Generic Model-Agnostic Convolutional Neural Networks for Single Image Dehazing

GENERIC MODEL-AGNOSTIC CONVOLUTIONAL NEURAL NETWORKS FOR SINGLE IMAGE DEHAZING

BY

ZHENG LIU, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2018) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Generic Model-Agnostic Convolutional Neural Networks for Single Image Dehazing
AUTHOR:	Zheng Liu B.Eng., (Automation Engineering) University of Electronic Science and Technology of China, Chengdu, China
SUPERVISOR:	Dr. Jun Chen
NUMBER OF PAGES:	xi, 48

To my family and friends

Abstract

Haze and smog are among the most common environmental factors impacting image quality and, therefore, image analysis. In this paper, I propose an end-to-end generative method for single image dehazing problem. It is based on fully convolutional network and effective network structures to recognize haze structure in input images and restore clear, haze-free ones. The proposed method is agnostic in the sense that it does not explore the atmosphere scattering model, it makes use of convolutional networks advantage in feature extraction and transfer instead. Somewhat surprisingly, it achieves superior performance relative to all existing state-of-the-art methods for image dehazing even on SOTS outdoor images, which are synthesized using the atmosphere scattering model.

In order to improve its weakness in indoor hazy images and enhance the dehazed image's visual quality, a lightweight parallel network is put forward. It employs a different convolution strategy that extracts features with larger reception field to generate a complementary image. With the help of a parallel stream, the fusion of the two outputs performs better in PSNR and SSIM than other methods.

Acknowledgements

First of all, I would like to take this opportunity to express my deepest and sincerest gratitude to my supervisor, Prof. Jun Chen, for his consistent and invaluable guidance and strong support throughout my Master program. He not only provided me with in-depth knowledge, but also inspired me with the spirit of exploring science and seeking truth. His kind encouragement and keen insights are highly appreciated and will always be remembered and cherished. This work would have not been possible without his help.

Secondly, I want to give my thanks to Prof. Keyan Wang of Xidian University and my colleague Sean for our discussions and advice they gave to me. It is my honor to collaborate with them and have their support.

Furthermore, I would like to thank Dr. Sorina Dumitrescu and Dr. Jiankang Zhang for being members in my defence committee. I appreciate their time reviewing my thesis and providing valuable feedback.

Last but not least, I am so lucky to have my parents and my girlfriend Qixue for their love and support. Without them, I could not finish my education and academic path. I appreciate the firm belief and the determination they gave to me.

To them, I dedicate this thesis.

Notation and abbreviations

GMAN	Generic Model-Agnostic neural Network
FCN	Fully Convolutional Network
CNN	Convolutional Neural Network
Adam	Adaptive moment estimation algorithm
ReLu	Rectified Linear unit
PN	Parallel Network
MSE	Mean Square Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
GPU	Graphics Processing Unit
DCP	Dark Channel Prior
MSCNN	Multi-Scale Convolutional Neural Network
GFN	Gate Fusion Network

SOTS Synthetic Objective Testing Set

Contents

List of Figures

Chapter 1

Introduction and Problem Statement

Modern applications rely on analyzing visual data to discover patterns and make decisions. Some examples could be found in intelligent surveillance, tracking, and control systems, where good quality images or frames are essential for accurate results and reliable performance. However, such systems could be significantly affected by environmentally induced distortions, the most common of which are haze and smog caused by dust or small water droplets in the atmosphere. And this problem occurs especially in cities. Therefore, a lot of research in the computer vision community has been dedicated to addressing the problem of restoring good-quality images from their hazy counterparts, Zhu *et al.* (2015); Cai *et al.* (2016); He *et al.* (2011); Berman *et al.* (2016) to name a few. That problem is commonly referred to as the *dehaze problem*.

The relationship between original images and hazy images (Narasimhan and Nayar, 2002) is approximately captured by the following equation known as the atmosphere scattering model (Figure 1.1):

$$I^i(x) = J^i(x)t(x) + A(1 - t(x)) \quad i = 1, 2, 3, \tag{1.1}$$

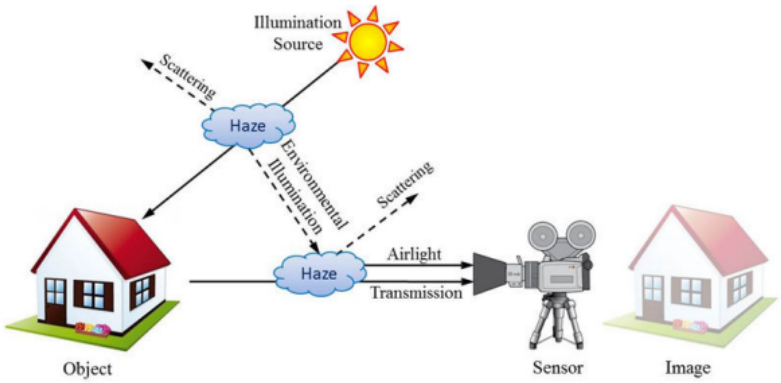


Figure 1.1: Schematic diagram of atmospheric scattering model.

where for a pixel in the ith color channel and spatially indexed by x, $I^i(x)$ is the intensity of the hazy pixel, $J^i(x)$ is the actual intensity of that pixel, and $t(x)$ is the medium transmission function that depends on the scene depth and the atmospheric scattering coefficient β. Parameter A in Equation (1.1) is the atmosphere light intensity, which is assumed to be a global constant over the whole image. Since all variables in Equation (1.1) are unknown except the hazy pixel intensity $I^i(x)$, dehaze is in general an undetermined problem.

Over the past couple of decades, many methods have been proposed to solve the dehaze problem. Those methods could be loosely grouped into two categories: *traditional* and *Machine Learning (ML)-based* methods. The likes of He *et al.* (2011), Zhu *et al.* (2015), and Tan (2008) are some examples of the first category. They solve the underdetermined problem by exploiting some form of prior information.

On the other hand, works such as Tang *et al.* (2014), Cai *et al.* (2016), Ren *et al.* (2016), and Li *et al.* (2017a) have followed a learning-based approach. They leverage the advances in classic and deep learning technologies to tackle the dehaze problem. Regardless how different those two categories may seem, they all aim to recover the original image by first estimating the unknown parameters A and $t(x)$ and then inverting Equation (1.1) to

determine $J^i(x)$:

$$J^i(x) = \frac{I^i(x) - A(1 - t(x))}{t(x)} \quad i = 1, 2, 3. \tag{1.2}$$

From the viewpoint of estimation theory, the methods in both categories fall under the umbrella of the plug-in principle[1], and they will all be referred to as plug-in methods. However, for the dehaze problem, the optimality of the plug-in principle is not completely justified. Indeed, it is unlikely that the problem of lossy reconstruction of the original image can be transformed equivalently to an estimation problem for parameters A and $t(x)$ (or their variants), at least when the two problems are subject to the same evaluation metric. Moreover, the actual relation between the original and hazy images can be fairly complex and may not be fully captured by the atmosphere scattering model. Due to this potential mismatch, methods that rely on the atmosphere scattering model (including but not limited to plug-in methods) do not guarantee desirable generalization to natural images even if they can achieve good performance on synthetic images.

Based on the aforementioned take on plug-in methods (and, more generally, model-dependent methods), this paper approaches the dehaze problem from a different, and more *agnostic*, angle; it presents a dehaze neural network that solely focuses on producing a haze-free version of the input image. It utilizes the recent advances in deep learning to build an encoder-decoder network architecture that is trained to directly restore the clear image, ignoring the parameter estimation problem altogether. The proposed method also has the potential of recognizing complex haze structures present in the training data but not captured by the atmosphere scattering model. See Figure (1.2) for a dehazing example. To

[1]Consider a parametric model $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ and a mapping $\tau : \Theta \to \mathbb{R}$. Suppose the observation comes from P_{θ^*}. The plug-in principle refers to the method of constructing an estimate of $\tau(\theta^*)$ by first deriving an estimate of θ^*, denoted by $\hat{\theta}$, then plugging $\hat{\theta}$ into $\tau(\cdot)$.

(a) (b)

Figure 1.2: Dehazing result of a synthetic example. Left: Hazy input. Right: Clear output.

the best of our knowledge, such view of the dehaze problem has never been explored except in the recent work Ren *et al.* (2018), where a so-called Gated Fusion Network (GFN) is introduced for image dehazing. It will be seen that our proposed network has several advantages over GFN, especially in terms of architecture complexity and input-size flexibility; moreover, certain characteristics of GFN are specifically tailored to the dehaze problem whereas the architecture of our network is more generic and consequently more broadly applicable.

The rest of this paper is organized as follows: In Chapter 2, many related work or methods for image dehazing are reviewed. Chapter 3 introduces the proposed Generic Model-Agnostic convolutional neural Network (GMAN) together with a detailed explanation of the network architecture and its building blocks. Chapter 4 illustrates details of the network's implementation based on GPU and deep learning framework, and also shows the experimental results of GMAN and the performance comparison with other methods. Chapter 5 introduces the proposed parallel network that improves dehazing performance and displays the experimental results. Finally, Chapter 6 makes the conclusion and discusses our future work of image restoration problem.

Chapter 2

Background and Related Work

Single image haze removal has been an ill-posed problem in the field of computer vision, and many remarkable methods have been put forward to solve it in the past decade. These can be divided into traditional and Machine Learning (ML)-based methods in general based on the techniques they use. The traditional ones mostly rely on analysing the physical model(1.1) and other prior information, while ML-based methods focus on using machine learning and deep learning techniques to offset the applicability of prior information mentioned above or estimate the haze-free images from Equation (1.2).

2.1 Traditional Methods

The early groundbreaking and infuential methods are from Tan (2008) and Fattal (2008). Tan (2008) exploits contrast of images to be the prior information under three hypothesis: firstly, hazy images have lower contrast than the clear ones; sencondly, atmosphere light intensity A only depends on scene depth, it is locally a constant value and the changes in

an image should be smooth; thirdly, the restored image has the same statistical characteristics with the natural clear image. With the help of Random Markov Field, $t(x)$ can be modeled and solved by maximizing the potential energy function. However, this image enhancement based method is not phisically-grounded and the result image may be oversaturated. Fattal (2008) employs independent component analysis to estimate the $R(x)$ and $l(x)$ decomposed from the target image $J(x)$ based on locally constant albedo.

He *et al.* (2011) proposed a significant idea which is called Dark Channel Prior (DCP) to tackle this problem. It make use of the fact that there is at least one channel of an image with extremely low intensity at some pixels. Therefore, the dark channel of $J(x)$ can be defined as:

$$J^{dark}(x) = \min_{c \in r,g,b}(\min_{y \in \Omega(x)} (J^c(y))). \tag{2.1}$$

It is assumed that the transmission map is constant in a local patch and A is given. And the $J^{dark}(x)$ of a haze-free image should be 0. Since A^c is always positive, $J^{dark}(x)$ can be expressed as:

$$J^{dark}(x) = \min_{c}(\min_{y \in \Omega(x)} (\frac{I^c(y)}{A^c})) \tag{2.2}$$

As a result, an estimation of $t(x)$ can be derived using Equation (2.1) and (2.2):

$$\hat{t}(x) = 1 - \min_{c}(\min_{y \in \Omega(x)} (\frac{I^c(y)}{A^c})), \tag{2.3}$$

where $\hat{t}(x)$ is denoted as the estimation of $t(x)$. In the next step of estimating A, the top 0.1% brightest pixels are chosen and the highest intensity among their corresponding pixel of the hazy image tend to be atmospheric light A. Thus, the clear image is easily recovered by Equation (1.2) exploiting the estimation of A and $t(x)$. To get a better result, soft matting (Levin *et al.*, 2006) is also adopted to do the refinement of $t(x)$. Obviously, DCP method

sets an excellent benchmark for this problem.

Even though He *et al.* (2011) obtain good effect on haze removal using DCP in most of circumstances, there are still some defect that could be improved. For example, the final image may look dim and DCP has limited performance in large white region like sky and huge walls in the scene. Zhu *et al.* (2015) put forward an algorithm based on the prior information of color attenuation that not only improve the performance on oversaturation problem and the detail quality of the image, but also speed up the processing time. It makes use of a new type of prior information and in particular, employs a regression model to estimate the depth map $d(x)$ of target image as below:

$$d(x) = \theta_0 + \theta_1 v(x) + \theta_2 s(x) + \epsilon(x), \tag{2.4}$$

where v is the brightness component of hazy image, s is the saturation component, θ_0, θ_1, θ_2 are coefficients of this linear model. $\epsilon(x)$ represents the random error of the model. Since Zhu *et al.* (2015) observe that the depth of image has inversly proportional relationship with the gap between brightness and saturation, the regression model is established and through supervised learning with the help of training data, these coefficients are estimated. Note that although this method exploits data training to get the optimal estimation, its innovative point is the type of prior information.

These typical traditional methods mentioned above have pretty good numerical and visual performance, respectively, however, they all require prior information in order to estimate the $t(x)$ or A. This process has one weakness that it hard to get most circumstances of the target image visual quality, which means there usually are limitations on some kind of images or just part of them. Furthermore, considering putting the algorithm into applications, it needs more information and proceedings to deliver the final hazy-free

images.

2.2 Machine Learning Methods

In recent years, Machine Learning and especially Deep Learning techniques have been making great progress with Convolutional Neural Network (CNN) becoming a powerful tool in the field of and computer vision. Thus, researchers take the advantage of those learning based thoughts and techniques to deal with image restoration problems. And many learning based methods have been proposed over the last few years.

Tang *et al.* (2014) first leverage machine learning algorithms to solve the image dehazing problem. They employ a Random Forest regression model to complete the job of estimating trasmission map $t(x)$. After doing the analysis of former prior information based methods, Tang *et al.* (2014) put forward the idea to let one of these prior information be others complementary with the help of a regression model. The inputs of their regressor are features extracted from hazy images and the outputs are their $t(x)$ values. With A estimated using the same criterion as He *et al.* (2011), the haze-free image will be reconstructed by (1.2). Another contribution of Tang *et al.* (2014) s' method is the generation of training data, which is based on two useful assumptions: (1). the image content is independent of depth map $d(x)$ or medium transmission map $t(x)$; (2). the depth value is locally constant. According to this, the training dataset can be composed of small image patches, and this influences many later ML-based methods.

As deep learning techniques and graphic programming hardware are developing rapidly, researchers start using convolutional neural network to solve computer vision problems. To tackle these low-level feature restoration problems like image denoising, super-resolution,

image deblurring and of course, image dehazing, it is proved that the parameters of a network can be learned when extracting features and reconstructing the result image. Cai *et al.* (2016) propose a CNN called DehazeNet with three main layers to estimate $t(x)$ of the input image. The first layer is a feature extraction layer based on Maxout unit; and the second one is a multi-scale mapping layer, which is achieved by setting different convolution kernel sizes (3×3, 5×5, and 7×7); following is the local extremum layer based on maxpooling operation. Before generating the result $t(x)$, Cai *et al.* (2016) employ a BRelu(Bounded Relu) function to do the nonlinear regression. The loss function of this network is MSE between output $t(x)$ and the ground truth transmission map calculated by the mapping relationship from the input RGB image. This method can keep the color of the sky and can also avoid the saturation problem. Furthermore, compared with former methods, it reaches the highest PSNR & SSIM value according to the result of Cai *et al.* (2016).

Similarly, Ren *et al.* (2016) also propose a CNN to estimate $t(x)$. However, the proposed network introduces a multi-scale structure which is composed of a coarse-scale network and a fine-scale network. Regarding the structure of those networks, they are very simillar in general. The coarse-scale network is used for obtaining the coarse transmission map structure and then the output will be sent after the upsampling layer of fine-scale network. The transmission map of hazy images will be fed to both networks, so the coarse output is additional information for fine-scale network. According to Ren *et al.* (2016)s, this character of their network can be benefit in visual performance of $t(x)$ estimation.

To avoid estimating A and $t(x)$ seperately, Li *et al.* (2017a) originally employ a transformation on Equation (1.1). Firstly, it can be rewritten as:

$$J^i(x) = \frac{1}{t(x)} I^i(x) - A\frac{1}{t(x)} + A, i = 1, 2, 3 \tag{2.5}$$

As shown in Equation (2.5), separate estimation will enlarge the errors in $J(x)$. So Li *et al.* (2017a) put forward a *Tranformed Formula*:

$$J^i(x) = K^i(x)I^i(x) - K^i(x) + b, i = 1, 2, 3, \tag{2.6}$$

where

$$K^i(x) = \frac{\frac{1}{t(x)}(I^i(x) - A) + (A - b)}{I^i(x) - 1}, i = 1, 2, 3 \tag{2.7}$$

b is the constant bias which set to 1 as default. Through estimation of $K(x)$ that depends on an relationship between $I(x)$ and $t(x)$, the reconstruction errors can be minimized. The whole model is called AOD-Net (see Fig. 2.1) and it contains two modules: K-estimation module and clear image generation module. Li *et al.* (2017a) employ a CNN which also adapts the multi-scale structure by applying concatenation operation to convolutional layers with different size kernels. Li *et al.* (2017a) illustrate that AOD-Net's joint estimation let $\frac{1}{t(x)}$ and A refine each other, thereby the output image has better lighting conditions and structural details. In terms of end-to-end structure, this method has a thoughtful idea of conbining transmission map and atmospheric light together, but since $t(x)$ appears as $\frac{1}{t(x)}$ inside $K(x)$, the error in $t(x)$ will be amplified. In addition, the clear image generation module requires the output of network and also input hazy image to derive the haze-free image, which means it is cannot generate result image in a single step. From Li *et al.* (2017a) s' experimental results, their network has improvement in PSNR and SSIM on the

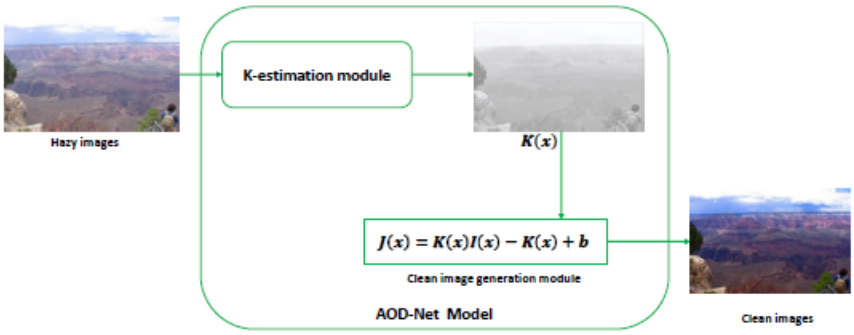


Figure 2.1: The schematic diagram of AOD-Net (adopted from original paper of Li *et al.* (2017a)).

test sets they build.

Ren *et al.* (2018) propose an end-to-end network called Gate Fusion Network (GFN) for image dehazing. It is the latest work that ignore the parameter estimation of this problem and achieve to build an end-to-end model, rather than simple end-to-end training. Unlike the methods introduced above, Ren *et al.* (2018) s' network generates hazy-free image directly after their fusion step. Firstly, the main structure of GFN is based on encoder-decoder structure which has 3 layers for each part. Shortcut connections are also employed to the last layer before the fusion layer. To better learn the pattern of haze and reconstruct original features, Ren *et al.* (2018) leverage a different way from our method which will be introduced later, they adapte multi-scale concept to GFN by applying it to three scales: coarsest level, finer level and finest level. Also the lower level's output will be the input of its upper level through upsampling and concanentation. Obviously, three loss functions are applied to each level. Another contribution of GFN is that it uses several inputs from the results of different image correction methods, which can be regarded as prior information, and this is the main motivation of fusion operation. Ren *et al.* (2018) derive white balanced, contrast enhanced and gamma corrected input to their model, and they can be expressed as

following I_{wb}, I_{ce} and I_{gc}. I_{wb} is obtained by Reinhard *et al.* (2001), I_{ce} and I_{gc} are:

$$I_{ce} = \mu(I - \tilde{I}), \tag{2.8}$$

where $\tilde{I}$ is the average luminance value, $\mu = 2(0.5 + \tilde{I})$, and

$$I_{gc} = \alpha I^{\gamma}, \tag{2.9}$$

where α is set to 1 and decoding gamma correction γ is set to 2.5 in Ren *et al.* (2018) s' experiment. At the end of GFN, outputs of these image enhancements are combined together and haze-free result is derived by them.

Compared with former methods, GFN has better PSNR and SSIM than most of them, however, there is still weakness and inconvenience of GFN. The first example is it cannot work well in corrupted images such as those who have severe fog. And the other ones is that when implementing GFN, there is a limitation in the size of the network's input. The *Width* and *Height* should be the mutiple of 8 otherwise it cannot be fed into the network, so the solution is to resize the input image according to the code. Obviously, this could lead to distortion of input and furthermore, influence the visual quality of desired output. In addition, the three enhancements are hard to be calculated when dealing with real world tasks and the processing time will increase, which is a significant cost.

As what I metioned above, our proposed Generic Model-Agnostic convolutional neural Network (GMAN) has advantages in several aspects over these related methods and the performance is also better. In the following chapters I will explain GMAN thoroughly and analyse its performance in details.

Chapter 3

The Proposed Algorithm

Since the single image haze removal is an ill-posed problem, a deep neural network based on convolutional, residual, and deconvolutional blocks is devised and trained to take on a hazy image and restore its haze-free version. The network globally has an encoder-decoder structure as the whole structure is shown in Fig. 3.1. To achieve haze removal and output the desirable clear image, these modules have their specialized function and are combined in an exquisite structure through experiments. In the following subsections, the network architecture, its building blocks, and the training loss function are discussed in more detail. In addition, the efficiency analasis of each parts are also included.

3.1 Network Architecture

The proposed network is a fully convolutional Network (FCN). It is used to restore a clear image from a hazy input one. Functionally speaking, it is an end-to-end generative network that uses encoder-decoder structure with down- and up-sampling factor of 2. Its first two layers are constructed with 64-channel convolutional blocks. Following them are two-step

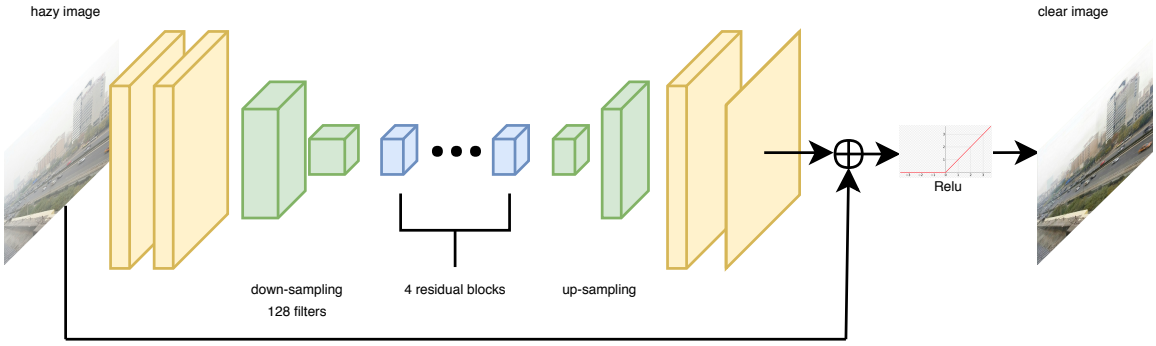


Figure 3.1: Structure and details of GMAN. The yellow blocks are convolutional layers, the green blocks are down-sampling layers and deconvolutional layers. We cascade 4 residual blocks shown as blue blocks, and the number of convolutional layers inside are 2, 2, 3, 4.

down-sampling layers that encode the input image into a $56 \times 56 \times 128$ volume. The encoded image is then fed to a residual layer built with 4 residual blocks, each containing a shortcut connection, see Fig. 3.3. This layer represents the transition from encoding to decoding, for it is followed by the deconvolutional layer that up-samples the residual layer output and reconstructs a new $224 \times 224 \times 64$ volume for another round of convolutions. The last two layers comprise convolutional blocks. They transform the up-sampled feature maps into an RGB image, which is finally added to the input image and thresholded with a ReLU to produce the haze-free version.

3.1.1 Encoder-decoder Structure

Since the proposed GMAN must have the function of exporting haze-free image, the model should be a generative model. Therefore, I adapt the idea of auto-encoder, which is a famous generative model, to carry out result image directly, and our model is illustrated in Fig. 3.2 On the one hand, in the part of encoder, the network employs fully convolutional layers to generate features of haze and original scene.

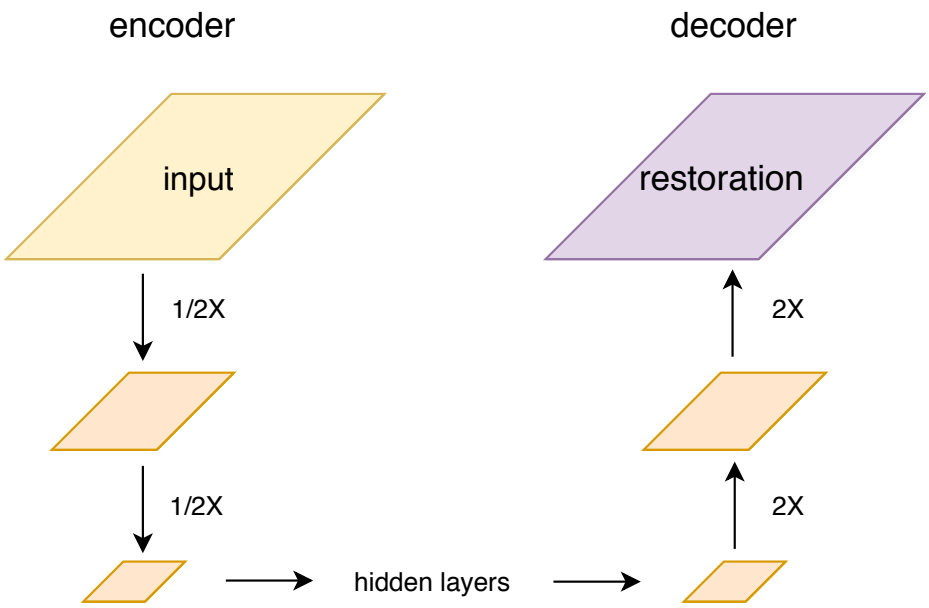


Figure 3.2: Encoder-decoder structure

And in order to reduce the dimenssion of feature map, which is important for allowing the network be deeper and filtering respectively useless information in the image, strides of convolution operations are set to be 2. Then throught two down-sampling layers, maps size is encoded to 56×56. Compared with max-pooling and average-pooling layer, this method for down-sampling can improve image quality and avoid decrease in output image resolution. On the other hand, deconvolutional layers are applied for up-sampling the samll feature maps and reconstructing missing data of haze-free RGB image. Using this structure, haze feature are discarded and original scene features are preserved, Furthermore, the generated image will be more smooth and stable, also, it will have a more steady training process. In short, GMAN gains several advantages in restoring the clear image through the compression and decompression of information. Between the encoder and decoder are hidden layers, I will illustrate how the hidden layers are composed in our GMAN later.

3.1.2 Residual Learning

As it has been a common consensus, the deeper the CNN is, the better expression ability it has. And He *et al.* (2016a) experimentally prove that deeper networks perform better when the time consuming complexity are the same. However, according to He *et al.* (2016b), a 56-layer plain network on CIFAR-10 dataset has higher training and test error than a 20-layer one, which indicates that adding more layers to a CNN cannot improve the performance when its depth reaches a certain extent. Instead, more layers could cause significant degradation of the network in classification task and other experiment results also support this phenomena. To tackle this problem, He *et al.* (2016b) propose a method in applying residual unit, which in GMAN is named residual block. This method is called residual learning, and it has surprisingly good performance in ImageNet task. He *et al.* (2016b) regard one of the former layers as identity layer and add it to a latter layer before the activation layer through short-cut connection, and through this way He *et al.* (2016b) hope to make the network possible to be deeper. Those residual blocks extract high level feature maps to preserve and transport information that is important for restoration to lower level ones in training period. From other point of view, the residual block can also be considered as a fusion operation.

The effect of residual learning, according to He *et al.* (2016b)'s result, avoid network performance's degradation and furthermore increase the accuracy in classification. The loss function converges fast and has lower error, at the mean time, it doesn't occur excessive overfitting. In the field of image classification problem, residual learning structure shows the ability of improving network's performance as its depth increase with and the network has smaller response variance. Another contribution of this method is offsetting gradient vanishing problem in deep neural networks.

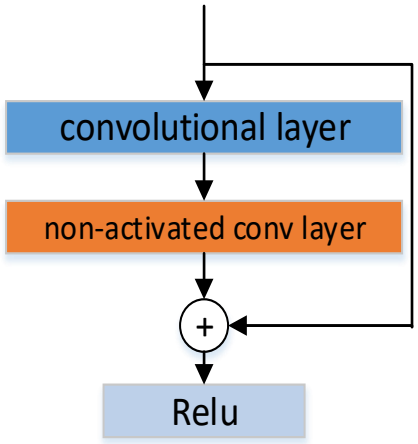


Figure 3.3: A residual block used in the middle layer of the proposed GMAN. In each block, the number of convolutional layers can be different. Relu is used as the activation function after the addition operator of every block.

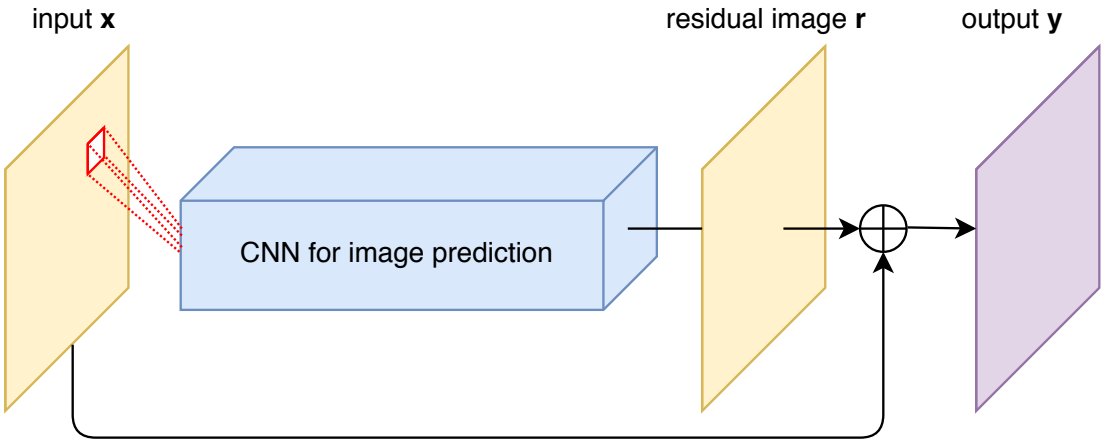


Figure 3.4: Global residual block: input **x**, residual image **r** and output **y** are all RGB images; blue block represents the proposed CNN; and after the addition of **x** and **r**, Relu is applied to get desired output.

The proposed GMAN employs residual learning on two levels, local and global level. In the hidden layers between encoder and decoder module, just right after down-sampling, the residual blocks are used to build the local residual layers. It takes advantage of the hypothesized and empirically proven Kim *et al.* (2016); Zhang *et al.* (2017); Szegedy *et al.* (2017); Ren *et al.* (2017) easy-to-train property of residual blocks (see He *et al.* (2016b)), and learns to recognize haze structures. Residual learning also appears in the overall architecture (see 3.4) of the proposed GMAN. Specifically, the input image is fed along with the output of the final convolutional layer to a sum operator, creating one global residual block. According to 3.4, the output can be expressed as $y = r + x$, so $r = x - y$. It is obvious that by adding input and output map together, residual image r is easy to be optimized because the prediction target of loss function is transformed to be residual image r and r is learned and the most of the value are close to zero. The main advantage of this global residual block is that it helps the proposed network better capture the boundary details of objects with different depths in the scene.

3.1.3 Loss Function: MSE and Perceptual Loss

To train the proposed GMAN, a two-component loss function is defined. The first component measures the similarity between the output and the ground truth, and the second helps produce a visually meaningful image. The following three subsections provide more information on each component and the total loss:

MSE Loss

Using PSNR to measure the difference between the output image and the ground truth is the most common way to show the effectiveness of an algorithm. Thus, MSE is chosen

to be the first component of the loss function, namely L_{MSE}. The optimal value of PSNR could be reached by minimizing MSE at pixel level, which is expressed as:

$$L_{MSE} = \frac{1}{N} \sum_{x=1}^{N} \sum_{i=1}^{3} \parallel \hat{J}(x_i) - J(x_i) \parallel^2, \tag{3.1}$$

where $\hat{J}(x_i)$ is the output of the network, $J(x_i)$ is the ground truth, i is the channel index, and N is the total number of pixels. Through MSE loss, the network learned to produce result image with maxmum similarity of ground truth image.

Perceptual Loss

In many classic image restoration problems, the quality of the output image is measured solely by the MSE loss. However, the MSE loss is not necessarily a good indicator of the visual effect. As Johnson *et al.* demonstrate in Johnson *et al.* (2016), extracting high level features from specific layers of a pre-trained neural network can be of benefit to content reconstruction. The perceptual loss obtained from high-level features can describe the difference between two images more robustly than pixel-level losses. In our experiments, it is proved that opimizing the perceptual loss can get slightly gain on SSIM value.

Adding a perceptual loss component enables the decoder part of GMAN to acquire an improved ability to generate fine details of target images using features that have been extracted (see 3.5). In the present work, the network output and the ground truth are both fed to pre-trained VGG16 network from Simonyan and Zisserman (2014); following Johnson *et al.* (2016), we use the feature maps extracted from layers $conv1_1, conv2_2, conv3_3$ (which will be simply referred to as layers 1, 2, 3) of VGG16 to define the perceptual loss L_p as

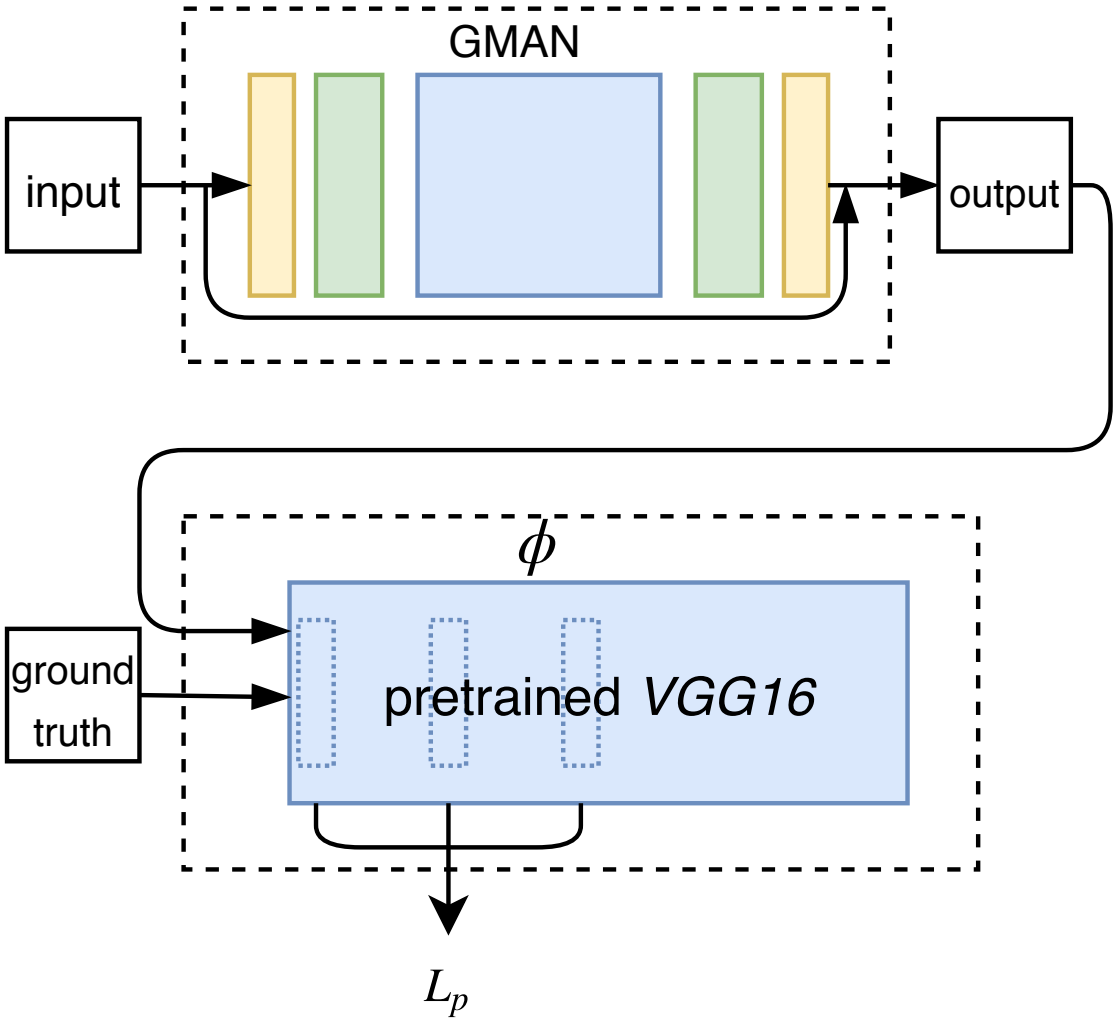


Figure 3.5: The flow chart of how perceptual loss L_p is calculated.

follows:

$$L_p = \sum_{j=1}^{3} \frac{1}{C_j H_j W_j} \parallel \phi_j(\hat{J}) - \phi_j(J) \parallel_2^2, \tag{3.2}$$

where $\phi_j(\hat{J})$ and $\phi_j(J)$ are the feature maps of layer j of VGG16 induced by the network output and the ground truth, respectively, and C_j, H_j, and W_j are the dimensions of the feature volume of layer j of VGG16.

Total Loss

Combining both MSE and perceptual loss components results in the total loss of GMAN. In order to provide some sort of balance between the two components, the perceptual loss is pre-multiplied with λ, yielding the following expression:

$$L = L_{MSE} + \lambda L_p. \tag{3.3}$$

Therefore, in the training process, L is optimized when its components are obtained respectively and the optimization unified since GMAN has end-to-end structure. Details of training and test process will be explained in the following chapter.

Chapter 4

Implementation and Experimental Result

This chapter first discribes the datasets GMAN uses for training and testing, including some pre-processing of input images. Then details of training process, for example parameter setting and hardware information, will be illustrated. After that, quantitative result of our method are provided, as well as visual quality comparison with other methods.

4.1 Dataset for Training and Testing

According to the atmosphere scattering model, the transmission map $t(x)$ and atmosphere light intensity A control the haze level of an image. Therefore, setting these two factors properly is important for building a dataset of hazy images. We use the OTS dataset from RESIDE (Li *et al.*, 2017b), which is built using collected real-world outdoor scenes. The whole dataset contains 313,950 synthetic hazy images, generated from 8970 ground-truth

images by varying the values of A and β (the depth information is estimated using algorithm from Liu *et al.* (2016)). Thus, for each ground-truth image, there are 35 corresponding hazy images.

We notice that the testing set of RESIDE, the SOTS, has 1000 ground-truth images, each with 35 synthetic hazy counterparts, that are all contained in the training data. This certainly can lead to some inaccuracies in testing results. Thus, the testing images were all removed from the training data (including their hazy counterparts), leading to a reduced-size training dataset of 278,950 hazy images (generated from 7970 ground-truth images).

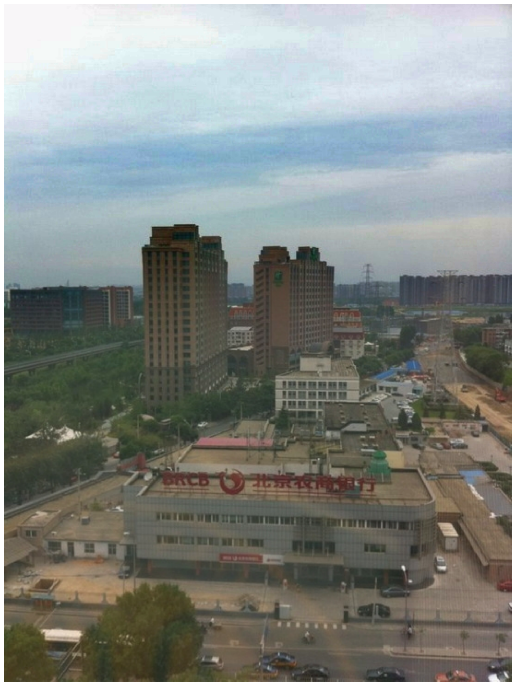
4.2 Training Details

The proposed GMAN is trained end-to-end by minimizing the loss L given by Equation (3.3). All layers in GMAN have 64 filters (kernels), except for the down-sampling ones which have 128 filters, with spatial size of 3×3. The network requires an input with size 224×224, so every image in the training dataset is randomly cropped in order to fit the input size. This restriction is only for the training phase, because the trained network can be applied to images of arbitrary size since it is a fully convolutional network. During training period, images in training dataset are randomly shuffled for every epoch in order to get better ability of generalization. The batch size is set to 35 to balance the training speed and the memory consumption on the GPU. For accelerated training, the Adam optimizer (Kingma and Ba, 2014) is used with the following settings: the initial learning rate of 0.001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The network and its training process have been implemented using *TensorFlow* software framework and carried out on an NVIDIA Titan Xp GPU. After 20 epochs of training, the loss function drops to a value of 0.0004, which is considered a good stopping point.

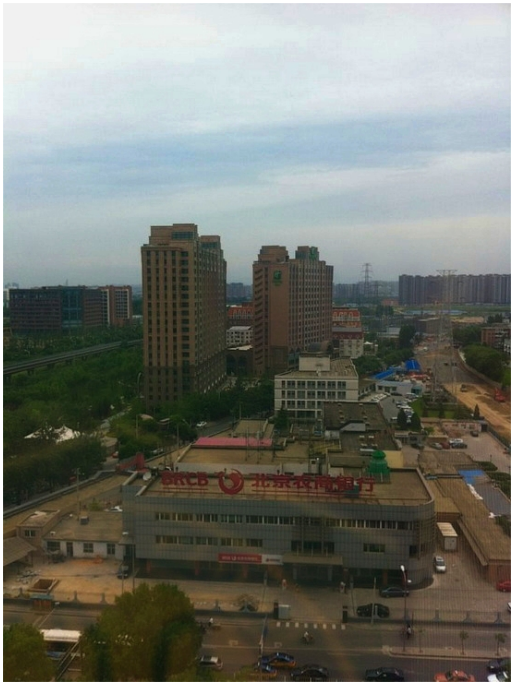
(a) Hazy

(b) DCP

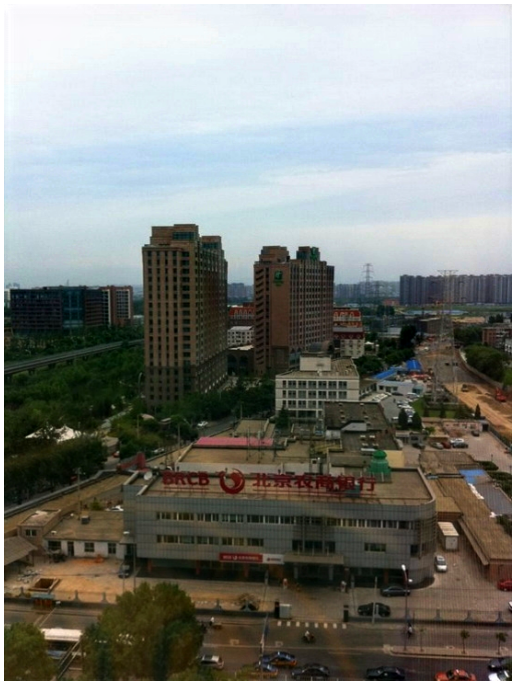
(c) DehazeNet



(d) MSCNN



(e) AOD-Net



(f) GFN

(g) GMAN

(h) Ground truth

Figure 4.1: Visual quality comparison of different dehaze methods. Examples are from synthetic hazy images

(a) Hazy

(b) DCP

(c) DehazeNet

(d) MSCNN

(e) AOD-Net

(f) GFN

(g) GMAN

Figure 4.2: Visual quality comparison of different dehaze methods. Examples are from natural hazy images

4.3 Experimental Results

The proposed GMAN achieves superior performance relative to many state-of-the-art methods. According to Table 4.1 [1] below, it clearly outperforms all other competing methods under consideration on the SOTS outdoor dataset (He *et al.*, 2011; Cai *et al.*, 2016; Ren *et al.*, 2016; Li *et al.*, 2017a). Moreover, as shown in Fig. 4.1 and 4.2, GMAN avoids darkening the image color as well as the excessive sharpening of object edges. In contrast, it can be seen from Fig. 4.1 and 4.2 that the DCP method (He *et al.*, 2011) dims the light intensity of the dehazed image, and causes color distortions in high-depth-value regions (e.g., sky); though MSCNN (Ren *et al.*, 2016) does well in these high-depth-value regions, its performance degrades in medium-depth areas of the target image. Hence, the proposed GMAN can overcome many of these issues and generate a better haze-free image.

	DCP	DehazeNet	MSCNN	AOD-Net	GFN	GMAN
PSNR	18.54	26.84	21.73	24.08	21.67	28.19
SSIM	0.7100	0.8264	0.8313	0.8726	0.8524	0.9638

Table 4.1: Performance comparison on the SOTS outdoor dataset.

	DCP	DehazeNet	MSCNN	AOD-Net	GFN	GMAN
PSNR	18.87	22.66	20.01	21.01	22.44	20.53
SSIM	0.7935	0.8325	0.7907	0.8372	0.8844	0.8081

Table 4.2: Performance comparison on the SOTS indoor dataset.

We have also tested our network on the SOTS indoor dataset (see Table 4.2, Fig. 4.3 and 4.4). In this case, the performance is not as impressive, and comes fourth after DehazeNet, GFN, and AOD-Net. Nevertheless, one can still see the great promise of the model-agnostic dehaze methods even on the indoor dataset. Indeed, also as a member of the family of

[1] In Tables 4.1 and 4.2, the performance results of other methods except GFN are quoted from Li *et al.* (2017b).

(a) light

(b) thick

(c) GMAN (light)

(d) GMAN (thick)

(e) ground truth

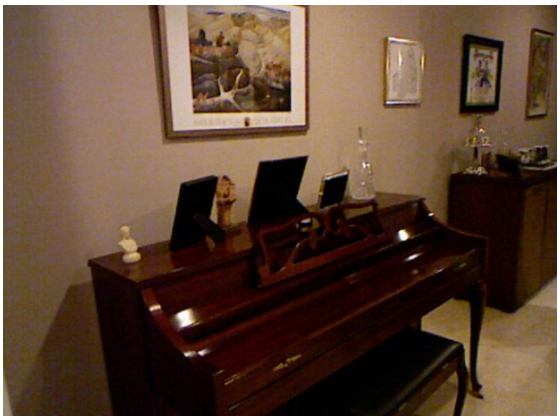
Figure 4.3: Dehazing results from SOTS indoor subset, example with high light intensity area.

(a) light

(b) thick

(c) GMAN (light)

(d) GMAN (thick)



(e) ground truth

Figure 4.4: Dehazing results from SOTS indoor subset, example with low light intensity.

model-agnostic networks, GFN is ranked second in terms of PSNR and ranked first (almost tied with the top-ranked DehazeNet) in terms of SSIM. Our preliminary results indicate that it is possible to design a more powerful model-agnostic network that dominates all the existing ones (especially those based on the plug-in principle) on both SOTS outdoor and indoor datasets by integrating and generalizing the ideas underlying GMAN and GFN. This line of research will be reported in a followup work.

Chapter 5

A Parallel Network

GMAN, the proposed convolutional neural network has the ability of learning to generate haze-free images in an end-to-end manner as I demonstrate in the above Chapters. As it is shown in Figures 4.1, 4.3 and 4.4 as well as Tables 4.1 and 4.2, although GMAN has achieved significant performance in outdoor images and ranked first considering PSNR and SSIM among these competing methods, as for indoor test set, GMAN doesn't has the effectiveness as it does on outdoor set. It can be observed that there are some kinds of artifact occurs in indoor images, which result in distortion of object contour details and color accuracy. And in regions near boundary line between different scene depth, block effect (see Fig. 5.1) may also occur.

Hence, we hope to make an advancement towards better performance on indoor images in the extended work. In the following sections, I will introduce the proposed network structure and demonstrate the improvement through experimental result.

Figure 5.1: Example of block effect that occurs in indoor image.

5.1 Motivation

The main criterion of quantitative performance comparison is the value of PSNR between output image of a model and ground truth image. According to Table 4.2, GMAN is in the fourth place behind DehazeNet, GFN and AOD-Net. And in examples of indoor performance, it can be found that block effect and not being sensitive to similar depth regions are main defects of our network. These effects that weaken the visual quality also result in cutting down PSNR and SSIM. In my opinion, this is due to the fixed kernel size of our network to some extent, which leads to smaller reception field. Hence, finding a way to give more complementary information from larger reception kernel is the most intuitive idea. Moreover, the latest method GFN (Ren *et al.*, 2018) employ a gate fusion structure to make corrections in the quality of image, so as I mentioned in Chapter 4, integrating fusion idea and model-agnostic network together have the potential of making progress. So another parallel lightweight network is built using different convolution strategy, and the refined output image will be the combination of two streams.

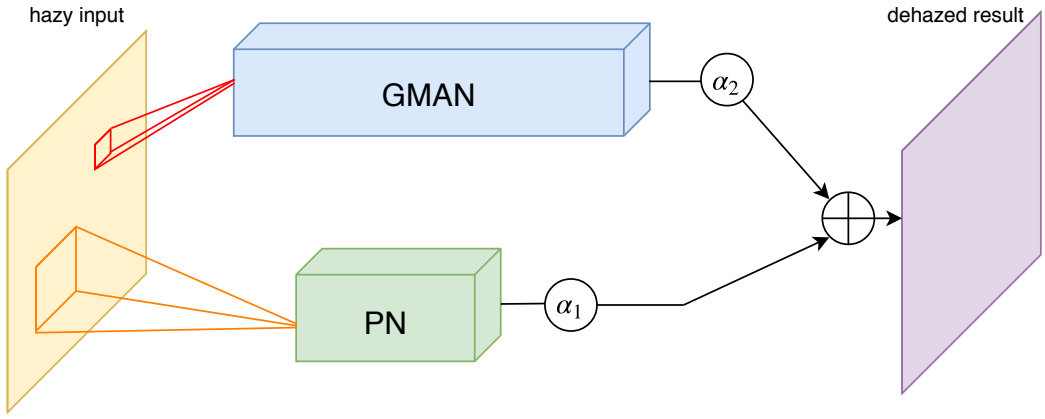


Figure 5.2: Example of block effect that occurs in indoor image.

5.2 Parallel Network

In this section, details and principle of the parallel network is explained. First of all, GMAN performs well on outdoor images and this advantage should be maintained. Thus, main structure of GMAN is still the principal part of the network. Beside GMAN is the parallel network (PN), it can be seen as another branch coming out from input hazy image, and this architecture is shown in Fig. 5.2. According to Fig. 5.2, it is a respectively shallow network compared with GMAN. Similarly, the PN also has an encoder-decoder structure because of that convolutional layers could down-size these feature maps and haze structure still need to be filtered. Regarding the convolutional layers, another type of them is employed. So I am introducing the dilated (atrous) convolution first, whose characters are main purpose of building PN, and details of PN are illustrated afterwards.

5.2.1 Dilated Convolution

Dilated convolution can also be called as 'convolution with holes', which is easier to understand that its core idea is inserting 0s into normal convolution kernels. Traditional

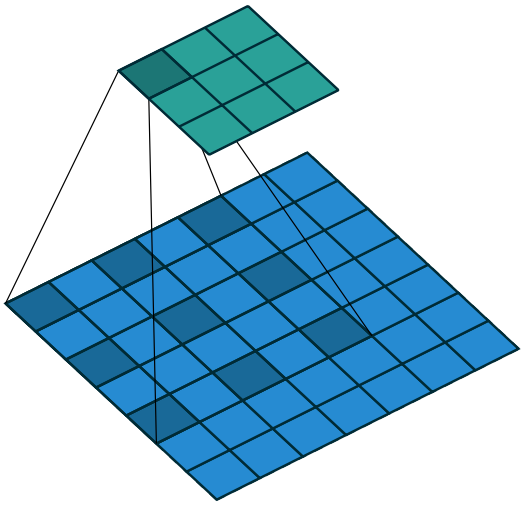


Figure 5.3: Dilated convolution: kernel size 3×3, dilation rate 2.

convolution exploits fixed kernel size and the parameters inside kernel are initialized by specific distribution. Since pooling layers and other down-sampling layers have been put into use for reducing map dimenssion and integrating relation between maps widely, the way to avoid loss in resolution and texture details of images appears to be more important. Also, pooling layer and convolutional layer with larger reception field are not learnable to generate a generalizable feature map. Yu and Koltun (2015) propose dilated convolution originally (see Fig. 5.3), it allows the exponential increase of reception field without spatial dimmensions loss in pixel level. Dilated convolution not only works well in pixel-level image predictions, but also shows surprisingly good effect in tasks that require global or sequence-to-sequence information reliability like: semantic segmentation, speech synthesis and machine translation.

A simple example of dilated convolution operation is shown in Fig. 5.3. the kernel size is 3×3 and dilation rate is set as 2. In the software *Tensorflow*, the dilation rate can be changed like a hyper-parameter. Except these 9 pixel points, value of the rest points

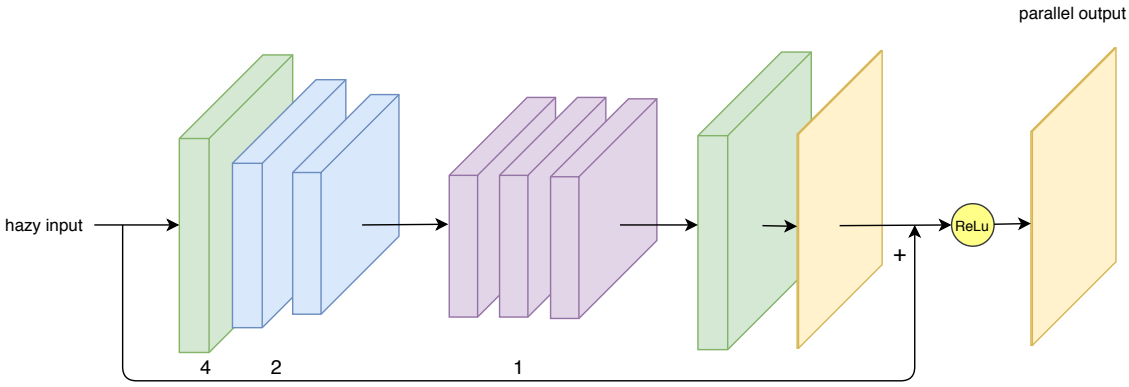


Figure 5.4: Structure details of PN. The dilation rate of these blocks are green: 4, blue: 2 and purple: 1, reception fields are accordingly: 9, 5 and 3. The last convoltional layer uses normal convolution with kernel size 3×3.

are filled with 0. Unlike padding blank pixels into a map, dilated convolution intends to skip some pixels and keep structures and global information. As a result of this action, the reception field then comes to 5×5 if padding method is 'SAME' in *Tensorflow*. Moreover, if padding set to 'VALID', the reception size can become larger. During implementation, the stride of dilated convolution in *Tensorflow* has the default value 1 and it cannot be changed. Compared with normal convolution, if the network needs a specific reception field, the number of parameters that dilated convolution exploits is respectively smaller.

5.2.2 Structure Details

Inside our parallel network, whose detail is shown in Fig. 5.4, almost all convolutional layers (except the last one that generate RGB image) employ the dilated convolution to extract features using larger reception fields and reconstruct a RGB image that is complementary to the output of GMAN. The first layer of PN has the dilation rate 4 and this reduce the size of the input image. Following are two layers with dilation rate 2, and then they are connected

with three layers with dilation rate 1. These 3 different rates lead to different effective kernel size as: 9×9, 5×5 and 3×3 since input kernel size of dilated convolution layers are unified to 3×3. To transfer the feature maps back to original size, a dilated deconvolutional layer which has dilation rate 4 is employed and the output size is 224×224. After deconvolution layer, I also use a normal convolutional layers to obtain the RGB image and also regard this layer as refinement operation. Although the PN is a shallow network with less than 10 layers, I still drag the input image to the last layer and do addition operation before ReLu function in order to calculate residual image which makes the network easier to be trained and optimized (same as global residual learning in GMAN). Until now, all layers except the last convolutional layer have 64 channels (number of filters), and along with GMAN's output, two haze-free images are generated from the two streams. Since these two outputs are very similar, I set two dynamic weights α_1 and α_2 to those images (see Equation 5.1) for balancing the contribution from two streams of our entire network. Then final estimation of the original haze-free image is the combination of two globally weighted map, and this process can be seem as a fusion module where shortcomings of GMAN are corrected and refined by PN's complementary output.

$$\hat{J}_{overall} = \alpha_1 \hat{J}_{GMAN} + \alpha_2 \hat{J}_{PN} \tag{5.1}$$

And through training GMAN and PN by minimizing the similarity between $\hat{J}_{overall}$ and ground truth image, α_1 and α_2 can learn to reach the optimal value automatically.

5.2.3 Experimental Result

In Chapter 4, the performance of GMAN on both outdoor and indoor test dataset are presented and discussed. From quantitative and visual comparison above it can be learned that

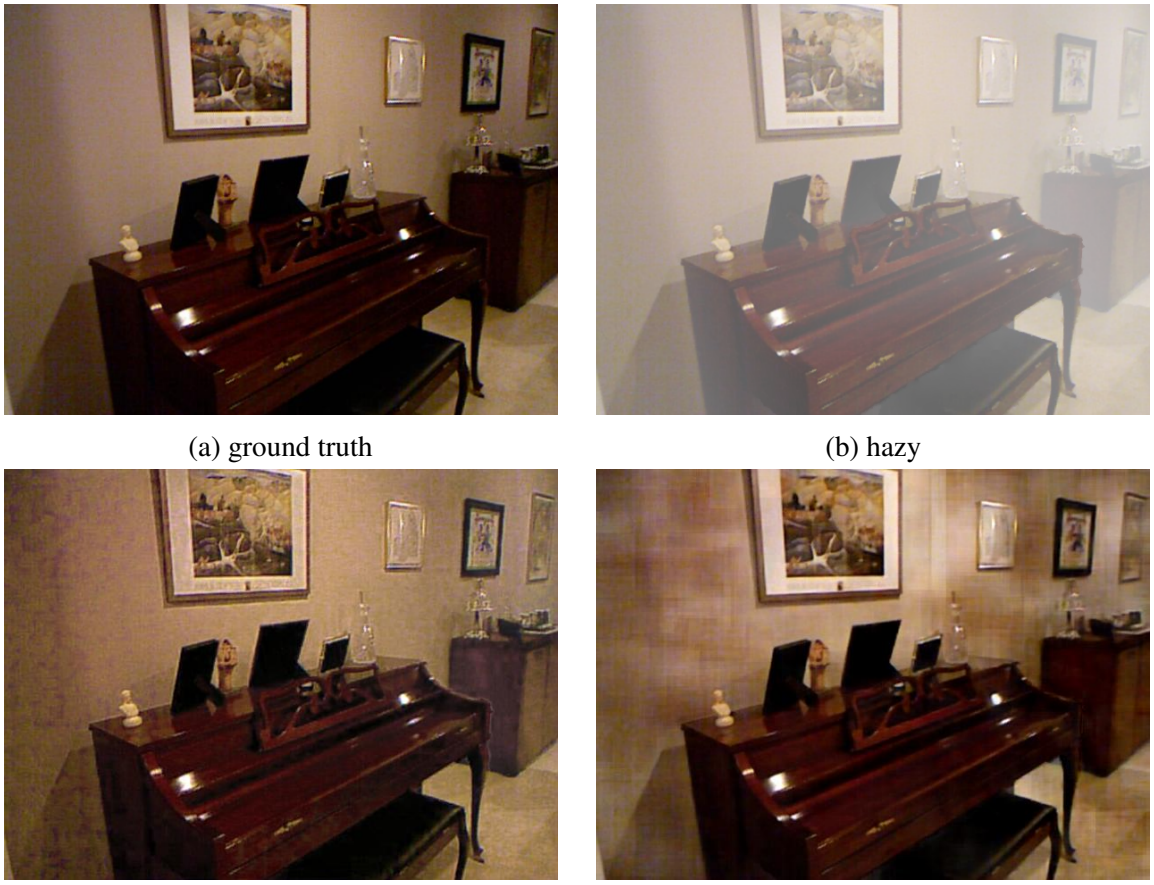
(a) ground truth (b) hazy

(c) GMAN
(22.31 / 0.8579)

(d) GMAN+PN
(23.35 / 0.9013)

Figure 5.5: Dehazing comparison example of SOTS between GMAN and GMAN+PN, also with (PSNR /SSIM).

GMAN has weakness on indoor dataset. Hence, this section will focus on improvement and characteristic that PN have on indoor training and test dataset specifically. First of all, during training process, obviousely the time consumption increases because the whole network has more parameters due to additional branch. Then as for the datasets used for training and testing, I employ the same datasets as those in GMAN's experiment in order to maintain the fairness of the performance comparison.



(a) ground truth (b) hazy

(c) GMAN
(21.70 / 0.7330)

(d) GMAN+PN
(23.69 / 0.8617)

Figure 5.6: Another dehazing comparison example between GMAN and GMAN+PN, also with (PSNR /SSIM).

During training period, Adam optimizer (Kingma and Ba, 2014) is used for minimizing the loss function with initialized learning rate 0.001. The training and testing process are conducted on *Tensorflow* framework and NVIDIA Titan Xp GPU. However, the stoping time in the training of this experiment appears to be later than GMAN, the network converges at around 30 epochs and the loss value decrease to 0.001.

In Fig. 5.5 and 5.6, the quality of the two stream network (GMAN with PN) is shown and compared with single GMAN result image. The advantage of PN can be seen especially in the first example, compared with ground truth image, details like reflection on the floor and light scene near the window can be restored properly in both results. However, on the one hand according to subfigure (c) of first example, GMAN does not remove the haze structure around the vase region, where GMAN+PN could achieve a better work. Moreover, the parallel network also restrains the light overexposure phenomenon that occurs in (c) and preserves better coler effect. On the other hand, in the second example, it keeps the advantages of PN, but it seems that the block effect has not been overcame completely. To better demonstrate the improvement of PN, PSNR and SSIM value of the two structure have also been displayed below images. Both examples show the apparent rising in these two criterion (increase more than 1.0 in PSNR and 0.05 in SSIM).

	DCP	DehazeNet	MSCNN	AOD-Net	GFN	GMAN	GMAN+PN
PSNR	18.87	22.66	20.01	21.01	22.44	20.53	23.03
SSIM	0.7935	0.8325	0.7907	0.8372	0.8844	0.8081	0.8890

Table 5.1: GMAN+PN performance comparison with existing methods.

With the new network structure, performance on test subset of SOTS can also be calculated. After applying GMAN with PN to it, Table. 5.1 shows the PSNR and SSIM testing result among 500 indoor images. According to Table. 5.1, without parallel network, GMAN stands in the fourth place both in PSNR and SSIM on indoor subset of SOTS. And

with the proposed PN, the PSNR comes to 23.03 while SSIM value reaches 0.8890, which is a big step moving forwards and the quantitative performance surpasses the DehazeNet (Cai *et al.*, 2016) to rank the first.

Hence, the visual example and the quantitative result prove that another parallel network using different type of convolutional operation could expand the reception field and achieve the concept of multi-scale refinement, thereby improve the quality of haze-free output. And with only less than 10 layers network branch, it can generate the complementary image without affecting the implementation efficiency.

Chapter 6

Conclusion and Discussion

In this thesis paper, I propose an end-to-end convolutional neural network called Generic Model-Agnostic Convolutional Neural Network (GMAN) to tackle the single image dehazing problem. And the proposed method explores a new direction. Firstly, according to its general structure, GMAN learns to capture haze structures in images and restore the clear ones without referring to the atmosphere scattering model through the encoder-decoder fully convolutional architecture. Unlike previous methods, GMAN also avoids the deemed-unnecessary estimation of parameters A and $t(x)$. Secondly, the proposed network also benefits from the residual learning strategy, locally and globally, which not only helps preserve more texture and detail information of original image, but help improve the efficiency of the network during training process. Finally, from the visual quality of experiment result, it can be proved that GMAN has the potential in generating haze-free images with better quality and the result images also show that it is capable of overcoming some of the common pitfalls of state-of-the-art methods, like color darkening and excessive edge sharpening. These advantages can also be reflected quantitatively, like Table. 4.1 and 4.2,

and the improvment over other methods verify another feature of our method: the perceptual loss jointly increase the similarity between desired image and ground truth when training the GMAN. Hence, the techniques in our method are effective and indespensable.

As I have mentioned in Chapter 1 above, solving the image dehazing problem can be benefit in several real world applications: intelligent surveillance, target tracking, viechle plate detection, etc. It is really important for algorithms to be light in memory usage, portable and easy to be applied, because there are many limitations in real world tasks like hardware capability and time or funds investment. GMAN, as an end-to-end network, has the property of being applied easily and has better generalization ability, which means there is no need to do other pre-adjustment about input image for a trained network. Furthermore, the software framework *Tensorflow* we use is suitble for large scale engineering project and tasks need multithreading computing. With high performance GPUs, it is also possible to fine tune the model with new data set in order to be adapted to the application's goal.

However, from the performance comparison with other methods, it can be revealed that our network has deficiency on indoor dataset. The cause of GMAN's shortcoming of dealing with indoor hazy has already been discussed in Chapter 5, and inspired by the analysis and existing model-agnostic idea, a parallel network is built to improve the performance. The parallel network exploits a new convolutional strategy to enlarge the reception field, which can provide more information of neiboring pixels in order to reduce block effect and improve PSNR and SSIM performance. Note that the parallel structure does not simply add layers (more parameters) to learn the features, with less than 10 layers, it can get a complementary image that assist GMAN refine the final output through training. Through the results of experiments, the ability of PN is proved because it reaches higher PSNR and SSIM value than other methods on indoor test set and keeps the visual quality at the mean

time. This improvement can also be seen in the examples (Fig. 5.5 and 5.6) I display. Therefore, GMAN with the help of a parallel network is believed to make progress on the basis of GMAN in image dehazing problem.

Moreover, due to the generic architecture of GMAN, it could lay the groundwork for further research on general-purposed image restoration. Indeed, we expect that through training and some design tweaks, our network could be generalized to capture various types of image noise and distortions. In this sense, the present work not only suggests an improved solution to the dehaze problem, but also represents a progressive move towards developing a universal image restoration method.

Bibliography

Berman, D., Avidan, S., *et al.* (2016). Non-local image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1674–1682.

Cai, B., Xu, X., Jia, K., Qing, C., and Tao, D. (2016). Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, **25**(11), 5187–5198.

Fattal, R. (2008). Single image dehazing. *ACM transactions on graphics (TOG)*, **27**(3), 72.

He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, **33**(12), 2341–2353.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer

and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer.

Kim, J., Kwon Lee, J., and Mu Lee, K. (2016). Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654.

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

Levin, A., Lischinski, D., and Weiss, Y. (2006). A closed form solution to natural image matting. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 61–68. IEEE.

Li, B., Peng, X., Wang, Z., Xu, J., and Feng, D. (2017a). An all-in-one network for dehazing and beyond. *arXiv preprint arXiv:1707.06543*.

Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., and Wang, Z. (2017b). Reside: A benchmark for single image dehazing. *arXiv preprint arXiv:1712.04143*.

Liu, F., Shen, C., Lin, G., and Reid, I. (2016). Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, **38**(10), 2024–2039.

Narasimhan, S. G. and Nayar, S. K. (2002). Vision and the atmosphere. *International Journal of Computer Vision*, **48**(3), 233–254.

Reinhard, E., Adhikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, **21**(5), 34–41.

Ren, S., He, K., Girshick, R., and Sun, J. (2017). Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, **39**(6), 1137–1149.

Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., and Yang, M.-H. (2016). Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer.

Ren, W., Ma, L., Zhang, J., Pan, J., Cao, X., Liu, W., and Yang, M. (2018). Gated fusion network for single image dehazing. *CoRR*, **abs/1804.00213**.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12.

Tan, R. T. (2008). Visibility in bad weather from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Tang, K., Yang, J., and Wang, J. (2014). Investigating haze-relevant features in a learning framework for image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2995–3000.

Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, **26**(7), 3142–3155.

Zhu, Q., Mai, J., and Shao, L. (2015). A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, **24**(11), 3522–3533.