

Learning a Patch Quality Comparator for Single Image Dehazing

Sanchayan Santra, Ranjan Mondal, and Bhabatosh Chanda

Abstract—In bad weather conditions like fog and haze, the particles present in the atmosphere scatter incident light in different directions. As a result, image taken under these conditions suffers from reduced visibility, lack of contrast, as a result, it appears colorless. Image dehazing method tries to recover a haze-free portrayal of the given hazy image. In this paper we propose a method that dehazes a given image by comparing various output patches with the original hazy version and choosing the best one. The comparison is performed by our proposed dehazed patch quality comparator based on Convolutional Neural Network (CNN). To select the best dehazed patch we employ binary search. Quantitative and qualitative evaluations show that our method achieves good results in most of the cases, and are, on an average, comparable with state-of-the-art methods.

Index Terms—Image Restoration, Dehazing, Defogging, Visibility Enhancement

I. INTRODUCTION

PRESENCE of fog and haze in the environment greatly reduces the visibility of the scene. Degradation of image of the scene is caused by the scattering of light by the particles present in the atmosphere. These particles attenuate the light reflected off an object, and thereby reduce its intensity before reaching the observer. The environmental illumination also gets scattered by these particles resulting in a translucent veil of light. Due to these two phenomena, the visibility of a scene is greatly reduced under haze condition. Image dehazing methods try to alleviate these problems by estimating a haze-free version of the given hazy image (Fig.1). These methods can play a crucial role as a pre-processing step for many computer vision applications, since most of them assume clear haze-free scene is provided as input.

The most challenging aspect of the problem is its depth dependent degradation, that means with increasing depth, the amount of degradation increases. As estimating depth from a single image is an ill-posed problem, a few researchers have tried to dehaze and image from the point of view of contrast restoration, assuming haze is uniform in the image [1], [2]. These methods work without taking into account how images form under haze. For this reason, these methods can't handle color changes due to haze. A separate class of methods exists that tries to solve this problem by modeling how images form under haze. These methods estimate the model parameters,



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

and then try to obtain an image without the added haze. Due to the depth dependent nature of the degradation, the model parameters also depend on the depth. So, to estimate depth earlier methods took the help of multiple images taken under different conditions [3]–[5]. Taking a different route, some methods tried to dehaze an image when depth map of the scene is known [6] or can be obtained before dehazing [7], [8]. Only more recent methods have focused on dehazing with only a single image as input. These methods achieve this by making stronger assumptions about the input and/or the output images. Tan [9] made an observation that haze-free images have more contrast than the hazy ones. So, in his method he tried to obtain a dehazed image by maximizing the local contrast. Although, the resulting images attain more visibility, they tend to contain saturated colors and look unnatural. Fattal [10] tried to estimate scene transmittance with the assumption that surface shading and scene transmittance are locally statistically uncorrelated. This method fails in case of fog and dense haze when surface shading and scene transmittance does not vary sufficiently. He et al. [11] have proposed dark channel prior to estimate scene transmittance. Dark channel prior is based on the observation that in haze-free images, in most of the local regions not covering the sky, pixels often have low intensity in at least one color channel. In case of hazy images the intensity of those color channels is mainly contributed by the airlight. Kim et al. [12] have also used the idea of contrast maximization but with one added penalty for pixel values going out of valid RGB range in the dehazed output. Although this works well, due to the nature of the penalty, they have employed linear search to find the optimum transmittance

Manuscript received aaaa XX, XXXX;

S. Santra, R. Mondal, and B. Chanda is with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, West Bengal, India, 700108 e-mail: (sanchayan_r@isical.ac.in; ranjan.rev@gmail.com; chanda@isical.ac.in).

value. Tang et al. [13] have tried to solve the problem in a learning framework. They took existing haze-aware features like dark channel, local max contrast, local max saturation, hue disparity to regress the transmittance in image patches. The training data for the regressor was generated by taking patches out of haze-free images. Fattal [14] adopted the idea of color line to image dehazing. In hazy condition, the local color line gets shifted in the direction of airlight. From this shift the transmittance is estimated. Zhu et al. [15] proposed color attenuation prior to model the scene depth. This prior is based on the observation that difference between the brightness and saturation can approximately represent the concentration of haze. So, they have modeled depth as a linear function of brightness and saturation. The parameters of this function is learned in a supervised fashion. With the recovered depth information they dehaze the given image. Cai et al. [16] also proposed a learning based framework to train a regressor to predict transmittance from patches. Instead of using hand-crafted features, they apply Convolutional Neural Network (CNN) to learn the features and predict the transmittance. Ren et al. [17] have also employed CNN to estimate scene transmittance. But rather than working at patch level, they estimate the transmittance map for the whole image. To be able to properly estimate the transmittance in the whole image they have used multi-scale CNN to capture both coarse and fine scale structures. The method proposed by Berman et al. [18] also works with whole images rather than patches. This is based on the observation that the colors of a haze-free image can be approximated by a few hundred colors and they form tight clusters in the RGB space. In case of haze, these cluster form lines (termed as haze-lines). These haze-lines are used to estimate the transmittance at different pixels. The above mentioned methods work with the assumption that the input images are taken in daytime. However, there are some methods that deal specifically with night time images [19]–[22] and also methods that work for both day and night time images [23]. For an overview of the methods the reader may refer to the survey by Li et al. [24].

In this paper we have proposed a daytime image dehazing method that finds the transmittance at each patch (and subsequently at each pixel) by comparing the dehazed version of the patch with the hazy one. This is motivated by the fact that comparing two patches and saying which one has more haze is easier than saying the haze level of a given patch. Now, this comparison is performed by our proposed module called *patch quality comparator*. This comparator, when given two patches as input, can indicate which one is of better quality in terms of haziness. We build the comparator in such a way, that the natural looking patches (e.g., without saturated colors, noise etc.) and the patches with less haze are declared as the better one. So, with the help of this comparator, we search for a transmittance value that can dehaze the given hazy patch and at the same time does not degrade the dehazed output by overdoing. Our main contribution is designing this comparator that efficiently guides this search and exploiting this comparator to solve single image dehazing problem.

The rest of the paper is organized as follows. Section II describes the haze imaging model and also the assumptions

made to solve the problem. Section III gives the idea behind the proposed approach. Section IV describes in detail the steps to dehaze a given image. In section V we state the experimental settings, and the obtained results are compared with that of the state-of-the-art methods in section VI. Finally, section VII concludes the paper.

II. BACKGROUND AND MOTIVATION

Light propagating through a scattering medium undergoes certain changes. Transformation of intensity is one of them. This change in intensity is modeled using the following equation [25] [26]

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

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

where $I(\mathbf{x})$ is the observed intensity, $J(\mathbf{x})$ is the intensity of light coming from the scene objects and before getting scattered, $t(\mathbf{x})$ is the scene transmittance denoting the amount of light that reaches the observer after getting scattered and A denotes the global environmental illumination. The scene transmittance $t(\mathbf{x})$ depends on the depth at position \mathbf{x} and the scattering coefficient (β) (Eq. 2). This scattering coefficient depends on the size of the scattering particle and the wavelength of the light. So, in case of RGB images if we use this model of image formation, then for each color channel $t(\mathbf{x})$ needs to be different. But, Nayar and Narasimhan [3] have shown that in case of fog and haze, the transmittance remains almost the same across the visible spectrum. Therefore, we assume $t(\mathbf{x})$ to be constant across the channels. So, for RGB images the Eq. (1) is treated as a vector equation, with $I(\mathbf{x})$, $J(\mathbf{x})$, and A as 3×1 vector and $t(\mathbf{x})$ as a scalar. Now, given an image $I(\mathbf{x})$ the image dehazing methods try to recover $J(\mathbf{x})$ at each pixel. The methods usually compute the value of ‘ A ’ for the whole image and estimate $t(\mathbf{x})$ at each pixel. Then the imaging model is inverted to compute $J(\mathbf{x})$. Since the mapping $J(\mathbf{x}) \rightarrow I(\mathbf{x})$ is not one-to-one, as $t(\mathbf{x})$ varies from pixel to pixel, the estimation of $J(\mathbf{x})$ independently at each pixel \mathbf{x} can be inconclusive. The value of $I(\mathbf{x})$ could be due to only $J(\mathbf{x})$ when $t(\mathbf{x}) = 1$ or due to only ‘ A ’ when $t(\mathbf{x}) = 0$. This confusion exists even if we know ‘ A ’ for the given image. To surmount this hurdle a simple assumption is adopted which is common in the literature [13], [14], [16]: within a small patch of the image the depth of the scene and consequently, the transmittance t is constant. This assumption is valid because a patch of the image corresponds to a small part of a single surface in the scene, which may be approximated by a relatively smooth surface except at the places where the patch is on the boundary between two surfaces. Therefore the following equation is utilized to estimate transmittance within a patch

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

But, if the patch we consider is very smooth, i.e., $I(\mathbf{x}) = \text{const}$ for all \mathbf{x} and contrast is nil, the effect of haze is neither apparent nor measurable. So, we may argue that using Eq. (3) is not advantageous in two cases: very smooth patches and patches with strong depth discontinuity. Therefore, when



Fig. 2. Haze is added in a patch. This haze patch is dehazed with t values less than 0.65 and greater than 0.65.

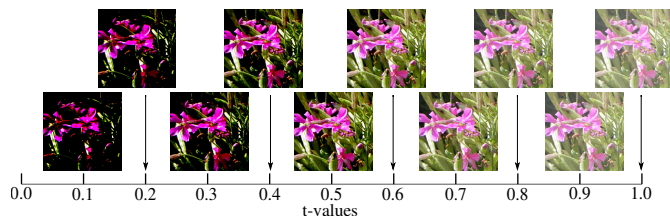


Fig. 3. Same haze patch is dehazed with different t 's. At $t = 1$ the dehazed patch is same as the original haze patch.

we process patches for estimating the parameters, we discard these two kinds of patches. The remaining ones are considered for estimating t . These estimates are then aggregated and regularized to get $t(\mathbf{x})$ for the whole image. These parameters are then used to invert the imaging equation (1). Here we concentrate only on the $t(\mathbf{x})$ estimation part; A is computed as it described by He et al. [11].

III. SOLUTION APPROACH

Our approach is built on the principle that for a given hazy patch there is a $t = t'$ that properly dehazes this patch. Dehazing this patch with $t > t'$ retains some haze in the dehazed output, and on the other hand, using $t < t'$ produces over contrasted, bad looking, unnatural output. So, dehazing a given patch with $t = t'$, we get the actual $J(\mathbf{x})$ by the following equation

$$J_{t'}^c(\mathbf{x}) = A - \frac{A - I^c(\mathbf{x})}{t'} \quad (4)$$

Where $c \in \{R, G, B\}$ is one of the color channels. If the same patch is dehazed with $t (\neq t')$, we can write

$$J_t^c(\mathbf{x}) = A - \frac{A - I^c(\mathbf{x})}{t} \quad (5)$$

From these two equation the following can be written

$$\Delta(\mathbf{x}) = J_t^c(\mathbf{x}) - J_{t'}^c(\mathbf{x}) = (A - I^c(\mathbf{x})) \left(\frac{t - t'}{tt'} \right) \quad (6)$$

So, depending on the value of $\Delta(\mathbf{x})$ we can say whether the dehazed output $J_t(\mathbf{x})$ is more than the actual $J(\mathbf{x})$ or less. Now this equation has two terms. The first term will always be positive. As, A is the environmental illumination, everything in the scene is illuminated by it. So, the value of $I(\mathbf{x}) (= r(\mathbf{x})A$, where $r(\mathbf{x}) \in [0, 1]$ denotes the reflectance property of the scene object) can't be more than A . Therefore the value of $\Delta(\mathbf{x})$ depends only on the relation between t and t' . If $t < t'$ then $\Delta(\mathbf{x})$ is negative. That means the dehazed output is less than the actual one, therefore darker. On the other hand, if $t > t'$ then the dehazed output is more than actual one: it can be further cleaned. This principle is also illustrated using an example as shown in Fig.2. We take a patch from an unhazed clear image. From this patch we generate a hazy patch using Eq. (3) with $t = 0.65$ and $A = [1, 1, 1]^T$. This generated hazy patch is then dehazed using the same A but with different t 's (e.g., 0.8, 0.7, 0.4, and 0.2). It is seen from the figure that dehazing the hazy patch with t less than the ideal (0.65) produces bad output and the patches dehazed with t greater than 0.65 are better than the original hazy patch. Some haze is still present in these hazy patches and it can be further cleaned. We say these are good dehazed patches. Now if we dehaze a given patch with different values of t , we get some good dehazed patches and some bad dehazed patches (Fig.3). If we arrange these dehazed patches, say, in ascending order based on the value of t that is used to obtain these patches, then starting from $t = 1$ down to $t = t'$, we get good dehazed patches, and bad dehazed patches for the remaining values of t . Thus the transition from good to bad dehazed patches occurs at $t = t'$ if we vary the value of t between 0 and 1. We use this fact to obtain the appropriate value of t for any given hazy patch. However, note that to be able to find the point of transition, we must be able to tell, without the knowledge of desired value of t , i.e. t' , whether a dehazed patch is a good dehazed patch or a bad one. For this purpose we build the patch quality comparator.

A. Patch Quality Comparator

The patch quality comparator we propose here compares two patches, say, a given patch and its dehazed version, and reports whether the dehazed patch is good or bad. If we know beforehand whether a dehazed patch is good or bad, we can use this information to train a classifier to perform this comparison. Now instead of using some handcrafted features and employing a two-class (good and bad) classifier to do this task, we use a CNN to learn the features and do the classification simultaneously. The proposed network takes two patches as input and produces two outputs to denote which one of the input is better. The ideal output is $[1, 0]^T$ if the first input is better and $[0, 1]^T$ otherwise. Here the assumption is that the two patches differ only in the amount of haze, and represent the same part of the same scene. The basic structure of the network is inspired from the common CNN based classifiers i.e. convolutional layers for feature extraction followed by dense layers for classification based on the extracted features. Our network is designed in the same way. As our network takes two patches as input, we process both of them by the

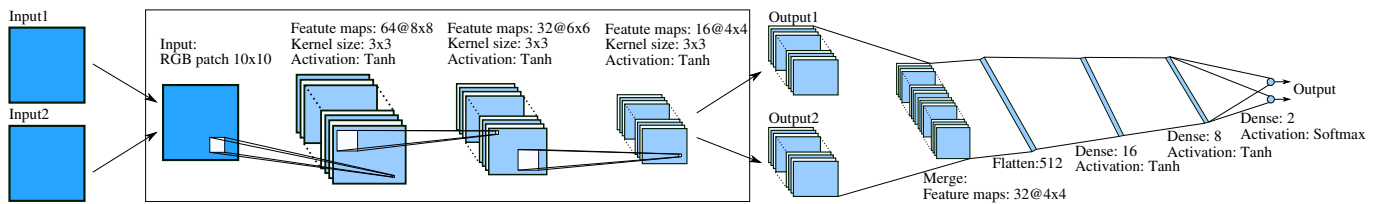


Fig. 4. Architecture of our Patch Quality Comparator

same set of convolutional layers so as to extract the same features from each of them. Another advantage of extracting the same set of features from both the inputs is that, this reduces the dependence of the network on the ordering of the training data. Now, the usual classification networks stack around 8 to 19 layers to classify an image. But in our case, we are processing small image patches (10×10). Therefore, we need only small number of convolutional layers to extract features. On the other hand, we have to take small number of dense layers to avoid overfitting of the classifier. We have used tanh function as the non-linear activation throughout the network except the last layer. In the last layer we use softmax activation function [27, p. 198] as we are training the network as a classifier. The use of tanh results in relatively large gradients (during backpropagation) leading to speed up of the optimization process. Moreover, the vanishing gradients problem associated with tanh is not likely to occur as the network is not very deep.

We train the comparator distinguish the haziness of patches making sure the following conditions are met:

- 1) The haze patch is better than a bad dehazed patch.
- 2) A good dehazed patch is better than the haze patch.

Using this comparator we find at what value of t the transition from good dehazed patch to bad dehazed patch occurs, by repeatedly dehazing a given patch using different values of t . As we have already mentioned, this point of transition gives the desired value of t for this patch. Now, Instead of arbitrary search for this point, we employ binary search to do this computation efficiently.

IV. IMPLEMENTATION OF THE METHOD

The proposed dehazing method has the following 4 main steps:

- 1) Computation of atmospheric light A .
- 2) Binary search for the t for each patch, based on the response of the comparator.
- 3) Aggregation and Interpolation of the estimated t .
- 4) Haze-free image recovery.

When a hazy image is given as input, first atmospheric light A is computed. Then from the input image, we take patches of size $\omega \times \omega$ with 50% overlap (both horizontally and vertically) and find t in each patch using the patch quality comparator, following binary search. Because of the overlap, a pixel receives more than one estimate of t . We take the average of these estimates while determining the value of t at that pixel. Now, it is quite likely that at some pixels the t is not estimated as they belong to either very smooth patches or

to patches with strong edges, which are discarded from being processed as stated earlier (see Section II). At those positions the values of t are interpolated. After this step we obtain $t(x)$ at every pixel in the image. Finally, the dehazed image is obtained by dehazing the image using Eq. (1) with this $t(x)$ and already computed A . Details of each step is provided in the following subsections.

A. Computation of Environmental Illumination

The environmental illumination ‘ A ’ is computed as it is described by He et. al. [11]. We describe it here briefly for the sake of completeness. Environmental illumination can be estimated from the color of the most haze opaque region, i.e., where value of the t is least or, in other words, depth is large. This region is detected with the help of the dark channel of the hazy image. Dark channel of an image I is given by

$$D(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{R, G, B\}} I^c(y) \right), \quad (7)$$

where I^c is a color channel of I and $\Omega(x)$ is a local patch centered at x . As dark channel approximates denseness of haze, the most haze opaque region selected by picking the top 0.1% brightest pixels in the dark channel. Within this region, the pixel with highest intensity in the input image is selected as the environmental illumination.

B. Transmittance finding using binary search

In this step we estimate t from a given patch. We find the ideal t ($= t'$) for a hazy patch using patch quality comparator following binary search strategy. The search is guided by the principle described in the Section III, that is, we get a good dehazed patch when $t > t'$, and a bad dehazed patch when $t < t'$. We need to find the point of transition from good dehazed patch to bad dehazed one, and this good/bad decision is taken by the patch quality comparator.

We begin the process with $t_e = 1$ and $t_b = 0$. We compute $t_m = (t_b + t_e)/2$. Then the input patch is dehazed with the value t_m , and the input hazy patch and the dehazed patch is compared. If the dehazed patch is bad then we can say that $t_m < t'$. Therefore t' lies in the range (t_m, t_e) . So, we set t_b to t_m . On the other hand, if the obtained dehazed patch is good, then $t_m > t'$. So, we set t_e to t_m as t' lies in the range (t_b, t_m) . This process is repeated (i.e. computing t_m , dehazing with new t_m , comparing new dehazed patch with the hazy one, and finally updating t_e or t_b) until $(t_e - t_b)$ becomes small enough. When the search stops, $t_m = (t_b + t_e)/2$ is declared as the desired t' for this patch. The above mentioned steps are written in algorithmic form in Algorithm 1. Here function dehaze()

dehazes a given patch with the provided t and A using Eq. (3). The `haze_patch_comparator()` is the function for Patch Quality Comparator described in Section III-A. It takes two patches as input and produces output depicting which one of the input patches is better.

Algorithm 1 t searching algorithm

Input: I_p, A, θ_t
Output: t_m

- 1: $t_e \leftarrow 1$
- 2: $t_b \leftarrow 0$
- 3: **while** $(t_e - t_b) > \theta_t$ and $t_e > t_b$ **do**
- 4: $t_m \leftarrow (t_e + t_b)/2$
- 5: $I_d = \text{dehaze}(I_p, t_m, A)$
- 6: $(a, b) = \text{haze_patch_comparator}(I_p, I_d)$
- 7: **if** $a > b$ **then** {dehazed patch is bad}
- 8: $t_b \leftarrow t_m$
- 9: **else** {dehazed patch is good}
- 10: $t_e \leftarrow t_m$
- 11: **end if**
- 12: **end while**
- 13: $t_m \leftarrow (t_e + t_b)/2$

C. $t(\mathbf{x})$ aggregation and interpolation

Thus the transmittance parameter t is estimated from the patches as described in the previous step. If we consider overlapping patches, a pixel is likely to receive more than one estimate of t . These values are combined to obtain a single value of t at each pixel. Since during processing we discard some patches that are either very smooth or have strong depth discontinuities depicted by presence of edge, there are pixels without any estimate of t . Value of t at those pixels needs to be interpolated. Now, we can't employ any generic interpolation technique (e.g. bilinear) as they won't be able to preserve the underlying image structure that the transmittance map is expected to follow. Therefore, we apply a Laplacian based interpolation similar to Fattal [14]. The interpolated result is obtained by minimizing the following function

$$\phi(t(\mathbf{x})) = \sum_{\mathbf{x}} s(\mathbf{x})(t(\mathbf{x}) - \tilde{t}(\mathbf{x}))^2 + \lambda \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N_{\mathbf{x}}} \frac{(t(\mathbf{x}) - t(\mathbf{y}))^2}{\|I(\mathbf{x}) - I(\mathbf{y})\|^2} \quad (8)$$

where $\tilde{t}(\mathbf{x})$ is the aggregated estimate obtained after processing the patches. $t(\mathbf{x})$ is the transmittance obtained through interpolated. $N_{\mathbf{x}}$ denotes the neighborhood of the pixel location \mathbf{x} . The presence map $s(\mathbf{x})$ is taken as 1 if the estimate of transmittance is available at \mathbf{x} and 0 if it is not. The regularization parameter λ controls the importance between the two terms. The first term is the error term that enforces the interpolated solution to be as close as the aggregated estimate. The second term is responsible for maintaining the smoothness in the transmittance map while interpolating $t(\mathbf{x})$ from the estimates available in the neighborhood. The smoothing is performed based on $\|I(\mathbf{x}) - I(\mathbf{y})\|^2$. The lower its value, the more similar are the neighboring $t(\mathbf{x})$ values. So, this term ensures that transmittance map becomes smooth where the input image is smooth, while allowing it to retain sharp profile

near the edges. Now for the whole image, the Eq. (8) can be written as

$$\Phi(t_o) = (t_o - \tilde{t}_a)^T S(t_o - \tilde{t}_a) + \lambda t_o^T L t_o. \quad (9)$$

Here \tilde{t}_a is $\tilde{t}(\mathbf{x})$ in vector form. Similarly, t_o is the vector form of $t(\mathbf{x})$. S is a diagonal matrix with $s(\mathbf{x})$ as its diagonal entries. L is a laplacian matrix of the graph constructed from the input image considering each pixel as a vertex and $1/\|I(\mathbf{x}) - I(\mathbf{y})\|^2$ as the edge weights between pixels \mathbf{x} and \mathbf{y} . Each vertex is connected to their neighbors. Now, minimizing Eq. (9) we obtain $t(\mathbf{x})$ for the whole image. The vector t_o that minimizes Eq. (9), is uniquely defined by the solution of the following linear equation,

$$(S + \lambda L)t_o = S\tilde{t}_a. \quad (10)$$

D. Haze-free image recovery

Once we obtain $t(\mathbf{x})$ at every pixel, we can dehaze the input image. We use this computed $t(\mathbf{x})$ along with the atmospheric light 'A' to obtain the dehazed result. Using the following equation we calculate the estimated dehazed image $J_e^c(\mathbf{x})$ as

$$J_e^c(\mathbf{x}) = A - \frac{A - I^c(\mathbf{x})}{\max\{t(\mathbf{x}), 0.0001\}}. \quad (11)$$

Note that, $J_e^c(\mathbf{x})$ values lying beyond the valid intensity range are clipped to the valid range. Second, we assume $t(\mathbf{x})$ should be greater than zero, otherwise no scene information would reach the observer or the sensor (camera). To ensure this, we clip the value of $t(\mathbf{x})$ arbitrarily at 0.0001 from lower. Third, unlike many other methods we do not employ any kind of post-processing technique.

V. EXPERIMENTAL DETAILS

In this section we describe in detail the setup we have used to train our comparator and to generate the dehazed images we report in the next section.

A. Training Data Generation

To train our patch comparator we synthesize hazy patches from clean haze-free patches. These clean patches are taken from 421 natural haze-free images. These images are members of a subset of the 500 fog-free images used by Choi et. al. [28]. We have discarded some images where already haze is present, specially at distant objects. We have extracted patches from these images while discarding very smooth ones and the ones with strong edges. This generates around 2.5 million patches. However, we don't use this directly to generate the training data. As we have observed, using all these patches to generate the training data may introduce bias in the trained comparator. This was due to the fact that many of the patches are "similar". To alleviate this situation, we have clustered the patches taking their RGB values and their first order gradient in horizontal and vertical directions as features. The patches are clustered using k-means with 1×10^5 cluster centers. Then the patches closest to the cluster centers are used as clean patches (only the RGB part) to generate the training data. Each training datum contains a patch pair and corresponding labels indicating the

better patch. Each pair consists of a haze patch and its dehazed version. To generate these patch pairs, we first generate a hazy patch from a clean patch using the haze imaging equation (3) with a random $t(=t')$ between 0 and 1 and two random values of 'A'. As 'A' is a 3×1 vector, we take 3 random values between 0 to 1 to get a 'A'. We dehaze this generated hazy patch with the corresponding A's and 30 different values of t . Half of the t 's are greater than t' and half of them are less than it. The values of t 's are taken in such a way that the values are concentrated near t' and are sparse as we go far from it. To achieve this we first divide each range $((0, t')$ and $(t', 1))$ into 5 varying length bins with smaller bins near t' and larger bins at distant. For example, we take bins of size $\frac{1}{2}, \frac{1}{4}, \frac{1}{8},$ and $\frac{1}{16}$ on either half of the length. From each of the bins, however, we sample equal number of t 's. Now as stated earlier we say the dehazed patches obtained with $t < t'$ are bad dehazed patches and dehazed patches obtained with $t > t'$ are good dehazed patches and label them accordingly. This process is done for each patch obtained from the haze-free images to generate the training data to train the comparator.

B. Parameter Settings

We have taken patches of size 10×10 to train our comparator and also to dehaze the test image. The patches with standard deviation less than 0.02 are discarded as smooth patch. To discard patches with depth discontinuity (i.e., having strong edge), we check if it contains pixels with gradient magnitude greater than 0.5. The comparator is trained with mean squared error loss for 50 epochs with batch size of 500 using the Adadelta [29] optimizer. This setup is build on Keras 1.2.2 [30] with Theano 0.9.0 [31] and libgpuarray backend. With this setup the training of the comparator is done on a machine with Intel® Core™ i7-4790 CPU @ 3.60GHz and Nvidia GTX 745. For computing the environmental illumination using dark channel, we have taken patches of size 15×15 . But for determining t from the patches, we have used 10×10 patches. For t -searching algorithm θ_t is taken to be 5×10^{-4} . Lastly for interpolation, the regularization parameter λ (see Eq. (8) and (9)) is taken to be 0.1.

VI. RESULTS

We have evaluated our method on synthetic dataset as well as outdoor hazy images and compared the results with that of the state-of-the-art dehazing methods and one contrast enhancement based method [2]. We report the results we obtain on D-Hazy [32] and Fattal [14] dataset. D-Hazy dataset¹ contains synthetically generated hazy images. Fattal dataset² contains both synthetic and real world images and provide environmental illumination (A) for each hazy image. Therefore, the results for the real world images given by Fattal [14] are reported using estimated 'A' as well as the provided 'A' values. To compare with Fattal [14] we have used the dehazed images provided by the author. For comparing with Pierre et al. [2], Ren et al. [17] and Berman et al. [18] we have

used the code provided by the authors in their default settings to generate the results. Out of these two methods, Berman et al. [18] relies on supplied 'A' value whereas Ren et al. [17] compute it from the given image. So, to obtain the results of Berman et al. [18] we have used the provided A values.

A. Quantitative Results

We quantitatively evaluate the results we obtain for synthetic hazy images with known ground-truth. For this purpose we have used synthetic images of the Fattal [14] dataset and D-Hazy dataset [32]. D-Hazy dataset is synthesized from Middlebury³ [33] and NYU Depth⁴ [34] dataset. The Fattal dataset contains both indoor and outdoor images, while the D-Hazy datasets contains images of various indoor scenes. We quantitatively evaluate the results obtained on these images using SSIM [35] and CIEDE2000 [36] metric as done by Ancuti et al. [32]. PSNR is not reported as it is not very effective in evaluating perceived image quality [37]. SSIM is a full reference image quality metric that is based on the hypothesis that the human visual system is highly adapted for extracting structural information. So, given an distorted image with its groundtruth, SSIM provides a score between $[-1, 1]$, with 1 denoting identical images. With this metric we can compare how much a dehazed image is structurally similar to the groundtruth. As color restoration is also crucial aspect of image dehazing, CIEDE2000 is employed to measure the color distortion. This compares two colors and gives a value between $[0, 100]$ with small values indicating similar colors. We report average CIEDE2000 value for the whole image. Here the results are reported separately for synthetic images of Fattal dataset, and Middlebury and NYU section of D-Hazy dataset.

For Fattal dataset the quantitative results are given in Table I while in Fig.5 we illustrate the visual results for 4 images: *church*, *flower2*, *lawn1*, and *couch*. The results show, the method of Pierre et al. [2] is able to clear only a small amount of haze and could not fix the color cast due to haze. This is due to the fact that this method does not take into account how images form under haze. This is also reflected in the quantitative results: this method has the highest CIEDE2000 values. Comparison among the other results show that color bias exists in the dehazed images obtained by Ren et al. [17] and also in our method (with atmospheric light estimated using Dark Channel Prior). This suggests that wrong estimation of atmospheric light may introduce color bias in the dehazed output because both the methods estimate 'A' from the image where transmittance is minimum and intensity is high. This does not happen if the actual A is supplied to the dehazing method. This is reflected in the CIEDE2000 values for the corresponding output images. In the output of Fattal [14] there are some local color distortions, which does not exists in the results of Berman et al. [18] and our method (given A). The Table I reveal that our method performs well against the competing methods.

¹http://www.meo.etc.upt.ro/AncutiProjectPages/D_Hazzy_ICIP2016/

²http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/

³<http://vision.middlebury.edu/stereo/data/scenes2014/>

⁴http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

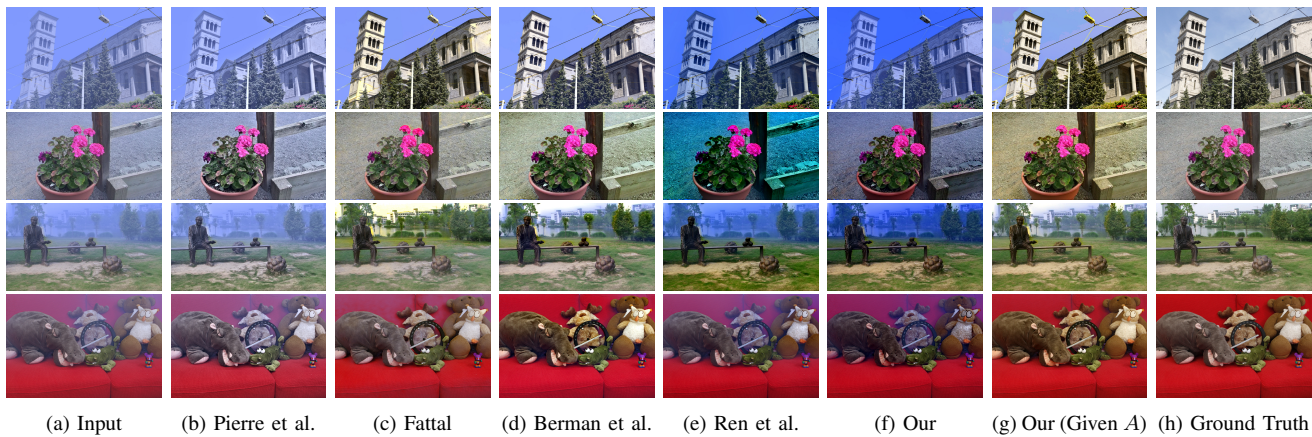


Fig. 5. Results of synthetic images of Fattal dataset on *church*, *flower2*, *lawn1*, and *couch*

TABLE I
QUANTITATIVE RESULTS OBTAINED ON FATTAL DATASET IN TERMS OF SSIM (HIGHER THE BETTER) AND CIEDE2000 (LOWER THE BETTER) METRIC.

	Pierre et al.		Fattal Color Line		Berman et al.		Ren et al.		Our		Our (Given A)	
	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000
church	0.887	19.69	0.964	6.342	0.952	10.069	0.871	20.257	0.84	25.424	0.93	7.077
couch	0.821	14.73	0.905	6.719	0.934	5.737	0.88	12.918	0.861	14.439	0.952	3.404
dolls	0.754	20.341	0.779	6.105	0.881	10.824	0.865	12.377	0.861	10.717	0.856	9.055
flower1	0.969	14.582	0.988	3.912	0.957	8.779	0.438	24.645	0.898	21.049	0.948	11.636
flower2	0.971	11.612	0.991	2.921	0.956	8.603	0.608	22.454	0.869	18.906	0.948	10.911
lawn1	0.869	21.753	0.972	6.653	0.962	8.229	0.831	21.003	0.84	21.455	0.967	6.196
lawn2	0.871	21.797	0.972	6.461	0.96	8.499	0.778	22.274	0.846	25.494	0.968	6.178
mansion	0.88	16.129	0.977	4.046	0.955	5.947	0.89	17.526	0.893	18.134	0.935	5.732
moebius	0.962	15.285	0.908	10.618	0.942	9.055	0.9	19.857	0.903	20.59	0.934	8.903
raindeer	0.842	13.743	0.948	4.104	0.882	11.028	0.814	15.492	0.86	13.348	0.969	2.923
road1	0.854	21.116	0.968	5.241	0.962	5.648	0.845	22.224	0.844	23.612	0.969	4.682
road2	0.819	23.501	0.962	7.119	0.95	8.865	0.881	20.175	0.839	22.652	0.952	6.334
Average	0.874	17.856	0.944	5.853	0.941	8.44	0.8	19.266	0.862	19.651	0.944	6.919

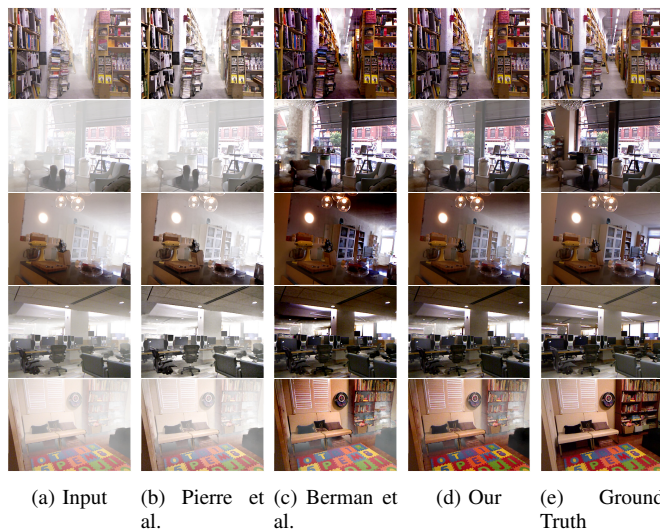


Fig. 6. Some Results of NYU portion of D-Hazy dataset

TABLE II
AVERAGE METRICS OBTAINED ON NYU PORTION OF D-HAZY DATASET.

	Pierre et al.	Berman et al.	Our	Our (Given A)
SSIM	0.7	0.73	0.73	0.794
CIEDE2000	24.283	13.33	13.78	13.036

with error for one image. The results of Berman et al. [18] show that it has tendency to over estimate haze at places and thereby turns the pixels black at those places, which is not present in the results obtained by our method. Also in the last image, the method of Berman et al. [18] has failed to properly remove the haze over the carpet which our method has done successfully. This is also reflected in the quantitative metric values reported in Table II.

For the Middlebury section of D-Hazy dataset, we have illustrated the results with four images (Fig.7): *Piano*, *Bicycle1*, *Motorcycle*, and *Flowers*. The results show that the method of Ren et al. [17] have failed to dehaze completely, particularly when the value of t is relatively low. For *Motorcycle* and *Bicycle1* images, the floor is dehazed more by our method because of it's similarity in color with airlight. For the same two image the method of Pierre et al. [2] is behaving in a queer manner. The cycle has been enhanced to white and the motorcycle image is completely black. Repeated run of the code did not change the output. For the *pipes* image this method is also giving black output. The results show that the

For the NYU section of D-Hazy dataset we only report the average values of the metrics in Table II, as the dataset contains 1449 images. To illustrate the performances we show 4 result images in Fig.6. The results show Pierre et al. [2] behaves the same way as noted in the previous paragraph: it is not able to clear dense haze. We could not get results for all the images using the method of Ren et al [17], as the provided code ends

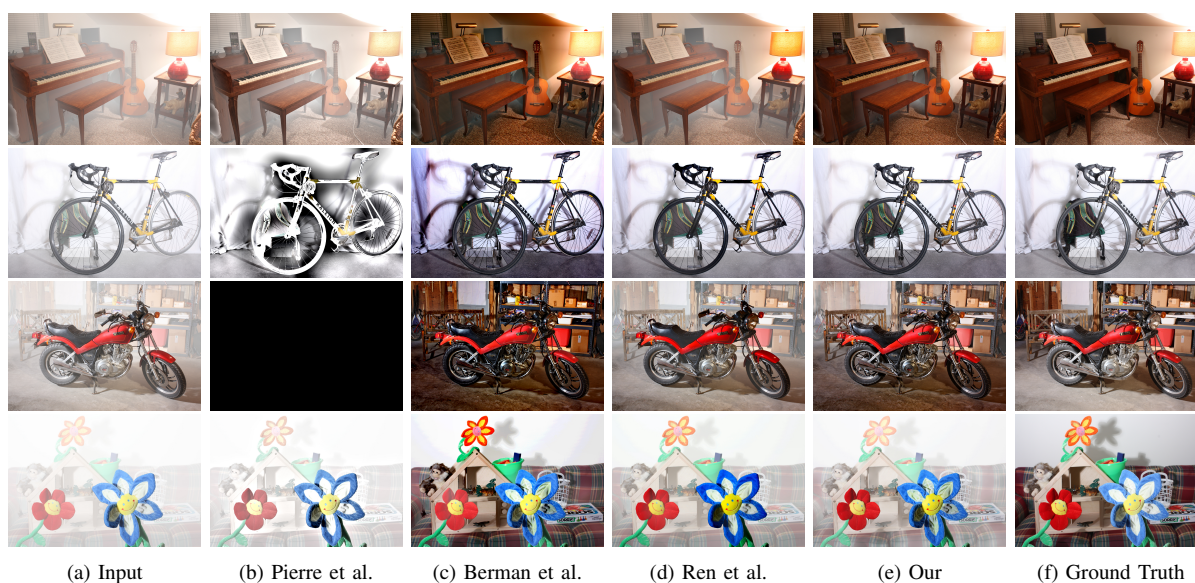


Fig. 7. Results of Middlebury portion of D-Hazy dataset on *Piano*, *Bicycle1*, *Motorcycle*, and *Flowers*.

TABLE III
QUANTITATIVE RESULTS OBTAINED ON MIDDLEBURY PORTION OF D-HAZY DATASET.

	Pierre et al.		Berman et al.		Ren et al.		Our	
	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000	SSIM	CIEDE2000
Adirondack	0.835	21.835	0.891	10.473	0.897	12.417	0.884	9.824
Backpack	0.901	12.24	0.842	12.511	0.879	9.818	0.85	13.582
Bicycle1	0.678	26.528	0.841	15.617	0.938	4.94	0.959	5.564
Cable	0.595	42.545	0.751	14.764	0.645	29.439	0.608	30.826
Classroom1	0.646	34.637	0.883	7.667	0.74	22.33	0.818	12.075
Couch	0.551	35.422	0.785	10.165	0.618	23.162	0.753	9.907
Flowers	0.757	27.462	0.889	8.316	0.783	21.328	0.814	19.58
Jadeplant	0.545	37.706	0.716	11.508	0.606	27.65	0.659	24.411
Mask	0.842	14.738	0.816	12.318	0.85	13.317	0.845	14.365
Motorcycle	0.018	37.864	0.633	18.235	0.819	14.893	0.79	15.918
Piano	0.643	28.187	0.814	9.263	0.715	17.346	0.89	5.626
Pipes	0.015	26.941	0.782	10.926	0.688	21.626	0.761	12.888
Playroom	0.703	25.18	0.815	10.386	0.776	15.074	0.863	8.201
Playtable	0.778	24.066	0.9	8.746	0.86	13.044	0.909	7.577
Recycle	0.904	15.963	0.925	10.87	0.952	7.8	0.94	9.357
Shelves	0.874	17.326	0.916	8.42	0.944	7.701	0.924	7.772
Shopvac	0.602	42.268	0.788	16.74	0.667	32.435	0.735	23.386
Sticks	0.925	14.118	0.953	6.423	0.961	5.398	0.93	7.34
Storage	0.769	27.53	0.869	8.452	0.824	18.971	0.855	15.748
Sword1	0.874	16.832	0.853	14.629	0.914	10.191	0.85	14.584
Sword2	0.821	19.907	0.913	9.82	0.885	13.997	0.881	14.769
Umbrella	0.872	16.507	0.917	9.672	0.909	12.203	0.903	12.728
Vintage	0.926	11.2	0.795	14.895	0.969	5.31	0.939	7.612
Average	0.698	25.087	0.838	11.339	0.819	15.669	0.841	13.201

method of Berman et al. [18] tends to over-enhance the results a bit. Table III reveals that on an average our method performs quite well.

B. Qualitative Results

For the real world images we qualitatively evaluate the results as we don't have ground truth for these images. We show the results for 8 different outdoor images (as shown in Fig.8 and Fig.9) to 11 persons in our lab and ask them to arrange in order of quality. Here we report the overall evaluation of the results as follows. The method of Pierre et al. [2] is not able to clear dense haze and color cast due to haze, and the method of Berman et al. [18] tend to over-

enhance the results. For both the methods these trends is also observed in the quantitative evaluations section. Also in line with observation made in the previous section, the colors of the results of Ren et al. [17] and our method is different from results of obtained by other methods because of different way of choosing atmospheric light. This also shows that the correct estimation of transmittance depends on atmospheric light. Fattal [14] achieves better results for *hongkong* and *herzeliya* images, but for *ny12* image the color bias is clearly visible. Blueish tint is present in the result for *ny12* image of Fattal [14], but that is not present in the output of our method with the given value of 'A'. Finally, For all other images our method achieves comparable results.



Fig. 8. Results of real world images of Fattal dataset on *dubai*, *florence*, *herzeliya*, *hongkong*, and *ny12*.

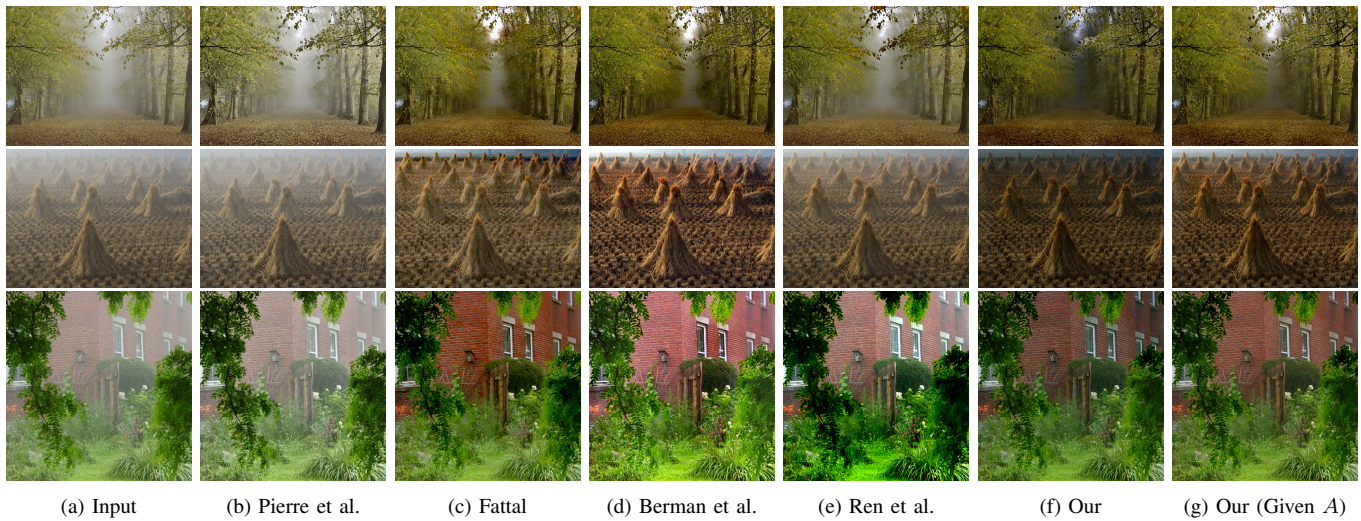


Fig. 9. Results of real world images of Fattal dataset on *forest*, *cones*, and *house*.

VII. CONCLUSION

In this paper we have proposed an image dehazing method that tries to estimate transmittance in each patch by comparing the dehazed version with the input hazy one. The comparison is done by our proposed patch quality comparator. With this CNN based comparator in our hand, we employ binary search to find transmittance in each patch. Although we have used the method of Dark Channel Prior to compute environmental illumination, the results show it is not always accurate. The output greatly improves with correct environmental illumination. This shows that the environmental illumination is crucial in dehazing an image, although it has not received

the required attention. The future work could be focused on accurate estimation of environmental illumination for both day and night time cases.

REFERENCES

- [1] J. Oakley and H. Bu, "Correction of Simple Contrast Loss in Color Images," *IEEE Transactions on Image Processing*, vol. 16, pp. 511–522, 2007.
- [2] F. Pierre, J.-F. Aujol, A. Bugeau, G. Steidl, and V.-T. Ta, "Variational Contrast Enhancement of Gray-Scale and RGB Images," *Journal of Mathematical Imaging and Vision*, vol. 57, pp. 99–116, 2017.
- [3] S. G. Narasimhan and S. K. Nayar, "Vision and the Atmosphere," *International Journal of Computer Vision*, vol. 48, pp. 233–254, 2002.

- [4] —, “Contrast restoration of weather degraded images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 713–724, 2003.
- [5] S. Shwartz, E. Namer, and Y. Schechner, “Blind Haze Separation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 1998–1991.
- [6] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski, “Deep photo: Model-based photograph enhancement and viewing,” *ACM Transactions on Graphics*, vol. 27, pp. 116:1–116:10, 2008.
- [7] S. G. Narasimhan and S. K. Nayar, “Interactive (de) weathering of an image using physical models,” in *IEEE Workshop on Color and Photometric Method in Computer Vision*, vol. 6, 2003.
- [8] N. Hautière, J.-P. Tarel, and D. Aubert, “Towards Fog-Free In-Vehicle Vision Systems through Contrast Restoration,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.
- [9] R. T. Tan, “Visibility in bad weather from a single image,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008, pp. 1–8.
- [10] R. Fattal, “Single Image Dehazing,” *ACM Transactions on Graphics*, vol. 27, pp. 72:1–72:9, 2008.
- [11] K. He, J. Sun, and X. Tang, “Single Image Haze Removal Using Dark Channel Prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 2341–2353, 2011.
- [12] J.-H. Kim, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, “Optimized contrast enhancement for real-time image and video dehazing,” *Journal of Visual Communication and Image Representation*, vol. 24, pp. 410–425, 2013.
- [13] K. Tang, J. Yang, and J. Wang, “Investigating Haze-Relevant Features in a Learning Framework for Image Dehazing,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2995–3002.
- [14] R. Fattal, “Dehazing Using Color-Lines,” *ACM Transactions on Graphics*, vol. 34, pp. 13:1–13:14, 2014.
- [15] Q. Zhu, J. Mai, and L. Shao, “A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior,” *IEEE Transactions on Image Processing*, vol. 24, pp. 3522–3533, 2015.
- [16] 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*, vol. 25, pp. 5187–5198, 2016.
- [17] 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 (ECCV)*, 2016, pp. 154–169.
- [18] D. Berman, T. Treibitz, and S. Avidan, “Non-local Image Dehazing,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1674–1682.
- [19] J. Zhang, Y. Cao, and Z. Wang, “Nighttime haze removal based on a new imaging model,” in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 4557–4561.
- [20] Y. Li, R. T. Tan, and M. S. Brown, “Nighttime Haze Removal with Glow and Multiple Light Colors,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 226–234.
- [21] C. Ancuti, C. O. Ancuti, C. D. Vleeschouwer, and A. C. Bovik, “Nighttime dehazing by fusion,” in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2256–2260.
- [22] D. Park, D. K. Han, and H. Ko, “Nighttime image dehazing with local atmospheric light and weighted entropy,” in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2261–2265.
- [23] S. Santra and B. Chanda, “Day/night unconstrained image dehazing,” in *23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 1406–1411.
- [24] Y. Li, S. You, M. S. Brown, and R. T. Tan, “Haze visibility enhancement: A Survey and quantitative benchmarking,” *Computer Vision and Image Understanding*, vol. 165, pp. 1–16, 2017.
- [25] H. Koschmieder, “Theorie der horizontalen Sichtweite,” *Beitrage zur Physik der freien Atmosphere*, pp. 33–53, 1924.
- [26] W. E. K. Middleton, “Vision through the atmosphere,” in *Geophysics II/Geophysics II*. Springer, 1957, pp. 254–287.
- [27] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
- [28] L. K. Choi, J. You, and A. Bovik, “Referenceless Prediction of Perceptual Fog Density and Perceptual Image Defogging,” *IEEE Transactions on Image Processing*, vol. 24, pp. 3888–3901, 2015.
- [29] M. D. Zeiler, “ADELTA: an adaptive learning rate method,” *arXiv e-prints*, vol. abs/1212.5701, 2012.
- [30] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.
- [31] Theano Development Team, “Theano: A Python framework for fast computation of mathematical expressions,” *arXiv e-prints*, vol. abs/1605.02688, 2016.
- [32] C. Ancuti, C. O. Ancuti, and C. D. Vleeschouwer, “D-HAZY: A dataset to evaluate quantitatively dehazing algorithms,” in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2226–2230.
- [33] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, “High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth,” in *German Conference on Pattern Recognition (GCPR)*, 2014, pp. 31–42.
- [34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor Segmentation and Support Inference from RGBD Images,” in *European Conference on Computer Vision (ECCV)*, 2012, pp. 746–760.
- [35] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.
- [36] G. Sharma, W. Wencheng, and E. N. Dalal, “The CIEDE2000 color difference formula: Implementation notes, supplementary test data, and mathematical observations,” *Color Research & Application*, vol. 30, pp. 21–30, 2004.
- [37] Z. Wang and A. C. Bovik, “Modern Image Quality Assessment,” *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, pp. 1–156, 2006.



Sanchayan Santra Sanchayan Santra is currently a research scholar in Indian Statistical Institute, Kolkata, India. Before that he was at Ramakrishna Mission Vivekananda Educational and Research Institute, Belur Math, West Bengal, India, pursuing his master’s degree. His research interest include Computational Photography, Computer Vision and Image Processing.



Ranjan Mondal Ranjan Mondal is currently pursuing his Ph.D after completing his master’s from Indian Statistical Institute, Kolkata. His research interest lie at the intersection of Deep Learning and Reinforcement Learning with Computer Vision and Image Processing.



Bhabatosh Chanda Bhabatosh Chanda received the B.E. degree in electronics and telecommunication engineering and the Ph.D. degree in electrical engineering from the University of Calcutta, Kolkata, India, in 1979 and 1988, respectively. He is currently a Professor with the Indian Statistical Institute, Kolkata, India. He received the Young Scientist Medal of the Indian National Science Academy in 1989, the Vikram Sarabhai Research Award in 2002, and the IETE-Ram Lal Wadhwa Gold Medal in 2007. He is a fellow of the Institute of Electronics and Telecommunication Engineers, the National Academy of Science, India, and the Indian National Academy of Engineering, and the International Association of Pattern Recognition.