate, setup of convolution layers similar to the original training in [18]. After several trials, we find the best performing training setup and manage to improve the PSNR and SSIM up to 2.2 dB and 9% respectively. The detailed description of the different setups and the achieved results will be further explained in chapter 4.
Figure 3.3: A sample aerial image is hazed and dehazed using AOD-Net trained on ground and aerial imagery datasets. The color histogram of the images show that synthetic haze appears as a color shift on image. It also shows that the adapted version is more robust to the color histogram suppression artifact. The PSNR and SSIM for fig. 3.3e are 10.3 dB and 17.5 % and for fig. 3.3g are 11.9 dB and 29% respectively.
3.2 Adapting AOD-Net on aerial imagery

While quantitative results are satisfactory, we notice an undesired artifact at the test time on our dehazed images. In fact, over-enhancement or color histogram suppression is often a mutual problem when dehazing the images [23]. Even though the CNN-based algorithms are trained to minimize the distance of the dehazed image and ground truth, the networkovershoots on some of the test images. By analyzing the histogram of the dehazed results, we can observe that despite the color histogram suppression effect is reduced after training on aerial images, it was not removed completely. Therefore, extra modifications are needed to regularize the network.

To modify the network and inspired by the state-of-the-art CNNs, we add different options like batch normalization, drop out layers, convolutional layers, and fully connected layers. As can be seen from fig. 3.4, the image structure is damaged after the above trials. In some cases, the dark area of the images which are mainly bushes or shadows are turned to be black (see fig. 3.4k). In the images taken from cities, the roads and roofs of the building are turned to be white, the green area is well dehazed though (see fig. 3.4i). An undesired color shift can also be seen in the results of the dehazed images when using extra convolutional and fully connected layers in the network (see fig. 3.4j).

For the additional modifications on the network, we focused on the loss function. We realized that directly minimizing the $\text{MSE}$ of the pixel values should work locally and is insufficient to prevent the color histogram suppression effect. However, the loss should keep the structure of the image not only locally but also globally. Therefore, we find it useful to transform our training process from spatial domain to frequency domain which removes the local dependency to the pixel values.

The first trial was to directly put the loss to be the $\text{MSE}$ of the images when transformed to the Fourier domain. In this way, although the results are much better than those of achieved by the previous modifications and the image structure remains undamaged, they are affected by a color shift artifact as can be seen from fig. 3.4l. Now that the global structure of the image is well preserved, we added the pixel value information ($\text{MSE}$ of the pixel values) to our loss maintain the color information. We defined our new loss function as the weighted sum of the $\text{MSE}$ between the dehazed and ground truth image in spatial domain and the frequency domain:

$$\text{Loss} = \text{MSE}(I_s, G_s) + \alpha \text{MSE}(I_f, G_f)$$

(3.3)

where $I_t =$ Dehazed image in spatial domain,
$G_t =$ Ground truth image in spatial domain,
$I_f =$ Dehazed image in frequency domain,
$G_f =$ Ground truth image in frequency domain,
$\alpha =$ Weight.

The weight $\alpha$ is selected in the range of 0.01 to 0.2 and the best results are achieved with $\alpha = 0.01$. A significant jump of 1.04 dB on the $\text{PSNR}$ and 7.6% on the $\text{SSim}$ of the test set can be seen when dehazed with the proposed method. The improvement in the
Figure 3.4: Samples of the damage on images when dehazing with new setup trials. The first and second row are the ground truth and the hazed images respectively. The thirds row shows the damages dehazed images where various options added to the network.

color histogram suppression effect can also be qualitatively and quantitatively seen from the image and KLD. Details on the experiment and sample results are further discussed in chapter 4.
Chapter 4

Experiments and Results
We explain our experiments in detail and discuss the results in this chapter. The chapter is divided into two main sections: In section 4.2, we describe how we adapt the AOD-Net to the aerial imagery domain by training the network on our aerial hazy image dataset and find the best setup. To this end, we perform qualitative and quantitative comparisons on performances of the adapted and the original network which is trained on indoor images. Later in section 4.2, we propose a new loss function which helps to balance the color distortion caused by the dehazing process.

For the quantitative evaluations, we employ two widely-used indices namely: PSNR and SSIm between the dehazed image and ground truth [18, 19, 7, 30, 23]. Additionally, Their color histograms are compared using KLD.

### 4.1 Evaluation Indices

PSNR [12] is the ratio between the maximum possible value of a signal and the power of noise affecting the signal. This ratio is often represented in logarithmic form because the range of the aforementioned values can be highly different.

\[
PSNR = 10 \log \left( \frac{(\max(I))^2}{MSE} \right),
\]

where \( I \) = Pixel values for dehazed images,
\( MSE \) = Mean Squared Error between dehazed and ground truth images.

The MSE between dehazed (I) and ground truth image (G) can be calculated using the equation below:

\[
MSE = \frac{1}{mn} \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} \|I(i,j) - G(i,j)\|^2,
\]

where \( G \) = Haze-free ground truth image,
\( i, j \) = Pixel locations over the images.

For color images, the MSE is taken over all pixel values of every channel and then is averaged. The PSNR ranges between 0 and \( \infty \). The higher the value of PSNR, the higher the ratio of signal to the distorting noise, therefore the closer get the images to each other.

As an improvement to PSNR and MSE, the index of SSIm [12] is proposed at Laboratory for Image and Video Engineering (LIVE) at the University of Texas, Austin. Since human visual system is sensitive to extract structures, SSIm is developed based on the degradation of the structural information [28].

\[
SSim(i, j) = \frac{(2\mu_I\mu_G + C_1)(2\sigma_{IG} + C_2)}{(\mu_I^2 + \mu_G^2 + C_1)\sigma_{IG}^2 + \sigma^2_G + C_2},
\]

where

\[
\mu_I = \frac{1}{mn} \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} I(i,j), \\
\mu_G = \frac{1}{mn} \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} G(i,j), \\
\sigma_{IG}^2 = \frac{1}{mn} \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} (I(i,j) - \mu_I)(G(i,j) - \mu_G),
\]

\( C_1 = \alpha \frac{\max(I)}{\mu_I}, \quad C_2 = \alpha \frac{\max(G)}{\mu_G}, \quad \alpha = \min(1, \max(\frac{\mu_I}{\max(I)}, \frac{\mu_G}{\max(G)})) \)
where \( \mu_I, \mu_G \) = The average pixel values in images, 
\( \sigma_I, \sigma_G \) = Variances of pixel values in two images, 
\( \sigma_{IG} \) = Covariance of pixel values in two images.

In eq. (4.3), the variables \( C_1 \) and \( C_2 \) are stabilizers in order to compensate for the division to a weak denominator. They can be computed using:

\[
C_i = (k_i L)^2, \tag{4.4}
\]

where \( i = 1, 2, \)
\( k_1 = 0.01, \)
\( k_2 = 0.03, \)
\( L = \text{Range of pixel values, (for 8-bit images } L = 255). \)

The SSIm can be in range \([-1, 1]\). It equals 1 when the two images are identical. We would use this index in percentage in our experiments.

The aforementioned indices are useful when comparing two images directly for their pixel values. In order to evaluate the color distribution alterations due to the hazing and dehazing process, we plot the color histogram of the images and compare them using KLD [9].

The KLD defines how much one probability distribution differs from the reference one. In our case we have normalized the histograms to be in the range of 0 and 1 and utilized the discrete KLD in order to compare the two distributions. The formulation is as:

\[
D_{KL}(P \parallel Q) = - \sum P(x) \log \left( \frac{Q(x)}{P(x)} \right), \tag{4.5}
\]

or alternatively:

\[
D_{KL}(Q \parallel P) = \sum P(x) \log \left( \frac{P(x)}{Q(x)} \right), \tag{4.6}
\]

where \( P(x), Q(x) = \text{Discrete distributions}. \)

Note that the reference image in all of our comparisons is the ground truth image. In another words, the dehazed images are evaluated depending their SSIm to the ground truth image. Also the histogram of the ground truth image is used as reference distribution in KLD.

### 4.2 Experiments Of Network Adaption On Aerial Imagery

We go through different experiment setups until we reach the successful case to adapt and improve the performance of AOD-Net on aerial images. The experiment setups are shown in table 4.1.
The total number of images we have from the flight over Oberpfaffenhofen, Bayern, is 674. In our first experiment, we use all of our images with their original size of 5616 × 3744 px in 3 haze levels (2022 hazy images in total). They are inserted into the network together with their ground truth versions for the training purpose. Even though the training process is done with the batches of two images, it is very slow and also GPU memory demanding. As we desire a time-efficient network, we simply change the modifications.

In the next setup, we crop our images to be 512 × 512 px, the haze levels are increased to 9 and the batch size rises to 32. Each image results in 81 cropped image, and in order to speed up our experiment by avoiding an enormous amount of data (674 × 81 × 9 = 401,346), we decrease our source images and set two new configurations:

- 40 source images, 9 levels of haze, resized to 512 × 512 px, batch size of 32, total number of images 29,160.
- 140 source images, 9 levels of haze, resized to 1024 × 1024 px, batch size of 32, total number of images 30,240.

<table>
<thead>
<tr>
<th>Setups</th>
<th>Flight Campaigns</th>
<th>Haze Levels and Model</th>
<th>Source Images</th>
<th>Image Size (in pixels)</th>
<th>Total Image Number</th>
<th>Batch Size</th>
<th>Success</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>same</td>
<td>3-H³</td>
<td>674</td>
<td>5616 x 3744</td>
<td>2,022</td>
<td>32</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>same</td>
<td>9-H</td>
<td>40</td>
<td>512 x 512</td>
<td>29,160</td>
<td>32</td>
<td>✓</td>
<td>14.89</td>
<td>52</td>
</tr>
<tr>
<td>3</td>
<td>same</td>
<td>9-H</td>
<td>140</td>
<td>1024 x 1024</td>
<td>30,240</td>
<td>32</td>
<td>✓</td>
<td>16.28</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>not same</td>
<td>9-H</td>
<td>40</td>
<td>512 x 512</td>
<td>29,160</td>
<td>32</td>
<td>✓</td>
<td>11.94</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>same</td>
<td>9-InH⁴</td>
<td>140</td>
<td>1024 x 1024</td>
<td>30,240</td>
<td>32</td>
<td>✓</td>
<td>10.75</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 4.1: Different experiment setups in training the network to be adapted on aerial images

These setups are designed in such a way to keep the training configuration of the original AOD-Net for a better comparison. As mentioned in section 3.1, the original AOD-Net is trained on 1445 images, hazed in 21 levels with the image size of 480 × 640 px [18].

¹In dB
²In %
³Homogeneous haze
⁴Inhomogeneous haze
We use Titan X GPU in our experiments. It takes around 5 days to train the network on around 27,000 training images and validate on 3,000 images for 10 epochs with the learning rate of 0.0001. The loss function of the network is mean squared error as in the original AOD-Net.

We create a test set for the comparison of the trained networks. The test set includes 17 images from different flight campaigns, collected from different years and locations in Germany. The last configuration which is shown in red in table 4.1 has the best PSNR and SSIm on the test set. It also outperforms the original AOD-Net which is trained on NYU depth V2 dataset.

We design a new dataset including source images from different flight campaigns to enable the network to perform on a varying range of depth. This setup does not improve the results, because it is not just the flight height of the images which changes, but also other features differ in different flight campaigns:

- Seasonal change which consequently results in color changes in vegetation such as farmland, trees, forests and generally. Additionally, a scene can look different on rainy or snowy days.

- Land-cover and land-use (e.g. the scene from rural to urban areas) can completely change the object types exciting in the image, such as buildings that can be seen mostly in the cities. Also, different types of lakes or rivers which appear in some specific areas in the image.

- The viewing angle of the camera can change the appearance of the scene (left, nadir or the right look).

- Flight height ranging from 2400 m to 940 m. It is not only the height of the plane which is not stable in different campaigns, but also the average height of the ground can be very dissimilar. Additionally, looking at each image individually, in areas where there are mountains and valleys at the same scene, the assumption of eq. (2.3) that we used to create our datasets will not work anymore. In other words, the distance of the maximum and minimum height in the image will be too large to be neglected.

Considering the aforementioned variations in different flight campaigns, and due to the limited number of source images for the sake of processing efficiency, the network is not able to learn the dehazing task properly with such diverse features. In the case of a complicated task, the network needs more training data and time to learn all the features and perform well. That is why the setup with the images from different flight campaigns failed to perform well on the test set.

The last setup we use to train the network uses the same configurations such as source image number, batch size, image size, etc. as experiment number 3 (see table 4.1). The only difference between these two setups is the haze which is inserted inhomogeneously. The
4.2 Experiments Of Network Adaption On Aerial Imagery

<table>
<thead>
<tr>
<th></th>
<th>AOD-Net Original</th>
<th>AOD-Net Adapted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR (in dB)</td>
<td>14.08</td>
<td>16.28</td>
</tr>
<tr>
<td>Average SSIm (in %)</td>
<td>50</td>
<td>59</td>
</tr>
</tbody>
</table>

Table 4.2: The average PSNR and SSIm of the test set when tested on original and adapted AOD-Net

quantitative and qualitative results do show an improvement in the test set which can be expected.

Since the inhomogeneous haze is more complex comparing to the homogeneous case, it needs more time and data for the network to be trained.

As a sample performance of the adapted network, we illustrate some synthetically hazed images together with their dehazed versions using original AOD-Net and adapted AOD-Net in fig. 4.1. In the first row, we have the ground truth images which are assumed to be haze-free. The second row includes the same images in various levels of hazy versions. The third row includes the dehazed images using the original AOD-Net and in the last row, the dehazed images using the adapted network are shown.

Applying adapted and original network to the images in our test set, we achieved the overall results as shown in table 4.2. According to the table, adapting the network improves the dehazing performance by 9% in SSIm and 2.2 dB in PSNR.

As can be seen from fig. 4.2, in both sets of dehazed images, the histogram suppression effect can be noticed soon. It is the case for almost all images in our test set. Observing the outputs of the state-of-the-art dehazing methods, color distortion of the dehazed images are one of the most common side effects of dehazing algorithms [23]. Two criteria can be focused by dehazing algorithm:

- Dehazing is to process a hazy image to look as sharp and clear as possible. In this case, the color histogram suppression effect of the network is an advantage.

- Dehazing is to reduce the distance of the output image to the ground truth image. In this case, the color histogram suppression effect is a drawback.

The contradiction in dehazing algorithms is the fact that even the images which we assume to be haze-free, can contain a small amount of haze. So, it remains an open question whether the over sharpening is an advantage or drawback to the network.

We put our focus on the second approach and further study the color histogram of the ground truth, hazed and dehazed images as illustrated in fig. 4.2. It can be seen that the histogram of the hazed image is narrowed and shifted to the front, which indicates an increase in the number of bright pixels and a decrease in the color contrast of the image. The histograms of the dehazed images for both original and adapted networks are suppressed to the darker zone which shows the color histogram suppression effect. The effect is a bit smaller for the adapted case. Additional to our observations from the histograms, we compute the KLD of the dehazed image histograms with those of the ground truth. As
KLD is a distance indicator, the outperforming results of the dehazed network can also be seen from this index, where the dehazed image of the adapted network has less distance to the ground truth one and is closer to it.

To overcome the aforementioned artifact and improve the performance of our dehazing network, many rearrangements can be considered to implement on the network, for example:

- Adding batch normalization
- Changing the learning rate
- Adding dropout layers
- Using different skip connections between the layers
- Employing various loss functions

As the first steps, we try the aforementioned modifications in different setups. Either the problem remains the same or we faced other problems like color distortion, blurring, ringing effects.

Focusing on the loss of the network, which is keeping the distance of the dehazed image to the ground truth one as less as possible, we come up with the possibility that MSE loss may not be sufficient. To be more precise, the MSE loss tries to keep each pixel of dehazed image as close as possible to the ground truth one. It works too locally and may not be representative of the final desired aim of the network. Therefore, we utilize another space in which, an image can be presented and processed. That clarifies our motivation to try processing the images in the frequency domain rather than the spatial domain.

We first try to minimize the distance of the images in the frequency domain. Otherwise stated, instead of MSE of the pixels, we define the loss function to be the MSE of the magnitude spectrum of the dehazed and ground truth images. As a result of this experiment, we see a logical pattern on our dehazed images. Based on the results, we propose a new method and defined the new loss function as eq. (3.3). The best results are acquired from $\alpha = 0.01$. The network is trained with the same configuration as the experiment 3rd from table 4.1, but the proposed loss is used. The trained network is tested on the test set of 17 images similar to previous experiments and the overall PSNR and SSim results are shown in table 4.3. There can be seen a significant jump on the results with 7.6% in SSim and 1.04 dB in PSNR.

The fig. 4.2 gives a general overview of the achieved results so far. The ground truth, hazy images, and their corresponding histograms are shown in the first row. As mentioned earlier in this section, the hazed image has less color contrast which defines the compression of its histogram. It also shifts to the front due to its relatively brighter pixels. The second row shows the dehazed images using original and adapted AOD-Net. As can be read from the caption, the PSNR and SSim indices of the dehazed image using the adapted network, shows an increase. It can be realized from their histograms that the suppression
4.2 Experiments Of Network Adaption On Aerial Imagery

Figure 4.1: Ground truth images, together with their hazy and dehazed versions. The first row: Haze-free images, the second row: Hazed version of the same images in different levels, the third row: Dehazed images using AOD-Net which is trained on NYU dataset, and the fourth row: Dehazed images using adapted AOD-Net
Table 4.3: The average PSNR and SSim of the test set when tested on adapted AOD-Net and proposed network

<table>
<thead>
<tr>
<th></th>
<th>AOD-Net Adapted</th>
<th>Proposed Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR (in dB)</td>
<td>16.28</td>
<td>17.32</td>
</tr>
<tr>
<td>Average SSim (in %)</td>
<td>59</td>
<td>66.6</td>
</tr>
</tbody>
</table>

effect of the adapted network is smaller than the original one. As a quantitative comparison index, the KLD of the dehazed images are computed concerning the ground truth image. The KLD of the dehazed image using the adapted method is less than the original one, meaning that the distance of its color distribution is less to the ground truth, and therefore they are closer. The last row in the figure belongs to the dehazed image using the proposed network and its color histogram. The PSNR and SSim of the dehazed image is notably greater than other dehazed images. Additionally, its color histogram follows a much similar pattern to the ground truth one. It can be proved by KLD of the dehazed image to ground-truth one, which is remarkably less than the other two results.

The positive effect of the new loss can also be noticed when trained on the inhomogeneous hazy images. As our last experiment, we trained the AOD-Net on inhomogeneously hazed images using the original structure and with the new loss function and compared the results which are shown in table 4.4.

Table 4.4: The average PSNR and SSim of the test set when tested on adapted AOD-Net and proposed network, both trained on inhomogeneously hazed data set

<table>
<thead>
<tr>
<th></th>
<th>AOD-Net classic</th>
<th>Proposed Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR (in dB)</td>
<td>10.75</td>
<td>12.5</td>
</tr>
<tr>
<td>Average SSim (in %)</td>
<td>40.17</td>
<td>47.10</td>
</tr>
</tbody>
</table>

According to the results, the modification improve the dehazing results by 6.93% in SSim and 1.75 dB in PSNR (see table 4.4).

The last experiment is done on an aerial image that is originally hazy and with no haze-free version. The image is captured in 2006 from the Alienz Arena Stadium in Munich. In the fig. 4.3, the natural hazy image together with three dehazed versions are illustrated. It can be noticed that the color attenuation effect in the dehazed image with adapted AOD-Net is less than the original version and the best qualitative results are achieved by the proposed network. The improvement can be noticed when focusing on the shadows.

The shadow areas are mostly dark in the image, during the color histogram suppression effect, they tend to seem black. This black color in the shadow areas leads to an information loss which can reduce the performance of image analysis tasks. Therefore, it is very crucial to keep the information in the shadow areas, as undistorted as possible.

When the proposed network is tested on the test set, the quantitative results show an improvement comparing to the original and even adapted network. However, a cartooning
4.2 Experiments Of Network Adaption On Aerial Imagery

Artifact is sometimes noticed in some of the dehazed images as can be seen in fig. 4.2.

To sum up, we manage to adapt a ground imagery dehazing method (AOD-Net) to the aerial domain by training it on the generated hazy aerial image dataset. We further modified the network and improved the loss to compensate for the color histogram suppression effect on the dehazed images. The results of the adapted and proposed network are then compared to the original version of the Mojgan MadadikhaljanAOD and achieved an overall improvement in PSNR and SSIM by 3.24 dB and 16.6% respectively.

As future projects, the training on the inhomogeneous dataset can be improved by more images and epochs. For each image different random DEMs can be generated as well. The inhomogeneous data can also be used to fine tune the homogeneous network.

As in the original AOD-Net, the network can perform better when directly inserted onto the image analysis task directly, so that the network will be jointly trained with the objective task.

A large-enough dataset can also be created using images from different campaigns including enough data from all differing features such as seasons, land-use.
Figure 4.2: Sample of the improvement of the adapted and proposed network with respect to original network.
4.2 Experiments Of Network Adaptation On Aerial Imagery

Figure 4.3: A sample aerial natural hazy image and its dehazed versions. The shadow areas are zoomed to better illustrate the color suppression effect.
Chapter 5

Conclusion and Future Works
Aerial images have a wide variety of applications due to their high resolution and large field of view. They are often used in cartography, surveying, disaster management, image recognition, classification, traffic monitoring, change detection, sustainable city planning, agriculture, etc. Despite the significant improvement in the quality of this imagery by employing modern cameras, there are remaining artifacts to be removed by enhancement techniques. Atmospheric condition at the time of capture is one of the factors which highly impact aerial image quality. The presence of the haze as a common atmospheric phenomenon can also damage the image quality and decrease the performance of image analysis systems. Haze as floating aerosols such as dust, water droplets, and smoke can degrade the visibility and transparency drastically. It can also obstacle the clearness of the objects in an image, leading to a color shift and consequently, has a negative effect on the performance of the automatic image analysis tasks like image recognition, classification, and tracking.

To overcome this undesired phenomenon, a large amount of prior knowledge-based and CNN-based dehazing methods focusing on ground imagery have been developed so far, while there is a lack when it comes to the aerial imagery domain. To address this issue, we developed a well-performing CNN-based dehazing method on aerial imagery in this project.

The core idea was to select an outperforming ground imagery dehazing method and adapt it to the aerial imagery. To do so, among the studied state-of-the-art ground image dehazing algorithms, we chose the well-known All-in-One Dehazing Network (AOD-Net). The motivation behind was that AOD-Net had superior results in most of the dehazing comparisons and directly recovers the haze-free image from a single input hazy image without any intermediate parameter estimation. It consists of a very simple network structure with only 5 convolutional layers making it more flexible for further modifications. Furthermore, as it is a CNN-based method, it has the potential to be trained on different imagery groups and therefore be adapted to various domains.

As a key element for the adaption procedure, we needed an aerial hazy image dataset which as far as we know, there is no publicly available one. Therefore, we created a synthetically-homogeneously and inhomogeneously hazed aerial image dataset with the images taken from DLR flight campaign of 2019 Oberpfaffenhofen, Germany.

As the physical model of the haze formation in the images, we used the atmospheric scattering model in which the depth data is a prerequisite. Depth is defined as the Euclidean distance of each pixel to its object point on the ground and to be computed, we need the coordinates of each pixel and its corresponding ground object point in the same coordinate system. To this end, the collinearity equation was employed to acquire the X and Y of the pixel points in the world coordinate system. The ground height of the object points as the input to the collinearity equation was computed using two different strategies. For the homogeneous scenario, the height of all object points is assumed to be the same as the average height of the region. It is because the flying height over the object is so high (1000 m) that we can neglect some meter (5 m in average) difference in the depth calculation. For the inhomogeneous case, we proposed a new method to generate random
Digital Elevation Model (DEM)s and insert them into the depth calculation process.

For the adaption purpose, the network is trained on the aerial image as well as NYU depth V2 ground image dataset. The trained networks are then tested on an aerial image test set. The dehazing performance of the network which was trained on the aerial images showed an improvement of 9% in Structural Similarity (SSim) and 2.2 dB in Peak Signal to Noise Ratio (PSNR).

As a common artifact in dehazing processes, the undesired color histogram suppression was improved to some extent, but not completely removed. Thus, further modifications are applied to overcome the aforementioned artifact. Inspired by the other existing dehazing networks, we tried different modifications by adding layers such as convolutional, dropout, fully connected layers. We also added batch normalization to our process but failed to get satisfying and improved results where the structure of most of the dehazed images was damaged severely and quantitative indices did not record any improvement.

Finally, we put our focus on the loss of the network. The loss of the original network was the Mean Squared Error (MSE) of the pixel values of the dehazed and ground truth image. Although it was performing efficiently to some extent, the procedure was working too locally and we came up with the idea that a domain transformation may avoid this pixel dependency during training. To this end, we consider the loss to be the MSE of the dehazed and ground truth image in the frequency domain. The dehazed results were quite satisfactory in the manner of preserving the structure of the image after being dehazed. The only remaining problem was a shift in the color information which is then compensated by inserting previous loss function to the proposed one and find the best linear combination of them. The best weighted sum of the two loss-functions then are tested on the test set and improvement of 7.6% in SSim and 1.04 dB in PSNR is achieved. The proposed network is also tested on natural hazy images and showed outperforming qualitative results in comparison to the adapted and original AOD-Net networks.

To conclude, we start from a ground imagery-based dehazing method and managed to successfully adapt it to the aerial domain. During this process, a synthetic hazy aerial image dataset is generated as training data. After the training on aerial images, we further modified the network to compensate for the color histogram suppression artifact. We managed to improve the overall PSNR and SSim of the dehazing performance up to 3.24 dB and 16.6% respectively.

As the following projects, the network can be inserted into an image analysis task such as classification and jointly trained on the target dataset. For the inhomogeneous hazy images, additional images or learning time can be considered to upgrade its dehazing performance.
Glossary

AI  Artificial Intelligence. 10
AOD-Net  All-in-One Dehazing Network. ix, x, 4–6, 8, 29–32, 37–42, 44–47, 51, 52

CGAN  Conditional Generative Adversarial Network. 7, 8

CNN  Convolutional Neural Network. ix, 4, 6, 10, 14, 17, 29, 33, 51

CNNs  Convolutional Neural Networks. 4, 6, 7, 10, 14, 33

DCP  Dark Channel Prior. 7

DEM  Digital Elevation Model. ix, 5, 19, 23–25, 45, 52

DLR  Deutsches Zentrum für Luft und Raumfahrt. vii, 5, 51

GPU  graphical processing unit. 39, 40

KLD  Kullback-Leibler Divergence. x, 6, 34, 37, 38, 41, 42, 44

LIVE  Laboratory for Image and Video Engineering. 37

MSE  Mean Squared Error. x, 6, 7, 13, 31, 33, 34, 37, 42, 52

NN  Neural Network. 6

NNs  Neural Networks. 6, 10

PSNR  Peak Signal to Noise Ratio. x, 6, 31–33, 37, 40–42, 44, 45, 52

SSIm  Structural Similarity. x, 6, 31–33, 37, 38, 40–42, 44, 45, 52

SVM  Support Vector Machines. 12
Bibliography


