

GuidedNet: Single Image Dehazing Using an End-to-end Convolutional Neural Network

Lucas T. Gonçalves, Joel O. Gaya, Paulo Drews-Jr, Silvia S. C. Botelho
Intelligent Robotics and Automation Group - NAUTEC
Center for Computational Science - C3
Universidade Federal do Rio Grande - FURG
Rio Grande, Brazil
Emails: {lucas.teixeira, joelfelipe, paulodrews, silviacb}@furg.br

Abstract—Poor visibility is a common problem when capturing images in participating mediums such as mist or water. The problem of generating a haze-free image based on a hazy one can be described as image dehazing. Previous approaches dealt with this problem using physical models based on priors and simplifications. In this paper, we demonstrate that an end-to-end convolutional neural network is able to learn the dehazing process with no parameters or priors required, resulting in a more generic method. Even though our model is trained entirely with hazy indoor images, we are able to fully restore outdoor images with real haze. Also, we propose an architecture containing the novel Guided Layers, introduced in order to reduce the loss of spatial information while restoring the images. Our method outperforms other machine learning based models, yielding superior results both qualitatively and quantitatively.

I. INTRODUCTION

Particles in the air interact with the light rays when capturing images in an external environment. In a hazy day, this interaction results in an image that differs from an image captured in a sunny day. The main phenomenon that cause this effect is called *scattering*. This effect causes an information attenuation increasing exponentially with the distance between the scene and the observer.

Scattering causes information loss and also adds noise to the image, producing *Forward Scattering* and *Backscattering*. The former occurs when light rays originated from the scene are scattered in small angles, reaching neighboring pixels, creating a blur in the image. This phenomenon is usually neglected due to its little impact [1]. In fact, light rays originated from outside the scene are scattered into the camera in hazy days, creating a partially opaque covering on the scene. Even on sunny days, distant objects in the scene present lower contrast and tend to be covered by a white mist.

Image dehazing is the procedure of taking a hazy image as input and removing the degradation effect caused by the scattering phenomenon, resulting in a haze-free image.

Originally, state-of-the-art dehazing techniques relied on an image formation simplified physical model, shown in Eq. 1. In this model, an image is described as a superposition of the scene’s radiance and the scattering effects. This model is widely adopted in classical hazy image modeling [2], [3] and is described as:

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

where I is the final hazy image, J is the scene radiance, A is the global atmospheric light and t is the medium transmission that determines the amount of light that reaches the observer. While these methods achieve good performance when the parameters and priors are well adjusted, they fall short when applied in a more generic context, such as real-time dehazing in unpredictable conditions. This occurs because these methods are based on strong assumptions and require the adjustment of several parameters related to the image formation, which are not always available.

Deep Neural Networks are constantly being used to successfully tackle image transformation problems, such as image denoising [4], [5], image colorization [6], [7], image super-resolution [8], depth prediction [9] and, recently, image dehazing with models that use convolutional neural networks (CNNs) by either estimating physical parameters [10]–[12] to apply in the image formation model showed in Eq. 1, or with an end-to-end model that yields directly the restored image [13].

One of the main problems with neural network models applied to dehazing is the loss of the original scene’s structural integrity. This adversity causes the visual results to be unrealistic and decreases significantly the output score when evaluating with structural similarity metrics such as the *SSIM* [14]. Nowadays, the main approach to handle this problem is the usage of *skip connections* [15] in the network architecture, and post processing using *guided filter* [16].

However, we propose in this paper a new approach that addresses the structural similarity issue. Our model is composed of a *single end-to-end* CNN to fully dehaze images. It uses a transformation neural network that takes a single hazy image as input and yields a fully restored image. The fact that our method does not require any additional pre or post-processing techniques is important, since it improves the model’s simplicity. Also, our model does not rely on human-developed priors or the physical model described in Eq. 1, enabling it to be used in a fully automated system. Furthermore, the usage of CNNs that are expensive to training but fast to execute enables us to operate in real time using an appropriate hardware.

We introduced *Guided Layers* in order to maintain the structural integrity of the image. These layers are based on the

guided filter [16], a technique often used in post processing on classic physical dehazing models. In this work, this technique is brought into the neural network as a layer instead of using it on post processing. The main advantage of the guided layer is that it tackles the structural integrity problem multiple times during the training instead of only once on the post-processing. This allows our network to backpropagate already filtered results with more structural integrity.

There are two main adversities when tackling the image dehazing problem using a deep learning approach: the lack of pairs of haze and haze-free image and the loss of structural integrity in the restored image. We deal with both of these problems. The first one is dealt using an image set with indoor images with precise synthetic haze. The latter is tackled using the proposed *guided layer* and the already consolidated *skip connections*.

Contributions: Our work introduces an *end-to-end CNN* designed to solve the single image dehazing problem to fully restore hazy input images. Our model achieves this without the requirement of adjusting parameters depending on the image’s conditions. Thus, obtaining a more generic method, allowing our model to be applied in a large range of situations. Also, we introduced the novel Guided Layers, which takes the *guided filter* into account when adjusting the weights of the network. In addition, our method is able to restore images taken of *real outdoor* hazy scenes, even though our training set is composed entirely of *indoor scenes* with *simulated* haze.

II. RELATED WORKS

Image dehazing task is generally approached with physical methods based on the simplified image formation model [17], [18], described in Eq. 1. Several dehazing methods [19]–[22] are based on priors to solve ill-posed problem using the image formation model. The most successfully is the *Dark Channel Prior* [20], where the minimum value between the color channels in a patch of the image is used as a transmission indicative. Although effective with the correct priors and parameters, this idea can fail in several situation and require the manual tuning of many critical parameters such as the patch size. Similarly, other methods based on priors present similar drawbacks.

Recently, methods were developed using a CNN to estimate the medium transmission t . [10] uses a CNN to estimate a medium transmission based on 16×16 RGB image patches. This method trains the network using as input synthetic hazy color images and, as the ground truth, a single transmission value for all pixels in the 16×16 patch. The authors propose to refine the transmission applying the *guided filter* [16] after the network’s last layer to fully restore an image. The refined transmission is applied in Eq 1 to obtain the final result. Differently, [23] divides the model into two separate CNNs. The first one roughly estimates the medium transmission based on the entire image. The second network locally refines the transmission. During the network training, each pixel is associated with a single transmission value. That approach achieves a more precise result than [10].

While those methods address the medium transmission estimation problem, they do not use the neural network to understand the image formation phenomenon. [11] proposed a dehazing network based on a reformulated light propagation model, where the transmission and the atmospheric light are estimated jointly and the restoration is optimized directly. However, he network still relies on the simplified model. Differently, [13] introduced an end-to-end model to solve the dehazing problem that learns the effects caused by haze. This method’s approach is the most similar to ours, however, they do not deal with the structural information loss in the neural network. Our method addresses this problem using a different network architecture, yielding superior results.

Using a CNN to only estimate a transmission map does not solve the limitation presented by the simplified physical models, since these methods still require priors to estimate parameters such as the global atmospheric light. Also, simplifications are used in these models, since neither the ambient light nor the minimum transmission are estimated. Furthermore, the simplified image formation model described in Eq. 1 is generally adopted.

Differences to this work: Previous methods adopt an already consolidated physical image formation model. Instead, our work proposes an approach using an end-to-end CNN that addresses the structural integrity problem. Our goal is to understand whether an end-to-end neural network integrated with *guided layers* is capable of learning the physical world’s complexity, being able to solve the image dehazing task without any additional hand tuned parameters or refining methods.

III. METHODOLOGY

One of the greatest challenges when developing deep learning models is to achieve a data set with a large quantity of input/ground-truth pairs in order to train the model properly. In our case, *hazy* and *haze-free* images pairs are required. It is also essential both images in each pair are captured under the same light conditions, making it really difficult to capture realistic images of the same scene with real haze.

In this section, we firstly explain how we proceeded to overcome the problem of gathering a large number of image pairs eligible to train our network. Then, we describe our network, explain the training process and give details about our model’s implementation.

A. Acquiring Data

Ideally, the image set used to train a neural network designed to solve the dehazing problem would be composed of image pairs of the same scene with and without haze, under the same light conditions. Also, such scene would be outdoors, since the haze effect usually does not occur indoors. However, capturing images in which this requirement is met is extremely difficult [24]. Besides, the pair amount needed to train a deep neural network is extensive, making it impractical to use a data set composed of real images.



Fig. 1. Examples of images generated by the haze simulator. First row: haze-free images captured indoors, used both as base image in the simulator and as ground-truth in the network training. Second row: scene’s depth map, used to estimate the transmission of object in the scene in order to apply the simulated haze degradation. Third row: hazy images, synthesized by the simulator and used as input for the network.

Since acquiring a data set with pairs of real haze-free and hazy images would be difficult, we decided to generate synthetic data. Using the image formation model described in Eq. 1, we can use the scene’s depth to estimate the transmission and apply simulated haze on haze-free images. Ideally, our data would be context-specific images (outdoors) with a precise depth map, in which we could generate synthetic hazy images nearly indistinguishable from real ones. However, current sensors are not yet able to correctly capture depth information outdoors, making our ideal data set unobtainable.

Outdoor images would permit us to create a larger and more diversified training set composed of outdoor hazy images, which is exactly the image type we intend to restore using the trained model. However, if we decided to use outdoor images with poorly-captured depth maps, we would end up with images with lack of spatial variation in haze intensity, making the simulated haze inaccurate.

Aware of these adversities, we decided to generate synthetic data using haze-free indoor images. This permits us to synthesize a much more realistic haze degradation effect, even though the information present in the scenes are out of context. We expect our model to learn to remove the haze phenomenon itself, independent of the context behind it. With that in mind, we preferred to synthesize a realistic haze effect in an unrealistic context instead of generating hazy images in a realistic context but with an inaccurate representation of the phenomenon our model needs to remove.

We need the scene’s atmospheric attenuation coefficients and the global illumination color in order simulate the haze phenomenon realistically. Those can be calculated based on patches with minimal transmission extracted from real hazy images. Also, we also need the scene’s depth map, which could be easily obtained using a *Kinect* camera, since we decided to use indoor images.

In order to synthesize the hazy images, we adopted the simulator proposed by [24]. It is able to generate a turbid underwater version of an input clear image, based on the

depth map and a turbidity patch. We were able to adjust this model to, instead of using a turbidity patch, use a haze patch, generating an input clear image’s hazy version. This is possible only because the underwater turbidity and haze phenomena are physically similar.

A set of hazy and haze-free image pairs was developed using the images captured with the *Kinect* camera, simulating the haze with the image formation model described in 1. Since all haze is synthetic, we are able to adjust simulator’s parameters to generate multiple hazy images originated from the same scene using several haze degradation levels.

Examples generated by the simulator can be seen in Fig 1. Images in the third row with synthetic haze were generated based on the indoor clear images in the first row.

B. Network Architecture

Our approach is to use a transformation neural network that, given a hazy image patch, is able to greatly reduce or even completely remove the degradation caused by the haze phenomenon.

The transformation neural network used in this work can be seen in Fig. 2. This network architecture is based on the *unet* architecture proposed in [15]. We decided to use an architecture similar to this one due to the fact that an *encoder-decoder* network has a large enough receptive field, even with a limited number of layers. Also, the *skip connections* enable some of the structural information from the input to be transferred to the output. This happens because the *skip connection* concatenates a feature map produced by an encoding layer directly to a feature map in the decoding part of the network.

The main difference from our architecture to *unet’s* is the presence of the *guided layers*, inspired by the *guided filter* [16]. This novel layer type was introduced in order to reduce the structural information loss, which is a common problem in the dehazing problem, due to the noise being spatially variant. Many models seek to reduce structural loss, some use an explicit stage used to refine the transmission estimative [10], [20], [22],

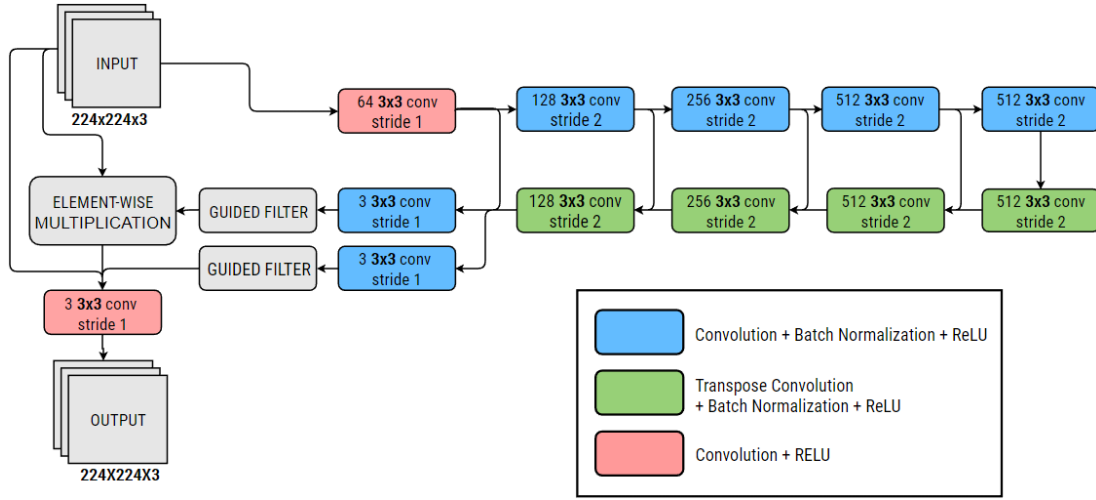


Fig. 2. Network architecture. All convolution and transpose convolution layers use *zero padding* set to one. Each convolution layer is described with a color and a text as follows: [number of feature maps] *conv* [filter dimensions] *stride* [stride used].

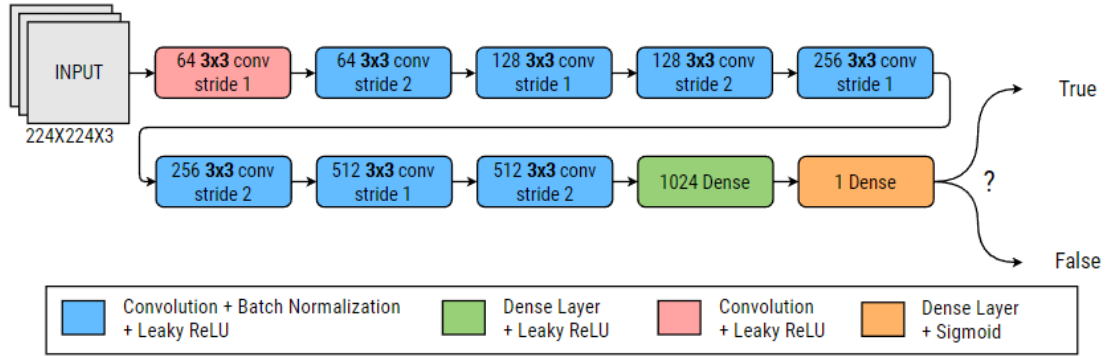


Fig. 3. Discriminator network’s architecture. All convolution layers use *zero padding* set to one. Each convolution layer is described with a color and a text as follows: [number of feature maps] *conv* [filter dimensions] *stride* [stride used]. The value in the dense layers represents the number of weights.

[25] and others use the architecture itself with *skip connections* [11], [23]. Our approach, although also using *skip connections*, differs from previous models because it uses the *guided filter* as a part of the network architecture. This means that its impact is considered when updating the weights during the training.

C. Guided Layer

In this work, we introduce a novel layer type based on [16]’s *guided filter*. The *guided filter* is a local linear model that receives two images as input: an image to be filtered and the guidance image. The model is used to transfer the guidance image’s structural information to the filtering output.

We implemented the guided filter [16] in Tensorflow as a function that can be used as a layer. Our model uses the guided layer with the network’s input image as the guide in order to reduce the loss of structural integrity of the last *decode layer*, which outputs an image with the same shape as the input. In the end, we concatenate the outputs of three layers: the guided layer, the input and the guided layer multiplied element-wise with the input. The result of this concatenation

is passed through a last convolution layer, resulting in the output haze-free image.

D. Training

Our model is implemented using Tensorflow [26]. The network’s input is a batch of 16 hazy images generated by the simulator and its ground-truth is this batch’s respective haze-free images. We trained the model using the *Adam* Optimizer [27] for 25 epochs, which is enough for the loss function to stabilize. The parameters used for the optimizer were:

- Learning rate: 10^{-4} ,
- β_1 : 0.9,
- β_2 : 0.999,
- ϵ of 10^{-8} .

Also, in order for the network to learn the desired robustness and invariance properties, we use data augmentation to enlarge the image set. We used four rotation degrees, each with two mirrored horizontally. With this setup, we multiply the size of our data set by eight.

The loss function used in order to adjust the network weights is a combination of the *feature loss* [28] with an adversarial loss network.

The *feature loss* seeks to ensure the output and ground-truth images have similar high-level features. This loss uses a classification neural network, thus, its first layers are trained to extract features semantically relevant. The loss function can be described as:

$$\ell_{feat} = \frac{1}{H_j W_j D_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2, \quad (2)$$

where $\phi_j(x)$ are the activations of the network's j -th layer used by the loss function when processing the image x and H_j , W_j and D_j are its dimensions. In this work, we adopted the 16-layer *VGG* [29] trained with the *ImageNet* dataset [30] as the *feature loss* network. The j value was set to the third convolutional layer.

Using adversarial loss networks, we aim to apply the concept of *generative adversarial networks* (GANs) [31] so that the network output ends up more similar to haze-free images. The adversarial loss uses a *discriminator* network to produce a single value representing the probability in which an image comes from the dataset rather than the network output. In our work, we used the discriminator network proposed in [8], which is illustrated in Fig. 3.

Since the discriminator function does not receive any information from the input, we use it combined with a loss function that considers the input. The combination produced is described in Equation 3:

$$\ell_{feat+GAN} = \ell_{feat} + \lambda \ell_{GAN}, \quad (3)$$

where λ is a parameter that determines the adversarial loss influence.

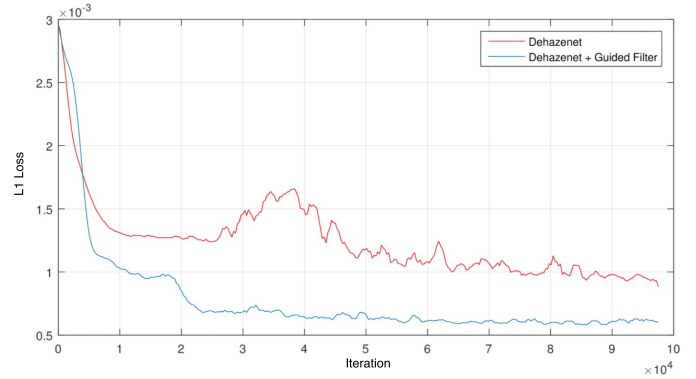
IV. RESULTS

The first step to test our method was to verify whether the Guided Layer would bring any improvement to a CNN architecture. With that in mind, we tested it with two dehazing neural network methods: Dehazenet [10] and *MSCNN* [23]. Both models estimate the transmission based on a hazy image to subsequently use it in the restoration process. We trained both models using the original work architecture and compared it to a version introducing the *guided filter* as a part of the network architecture (as seen in Fig 2, where the guide image is the same as the input).

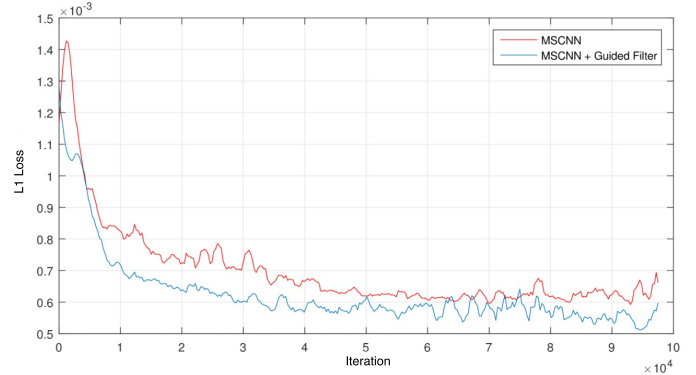
Both models were trained in a set of image and transmission map pairs with 224×224 dimensions. Of these pairs, 90% were in the training set and the remaining 10% in the validation set. The loss function used in this evaluation was the L1 Loss, a loss function that seeks to minimize the difference between the output and the ground-truth image pixel-wise. This function can be described as

$$\ell_{L1} = \frac{\|\hat{y} - y\|_1}{HWD}, \quad (4)$$

where \hat{y} is the network output, y is the ground-truth image and H , W and D are their height, width and depth, respectively.



(a) Dehazenet



(b) MSCNN

Fig. 4. The L1 loss value while training the (a) Dehazenet [10] and (b) MSCNN [23] compared to their respective versions with the guided layer. These experiments used $\epsilon = 10^{-3}$ and a 40×40 window in the guided layer.

The networks were trained for 75 epochs using the *Adam* Optimizer [27], with a learning rate of 10^{-5} , exponential decay rate β_1 of 0.9 and β_2 of 0.999. The numerical stability constant value ϵ adopted was 10^{-8} .

The results, as shown in Fig 4, reveal that the *guided layer* addition caused both models to have an increased performance during the training. This indicates the usage of this layer can be advantageous even in methods where the networks already uses *skip connections* to preserve the image's structural integrity, such as the *MSCNN* [23].

In this work, we decided to evaluate our method's results quantitatively using synthetic images and qualitatively using real images. We compared our method with the main physical models: CLAHE [32], Tarel [19], Ancuti & Ancuti [33], He et al. [20], Meng et al. [21], Fattal [22]. Our method was also compared with the main machine learning based methods: MSCNN [23], Dehazenet [10], AOD [11] and DeepDive [13].

All results presented in this Section were obtained using the parameters described in Section III. Our model's inputs are hazy synthetic $224 \times 224 \times 3$ images. These images were generated from a combination of the images acquired using the Kinect camera and the *NYU-Depth V2* dataset [34]. In order to generate the synthetic haze in this haze-free *RGBD* images, we used a haze version of the simulator proposed by [24].



Fig. 5. Comparative using the Flags image. In this figure, (a) represents the input, (b) is an image made available by the authors, (c), (d) and (e) are images generated by the code made available by the respective authors and (f) is our model’s output.

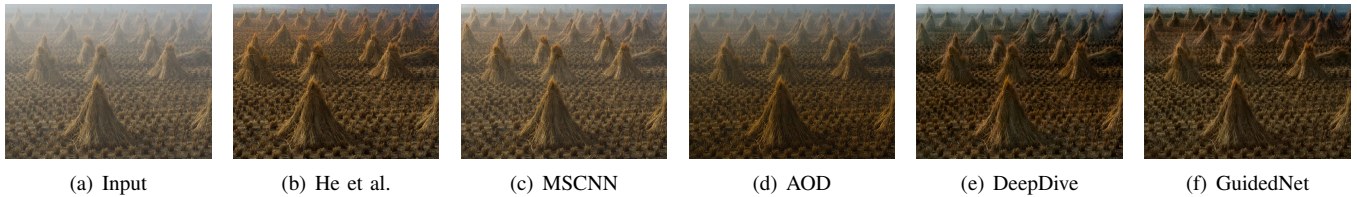


Fig. 6. Comparative using the Cones image. In this figure, (a) represents the input, (b) is an image made available by the authors, (c), (d) and (e) are images generated by the code made available by the respective authors and (f) is our model’s output.

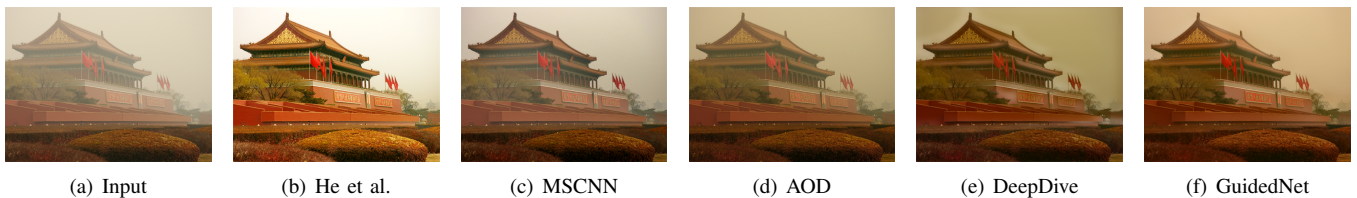


Fig. 7. Comparative using the Tiananmen image. In this figure, (a) represents the input, (b) is an image made available by the authors, (c), (d) and (e) are images generated by the code made available by the respective authors and (f) is our model’s output.

A. Qualitative Analysis

Even though our network was trained entirely with indoor hazy images, we used only real outdoor images for the qualitative analysis. All image outputs were produced by a model trained only *once*, without the need to adjust any parameters or any pre or post-processing stages.

The results obtained using the physical methods proposed by [20] are more pleasant visually, however, these methods require prior parameter adjusting to restore each image. Our method results were obtained using a neural network trained only *once*, allowing it to be used in applications requiring real-time response, such as embedded systems in numerous areas.

As seen in Figure 5, many of the models were able to produce visually satisfying results. However, most of them failed when dehazing the white sign on the top right, darkening it. This happened because most methods assume whitened areas have a greater chance to be haze, yielding over-corrected results in white or gray regions. Our model was able to successfully distinguishing real white objects from haze.

An interesting case is shown in Figure 6. The scene presents

basically the same structure distributed at various distances from the camera with distinct degradation levels. In this image, our method produced visually satisfying images and the visibility was considerably increased. In this example, the *MSCNN* [23] and *AOD* [11] were not able to successfully dehaze the entire image, only the parts near the camera.

Lastly, Figure 7 shows our method is capable of maintaining the input’s structural integrity and also displays its robustness when treating a haze phenomenon that does not scatter color channels equally. Also, differently from the physical methods, our model does not saturate the sky, maintaining the original exposure.

B. Quantitative Analysis

Two methods were used in our quantitative analysis: the structural similarity index (SSIM) [14], a metric that considers the similarity between the structures in the evaluated and reference images in grayscale. Since the color correction is one of the most important parts in image dehazing, we also evaluate the results using the CIEDE2000 [35], a metric that evaluates the color difference between the images.

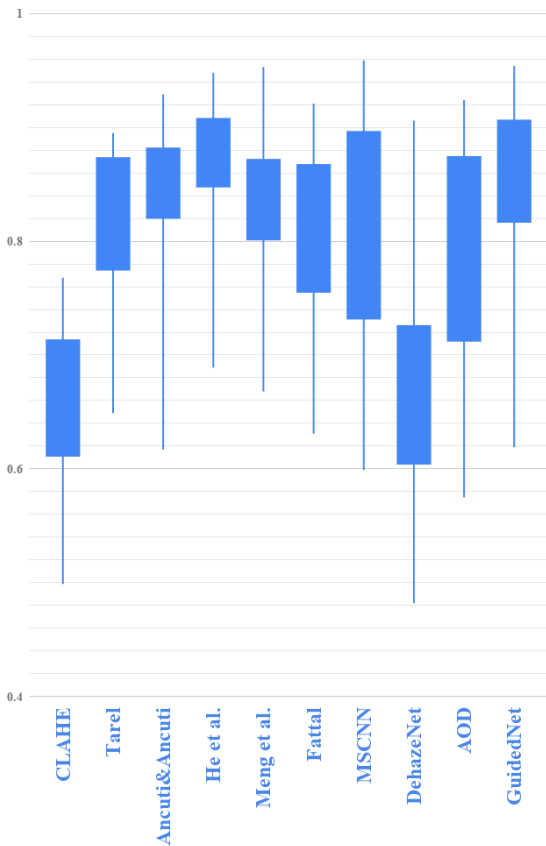


Fig. 8. Quantitative analysis using the Middlebury part of the D-Hazy dataset using the SSIM metric. The values used to compute the CLAHE, Tarel, Ancuti & Ancuti, He et al., Meng et al. and Fattal values are available in [36]. The values shown for the MSCNN, Dehazenet and AOD methods were computed using the implementation provided by the authors. The last candle in the boxplot corresponds to the method proposed.

The SSIM metric yields a value in the $[-1, 1]$ interval, where 1 is the best result with two identical images. The CIEDE2000, however, produces values between 0 and 100, where smaller values represent better color preservation. The dataset used to perform this comparison was the D-Hazy [36], which is composed by images with synthetic haze originated from the *Middlebury* [37] and *NYU-Depth V2* [34] datasets. However, the images originated from the *NYU-Depth V2* dataset were not used, since they were used to generate our training set.

Our work outperformed every other machine learning based method compared, as can be seen in Figures 8 and 9. These figures display the results using the box and whisker plot, where the line displays the minimum and maximum value and the box limits display the upper and lower quartiles. Even though our method's performance was slightly worse than the physical based method proposed in [20], it is important to remember their model has parameters that should be adjusted for each image in order to obtain these results, while our model uses a single network trained only once to treat all the images,

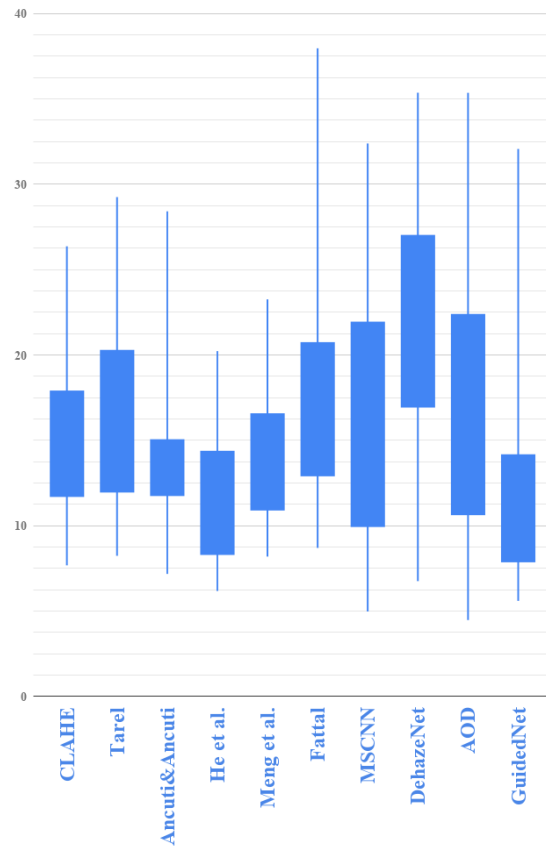


Fig. 9. Quantitative analysis using the Middlebury part of the D-Hazy dataset using the CIEDE2000 metric. The values used to compute the CLAHE, Tarel, Ancuti & Ancuti, He et al., Meng et al. and Fattal values are available in [36]. The values shown for the MSCNN, Dehazenet and AOD methods were computed using the implementation provided by the authors. The last candle in the boxplot corresponds to the method proposed.

resulting in a model that could be used in a much wider range of situations without the need to adjust any parameters. If compared to the *AOD* [11], which takes a similar approach, our method presents superior results in both metrics.

V. CONCLUSIONS

In this work, we demonstrated that neural networks are capable of solving the dehazing task without any additional pre or post-processing, resulting in a generic method that can be implemented in systems for real-time application. Also, it was shown the *guided layers* and *skip connections* are able to soften the structural integrity problem, yielding results with more structural information and more visually pleasant.

We were able to achieve results superior to the other machine learning based methods, including [13], a model that uses a similar approach to the dehazing problem.

The beneficial results obtained using the novel *guided layers* to this application indicate it can be used to improve results in similar image transformation tasks, such as colorization, denoising and super-resolution.

Also, despite being trained entirely with indoor images with simulated haze, our model was able to successfully restore external images with real haze. We believe, however, that the next step to improve our method's results is to obtain a dataset with scenes more similar to outdoors and with a more complex haze simulation, providing a sensible context with high precision haze simulation.

ACKNOWLEDGMENT

The authors would like to thank the Brazilian Petroleum Corporation - Petrobras and the Brazilian National Agency of Petroleum, Natural Gas and Biofuels (ANP) to Funding Authority for Studies and Projects (FINEP) and to Ministry of Science and Technology (MCT) for their financial support through the Human Resources Program of ANP to the Petroleum and Gas Sector - PRH-ANP/MCT. This paper is also a contribution of the Brazilian National Institute of Science and Technology - INCT-Mar funded by CNPq Grant Number 610012/2011-8 and partly funded by CAPES and FAPERGS.

REFERENCES

- [1] Y. Y. Schechner and N. Karpel, "Recovery of underwater visibility and structure by polarization analysis," *IEEE Journal of oceanic engineering*, vol. 30, no. 3, pp. 570–587, 2005.
- [2] H. Koschmieder, "Theorie der horizontalen sichtweit," *Beitr. Phys. Freien Atm.*, vol. 12, pp. 171–181, 1924.
- [3] E. J. McCartney, *Optics of the atmosphere: scattering by molecules and particles*. John Wiley and Sons, Inc., 1976.
- [4] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 769–776.
- [5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.
- [6] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *ECCV*, 2016.
- [7] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 119:1–119:11, Jul. 2017.
- [8] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2016.
- [9] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3D Vision (3DV), 2016 Fourth International Conference on*. IEEE, 2016, pp. 239–248.
- [10] 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, no. 11, pp. 5187–5198, 2016.
- [11] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4770–4778.
- [12] 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*. Springer, 2016, pp. 154–169.
- [13] L. T. Goncalves, J. O. Gaya, P. Drews-Jr, and S. S. C. Botelho, "Deepdive: An end-to-end dehazing method using deep learning," in *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2017, pp. 436–441.
- [14] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [16] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European conference on computer vision*. Springer, 2010, pp. 1–14.
- [17] J. S. Jaffe, "Computer modeling and the design of optimal underwater imaging systems," *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, 1990.
- [18] B. McGlamery, "A computer model for underwater camera systems," in *Ocean Optics VI*, vol. 208. International Society for Optics and Photonics, 1980, pp. 221–232.
- [19] J.-P. Tarel and N. Hautiere, "Fast visibility restoration from a single color or gray level image," in *12th International Conference on Computer Vision*. IEEE, 2009, pp. 2201–2208.
- [20] 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, no. 12, pp. 2341–2353, 2011.
- [21] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 617–624.
- [22] R. Fattal, "Dehazing using color-lines," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 13, 2014.
- [23] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.
- [24] A. Duarte, F. Codevilla, J. Gaya, and S. S. C. Botelho, "A dataset to evaluate underwater image restoration methods," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–6.
- [25] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, p. 72, 2008.
- [26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.
- [27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [28] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [32] K. Zuiderveld, "Contrast limited adaptive histogram equalization," *Graphics gems*, pp. 474–485, 1994.
- [33] C. O. Ancuti and C. Ancuti, "Single image dehazing by multi-scale fusion," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3271–3282, 2013.
- [34] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.
- [35] G. Sharma, W. Wu, and E. N. Dalal, "The ciede2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research & Application*, vol. 30, no. 1, pp. 21–30, 2005.
- [36] C. O. Ancuti, C. O. Ancuti, and C. De Vleeschouwer, "D-hazy: a dataset to evaluate quantitatively dehazing algorithms," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2226–2230.
- [37] N. Nestic, P. Westling, X. Wang, Y. Kitajima, G. Krathwohl, and D. Scharstein. (2014) Middlebury dataset. [Online]. Available: <http://vision.middlebury.edu/stereo/data/scenes2014/>