Effects of Haze Removal for Image Segmentation - Self Driving Cars

Ved Vyas
Khoury College of Computer Science
Northeastern University
Boston, MA 02115
vyas.ve@northeastern.edu

Aditi Pathak
Khoury College of Computer Science
Northeastern University
Boston, MA 02115
pathak.ad@northeastern.edu

Abstract

Fully autonomous cars aren't a reality yet, but they are not very far away in the future. However, technology has enabled several advanced driver assistance features, most of which require environmental perception. There are various techniques that are used for environmental perception like image segmentation. It is the process of assigning a label to every pixel in an image such that pixels with similar characteristics share the same label. This gives ideal results when the road users and surroundings are clearly visible to the camera, however bad weather conditions like haze, snow, rain, or fog can degrade the quality of segments. So we need a system in place to make sure that we can still collect valuable data in real time. In this paper we discuss the performance of a dark channel prior based image defogging technique on a set of densely hazy images. An image segmentation model using U-net architecture is trained on around 2900 images. We used this model to observe the effect of haze on segmentation by comparing segments in hazy vs non-hazy images. Experimental results show that the segmentation results improve after haze removal from the images, although haze removal itself is a challenging task, especially in images with high density haze.

1 Introduction

Recently, the news of soon to launch Tesla's self-driving car spread like wildfire. The recent advancements in technology have turned a distant reality into something plausible in the near future. Semantic Segmentation can now be used to get relatively accurate solutions that can be used in self-driving cars, all credit to the improvements in deep learning [10]. However, even with these advancements, there is a unique challenge that has to be dealt with for safe driving - extreme weather conditions.

Image segmentation works efficiently in ideal scenarios but to make it effective, we have to consider unfavorable scenarios. During such a condition, it becomes difficult for an image segmentation model to identify objects and boundaries. In this project, we explore the effect of one such case in which there is haze/fog in the environment on the U-net model [11]. This model was first introduced for image segmentation for biomedical images and won the ISBI cell tracking challenge 2015 and provided a noticeable increase in speed.

In certain weather conditions, the presence of water droplets and particles in the air can degrade the image quality of the scene. The atmospheric scattering and absorption by these particles give rise to smoke, haze, and fog. As a result, the radiance reaching the camera is reduced along the line of sight. Also, the camera captures airlight, which is described as ambient light diffused or scattered by the dust particles limiting visibility as the object tends to blend with the background sky. The resulting images have less contrast and color fidelity. Therefore the presence of haze in images can affect the

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

performance of outdoor computer vision applications. Haze removal is challenging and has a lot of work done on it so far, as in [1], [2], [3], [4], [5], [6], [7]. Haze removal often faces the trade-off between time consumption and restoration quality with different haze or fog densities. We evaluate a proposed dark channel prior defogging method that is shown to manage the trade-off well [2]. The method uses dark channel prior, which is an image prior, in a manner that is fast in computational speed.

After this, we train a U-net model on cityscapes data [12]. To achieve our original goal, we then run this model on the restored images we have obtained from image defogging and compare it with the results from running the model on corresponding hazy images.

There have been different studies that connect the ideas of haze removal and image segmentation drawing a connection between them [13]. This connection concludes that de-hazed or defogged images are better for image segmentation. Our final goal is to explore this relationship and see if there is any veracity in this conclusion.

We have referenced the open-source codes referencing the works in [22] [16].

2 Related work

Automated cars use various techniques to identify objects while collecting real-time data. One of these techniques is image segmentation. The researchers in [18] talk about how we need high-accuracy models for automated cars. However, the trade-off that comes with higher accuracy is high computational requirements. This paper talks about the U-net model which delivers high accuracy with low computational requirements.

Image segmentation is important for image processing and computer vision with applications such as scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and image compression, among many others [20]. Various algorithms for image segmentation have been developed in the literature.

Researching further into the model we came across [11] which introduces the U-net. This model utilizes a contracting path for context and an expanding path for precise localization. Both of these papers together help us in understanding the process of segmentation. However, the model focuses on ideal conditions.

Research has found that unfavorable weather conditions directly affect the performance of image segmentation [13]. Consequently, U-net's performance would suffer. In the real world, where a small error or delay of a fraction of a second can lead to an accident, image segmentation must be very accurate and quick.

In order to present a clear image and overcome this issue, we have to remove the haze which would have a direct correlation with the accuracy of segmentation. [19] proposes such a method based on the dark channel prior. The authors in [19] suggest searching a sky area to estimate the transmission map. It is then used to generate a clear image by estimating the intensity. Through this, we can estimate the intensity of haze and obtain a clear image. Using this method can also provide a high-quality death map as a byproduct. The authors in [18] also highlight the speed of this method suggesting that they could process 50 images (720 x 480 pixels) per second on a PC with Intel i5 2.53-GHz CPU and a NVIDIA GeForce 310M GPU.

Image Segmentation has been explored as a solution to remove haze[21]. It proposes an algorithm that involves a new adaptive CLAHE process, in which a stronger enhancement is applied to the areas of weaker variation. The researchers found that using image segmentation for a better quality image improves performance and enhances the image.

On the other hand, this also begs the question of a difference between the efficiency of image segmentation on different results on hazy and clear data. Any adverse conditions would deteriorate the results of image segmentation as suggested by [13].

The main idea that brings all of these papers together is the two main goals for successful environmental perception: delivering high-accuracy results and using low computational resources. Both U-net and the proposed haze removal technique deliver desired results with those goals.

3 Methods

3.1 Single Image Defogging

The method used for image defogging is based on the concept of dark channel prior. Image priors are information extracted from the image that can improve the results of image processing tasks. Dark channel prior is rooted in the observation from non-hazy outdoor images that there is at least one channel in non-sky regions with pixels with fewer intensity values, almost close to zero [2]. The reasons for the existence of dark channels can be attributed to factors like the presence of dark objects or colorful objects, and the presence of shadows in the image. But as a foggy image has more brightness due to atmospheric scattering and absorption of light, the dark channel has pixels with higher intensities than a clear image. Hence, the dark channel pixels in a foggy image approximately but very closely reflect the fog density. This insight has been used in the proposed method in creating the transmission map (which tells how much light reached the camera without scattering) required to restore a foggy image. However, due to the nature of the concept, this method cannot be applied to images having large white patches (like a white background or snowy areas with no shadows cast on them) [1].

The dark channel prior is mathematically defined as:

$$I^{dark}(x) = \min_{c\epsilon(r,g,b)} (\min_{y\epsilon\Omega(x)} (I^c(y))) \tag{1}$$

where, $\Omega(x)$ is a local patch of the image centered at x, $I^c(x)$ represents intensity of pixel at location x in color channel c and $I^{dark}(x)$ represents intensity of pixel at location x in dark channel.

The image defogging model which is used to remove haze is modeled as:

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{2}$$

where, J(x) is the clear image, I(x) is the image with haze or fog, t(x) is the transmission map and A(x) is the atmospheric light intensity, which is the ambient light.

Using the above model, we can restore a hazed image back to a clarified image as follows:

$$J(x) = (I(x) - A)/\max(t(x), t_0) + A \tag{3}$$

So the process flow of the proposed method looks like:

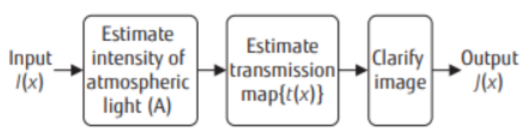

Figure 1: Step by step process flow of fast single image defogging method.

The transmission map is obtained from the dark channel of the local patches around every pixel, also called coarse map. In our method, this coarse map is refined by using a fine map. A fine map contains the edge information from the foggy image.

The coarse and fine maps are formulated as:

$$M^{coarse}(x) = \min_{c\epsilon(r,g,b)} (\min_{y\epsilon\Omega(x)} (I^c(y))) \tag{4}$$

$M^{coarse}(x)$ represents a coarse map.

$$M^{fine}(x) = \min_{c\epsilon(r,g,b)} (I^c(x)) \tag{5}$$

$M^{fine}(x)$ represents a fine map.

Finally, the coarse map and the fine map are combined as below to obtain the transmission map:

$$M^t(x) = \min(\max_{y \epsilon \Omega(x)}(M^{coarse}(y)), M^{fine}(x)) \tag{6}$$

where, $M^t(x)$ is the estimated transmission map.

The methodology uses the local information for every pixel in the image. So the processing can be parallelized for each pixel, hence giving computational advantage and saving time in real-time applications like self-driving cars compared with other conventional methods of [1], [4], [5]. With the advancement in machine learning, state-of-the-art methods use CNN-based architectures to predict the transmission map and improve quality as shown in the works of [3], [6]. The deep neural network-based architectures have the ability to outperform the conventional methods but require large data to be trained and have low computational speed. On the contrary, our method can be directly applied on a single image and give good results fast [7].

In our implementation, we developed and improvised codes for computation of atmospheric light, added refinement to transmission map and finally restored the scene radiance using above mentioned formulas and basic python libraries.

3.2 U-net model for image segmentation

The U-Net is used for semantic segmentation. It comprises an expansive path and a contracting path. The contracting path adheres to the standard convolutional network architecture. Two 3x3 convolutions (unpadded convolutions) are applied repeatedly, and after each one, a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 are used for downsampling. The number of feature channels is doubled at each downsampling step. Every step in the expansive path entails upsampling the feature map, followed by a 2x2 convolution (also known as an "up-convolution") that cuts the number of feature channels in half, a concatenation with the equally cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU [11].

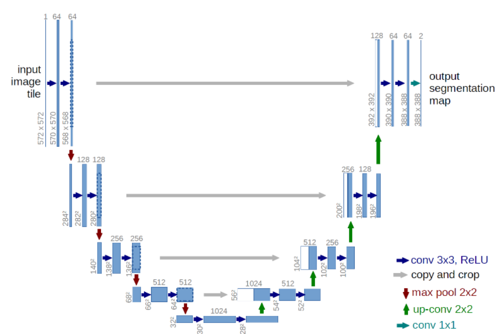

Figure 2: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

This is a modified and extended architecture of the fully convolutional network mentioned in [14]. U-net works with very few training images and yields more precise segmentations; see Figure 2. The main difference in [14] and U-net is to supplement a usual contracting network with successive layers, where pooling operators are replaced by upsampling operators. Therefore, the resolution of the output is increased because of these layers.

The network is trained using Caffe's stochastic gradient descent implementation using the input photos and their related segmentation maps. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration.

The output image is smaller than the input by a constant border width as a result of the unpadded convolutions. In our implementation of SGD, Adam optimizer is used for model optimization by coordinating the network's forward inference and backward gradients to form parameter updates that attempt to improve the loss [15].

To train our segmentation model we utilize PyTorch's segmentation model API that provides encoders that have pre-trained weights for faster and better convergence.

Once we have trained our image segmentation model, the Dice coefficient is used as a statistical validation metric to evaluate the performance. The equation for this concept is:

$$2 * |(X \cap Y)|/(|X| + |Y|) \tag{7}$$

where, X and Y are two sets, |X| and |Y| means the number of elements in set X and set Y, ∩ is used to represent the intersection of two sets.

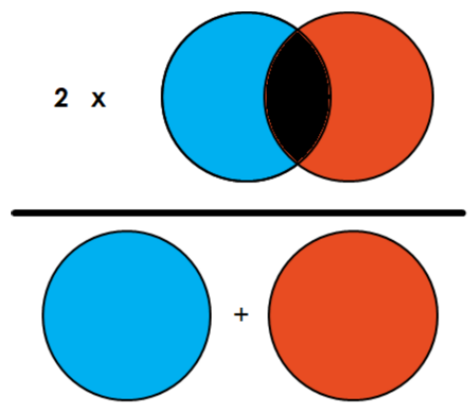

Figure 3: A diagrammatic representation of dice coefficient.

4 Experiments

4.1 Single Image Defogging

The dataset we used to implement the proposed method is the dense haze dataset [8]. It contains 55 pairs of densely hazy and corresponding clean images of outdoor scenes. The images are of the dimension 1600 X 1200. Our aim was to tune the parameters and obtain good quality clarified images and observe the performance of the method on densely hazed images compared to hazed images.

We tested our implementation for multiple images from our dataset [8], experimented with the parameters like window size and compared the restored image qualities with respect to the clear images. To compare the qualities of the images, we used image quality measurement metrics like SSIM, PSNR. The structural similarity index is a perception based model [9] and PSNR is signal to noise ratio. Higher the values of SSIM and PSNR implies less image distortion.

Table 1 shows the performance of the method on 3 images (selected randomly) in terms of SSIM and PSNR. Figure 4,5,6 shows the hazed images, dehazed images and clear images.

Table 1: Evaluating proposed dehazing method using SSIM and PSNR.

Image	**Image 1**	**Image 2**	**Image 3**
SSIM (clear image, hazy image)	0.36	0.37	0.35
SSIM (clear image, dehazed image)	0.45	045	0.39
PSNR (clear image, hazy image)	9.96	8.8	12.0
PSNR (clear image, dehazed image)	15	11.6	12.8

Figure 4: Image 1: Hazy Image, Dehazed Image, Clear Image (left to right)

Figure 5: Image 2: Hazy Image, Dehazed Image, Clear Image (left to right)

Figure 6: Image 3: Hazy Image, Dehazed Image, Clear Image (left to right)

It is observed that dehazed image is better than the hazed image in terms of image quality, but visibly the difference is not significant.

We also evaluated the impact of window size on the defogging method. We observed that for our images changing the window size from 20 to 600 has no effect on changing the results.

The method, when evaluated on a relatively less densely hazed image, performs better, as seen in Figure 7.

Figure 7: Hazy Image, Dehazed Image (left to right)

However, the method is not valid with images having scene objects similar to the airlight as discussed before. Figure 8 confirms this limitation, as there is only an increase in the PSNR value of comparison of dehazed and hazed image with clear image. SSIM value remains same (0.19).

Figure 8: Hazy Image, Dehazed Image, Clear Image (left to right)

This method in our work was not implemented on a CUDA GPU, hence the processing time was very high for each image and for increasing window size. But parallel processing can definately decrease the run time significantly.

4.2 U-net model for image segmentation

We use a second dataset to train a U-net segmentation model [12]. It has 2975 training images and 500 validation images. The resolution of each image is 256x512 pixels and each file is a composite with the original photo and a labeled photo on the left half and right half of the image respectively. It is created from labeled videos taken from vehicles driven in Germany and has still images from the original videos.

We have trained a U-net model on the training images and checked its efficiency against our validation images. During our training, we experimented with hyperparameters such as batch size, epochs, and learning rate. To quantify the results of our model we use the dice coefficient.

Figure 9: VIsualization of results of U-net model on cityscapes data.

Finally, we use the set of images from haze dataset [8] which includes both haze and clear images to see the difference in the image segmentation results.

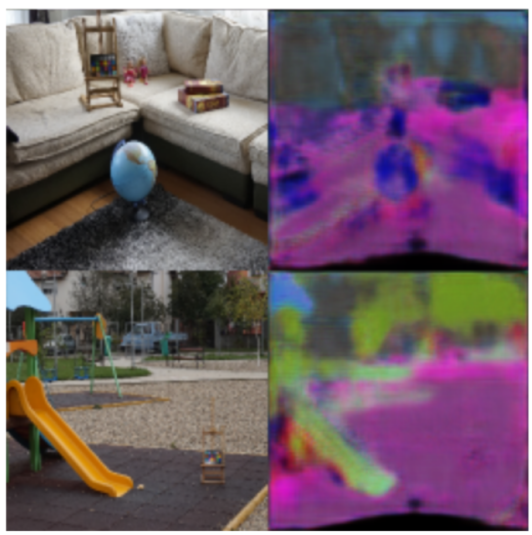

Figure 10: Visualization of the results on haze-removed data.

The above images help us visualize the results. For quantifying our results, we use dice score. The first set of dice scores is from the test set of our model:

[0.4035002918028487, 0.3901761007815238, 0.40926223021097247, 0.3868879128226437, 0.3936224809626502, 0.4033895610885987, 0.3966571258664878, 0.3635251790776557] Average = 0.3933776103266726

[0.2822095925192266, 0.5730872423420391, 0.5456221333355628, 0.37458674259234737, 0.5466175061026143, 0.44117433868339045, 0.3974517022586546, 0.48654446025805165] Average = 0.45591171476148584

5 Conclusion: Discussion and Future Scope

The difference in dice scores and visualizations for the two datasets follows our original hypothesis that haze creates problem in image segmentation which would directly affect self-driving cars ability to function safely.

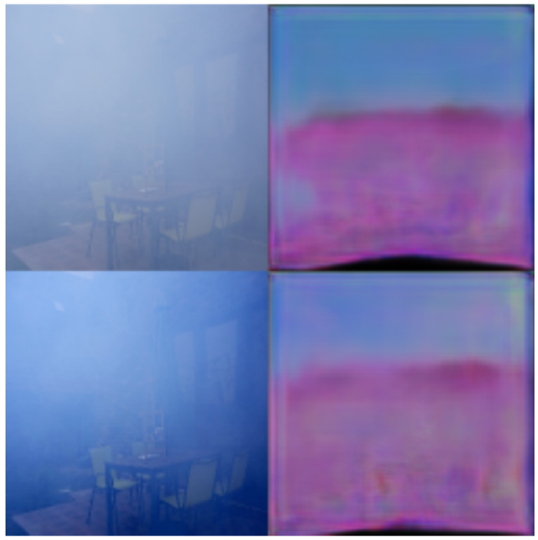

Figure 11: visualization of the results on haze data.

The dice score increases significantly in the second case since the model is not able to find edges because of the fog, this leads to more similarity between the original image and the segmented image. However, that is not true for our test data of cityscapes. Additionally, the dice scores should ideally be closer to 1 but we get a low dice score, one of the reasons for this is the lack of computational power to ideally fit the model.

To resolve this issue and include it in our future work, we can implement more than one segmentation model and combine their results to achieve higher dice scores. These models can include the implementations of PSPNet, Unet++, FPN, and the latest DeepLab V3+. Further future improvement can include state-of-the-art haze removal techniques that can encompass other weather conditions too. For example, snow, rain, low light, etc. We would also like to try vectorization in our code using packages like cpython.

Overall, we have achieved close results to our original hypothesis, which could be improved further with more computational power and implementation techniques.

References

[1] He, Kaiming et al. "Single Image Haze Removal Using Dark Channel Prior." IEEE Transactions on Pattern Analysis and Machine Intelligence (2011): n. pag.

[2] Tan, Zhiming et al. "Fast Single-image Defogging." (2014).

[3] Cai, Bolun et al. "DehazeNet: An End-to-End System for Single Image Haze Removal." IEEE Transactions on Image Processing 25 (2016): 5187-5198.

[4] Fattal, Raanan. "Single image dehazing." ACM SIGGRAPH 2008 papers (2008): n. Pag.

[5] Kratz, Louis and Ko Nishino. "Factorizing Scene Albedo and Depth from a Single Foggy Image." 2009 IEEE 12th International Conference on Computer Vision (2009): 1701-1708.

[6] Rashid, Huzaifa et al. "Single Image Dehazing using CNN." IIKI (2018).

[7] Lee, Sungmin et al. "A review on dark channel prior based image dehazing algorithms." EURASIP Journal on Image and Video Processing 2016 (2016): 1-23.

[8] `https://data.vision.ee.ethz.ch/cvl/ntire19//dense-haze/`

[9] Wang, Zhou et al. "Image quality assessment: from error visibility to structural similarity." IEEE Transactions on Image Processing 13 (2004): 600-612.

[10] Siam, Mennatullah et al. "Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges." 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (2017): 1-8.

[11] Ronneberger, Olaf et al. “U-Net: Convolutional Networks for Biomedical Image Segmentation.” ArXiv abs/1505.04597 (2015): n. Pag.

[12] https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs

[13] Zhang, Jian et al. “A new haze removal approach for sky/river alike scenes based on external and internal clues.” Multimedia Tools and Applications 79 (2019): 2085-2107.

[14] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation (2014), arXiv:1411.4038 [cs.CV]

[15] Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” CoRR abs/1412.6980 (2014): n. Pag.

[16] https://www.kaggle.com/code/michaelcripman/road-segmentation-unet-pytorch-implementation

[17] https://github.com/qubvel/segmentation_models.pytorch

[18] Gautam, Sanchit et al. “Image Segmentation for Self-Driving Car.” 2022 2nd International Conference on Intelligent Technologies (CONIT) (2022): 1-6.

[19] Tan, Zhiming et al. “Fast Single-image Defogging.” (2014)

[20] Minaee, Shervin et al. “Image Segmentation Using Deep Learning: A Survey.” IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2020): 3523-3542.

[21] C. Wang and B. Zhu, "Image Segmentation and Adaptive Contrast Enhancement for Haze Removal," 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), 2020

[22] https://github.com/kindraywind/SingleImageHazeRemover