Explore National Parks at home: Generate Haze-Free Images

Jiaqi Jiang jiangjq@stanford.edu Kaiwen Wang kaiwenw@stanford.edu
Lijing Wang lijing52@stanford.edu

Abstract

Dense haze caused by air pollution may block the view of the National Parks. In this project, we take outdoor images as inputs to 1) estimate their air pollution levels 2) generate haze-free images. We compare the performance of deep learning method with the Dark Channel Priors algorithm which is a traditional dehaze method. The deep nets show better performance at haze level prediction while producing satisfactory dehazed images.

1. Introduction

Every year, thousands of visitors visit U.S. National Parks. Many of their pictures may be blurred because of haze. However, monitoring the air pollution level usually depends on special devices, which might not be easily accessible to the public. In this project, we take we take outdoor images as inputs to predict air pollution levels and generate haze-free images.

2. Dataset

A. FRIDA Dataset

This dataset contains 3024 synthetic images (Figure 1) with 88 different scenes [5] and their haze level labels based on the algorithms and dataset provided by the FRIDA(Foggy Road Image DAtabase) and FRIDA2[8] [7] dataset.

B. National Parks Dataset

This dataset contains 145,803 images(Figure 2) from different national parks with 11 fixed cameras in 2016. Cameras take pictures every 15 mins and we only keep images from 7am to 6pm at the daytime. All pictures are crawled from `https://npgallery.nps.gov/AirWebCams/`and resized into 512*512*3 with the help of Google Compute Engine.

Since the national park dataset is limited in the number of different scenes, we would like to first train a model on the synthetic dataset and generalize to the national park

Figure 1: Low haze and high haze samples with size 64*64*3

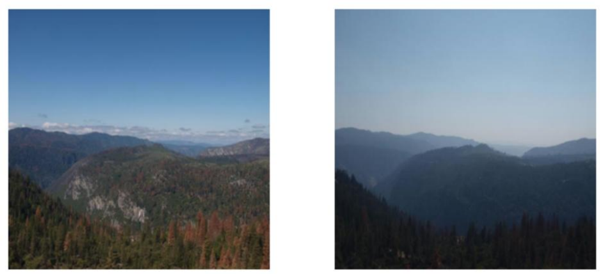


Figure 2: Low ozone and high ozone pollution images of Yosemite National Park with size 512*512*3

dataset. The synthetic dataset is also augmented to provide more taining examples.

Our output label is the matched hourly Ozone level from Clean Air Status and Trends Network (CASTNET) in Environmental Protection Agency `https://java.epa.gov/castnet/clearsession.do`. However, in the real world, the pollution levels are normally distributed(Figure 3), which would lead to high uncertainty in the end-member cases.

3. Approach

3.1. Dark Channel Prior

In this section, we introduce a popular method in haze removal: dark channel priors and its correponding atmo-

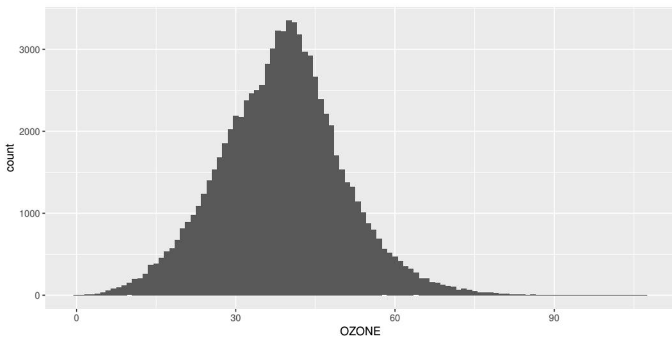


Figure 3: The histogram of Ozone level (ppb)

spheric scattering model [6] to understand how the haze impacts the image.

Each haze image consist two parts:

$$I = J * t + A * (1 - t)$$

I is the hazy image, t is the transmission map, J is the scene radiance and A is the atmospheric light. Haze removal is to get J with estimation of A and t. To solve that, we define dark channel:

$$J^{dark} = \min_{\Omega}(\min_{r,g,b} I)$$

For outdoor haze-free images:

$$J^{dark} \rightarrow 0$$

therefore, $t = 1 - \min_{\Omega}(\min_{r,g,b} \frac{J}{A})$, A can be estimated by brightest pixels in dark channels. Therefore, we can remove haze first with the estimation of t. That is the most popular dehaze method without using any deep learning methods.

Figure 4: Haze removal example with Dark Channel Prior

3.2. Haze Prediction

3.2.1 Linear Regression

We use the mean of the 'removed' haze and feed it into a linear regression model to predict the haze level. As a direct way to make prediction, this method is not doing well since it missed the depth information. To better utilize the information form the image, we try the CNN method.

3.2.2 Convolutional Neural Network

The second method we try is CNN, since it is good at extracting features from an image. We use 4 convolution layers and max pooling layers shown in Figure 5 and 2 fully

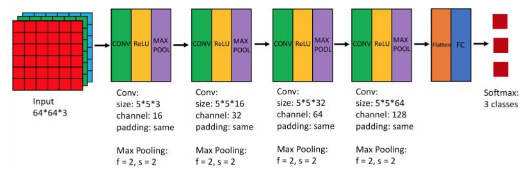


Figure 5: CNN structure

connected layers at the end. The prediction of haze doesn't rely on local features too much, so we use relatively large filters.

We first trained the CNN on FRIDA dataset since this dataset has diverse scenes. Before training, we resized the image to 64×64 to reduce the input size. Because details of the image are irrelevant to haze level, it won't hurt too much to the prediction. FRIDA data set was divided into three categories: 70% training set, 15% dev set and 15% test set, then trained the CNN to classify the images into three haze levels. We tuned hyperparameters such as learning rate, mini-batch size and bn momentum to optimize our CNN. Besides, early stop was used to reduce overfit. And metric we used is accuracy of prediction.

When the CNN was well-trained on FRIDA data set, we transferred to National Parks dataset. Since this dataset has only one scene for each national park, we used weights of pre-trained CNN as a starting point to finetune the parameters in order to avoid fitting to a certain scene. For this task, we replaced the last layer with 6 neutrons since we are going to classify images into 6 haze levels.

Our code in this section could be found at `https://github.com/Douphoton/CS230_project`

3.3. Dehaze

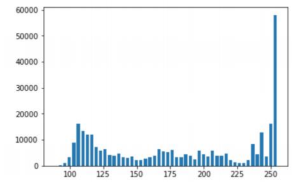
3.3.1 Neural Style Transfer

Neural Style Transfer is one of the most fun techiques in deep learning [2]. It merges two different images based on the loss we defined. In this section we will use that idea to perform haze removal.

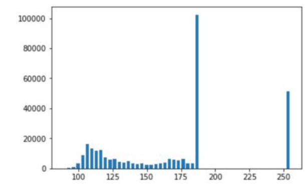
We are going to generate a image G with the content C of our haze image and the style S of the corresponding haze-free dark channel prior.(Figure 6)

Figure 6: Neural Style Tranfer

For the dehaze problem, we would like to preserve more lower-level features in order to generate an image with clear



(a) The histogram of Dark Channel Prior of Content Image



(b) Adjusted histogram of Dark Channel Prior of Content Image

(c) Dark Channel Prior of Content Image

(d) Adjusted Dark Channel Prior of Content Image

Figure 7: Adjust the histogram

edges. We set the image C (also our haze image) as the input to the pretrained VGG network, and run forward propagation. The content cost function is calculated between the content and generated images of 'conv1_2' and 'input' layers.

$$J_{content}(C,G) = \sum_{layers} \frac{1}{4 * n_H * n_W * n_C} \sum_{all\ entries} (a^{(C)} - a^{(G)})^2$$

n_H, n_W and n_C are the height, width and number of channels of the layer we have chosen.

For Style cost, we cannot just use the dark channel prior of the content image. We can see the histogram of the dark channel prior of the content image is not clustered at 0(Figure 7). Haze-free dark channel prior is clustered at 0. Therefore, we adjust the histogram and remove larger values (except completely white pixels) to get a target dark channel prior of our Style.

After we have adjusted dark channel prior as the Style image, we can define our style cost by calculating the dark channel of style image and generated image.

$$J_{style}(S,G) = \sum_{layers} \frac{1}{4 * n_C * n_H * n_W} \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} (DC_{ij}^{(S)} - DC_{ij}^{(G)})^2$$

DC is short for dark prior channels. And we use 'input','conv1_1','conv2_1','conv3_1' layers with different weights.

We first randomly generate G and then use Adam optimizier to update pixels on our G to minimize the content and style cost.

Our code for Neural Style Transfer could be found at `https://github.com/lijingwang/deepDehazeNets/blob/master/NSTdehaze.ipynb`

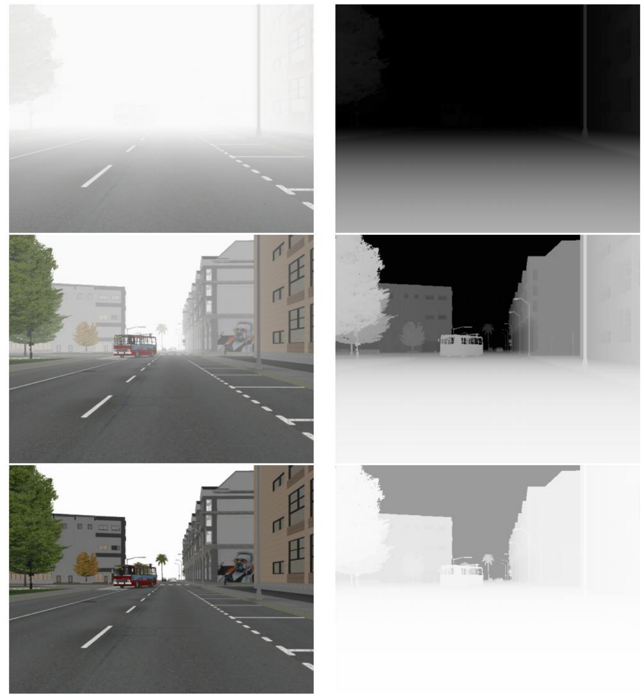


Figure 8: Left:Augmented Images with $\beta = 1e-4, 1e-5, 1e-6$.Right:Corresponding transmission maps of Augmented Images

3.3.2 Deep Neural Nets for Dehaze

In this section, we want to build an End-to-End Dehaze Networks with inputs haze images and outputs corresponding transmission maps. In order to get transmission maps, we need corresponding haze-free images. FRIDA2 is what we are looking for. FRIDA2 is a part of FRIDA dataset and has 66 diverse road scenes. And each image without fog is associated 4 foggy images and a depthmap. Before training our networks, we need to do data augmentation to get more training images.

Data Augmentation We want to physically simulate fog on images(Figure 8). The transmission map in reality is related to the depth map and it satisfied the exponential law [4].

$$t(d) = e^{-\beta * d}$$

$t(d)$ is the transmission map with the covariate d (depth). β is controlling the thickness of the fog.

We tune β from 1e-4 to 1e-6. As the β decreases, fogs density decreases and the visibility increases. Therefore, we can add physical haze on images and have the ground truth for the transmission map.

Using physical model, we augment 100 images for each diverse road scene with β from 1e-4 to 1e-6, in total 6000 images. Together with associated 4 foggy images for each scene, we have 6264 images.

Deep Neural Nets We divided our augmented dataset into 3 parts: training sets, 54 scenes, 5616 images; dev sets, 6 scenes, 624 images; test sets, 6 scenes, 624 images. Our inputs are haze images and outputs are corresponding transmission maps.

For dark channel prior, we calculate the minimum value over 3 color channels. And it can be similarly done with Maxout Layers [1] [3] in deep learning. This maxout operation is performed in the filter/channel dimension and it calculate the maximum value over different channels. Therefore, we first perform convolutional 2D networks(Figure 9) to get more channels and reduce dimensions with maxout.

After maxout, we want to extract more features to get our transmission map. Each pixel value on a transmission map is only related to close neighbours. So we do not need to mix pixels up. Instead, we do 2D convolutions around neighbours with more filters. 1*1 convolution network is to reduce dimension on filter dimensions so that we can perform sigmoid function for each pixel. Sigmoid out is our prediction for the transmission map. For each layer, we perform Batch Normalization to reduces the amount that hidden unit values shift around.

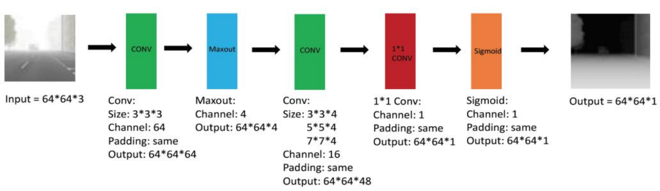


Figure 9: Structure of our Deep Neural Nets

Our Training Loss is defined as below:

$$L = \frac{1}{64 * 64}(sigmoid\ output - transmission\ map)^2$$

Based on the loss function we have, we use the Adam optimizer to minimize training and dev loss.

Our code for Deep Neural Nets for Dehaze could be found at `https://github.com/lijingwang/deepDehazeNets`

4. Result and discussion

4.1. Haze Prediction

4.1.1 Linear Regression

The accuracy for 3 classes classification with knn of haze level is 0.64(Figure 10).

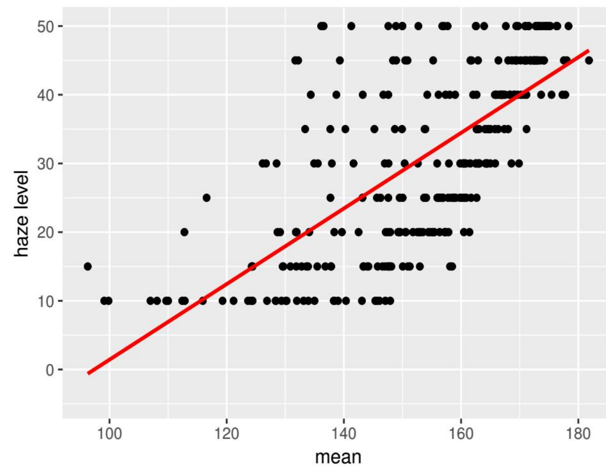
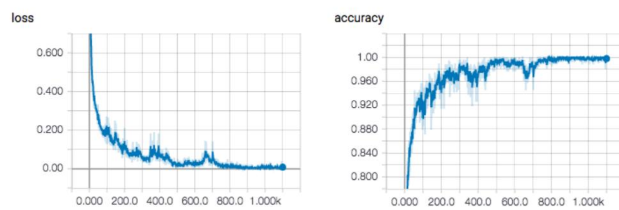


Figure 10: Linear Regression x: mean of removed haze, y: haze level



(a) Loss (b) Accuracy

Figure 11: Performance on training set

4.1.2 Convolutional Neural Network

For FRIDA dataset, we used Adam optimizer with a learning rate of 1e-3, batch-norm momentum of 0.9, training for 100 epochs, got 99.9% accuracy on training set and 98.8% accuracy on test set. The loss and accuracy as a function of training step are shown in Figure 11.

For Naitonal Parks dataset, the best accuracy on test set is 85% up to now. Reasons why performance is not good as FRIDA dataset are: 1) We have more haze levels 2) Numbers of images in each haze level are quite different. Some haze level only has few images therefore the accuracy on these levels could be unsatisfied. 3) Compared with FRIDA dataset where the brightness is same for all images, images of Natinal Parks have different brightness since they were taken at different time.

4.2. Dehaze

4.2.1 Neural Style Tranfer

There are two cars and one road sign in our generate image(Figure 12). However, they are impossible to be noticed in content image. This method works to find out more details hidden in content image.

Figure 12: Generated Image after 100 iterations

It also has some downsides. It is not corresponding to our original colors and not very clear on existing details. Therefore we build a more sophisticated network to perform haze removal with the end-to-end principle.

4.2.2 The End to End Dehaze Deep Neural Nets

We tune the hyperparameter(Figure 13) : learning rate to find out the most appropriate one in our training. 0.01 is too

Hyperparameter	Dev Set Loss after 10 epochs
experiments/learning_rate/learning_rate_0.01	0.0137874
experiments/learning_rate/learning_rate_0.001	0.0111132
experiments/learning_rate/learning_rate_0.0001	0.0245682

Figure 13: Hyperparameter Tunning

fast and it seems to oscillate around local minimal. 0.0001 is too slow to achieve a lower loss. Therefore we pick 0.001 as the learning rate together with 16 channels and batch size 32 in our training.

We train our network for 50 epoches(Figure 14) and development set accuracy achieved the best result at epoch 20, so we early stop at epoch 20.

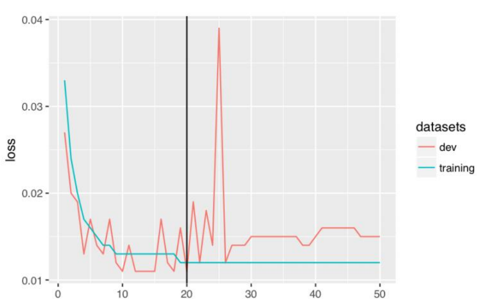


Figure 14: Loss on Training Set and Dev Set

Our best result gives 0.012 loss on training set, 0.011 loss on development set and 0.015 loss on test set.

Therefore, we can use our networks to predict transmission map and dehaze images. We can see our predicted transmission map(Figure 15) is similar to the real one. And ours reveals more details after dehazing.

White lines on the ground are unrecognizable on the real transmission map but not ours. It is not surprising because we have these lines in our input image and white lines can be treated as low transmission areas.

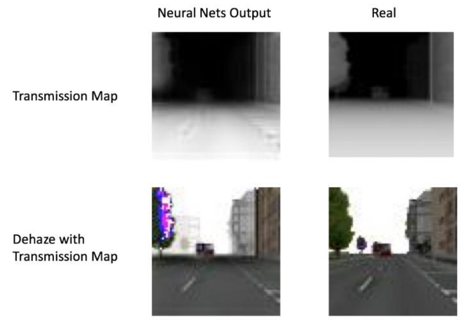


Figure 15: Dehaze with predicted transmission map

After evaluated our network on synthetic examples, we generalize to the National Parks dataset(Figure 16):

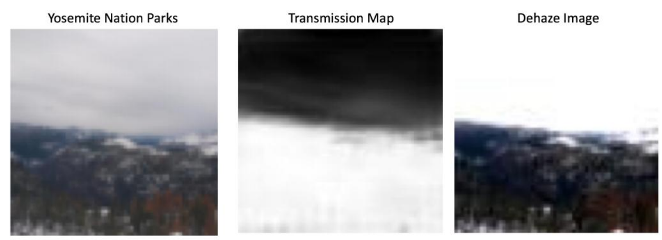


Figure 16: Dehaze on Yosemite Images

Our dehazing result show satisfactory performance at different scenes though the recovered sky is not very realistic. One possible explanation is that we previously trained our network with white haze and air lights. Although it is a good estimation in the synthetic examples, it is not well representing the light condition in the national parks. To improve that in the future, we need to dehaze with different air lights in outdoor scenes.

5. Conclusions and Future Works

In this project, we successfully predict the air pollution level from an outdoor image. We also achieved desirable result in image dehazing. Our future work includes:dehaze with different air lights in outdoor scenes;fine tune parameter on national park dataset;train GANs to generate images on requested weather conditions.

6. Contributions

All team members contributed to the progress of the project. Specific work assignment is as follows. Jiaqi Jiang implemented CNN using TensorFlow and helped write up the milestone. Kaiwen Wang performed haze removal with dark channel prior, made poster and drafted the final write up. Lijing Wang collected and cleaned National Parks Data, performed haze removal by Neural Style Transfer,wrote up the milestone, and performed data augmentation.

References

[1] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[3] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[4] S. Kahraman and R. De Charette. *Influence of Fog on Computer Vision Algorithms*. PhD thesis, Inria Paris, 2017.

[5] Y. Li, J. Huang, and J. Luo. Using user generated online photos to estimate and monitor air pollution in major cities. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, ICIMCS '15, pages 79:1–79:5, New York, NY, USA, 2015. ACM.

[6] J. Sun, K. He, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:2341–2353, 08 2010.

[7] J.-P. Tarel, N. Hautière, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, Summer 2012. http://perso.lcpc.fr/tarel.jean-philippe/publis/itsm12.html.

[8] J.-P. Tarel, N. Hautière, A. Cord, D. Gruyer, and H. Halmaoui. Improved visibility of road scene images under heterogeneous fog. In *Proceedings of IEEE Intelligent Vehicle Symposium (IV'2010)*, pages 478–485, San Diego, California, USA, 2010. http://perso.lcpc.fr/tarel.jean-philippe/publis/iv10.html.