DeepDive: An End-to-End Dehazing Method Using Deep Learning

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

***Abstract*—Image dehazing can be described as the problem of mapping from a hazy image to a haze-free image. Most approaches to this problem use physical models based on simplifications and priors. In this work we demonstrate that a convolutional neural network with a deep architecture and a large image database is able to learn the entire process of dehazing, without the need to adjust parameters, resulting in a much more generic method. We evaluate our approach applying it to real scenes corrupted by haze. The results show that even though our network is trained with simulated indoor images, it is capable of dehazing real outdoor scenes, learning to treat the degradation effect itself, not to reconstruct the scene behind it.**

I. Introduction

When capturing outdoor images, we are often faced with a partially opaque covering over more distant regions of the scene. This effect is known as haze and it occurs when light propagates through the particles in suspension. In this situation, the light rays can be *absorbed* or *scattered*. Both of these phenomena cause an information attenuation which increases exponentially with distance. This effect is described by the Beer-Lambert law. In addition, *scattering* also adds noise to the image, producing two effects: *Forward Scattering* and *Backscattering*. The first one occurs when light rays coming from the scene are scattered in small angles, reaching neighboring pixels, which creates a blur in the image. This phenomenon is usually neglected because of its little impact. In the second one, light rays coming from outside of the scene are scattered into the camera creating a partially opaque covering on the scene.

An image dehazing procedure takes a hazy image as input and removing the degradation effect, resulting in a haze-free image. There are several approaches to this task.

Most dehazing methods rely on a simplified physical model of the image formation, as shown in Eq. 1. In this model, the image is described as a superposition of scene radiance and the scattering effects and is widely adopted to hazy image modeling [1].

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

where **I** is the 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. These methods attempt to estimate a transmission map using heuristics and priors developed through the observation of haze-free images. This transmission map is then used to restore the image. [2] assumes that the transmission can be defined as the source of covariance. Later, [3] revealed that this assumption only works for low degradation levels and proposed a method to estimate the transmission that uses a *color line assumption* [4]. [5] proposed a model based on the *Dark Channel Prior* that obtains good results. However, the method fails in objects that presents high grayscale levels.

Even though these methods achieve satisfying results, they are based on strong assumptions and require diverse parameters related to the image formation, which are not always available. It is due to the unpredictability of the scene's conditions, causing them to fail in situations where the priors used are not true, such as underwater environments [6], [7] or scenes where the haze is not perfectly white.

Convolutional Neural Networks (CNNs) have been applied successfully to complete many image processing tasks, such as image denoising [8], image colorization [9], image super resolution [10], depth prediction [11] and recently, [12], [13] adopted CNNs to estimate a transmission map in order to dehaze images. Based on it, we developed a new *end-to-end* deep learning model trained entirely with hazy and haze-free image pairs to successfully dehazing images.

Unlike previous approaches that use deep learning, our model takes a hazy image as its only input and outputs a fully restored image, requiring no additional pre or post-processing techniques. Also, it does not rely on human-developed priors. We take advantage of the capability of CNNs to automatically learn complex input-output relations based on data observation, allowing more complex heuristics which were unable to be noticed by humans to be learned. This could result in better restoration results in a wider range of situations.

Contributions: We propose a new end-to-end solution to the single image dehazing problem, directly using a CNN to fully restore hazy input images. Also, despite our model being trained entirely with pairs of simulated hazy indoor images, it is able to restore images from real *outdoor* hazy images. This shows that our proposal is able to learn the phenomenon itself, making our model effective for a larger range of scenes.

The remainder of this paper is organized in the following

way: Section II presents a more in-depth description about the main dehazing models; III explains the methodology used to train the network, to acquire data for training and the implementation details; Section IV evaluates our approach on real hazy scenes and presents a comparison with other methods. Finally, in Section V, we summarize the paper's contributions and present the future research directions.

II. Related Works

Neural networks have already been used in the process of single-image dehazing [12], [13]. These methods use CNNs to learn the process of given an image, estimate a medium transmission map. The network architecture used in [12], for example, takes a small $16x16$ patch and estimates a transmission for one pixel in that patch, feeding multiple patches through the network to estimate one image's transmission map. This approach allows the usage of a simpler architecture, with less weights compared to a network used for *end-to-end* restoration.

Ren et al. [13] proposes a coarse-to-fine CNN for transmission estimation. The Multi-Scale CNN network is composed by two subnetworks: (i) for coarse transmission estimation and (ii) for fine transmission estimation. The first network is trained first, and its output is used to feed the latter. The network uses large convolutional filters, such as 11×11, 9×9 and 7×7.

A CNN is also used to estimate a transmission map in [14], however, this map is used with another purpose. It is used to estimate the distance between the objects and the observer. This information is then used to achieve vision-based obstacle avoidance applied to Autonomous Underwater Vehicles.

Although computationally cheaper and faster, estimating only the transmission map with a CNN does not solve the entire problem present in physical models. It still requires priors to estimate the medium parameters as the global atmospheric light. Furthermore, it uses simplifications, since it does not estimate the ambient light neither the minimum transmission and also uses the simplified model described in Eq. 1.

Differences to this work: Previous methods use a transmission map and a simplified model to dehaze an image. Instead, we propose a pure learning approach, where our network is the only thing between a hazy image and its haze-free version. Our goal is to understand whether an end-to-end neural network is capable of learning the entire process of image dehazing, without the support of additional methods and computations.

III. Methodology

Deep learning is a branch of *machine learning* based on data-driven models composed by many processing layers of non-linear transformations. A *network* is a composition of this layers. Our method is divided in two main parts: Subsection III-A explains the process of training a convolutional neural network to dehaze images without prior information. Then, subsection III-B describes how we surpassed the problem of acquiring data to train our network. Lastly, subsection III-C explains the implementation details in our algorithm.

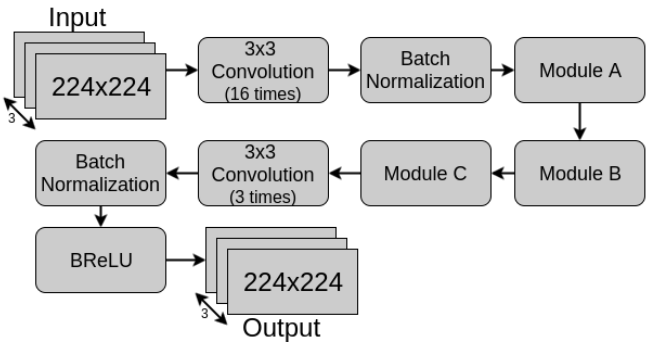


Fig. 1. The network's full architecture, it is composed chained inception modules combined with some convolutions.

A. Training CNNs for Image Dehazing

Our approach is to train a CNN to the point that it is able to, given a hazy image patch, produce another version of the same scene where the haze is reduced or even removed. This is a complex task, so we trained our network for 120 epochs. We used a learning rate of 1×10^{-5}. The model was trained with batches of 32 square patches. After each batch is processed, we evaluate the model's performance comparing its output with the clear ground truth patches using a loss function L.

We use L as the Mean Squared Error function combined with the Feature Loss [15], which is a loss function focused in securing that the compared images have similar features. This function uses a neural network that has its first layers designated to extract the features and the last layers to output a loss value. The network's output is put into the function ℓ described in Eq. 2.

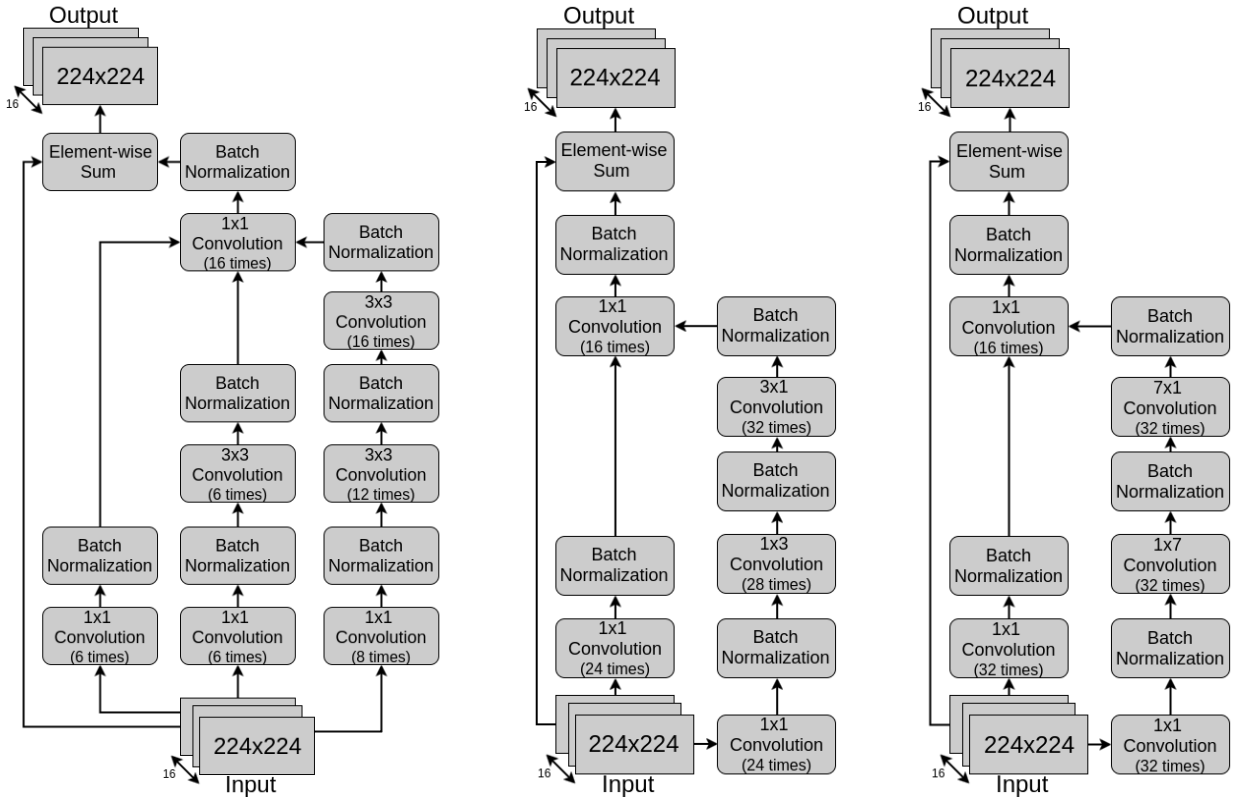
$$\ell_{feat}^{\phi,j} = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2, \tag{2}$$

where ϕ_j is the *jth* network layer, C, H and W are the feature map's depth, height and width, respectively. $\hat{y}$ represents the desired output and y is the layer's output.

Lastly, we use the Adam [16] Optimizer to readjust the weights, repeating this process to minimize the loss L. The parameters used in the optimizer were $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$.

The architecture of our dehazing CNN is based on [17]'s classification model. However, we have made some adjusts to best fit it to our problem. In our case, the desired output is an image with *the same resolution* as the input, therefore, it is desirable to keep as much as information from the input as possible. Aware of that, we do not use any type of *dimension reduction* operations, such as pooling. Also, since all our feature maps present the same resolution as the input, it is necessary to reduce the number of feature maps per layer in order to maintain the memory use and processing time at reasonable levels. Our network is also much shallower than state-of-the-art classification architectures.

Our network is composed of inception modules similar to the ones presented in the Inception-ResNet-v2 [18] architecture. However, our modules are built exclusively with 16



(a) Module A (b) Module B (c) Module C

Fig. 2. These modules are based on the Inception-Res-V2 design [17]. Although we use less channels, due to GPU's memory constraints.

feature maps as an input and expect 16 feature maps as the output. The detailed structures of our modules are shown in Fig 2. Overall, our network, as presented in 1, has the following structure: We take a $W \times H \times 3$ image as an input and apply sixteen 3×3 convolutions on it, followed by a batch normalization [19] layer. The resulting feature maps are run through three consecutive inception modules, then, we apply three 3×3 convolutions to the resulting feature maps, ending up with an $W \times H \times 3$ output image, where W and H represent the original image's width and height, respectively. Finally, we apply the BReLU [12] activation function to the output.

Convolutional Neural Networks require large sets of labeled data to be trained. In our case, *pairs* of hazy and haze-free images are required. It is crucial that both images in each pair are composed by the same scene captured under the same lightning conditions. The problem is: the feasibility of gathering these pairs of images in a quantity large enough to train a neural network is low. How can we capture a large number of pairs that meet the requirements to train our network? Subsection III-B explains how we overcame this obstacle and achieved a dataset that fits our needs.

B. Data Used for Training

Knowing the adversities of collecting data that suit our needs, we decided to generate synthetic data, capturing haze free images and applying simulated haze. In order to accomplish that, we need the scene's global illumination color, atmospheric attenuation coefficients and it's depth map. The global illumination color and the attenuation coefficients can be calculated based on patches extracted from real hazy images containing only regions where the transmission is minimal. Since we capture images with the Kinect camera, we are also able capture the scene's depth map.

Duarte et al. [20] proposed a simulator that, given an clear image, generates a turbid underwater image of the same scene based on the depth map and a turbidity patch. Our data is synthesized using the same model, but, instead of a turbidity patch that represents the water medium, we use a hazy patch.

Based on the data available, we are able to simulate haze effects using the image formation model described in Eq. 1, using the depth map acquired using the Kinect camera to estimate each pixel's transmission. Thus, we are capable of generating more than one hazy pair for each clean image, applying several levels of degradation produced by the haze.

Fig 3 shows some image samples synthesized using the simulator. These images are examples of the images we used in the training of our network.

An ideal dataset for our network would be composed entirely of outdoor images with their corresponding high quality depth maps. However, we were unable to find such dataset in the literature, since all currently available outdoor datasets with depth maps present problems that make them unusable

Fig. 3. Examples of images generated by the simulator to train our network. First row: Clear images used as ground-truths. Second row: Transmission maps used to estimate the depth of the objects present in the scene. Third row: Hazy images synthesized by the simulator used as the network input.

to train our network. These problems are low resolution depth maps, limited maximum range and mismatching alignment between the depth map and the optical image.

We end up with two options: (a) using a dataset composed of indoor images or (b) using outdoor images with unknown depth. The first option allows us to produce images with more realistic degradation effects, but these images are considerably different from the outdoor images we intend to restore. The second option permits us to create a larger, more diversified dataset composed with images closer to outdoor hazy images, but result in a lack of spatial variation in haze intensity.

It is preferable to synthesize a realistic haze effect in an inaccurate context than generating hazy images in the right context with an imprecise representation of the haze degradation. It is due to our goal is to learn to remove the haze itself and not to recreate what is behind it. Thus, we adopted indoor images with accurate depth maps to the generation of our datasets. Although we use the depth information to *generate* the dataset, it is important to note that our deep learning model is trained entirely with RGB images only.

C. Network's Implementation

Image processing tasks are already very costly computationally and when training a deep neural network where each layer processes multiple images, the usage of *Graphic Processing Units*(GPUs) is crucial. For this reason, our model is implemented entirely using Tensorflow [21], a framework that enables us to implement GPU-friendly algorithms, boosting our training performance. Using nVidia's Titan X with the *Pascal* architecture, we are able to run large scale experiments, speeding up the project's development.

Fig. 4. Left: Hazy image used as input. Right: Our model's dehazed output.

IV. Results

Even though our network is trained entirely with simulated images, the data adopted in the experimental results contains only images captured in real hazy environments. It is important to remember that we do not apply any type of correction or calibration to the hazy image before or after it is processed by the neural network.

Due to the usage of nVidia's Titan X GPU, our model takes 0.055 seconds to dehaze a 224×224 image. This fast processing permits our model to be used in live low-resolution videos, dehazing in real-time, which can be useful in systems that operate in environments with poor visibility.

Object distance is an important factor when talking about haze. The degradation effect caused by haze increases exponentially depending on the object's distance, due to the amount of particles between the object and the observer. Fig 4 is a perfect example of this phenomenon, where the signs that are farther experience degradation due to the haze and the objects near the observer are perfectly visible.

Fig 5 presents a scene with objects also distant from the camera, but with a lot of occlusion due to the haze. Even though, the scene behind the haze becomes considerably more visible. The improvement is more noticeable in areas with a lot of degradation, such as the buildings, which are the farthest objects from the observer.

Fig. 5. Left: Hazy image used as input. Right: Our model's dehazed output.

Fig. 6. Top: Hazy image used as input. Bottom: Our model's dehazed output.

To exemplify our model's capacity to detect the objects' distances, Fig 6 shows a slightly hazy image with objects in distinct distances from the camera. Our model not only successfully dehazes the mid-range objects (bigger structure) but detects the closer objects (bush) do not need restoration and more distant objects (small house in the back) need a stronger restoration.

Meanwhile, Fig 7 shows an image with distant objects with a little degradation. In that case, our model improves the visibility, increasing image contrast and level of detail.

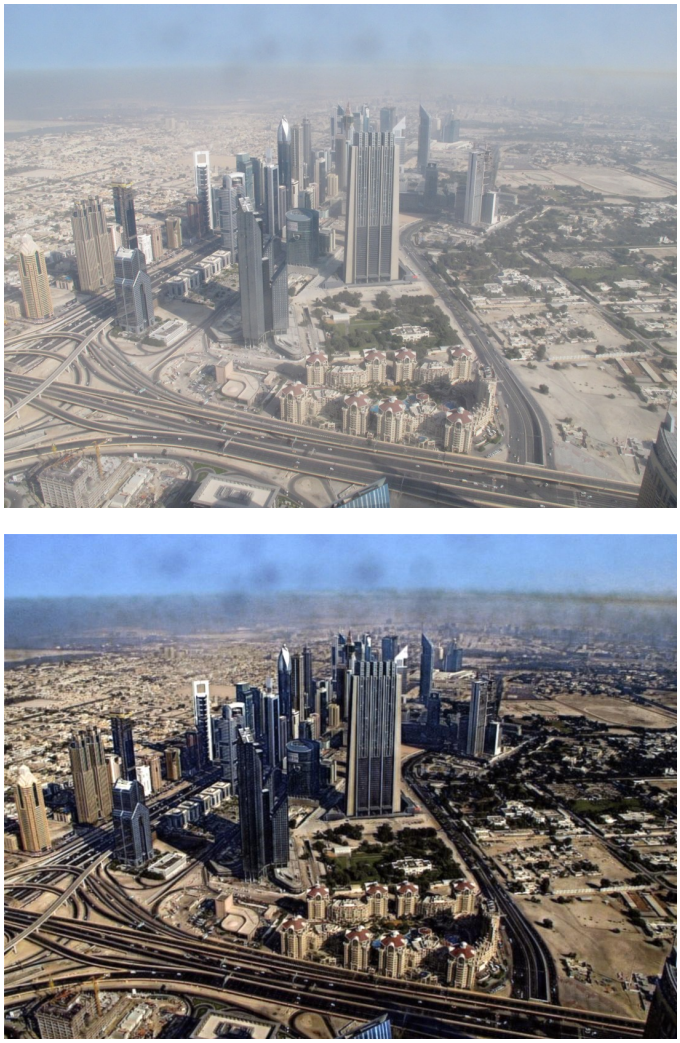


Fig. 7. Top: Hazy image used as input. Bottom: Our model's dehazed output.

Since our training set is composed entirely of indoor images, simulating realistic hazy images with haze-free pairs to train our network allowed our model to learn to remove the haze effect itself. Also, it learned the correlation between level of degradation and distance, even though the network is not presented the scene's depth map.

V. Conclusions and Future Work

Neural networks are capable of learning several image processing tasks, and dehazing is one of them. For the completion of this task to be possible, it is important that (i) the network

architecture has a large capacity, (ii) the training set is large enough and (ii) the image database is composed of pairs of haze-free and *realistic* hazy images. A well engineered haze simulator and convolutional neural networks implemented on GPUs that are specifically designed for large processing tasks fulfill these requirements.

In this paper we proposed a novel *end-to-end* convolutional neural network architecture method for image dehazing. Based on a hazy image and no previous knowledge about the environment or the objects distances, our approach is capable of removing the haze in the scene, significantly increasing visibility and level of detail. Also, this entire process was achieved with a image database containing only *indoor* images, indicating that our method is able to remove the haze effect.

As future work, we intend to improve our network's architecture and synthesize an even more realistic dataset, in order to increase our method's performance. Furthermore, we believe that our model can be applied to *underwater turbid images* merely by changing the training data, since the degradation effects caused by haze and water are very similar.

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] 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, no. 11, pp. 3522–3533, 2015.

[2] R. Fattal, "Single image dehazing," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, p. 72, 2008.

[3] ——, "Dehazing using color-lines," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 13, 2014.

[4] I. Omer and M. Werman, "Color lines: Image specific color representation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–II.

[5] X. T. K. He, J. Sun, "Single image haze removal using dark channel prior," *IEEE Conference on Computer Vision and Pattern Recognition - CVPR*, 2009.

[6] P. Drews-Jr, E. Nascimento, F. Moraes, S. Botelho, and M. Campos, "Transmission estimation in underwater single images," in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 825–830.

[7] P. Drews-Jr, E. Nascimento, S. Botelho, and M. Campos, "Underwater depth estimation and image restoration based on single images," *IEEE Computer Graphics and Applications*, vol. 36, no. 2, pp. 24–35, 2016.

[8] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 769–776. [Online]. Available: http://papers.nips.cc/paper/3506-natural-image-denoising-with-convolutional-networks.pdf

[9] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *ECCV*, 2016.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016. [Online]. Available: http://arxiv.org/abs/1609.04802

[11] 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.

[12] 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.

[13] 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.

[14] J. O. Gaya, L. T. Gonçalves, A. C. Duarte, B. Zanchetta, P. Drews, and S. S. Botelho, "Vision-based obstacle avoidance using deep learning," in *Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016 XIII Latin American*. IEEE, 2016, pp. 7–12.

[15] 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.

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, D. Blei and F. Bach, Eds. JMLR Workshop and Conference Proceedings, 2015, pp. 448–456. [Online]. Available: http://jmlr.org/proceedings/papers/v37/ioffe15.pdf

[20] A. Duarte, F. Codevilla, J. O. Gaya, and S. S. Botelho, "A dataset to evaluate underwater image restoration methods," pp. 1–6, 2016.

[21] 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.