Image Dehazing by Joint Estimation of Transmittance and Airlight using Bi-Directional Consistency Loss Minimized FCN

Ranjan Mondal, Sanchayan Santra, Bhabatosh Chanda
Electronics and Communication Sciences Unit
Indian Statistical Institute, Kolkata, India
{ranjan15_r, sanchayan_r, chanda}@isical.ac.in

Abstract

Very few of the existing image dehazing methods have laid stress on the accurate restoration of color from hazy images, although it is crucial for proper removal of haze. In this paper, we are proposing a Fully Convolutional Neural Network (FCN) based image dehazing method. We have designed a network that jointly estimates scene transmittance and airlight. The network is trained using a custom designed loss, called bi-directional consistency loss, that tries to minimize the error to reconstruct the hazy image from clear image and the clear image from hazy image. Apart from that, to minimize the dependence of the network on the scale of the training data, we have proposed to do both the training and inference in multiple levels. Quantitative and qualitative evaluations show, that the method works comparably with state-of-the-art image dehazing methods.

1. Introduction

Haze and fog reduces the visibility of outdoor scenes. For this reason, distinguishing objects from distance becomes difficult. Haze occurs when light falls on atmospheric particles and gets absorbed and scattered by them. This causes deterioration in the quality, particularly contrast, of the captured image. The strategy for eradicating the effect of haze from such degraded images is known as Image Dehazing (Fig. 1). Image dehazing is a tricky problem to solve due to the direct dependence of the haze density on the depth of objects. Diverse methods have been suggested to tackle the problem with impressive outcomes [23, 18, 15, 14]. Estimating the scene transmittance and environmental illumination has been established as the key to solve this problem. In recent times, single image dehazing has been receiving a lot of attention due to its practical significance. Due to the ill-posed nature of the problem, the methods mainly depend on statistical priors and physical cues. The recent success of Convolutional Neural Networks (CNN) in the field of computer vision [19, 10, 12] have inspired its use in image dehazing [8, 24, 20]. The main advantage of CNN is that they can learn features from data. This enables feature learning based-on the hazy image formation mechanism.

Figure 1. Hazy image and its dehazed version obtained by our method

Being inspired by these findings, we propose a Fully Convolutional Neural Network (FCN) based approach for single image dehazing. This method was originally proposed for NTIRE 2018 challenge on image dehazing [1]. The method works by dividing a hazy image into patches and inferring the haze parameters for each of these patches using the trained FCN. The main contribution of the work can be summarized as three parts:

1. design of the FCN-based estimator network,
2. formation of the bi-directional consistency loss and
3. the multi-level approach to network training and inference.

These are introduced to mitigate some of the problems of existing CNN based methods and are described in detail in Section 4. Out of the remaining sections, Section 2 presents a tour of the existing methods. The image formation model under haze is outlined in Section 3. Our dehazing method is

described in Section 5. In Section 6 we report the results that we obtain using the proposed method and also the comparisons with the existing methods. Concluding remarks have been made in Section 8.

2. Related Work

Single image dehazing is an ill posed problem. So, the methods resort to additional prior information for dehazing an image. The first thing one may observe about hazy images is the lack of contrast. Hence, an attempt was made by Tan [27] to dehaze an image by maximizing local image contrast. Kratz and Nishino [18] have formulated the image dehazing problem as a Factorial MRF framework. Scene depth and albedo are formulated as independent latent layers and are estimated by maximizing the posterior probability using EM approach. He *et al*. [15] suggested a simple yet effective prior, called the dark channel prior, to estimate haze depth. This prior says that haze free images, in general, have value close to zero in at least one of the color channels over a small patch and this increases with depth of the haze. This was utilized to estimate the scene transmittance. A major drawback of the methods mentioned till now is their their computation time. Tarel and Hautière [30] proposed to remedy this situation by a fast visibility restoration method. They achieve this by estimating the atmospheric veil based on the bound of its possible values using a filter based approach. Haze-free image is obtained by removing this atmospheric veil. Ancuti *et al*. [4] have also proposed a fast method that can identify haze regions based on hue disparity of the original image and its semi inverse. This helps is easy estimation of transmittance and airlight. Meng *et al*. [22] have extended the idea of He *et al*. [15] in computing the initial estimates of transmittance. The estimates are computed from the boundary constrain enforced by the radiance cube. These estimates are combined with a weighted L_1-norm based contextual regularization to reduce halo artifacts. Tang *et al*. [28] proposed to put together the hand-crafted haze relevant features and learn the mapping between the features and the transmittance. The work of Fattal [14] is built around the color line prior. This prior is based on the observation that the color of pixels in a small patch form a linear structure in the RGB space. This line goes through origin for clear images, but it gets displaced due to haze. Help of this cue is taken to estimate the transmittance. On the other hand, the method proposed by Berman *et al*. [7] relies on the observation that color of a haze-free image form a few hundred tight clusters in the RGB space. The presence of haze elongates these clusters to lines. These lines, termed haze lines, facilitate the estimation of transmittance. Zhu *et al*.'s [32] work is based on the observation that the difference between the brightness and the saturation approximately represents the concentration of haze. This information is modeled with the depth as a linear function of brightness and saturation. The parameters of this model is learned to infer depth from hazy images. Cai *et al*. [8] have proposed a method to predict transmittance from patches by learning a regressor. Instead of using hand-crafted features, they learn the features under a CNN framework of a custom design. They have employed special *Maxout* and bilateral rectified linear unit for feature computation. Instead of working on image patches Ren *et al*. [24] estimates transmittance from full images using CNN. They make use of coarse- and fine-scale network to extract features from multiple scales. Li *et al*. [20] have proposed an end-to-end trainable dehazing network that can be easily embedded into other deep learning models to improve the performance of other computer vision task like object recognition in hazy images. They have proposed to use a modified version of the haze imaging equation that unifies the parameters to a single variable. Then they use a CNN to predict this unified parameter. This is the only CNN based method that emphasizes on the estimation of both transmittance and environmental illumination, explicitly. In that sense our work is similar to this method, but some novel attributes to make it superior.

All the above discussed methods restricted themselves to dehaze images which are capture during the daytime. But the degradation caused by the haze, or more generally, the scattering of light by the particles present in the medium, can occur in various other situations, such as, images taken at night or underwater. A method proposed by Ancuti *et al*. [3] attempted to address the problem of dehazing night-time images by relaxing the imaging model and dealing with specific degradation like glows around light sources. Although handling night-time images require special processing, there are methods which work independent of the actual time of capturing the image [25]. Even though the cause of degradation is similar for underwater images, methods proposed for daytime images fails to perform in many occasions. This emphasizes the need for tailor-made methods for images captured underwater [13].

3. Imaging Model under Haze

Light propagating through a hazy atmosphere, gets scattered in various directions by the floating particles. As a result, depending on the aerosol density, the intensity of light decreases as it passes through the medium. This phenomenon is modeled by the following equation [17]:

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A, \tag{1}$$

where $I(\mathbf{x})$ is the observed intensity of light at a pixel $\mathbf{x}$ and $J(\mathbf{x})$ is the intensity (radiance) at the same pixel $\mathbf{x}$ without the effect of haze. 'A' is the global environmental illumination. The term $t(\mathbf{x})$ characterizes the scene transmittance. It indicates the proportion of light that reaches camera from

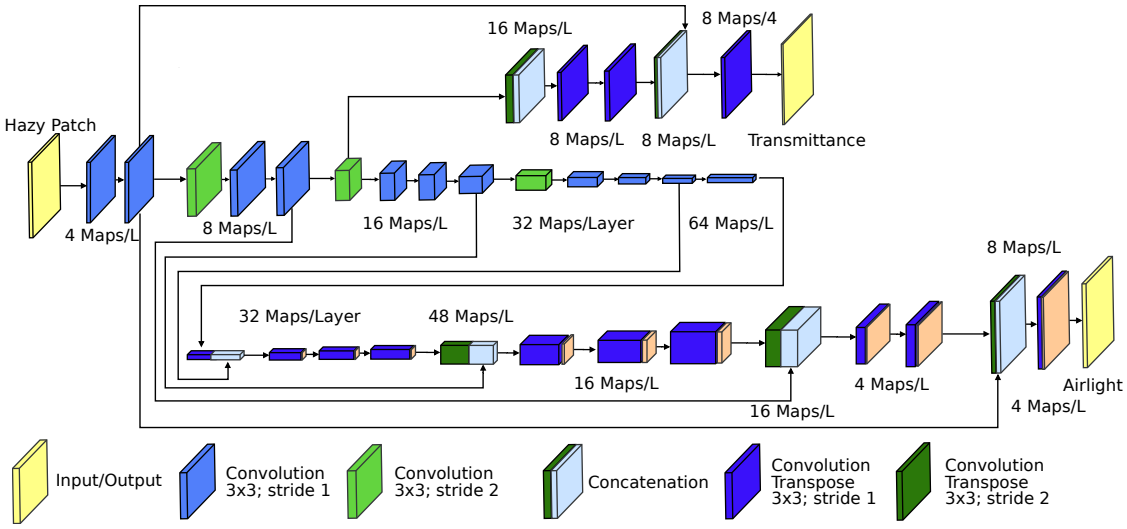


Figure 2. Proposed estimator network

the object. The transmittance is defined as follows,

$$t(\mathbf{x}) = e^{-\beta d(\mathbf{x})}. \tag{2}$$

Here β is the scattering coefficient and $d(\mathbf{x})$ denotes the depth of pixel $\mathbf{x}$. The first part of Eq. (1) is known as the *direct transmission* and the second part is called the *airlight*. For RGB images, this equation is considered as a vector equation with $I(\mathbf{x})$, $J(\mathbf{x})$ and $A(\mathbf{x})$ as 3×1 vector and $t(\mathbf{x})$ as a scalar.

Although it is common to assume that the environmental illumination is constant throughout the image. But this is only valid if the sky is overcast [23]. So, to take into account the other situations we relax this assumption to get the following relaxed model,

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))A(\mathbf{x}). \tag{3}$$

In this, environmental illumination can vary from pixel to pixel. Now given a hazy image, image dehazing methods try to estimate both transmittance and environmental illumination from it and try to obtain the haze free image by inverting the imaging model (Eq. (3)). But estimating $t(\mathbf{x})$ and $A(\mathbf{x})$ independently can be hard due to the way they are related in the imaging equation. For example, when $t(\mathbf{x})$ is close to 1, the effect of $A(\mathbf{x})$ becomes negligible in the hazy image. For this reason, we estimate $(1 - t(\mathbf{x}))A(\mathbf{x})$, denoted by $K(\mathbf{x})$, as a whole. So, in our method we use the following version of the equation (3)

$$I(\mathbf{x}) = J(\mathbf{x})t(\mathbf{x}) + K(\mathbf{x}), \tag{4}$$

and try to estimate transmittance ($t(\mathbf{x})$) and airlight ($K(\mathbf{x})$) in order to recover its haze-free version.

4. Motivation and Proposed Solution

In the following subsections we justify and describe the path we have chosen to reach a solution.

4.1. Estimator Network

Most of the existing CNN based dehazing methods work with small patches by assuming that the transmittance to be constant within a patch. They estimate the environmental illumination separately. But in our method we estimate both $t(\mathbf{x})$ and $K(\mathbf{x})$ from each patch. Estimating the airlight, and consequently the environmental illumination, from a small patch is error prone, as it is difficult to predict whether the colors are due to color of illumination or the color of the object. So, working with bigger patches is inevitable. However, this only reduces the chance of confusion. On the other hand, in bigger patches the constant transmittance assumption is violated. As a result, estimating transmittance map of same size as the input patch becomes necessary for bigger patches. Fully convolutional networks (FCN) have shown promise for the problems where the output size is same as the input, for example, in semantic segmentation of images [21]. For this reason, we design a FCN based estimator network to estimate $t(\mathbf{x})$ and $K(\mathbf{x})$ from RGB hazy patches.

The proposed FCN is two-way forked model that jointly estimates the scene transmittance and airlight (Fig. 2). There are two separate paths to estimate the two parameters: the transmittance estimation path and the airlight estimation path. The path to estimate airlight has more depth than the transmittance estimation path. Success of the earlier methods point out the fact that transmittance can be well estimated from small patches. So, the receptive field, which is the effective size of a convolution kernel on the input

layer, can be kept small for computing transmittance. But to estimate airlight the network needs to see a broad portion of the image. Therefore, the airlight estimation path needs to go much deeper to increase the size of the receptive field. We have taken convolutions with shift of 2 pixels instead of 1 to increase the size of the receptive field while keeping the number of layers less. Without this shift of 2 pixels, we would require more layers to get receptive field of similar size. The convolution layer are matched by same number of *convolution transpose* layers in each path. Similar to the FCN of semantic segmentation [21], we have added some skip connections to retain small scale features and therefore fine details in the output. This also helps in the propagation of the gradient during the training of the network. The skip connections are added between the layers where we are reducing the feature dimension with stride 2 and where we are increasing the feature dimension with stride 2 (in convolution transpose). In the path for estimating airlight, '*elu*' is used as a activation function after each convolution layer except the last layer, whereas in the transmittance estimating path 'sigmoid' activation function is used. In both the output layers we have used 'sigmoid' activation function. Batch-Normalization layer is employed in the last few layers of airlight path to reduce the chance of over-fitting. Note that the network is designed in such a way that it can take input whose dimension is integer multiple of 128, e.g., 128×128, 256×256 and so on. This is possible due to the fully convolutional nature of the network.

4.2. Bi-directional Consistency Loss

To train the proposed network we have designed a new loss based on Eq. (4) instead of using l_2 loss from the ground truth parameter values. The loss is designed in such a way that the network is (ideally) able to do the following,

- Generate the hazy input image from the clear image
- Obtain the clear image by dehazing the input image.

We define the loss (L) as follows,

$$L = \frac{1}{N}\sum_{\mathbf{x}} \big(L_1(\mathbf{x}) + L_2(\mathbf{x})\big) \text{ where,} \tag{5}$$

$$L_1(\mathbf{x}) = |I(\mathbf{x}) - J(\mathbf{x})t(\mathbf{x}) - K(\mathbf{x})| \tag{6}$$

$$\text{and } L_2(\mathbf{x}) = \left| J(\mathbf{x}) - \frac{I(\mathbf{x}) - K(\mathbf{x})}{\max\{t(\mathbf{x}), \epsilon\}} \right|. \tag{7}$$

Here $I(\mathbf{x})$ and $J(\mathbf{x})$ are Input hazy image and ground truth clean image respectively. N is the number of pixels in each image, $K(\mathbf{x})$ is the estimated airlight and $t(\mathbf{x})$ is the estimated the transmittance we obtain using our estimator network. This imaging model inspired loss has certain advantages. First of all, this loss only requires a pair of hazy and haze-free images, apart from the network outputs. Ground-truth parameter values are not necessary. This design also helps in joint estimation of the parameters that conforms to the imaging equation (3). Besides, dehazed output is sensitive to the value of $t(\mathbf{x})$, especially when it is small as a result of which, a small error in $t(\mathbf{x})$ can produce large deviations in the dehazed output. The proposed formulation avoids this pitfall by computing error using the clear image. Also, the bi-directional dependency ensures that the correct estimates are obtained from the network, and do not get stuck at trivial solutions like $t(\mathbf{x}) = 0$ and $A(\mathbf{x}) = I(\mathbf{x})$.

4.3. Multi-level Training

One of the weakness that is inherent in CNNs is that it works with a fixed image dimension and resolution. The dimension problem is usually tackled by resizing the input image. Note that blind resizing may not maintain the aspect ratio and can cause the network to perform poorly. Moreover, if the resolutions (physical area taken by a single pixel) of training and testing images does not match, the network performance can degrade. But we neither have control over the dimension of an input image nor any information about its resolution. For these reasons we take a multi-level approach in both training and application steps. Here we describe the training procedure; the application part is described in the next section.

From the training data we extract overlapping patches from the both clear and corresponding hazy images. We start with a patch of size $P \times P$ in the first level, where $P = \min\{H, W\}$ for a given image of size $H \times W$. In the second level, we extract patches of size $\frac{P}{2} \times \frac{P}{2}$. In the third level patch size becomes $\frac{P}{4} \times \frac{P}{4}$. This halving process is repeated until the patch size falls below 128×128. Therefore, the maximum achievable level is given by,

$$l = \lfloor (\log_2(\min(H, W)) - \log_2(128)) + 1 \rfloor. \tag{8}$$

All the extracted patches are resized to 128×128 before they are used. As we have a corresponding clear image for each of the hazy images, we have a clear patch corresponding to each of the hazy patches. These patch pairs are used to train our network.

5. Dehazing Steps

Our method take the following steps to dehaze an image.

1. Image downscaling and multi-level estimation of transmittance $t(\mathbf{x})$ and airlight $K(\mathbf{x})$,
2. Aggregation of $t(\mathbf{x})$ and $K(\mathbf{x})$,
3. Regularization using guided filter, and
4. Recovery of haze-free image.

Each step is described in detail in the following subsections.

5.1. Image Downscaling and Multi-level estimation of $t(\mathbf{x})$ and $K(\mathbf{x})$

Before doing any kind of processing, we first downscale the input image. If the number of both row and column of the image is greater than L, the input image is scaled with a scaling factor of $k = \frac{L}{\min(H,W)}$ for an image size of $H \times W$. In our experiment, we have taken $L = 850$. So, the scaled image dimension becomes $\lfloor kH \rfloor \times \lfloor kW \rfloor$. This resized image is used in the subsequent steps. This is done to keep the resource requirements low. Next we estimate $t_i(\mathbf{x})$ and $K_i(\mathbf{x})$ at i-th level. Here we have done the computation in only three levels. The patches are of size 256×256, 384×384 and 512×512 in level one, two and three respectively. But this is only possible if the image dimensions are more than the patch sizes. So, for the images that we don't downscale, we take patches of shape 128×128, 256×256 and 384×384. In this situation, we skip a level if the corresponding patch size does not fit into the image. Now, at each level, we take overlapping patches of the specified size, resize them to 128×128 and feed them to our estimator network. The obtained $t(\mathbf{x})$- and $K(\mathbf{x})$-maps from the network are resized back to their actual sizes. Note that the actual size will depend on the level the operation is being done. Then in each level we aggregate the patches to form $t(\mathbf{x})$- and $K(\mathbf{x})$-maps of size $\lfloor kH \rfloor \times \lfloor kW \rfloor$, by averaging the estimates in the overlapping portions. After this step, we get transmittance and airlight maps for each of the levels.

5.2. Aggregation of $t(\mathbf{x})$ and $K(\mathbf{x})$

In the previous step of multi-level estimation, we have obtained transmittance and airlight map for each level. We have to aggregate them to form single transmittance and airlight map which is utilized in the subsequent steps. To aggregate them, we take weighted average of the estimates obtained at each level to generate $t(\mathbf{x})$ and $K(\mathbf{x})$ as follows:

$$t(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(t)} t_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(t)}}, \tag{9}$$

$$K(\mathbf{x}) = \frac{\sum_{i=1}^{l} w_i^{(K)} K_i(\mathbf{x})}{\sum_{i=1}^{l} w_i^{(K)}}. \tag{10}$$

Here $w_i^{(t)}$, $w_i^{(K)}$ are the weights that we use to aggregate $t(\mathbf{x})$'s and $K(\mathbf{x})$'s respectively. $t_i(\mathbf{x})$ and $K_i(\mathbf{x})$ denote the estimates we have obtained at level i and l denotes the number of levels we oprate on. In our experiment, we have taken all the weights to be 1. Although different weights may also be used.

5.3. Regularization using Guided Filter

We have obtained transmittance and airlight maps of size $\lfloor kH \rfloor \times \lfloor kW \rfloor$ after aggregation. But due to the patch based processing, these maps usually contain halos at the border of the patches. So, these needs to be refined before we use the estimates to recover the haze-free image. For this purpose, we need a edge-preserving smoothing filter that smooths the estimates but at the same time respects the object boundaries present in the image. We have used Guided Filter [16] for this purpose because of its efficiency. The Guided Filter filters a given input image while considering the content of a guidance image. We utilize this as an edge-preserving smoothing filter. For smoothing the airlight, we have separately smoothed each of its color channel with corresponding channel of the hazy image as the guidance image. For smoothing the transmittance, we have used the gray-scale version of the hazy image as the guide.

5.3.1 Recovery of haze-free image

Before we had started processing the image, we had downscaled it to reduce the processing load. But the output dehazed image is supposed to have the same dimension as the input image. So, we resize the smooth transmittance map and airlight map back to the original image size e.g. $H \times W$. After that, following Eq. (4), we obtain the dehazed image as follows,

$$J'(\mathbf{x}) = \frac{I(\mathbf{x}) - K(\mathbf{x})}{\max\{t(\mathbf{x}), \epsilon\}}. \tag{11}$$

Here $J'(\mathbf{x})$ is the estimated dehazed image. Note that we have clipped the value of J' between 0 and 1 so that the output stays within the valid range of image intensity.

6. Evaluation

In this section we describe the experimental settings under which we get the results and then compare the results with state-of-the-art methods. We have reported our results on both synthetic and real-world images.

6.1. Experimental Settings

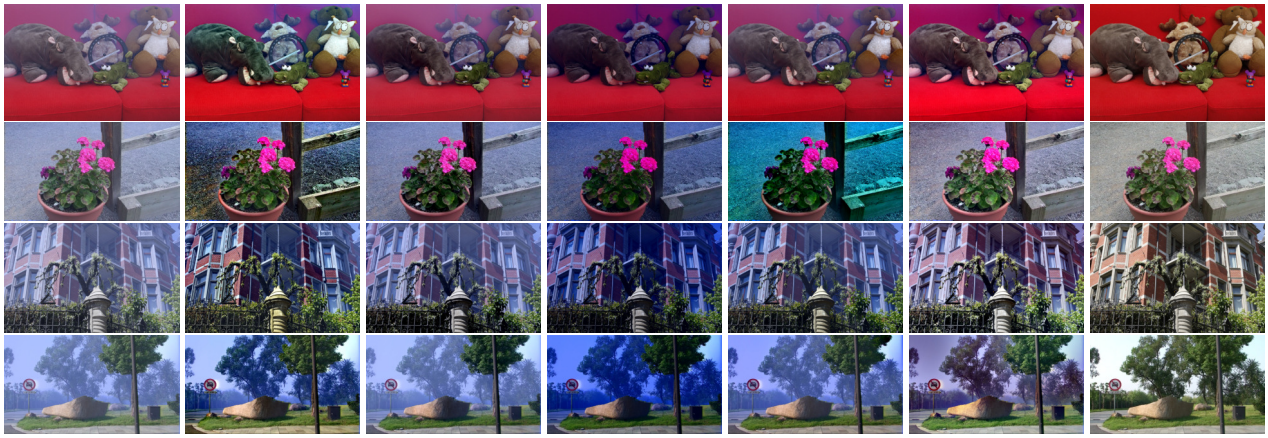
The whole experiment is done on a 3.6GHz quad core machine with 32 GB RAM and one Nvidia GeForce GTX 745 GPU, with Ubuntu 16.04 running on top of them. The estimator network is trained with the help of Keras [9] deep learning library with tensorflow backend and the outdoor training images of the NTIRE dehazing dataset [5, 6]. We train the network for 300 epochs with a batch size of 10 with the help of Adagrad optimizer [11]. The trained model is available from our website[1].

To evaluate our results, we have compared with Cai *et al*. [8], Ren *et al*. [24], Berman *et al*. [7] and Li *et al*. [20]. We have used used codes provided by the authors, in their default settings to generate the results. For Berman *et al*. [7], we have taken the environmental illumination to be $[1, 1, 1]^T$,

[1] http://san-santra.github.io/cvpr18w_dehaze

Table 1. Quantitative comparison of PSNR, SSIM, CIEDE2000 values on Fattal dataset

Image	Berman *et al*. [7]	Cai *et al*. [8]	Li *et al*. [20]	Ren *et al*. [24]	Ours
church	**15.69**/0.88/**16.91**	14.64/0.82/20.45	9.44/0.61/34.64	14.18/0.85/20.26	14.47/**0.89**/24.4
couch	17.28/0.86/14.18	16.71/0.82/14.34	16.79/0.82/17.33	18.02/**0.87**/1**2.92**	**19.54**/0.84/12.94
dolls	15.71/0.8/15.74	16.26/0.81/12.43	**17.24/0.82/10.88**	16.95/0.83/12.38	14.91/0.81/13.51
flower1	12.15/0.71/20.99	19.81/0.94/16.72	12.21/0.79/29.42	9.08/0.42/24.65	**21.35/0.94/14.72**
flower2	11.86/0.67/21.17	19.44/0.91/15.37	13.13/0.78/25.27	10.82/0.59/22.45	**22.75/0.94/11.39**
lawn1	14.78/0.83/**17.93**	13.8/0.81/23.01	11.33/0.67/31.74	14.38/0.8/21.0	**16.17/0.86**/20.22
lawn2	**15.32**/0.85/**17.81**	13.61/0.81/22.47	10.98/0.66/31.7	13.3/0.76/22.27	14.91/**0.86**/20.92
mansion	17.34/0.87/15.84	17.39/0.84/17.42	14.23/0.69/24.01	17.7/0.87/17.53	**21.89/0.92/13.65**
moebius	14.59/0.83/22.4	**19.18/0.94**/16.38	13.21/0.76/27.61	16.38/0.89/19.86	18.22/0.89/**15.29**
raindeer	16.6/0.8/15.28	17.87/0.84/13.73	16.54/0.79/18.5	16.83/0.8/15.49	**22.66/0.89/10.71**
road1	**16.33**/0.87/19.06	13.73/0.79/22.2	11.75/0.65/29.32	14.13/0.82/22.22	16.17/**0.89**/18.42
road2	**18.23/0.89/16.83**	13.22/0.77/23.43	11.95/0.61/30.96	16.45/0.86/20.18	15.89/0.9/20.79
Average	15.49/0.82/17.84	16.31/0.84/18.16	13.23/0.72/25.95	14.85/0.78/19.27	**18.24/0.89/16.41**



(a) Hazy Image (b) Berman *et al*. (c) Cai *et al*. (d) Li *et al*. (e) Ren *et al*. (f) Ours (g) Ground Truth

Figure 3. Comparison of outputs using *Couch*, *flower2*, *mansion* and *road1* images of fattal dataset

as the does not calculate it by itself. We could not generate the results of Li *et al*. [20] on NTIRE dehazing dataset, due to the resource requirements of the code on large images.

6.2. Synthetic images

Here we report our results on hazy images that have been generated by adding haze to clear images. Therefore, for these images we have the ground truth clear images. So, full-reference metrics like PSNR and SSIM [31] can be employed to evaluate the results. We have also reported the average CIEDE2000 [26] values to evaluate the performance of color restoration. For PSNR and SSIM higher value indicates the result is closer to the reference (haze-free) image. Whereas, a low CIEDE2000 value indicates that the colors are similar to the reference image. With these metrics in hand, here we report the results we obtain on Fattal dataset [14] and validation images of the NTIRE dehazing dataset [5, 6]. Both the dataset contain indoor and outdoor images. Although there are other synthetic image dataset [29, 2], we have chosen these two considering the fact that, fattal dataset [14] has images with non-white airlight and NTIRE dehazing dataset [5, 6] has been generated by professional haze machines. We have quantitatively evaluated all the images of the selected dataset (Table 1 and 2) and for visual comparison we have shown the results on 8 images.

Fig. 3 shows the results on Fattal dataset. In these images, method of Berman *et al*. is able to clear the haze, but it also has a tendency to over-enhance the results. This behavior is not observed in other methods. Cai *et al*. produces better looking results than Li *et al*. in all the cases, maintaining the colors more accurately. Ren *et al*. performs to some extent similar to Cai *et al*., except the *flower2* image where the colors are distorted. Our method is not only able to clear the haze but also able to maintain the image brightness, thereby producing results close to the ground truth. This is also reflected in the quantitative results in Table 1.

For the validation images of NTIRE dataset the similar trend can be observed (Figure. 4). Berman *et al*. clears the haze but produces over-contrasted output with saturated colors. In these images Ren *et al*. performs a little better than

Table 2. Quantitative comparison of PSNR, SSIM, CIEDE2000 values on NTIRE hazy dataset

	Image	Berman *et al*. [7]	Cai *et al*. [8]	Ren *et al*. [24]	Ours
Indoor	26	12.42/0.65/20.15	10.17/0.69/24.64	11.02/0.72/22.36	**15.71/0.78/13.86**
	27	14.8/0.66/18.03	14.51/0.67/17.74	17.61/0.77/12.31	**21.94/0.77/8.25**
	28	13.3/0.62/19.24	13.39/0.72/17.7	13.11/0.72/17.06	**16.15/0.73/13.71**
	29	14.67/0.67/15.73	11.91/0.55/20.78	17.6/0.84/11.43	**21.88/0.83/9.33**
	30	13.93/0.61/19.09	15.53/0.71/15.16	16.79/0.73/14.21	**20.66/0.73/12.19**
Outdoor	36	16.92/0.58/14.43	16.59/0.64/13.17	19.46/0.68/11.84	**23.23/0.68/7.6**
	37	14.99/0.52/15.14	15.76/0.57/15.36	17.73/0.6/13.27	**21.4/0.63/8.53**
	38	15.55/0.64/16.92	13.25/0.6/21.85	16.21/0.66/19.02	**22.4/0.69/8.52**
	39	17.65/0.62/16.43	12.78/0.57/20.71	15.75/0.61/16.74	**19.95/0.64/10.84**
	40	17.04/0.61/15.06	16.53/0.67/11.62	18.67/0.7/11.96	**22.2/0.71/7.85**
	Average	15.13/0.62/17.02	14.04/0.64/17.87	16.39/0.7/15.02	**20.55/0.72/10.07**

(a) Hazy Image (b) Berman *et al*. (c) Cai *et al*. (d) Ren *et al*. (e) Ours (f) Ground Truth

Figure 4. Comparison of outputs using validation images of NTIRE dehazing dataset

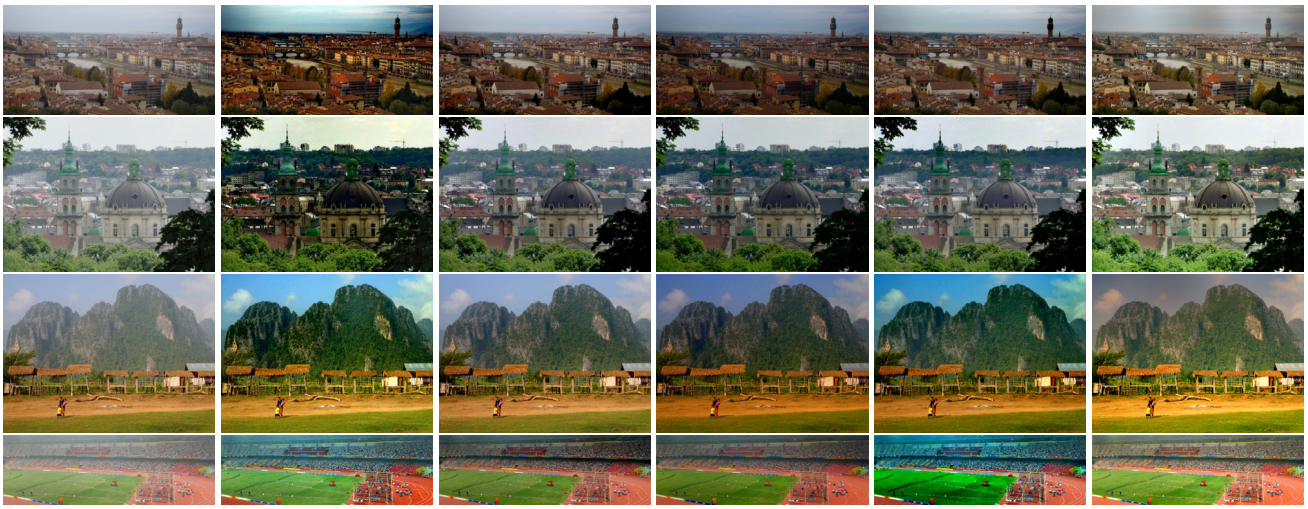
Cai *et al*. in removing the haze. But, output obtained by our method is much closer to the ground truth. This observation is also validated by the quantitative results in Table 2.

6.3. Real World Images

We have qualitatively evaluated the real world images as we don't have ground truth for these images. We have used 4 real world benchmark images used for image dehazing: *florence*, *lviv*, *mountain* and *stadium*. Berman *et al*. [7] produces dark looking results similar to the results obtained in synthetic images, except the *stadium* image. Some haze still remains in the outputs obtained by Cai *et al*. [8] specially at the areas with dense hazes. Li *et al*. is able to clear the haze but produces relatively darker images. Ren *et al*. tends to distort colors in the outputs. It is specially noticeable in *stadium* and *mountain* image. Our method produces the brightest looking results while making the image haze-free at the same time without introducing color distortions.

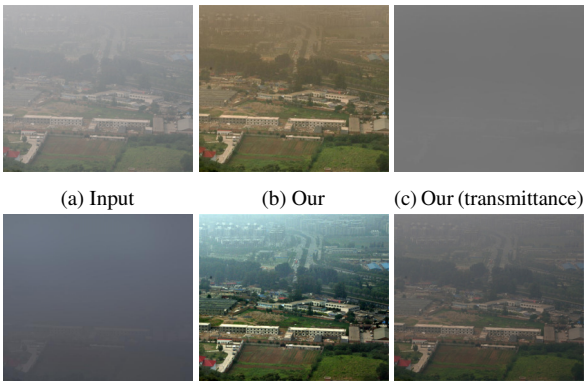
7. Failure Case

Although our method works well in many images, it fails to work in some images. Here we provide two such cases:



(a) Hazy Image (b) Berman *et al*. (c) Cai *et al*. (d) Li *et al*. (e) Ren *et al*. (f) Ours

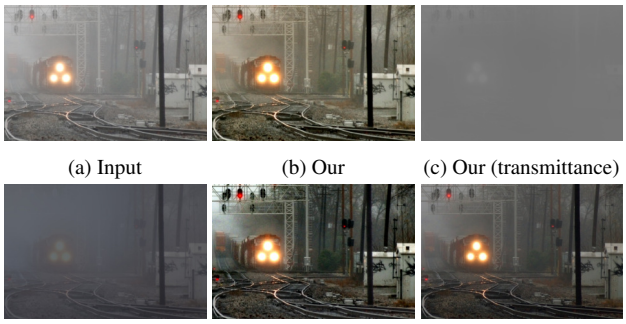
Figure 5. Results on real world images: *florence*, *lviv*, *mountain* and *stadium*



(a) Input (b) Our (c) Our (transmittance)

(d) Our (airlight) (e) Berman *et al*. (f) Li *et al*.

Figure 6. Failure on *canon7* image



(a) Input (b) Our (c) Our (transmittance)

(d) Our (airlight) (e) Berman *et al*. (f) Li *et al*.

Figure 7. Failure on *train* image

canon7 (Fig. 6) and *train* (Fig. 7) image. In both the cases our method have failed to completely clear the haze, specially when the haze is quite thick. This happens because our method fails to correctly estimate the transmittance. Berman *et al*. [7] performs better in this regard but introduces a different color. Li *et al*. [20] is able to clear only a small amount of haze. Our method have also failed to estimate the airlight properly. As a result, we see that the haze has turned yellowish in the results. This is not the case for Li *et al*. and *train* image of Berman *et al*.

8. Conclusion

In this paper, we address the problem of image dehazing using a fully convolutional neural network. We have proposed to mitigate the some of the problems encountered by the existing methods. For example, estimation of airlight, training the network without ground truth transmittance or environmental illumination and scale dependence of the CNNs. We have proposed to estimate airlight along with transmittance. For training the network, we have defined a custom loss that minimizes the error of getting the clear image from the hazy image and vice versa. The scale problem of CNNs have been tackled using a multi-level approach. Although we have achieved results comparable to that of the state-of-the-arts, the proposed approaches may not be the best ones. There is still a lot of room for improvement. How the multi-level training is effecting the performance of the network is not investigated. We hope to address these questions in the future.

References

[1] C. Ancuti, C. O. Ancuti, R. Timofte, L. Van Gool, L. Zhang, M.-H. Yang, et al. Ntire 2018 challenge on image dehazing: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 1

[2] C. Ancuti, C. O. Ancuti, and C. D. Vleeschouwer. D-HAZY: A dataset to evaluate quantitatively dehazing algorithms. In

2016 IEEE International Conference on Image Processing (ICIP), pages 2226–2230, Sept. 2016. 6

[3] C. Ancuti, C. O. Ancuti, C. D. Vleeschouwer, and A. C. Bovik. Night-time dehazing by fusion. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2256–2260, Sept. 2016. 2

[4] C. O. Ancuti, C. Ancuti, C. Hermans, and P. Bekaert. A Fast Semi-inverse Approach to Detect and Remove the Haze from a Single Image. In *Computer Vision ACCV 2010*, Lecture Notes in Computer Science, pages 501–514. Springer, Berlin, Heidelberg, Nov. 2010. 2

[5] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images. *ArXiv e-prints*, Apr. 2018. 5, 6

[6] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images. *ArXiv e-prints*, Apr. 2018. 5, 6

[7] D. Berman, T. Treibitz, and S. Avidan. Non-local Image Dehazing. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1674–1682, June 2016. 2, 5, 6, 7, 8

[8] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, Nov. 2016. 1, 2, 5, 6, 7

[9] F. Chollet et al. Keras. https://keras.io, 2015. 5

[10] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a Deep Convolutional Network for Image Super-Resolution. In *Computer Vision ECCV 2014*, Lecture Notes in Computer Science, pages 184–199. Springer, Cham, Sept. 2014. 1

[11] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011. 5

[12] D. Eigen, C. Puhrsch, and R. Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014. 1

[13] S. Emberton, L. Chittka, and A. Cavallaro. Hierarchical rank-based veiling light estimation for underwater dehazing. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 125.1–125.12. BMVA Press, September 2015. 2

[14] R. Fattal. Dehazing Using Color-Lines. *ACM Trans. Graph.*, 34(1):13:1–13:14, Dec. 2014. 1, 2, 6

[15] K. He, J. Sun, and X. Tang. Single Image Haze Removal Using Dark Channel Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, Dec. 2011. 1, 2

[16] K. He, J. Sun, and X. Tang. Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013. 5

[17] H. Koschmieder. Theorie der horizontalen sichtweite. *Beitrage zur Physik der freien Atmosphare*, pages 33–53, 1924. 2

[18] L. Kratz and K. Nishino. Factorizing Scene Albedo and Depth from a Single Foggy Image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1701–1708, Sept. 2009. 1, 2

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1

[20] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. AOD-Net: All-In-One Dehazing Network. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 4770–4778, Oct 2017. 1, 2, 5, 6, 8

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015. 3, 4

[22] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan. Efficient Image Dehazing with Boundary Constraint and Contextual Regularization. In *2013 IEEE International Conference on Computer Vision*, pages 617–624, Dec. 2013. 2

[23] S. G. Narasimhan and S. K. Nayar. Vision and the Atmosphere. *International Journal of Computer Vision*, 48(3):233–254, July 2002. 1, 3

[24] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single Image Dehazing via Multi-scale Convolutional Neural Networks. In *European Conference on Computer Vision*, pages 154–169. Springer, 2016. 1, 2, 5, 6, 7

[25] S. Santra and B. Chanda. Day/night unconstrained image dehazing. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 1406–1411, Dec. 2016. 2

[26] G. Sharma, W. Wencheng, and E. N. Dalal. The CIEDE2000 colordifference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, Dec. 2004. 6

[27] R. T. Tan. Visibility in bad weather from a single image. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 2

[28] K. Tang, J. Yang, and J. Wang. Investigating Haze-Relevant Features in a Learning Framework for Image Dehazing. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2995–3002, June 2014. 2

[29] J. P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer. Vision Enhancement in Homogeneous and Heterogeneous Fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, 2012. 6

[30] J. P. Tarel and N. Hautire. Fast visibility restoration from a single color or gray level image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2201–2208, Sept. 2009. 2

[31] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004. 6

[32] Q. Zhu, J. Mai, and L. Shao. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, Nov. 2015. 2