RAIN AND HAZE REMOVAL

Neil Arakkal, Rahul Krishnamoorthy, Sutej Kulkarni, Vishal Ilangovan Bhuvaneswari

ABSTRACT

In this project, we will be performing Hazy Image Recovery using the All In One Image Dehazing (AOD) convolutional neural network (CNN) to dehaze an image and an end-to-end Deep Neural Network architecture to remove rain streaks from single images. The AOD network directly generates the dehazed image through a light-weight CNN, without estimating the transmission matrix and the atmospheric light separately as most previous models have done. Such a network makes it easy to embed the AOD network into other deep models for improving high-level tasks on hazy images. For the rain removal task, we implement a deep Residual neural network (Resnet) to make the learning process easier by reducing the mapping range between the input and the output. Further, we use high-frequency detail images for training the network instead of using the images directly. By doing so we focus on the high-frequency rain streaks in the image. Previously developed methods are based on separating the rain streaks from the images using only low-level features. These methods face difficulty in removing the rain streaks when the structure of the rain is similar to the object in the scene.

Index Terms— AOD Network, Convolutional Neural Network, ResNet

1. INTRODUCTION

Outdoor scenes are often affected by fog, haze, rain, and smog. Poor visibility in the atmosphere is due to suspended particles. Haze degrades image quality and limits visibility especially in outdoor settings. This consequently affects performance on other high-level tasks such as object detection and recognition. These all make single-image haze removal a highly desirable and crucial requirement. The presence of rain streaks or droplets in the images or videos can obstruct the view of the scene and is very undesirable. Moreover, they can affect the efficiency of outdoor vision systems. Rain removal is a much-needed technique in these scenarios.

In this paper, we propose an efficient end-to-end dehazing Convolutional Neural Network (CNN) model, called All-in- One Dehazing Network (AOD-Net). AOD-Net is designed based on a re-formulated atmospheric scattering model. A popular technique to achieve haze removal is to recover a transmission matrix, towards which various statistical assumptions and sophisticated models have been adopted. This involves some pre-processing techniques such as guided-filtering or soft matting will further distort the hazy image generation process, causing sub-optimal restoration performance. And also, the estimation of the transmission metric and atmospheric light may further amplify the error when applied together.

The previously developed state-of-the-art image processing systems to remove rain particles use only low-level features. But these systems find it difficult to identify and separate rain particles when the objects in the scene have the same orientation and structure of the raindrops. Removing rain from videos is relatively easier compared to the processing single images since videos have temporal information between frames and this can be leveraged to remove rain streaks form videos. Deep Neural Networks (DNN) were previously used for high-level vision tasks such as image classification. In this project, we implement a Deep Residual Network (Resnet) to process single images to remove rain artifacts. We make use of the high-frequency nature of the rain streaks in images to filter them out and use them as input to the neural network.

2. LITERATURE SURVEY

2.1. Related work for Image Dehazing

[1] made use of a multi-scale deep neural network for single image dehazing by making use of the images and their corresponding depth maps and learning the correlation. Their proposed algorithm uses a coarse-scale net that predicts the holistic transmission map by looking at the entire image and then uses another fine-scale net which makes a local refinement to the results. Their proposed algorithm was able to produce really fine dehazed images but had a few drawbacks as well. Even though the algorithm was able to produce dehazed images, the performance deteriorates when the lighting gets worse. When the amount of lighting in the scene decreases, the method does not produce reasonable results which we aim to improve in our implementation.

[2] takes a hazy image as input and produces its medium transmission map which is thereby used to produce a dehazed image by making use of the atmospheric scattering model. Their method makes use of a deep convolutional neural network where the layers are specifically tailored to embody established assumptions in this field. They make use of max out

layers to produce haze relevant features and also made use of a novel nonlinear activation function which they say improves the quality of the produced images. One of their drawbacks is that they do not make use of an end to end module which just produces the final dehazed images. Instead, their method produces a transmission map that has to be then operated on to produce the dehazed images.

[3] proposes a flexible cascaded convolutional neural network for single hazy image restoration. It considers the medium transmission and global atmospheric light jointly by two networks. The medium transmission estimation makes use of a deep densely connected convolutional neural network and the global atmospheric estimation model uses a lightweight convolutional neural network that is cascaded together with the medium transmission estimation model and thereby shares common features. The final haze-free image is obtained by atmospheric scattering model inversion which they say produces more accurate results. This method is not end-to-end and makes use of two cascaded convolutional neural networks and even after the cascaded convolutional neural networks, it has to go through additional steps to produce haze-free images.

All these approaches make use of the atmospheric scattering model in one way or another to produced haze-free images. It can be inferred from these methods that the atmospheric scattering model is the key to producing haze-free images so our implementation will be making use of the atmospheric scattering model.

2.2. Related work for Rain removal from images

The greater the depth of the neural network, the greater is the number of features learned by the network and lower is the risk of overfitting the data. This usually results in increased performance. But networks with increased depth face the problem of gradient vanishing i.e, the calculated gradients vanish as they are backpropagated. This can lead to saturation of the performance or even degradation. Deep Residual Learning [4], proposed a network architecture that can be trained with a large number of layers and yet result in compelling performance. ResNet contains residual blocks with identity shortcut connections that skip one or more layers. These residual blocks are stacked to form layers of the network. The residual block allows the network to fit a residual mapping instead of directly fitting the underlying mapping. This reduces the mapping space from the input to the output and hence makes it easier for the network to learn the mapping faster.

[5] proposes a method of single-image rain removal by formulating it as a layer decomposition problem. This method imposes priors for both the background and rain streak layers using Gaussian mixture models learned on small patches that can accommodate a variety of background appearances as well as the appearance of rain streaks. A structure residue recovery step further separates the background residues and improves the decomposition quality of the layers priors.

[6] proposes a single-image dictionary learning-based algorithm based on a non-linear generative model known as the *screen blend model*. It sparsely approximates the patches of two layers by very high discriminative codes over a learned dictionary with strong mutual exclusivity property. These discriminative sparse codes lead to accurate separation of two layers from their non-linear composite.

3. TECHNICAL APPROACH

This section is again subdivided into two parts where the first subsection offers details about the technical approach for the image dehazing problem while the second subsection offers details about the same for rain removal from images.

3.1. Technical approach for Image dehazing

AOD assumes that hazy images are generated based on the atmospheric scattering model described below. According to the ASM:

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

Here, I(x) is the observed hazy image, J(x) is the original image, A is the Global Atmospheric lighting and t(x) is the transmission matrix. Global Atmospheric lighting refers to the natural light of the atmosphere across the entire scene. The transmission matrix represents the amount of light that reaches the camera from the object. It is calculated as below:

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

Here, β is the Scattering Co-efficient (Non-negative) and d(x) is the distance between the object and observer. The idea is that the atmosphere scatters light coming from the object before it reaches the camera. The amount of light scattered depends on the atmospheric properties (captured by β) as well as the distance of the object from the camera (captured by d(x)). Rest of the light is transmitted (reaches) the camera. Since β and d(x) are both non-negative, the value of t(x) is in the range (0,1]. 0 means no light from the object reaches the camera (all light is scattered) and 1 means all the light from the object reaches the camera (no scattering). A higher value of β represents an atmosphere that tends to scatter light more. Moreover, as the object moves further away from the camera, more light is scattered. Now, given values of transmission (for each pixel) and the global atmospheric light, the light received by the camera can be calculated. If t(x) is 1, then the camera sees the light from the object perfectly (no scattering). If t(x) is 0, then the camera sees only the atmospheric light. Else, the camera sees a linear interpolation between the light from the object and the atmosphere.

3.1.1. *Problem Formulation*

Existing work on de-hazing is based on individually estimating t(x) and A. The key problem with this is that the estimation errors may accumulate. AOD aims to estimate both the parameters in a unified manner with the following reformulation of Equation 1

$$J(x) = K(x)I(x) - K(x) + b \tag{3}$$

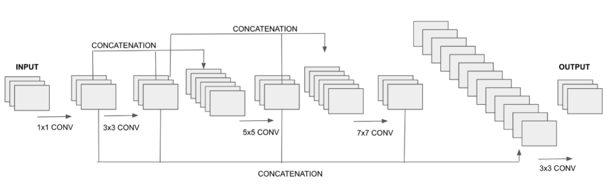
$$K(x) = \frac{\frac{1}{t(x)}(I(x) - A) + (A - b)}{I(x) - 1} \tag{4}$$

Here, t(x) and A are integrated into a single variable K(x) which is dependent on input I(x). Once, K(x) is estimated, the de-hazed image can be calculated using equation (3). Hence, the aim now is to estimate K(x) such that it minimizes the mean squared error (MSE) between the de- hazed and original image. This is done by training a CNN.

3.1.2. *AOD Network Design*

The network has two modules. The first module estimates K(x). Note that, for an image of size W x H x 3, K(x) also has a size of W x H x 3. The second module involves element-wise operations to generate the de-hazed image using equation 3. The K-Estimation module is critical as it is indirectly estimating the atmospheric light and transmission matrix (implicitly calculating the depth of each pixel in the image). It consists of 5 convolutional layers (with ReLU activations). Each layer has 3 filters but a different kernel size (to generate multi-scale features). Moreover, activations of previous layers are concatenated with intermediate layers before taking convolutions as this compensates for information loss during convolutions.

Fig. 1. AOD-Net Architecture

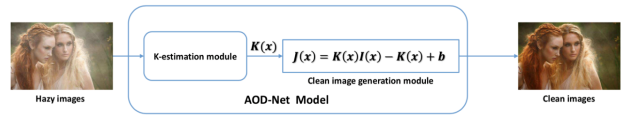


The network as shown in Figure 1 takes in an input image and returns an output image after passing through the AOD network architecture. The hazy image passes through the proposed pipeline and a dehazed image is obtained as shown in Figure 2.

3.1.3. *AOD Training*

Weight initialization is done with a Gaussian distribution in which the mean is 0 and the standard deviation is 0.02. We

Fig. 2. AOD-Net Prediction Pipeline



are applying the Rectified Linear Unit(ReLU) activation function after each convolution. ReLU activation with a maximum value of 1.0 is applied after the element-wise multiplication layer. We pass the training set and compute the gradients for each one of them. We clip the gradients on norm 0.1 and update the training set by applying the clipped gradients.

Initially, we define the number of batches to be processed, and for each batch, we run the optimizer operation and store the corresponding loss (MSE) achieved. Then we print the batch loss every 1000 iterations to check if the loss is decreasing. We run the training for 10 epochs and after each epoch, we save the weights. To evaluate performance on the validation dataset, we repeat the same process except that the optimizer operation is not run. After 100 batches, we display the hazed, original, and dehazed images for visual examination. We used a learning rate of 2e-4 with an adam optimizer and trained our dataset for a total of 10 epochs. We divide our dataset into 16380 training images and 1820 testing images.

3.2. Technical approach for rain removal from images

In this part, we implement a deep detail network using ResNet. Using ResNet reduces the mapping range between the input and the output and makes learning easier for the network to remove rain droplets in images. Using Convolutional Neural networks overcomes the difficulty faced by the previously developed methods in differentiating the rain content from the objects in the scene.

3.2.1. *Deep Detail Network*

Let **M** and **N** represent the rainy and the clear images respectively. The intuitive direction would be to train the deep network architecture with the **M** and minimize the loss function given by Equation 5.

$$\mathcal{L} = \sum_i \|h(X_i) - Y_i)\|_F^2 \tag{5}$$

Training the network directly on the images did not produce satisfactory results. The derained image seemed to have a color shift since the mapping range pans all possible pixel values and the network's regression function under fits the data. Also, training the network directly on images results in gradient vanishing even after the use of regularization methods.

To overcome the above-stated drawback with training the network with images directly, the detail layer of the images was given as input to the network. The image was decomposed into two parts: the detail layer and the base layer. The base layer of the image is the low-pass filtered component of the image. This was obtained using a guided filter. Subtracting the base layer from the image results in the detail layer which consists of the high-frequency rain components. This reduces the range of values in the input space.

To further reduce the range of the solution space, negative residuals were used. The negative residual is defined as the difference between the clear image and the rainy image. The rainy image consists of bright rain spots and the difference results in negative values. Hence the name negative residual. Using the detail layer of the image and the negative residual as input the network significantly reduces the mapping range. Therefore, the network is trained to approximate the detailed images to the negative residual images. The loss function previously defined can be modified as shown in equation 6.

$$\mathcal{L} = \sum_i \|h(X_i) + X_i - Y_i)\|_F^2 \quad (6)$$

Negative residual is defined as **N - M**. Using ResNet along with Negative Residual mapping helps better distinguish the rain streaks in the image. The skip connection of the ResNet also helps in propagating lossless information through the entire network.

3.2.2. Network Architecture

The deep Detail network has an end-to-end architecture that takes in rainy images as the input and produces clear mages. The input images are split into base images and detailed images were obtained by passing them through a low-pass guided filter. These base images are used to obtain the detailed images which are then fed to the network. The implemented ResNet has a depth of 26 layers without pooling layers to preserve the spatial information. The pipeline for rain removal from images can be explained using equations 7 to 11.

$$X^0_{detail} = X - X_{base} \quad (7)$$

$$X^1_{detail} = \sigma(BN(W^1 * X^0_{detail} + b^1)) \quad (8)$$

$$X^{2l}_{detail} = \sigma(BN(W^{2l} * X^{2l-1}_{detail} + b^{2l})) \quad (9)$$

$$X^{2l+1}_{detail} = \sigma(BN(W^{2l+1} * X^{2l}_{detail} + b^{2l+1})) + X^{2l-1}_{detail} \quad (10)$$

$$Y_{approx} = BN(W^L * X^{L-1}_{detail} + b^L) + X \quad (11)$$

Where l ranges from 1 to $\frac{L-1}{2}$ with L indicating the total number of layers in the network. W indicates the weights and b, the biases for each layer. BN in the equations indicate batch normalization to alleviate the internal covariate shift. σ refers to a rectified linear unit (ReLU) which introduces non-linearity between layers. For the first layer, filters of size $c \times s1 \times s1 \times a1$ were used to generate $a1$ feature maps. s represents filter size and c represents the number of image channels. For layers 2 through$L1$, filters are of size $a1 \times s2 \times s2 \times a2$. For the last layer, we use filters of size $a2 \times s3 \times s3 \times c$ to estimate the negative residual. The de-rained image is obtained by directly adding the estimated residual to the rainy image X.

3.2.3. Training

The network was trained with 1100 image pairs. Random image patches of size 64 $\times$ 64 were generated from the images and were used for training. A stochastic gradient descent optimizer with a learning rate of 0.15 was used for the first 100,000 epochs. For the next 100,000 epochs, the same optimizer but with a learning rate of 0.1 was used to minimize the loss function.

4. EXPERIMENTAL RESULTS

The experimental results for dehazing images and rain removal are included as separate sections since separate experiments were conducted for each process. The results section expands on how the images are dehazed or how the rain droplets were removed from images. In addition to that SSIM index between the distorted image - clear image and distortion removed - clear image pairs were also recorded to quantify how well the pipelines are effective in removing these distortions.

4.1. Experimental results for Haze removal

To find out how well the model has generalized for the task of haze removal from images, a test set was created which does not include the same images that were used for the training process. If the algorithm can remove haze from these hazy images, then it can be concluded that the model has trained well for this operation. Figure 3 shows the hazy, dehazed, and the corresponding ground truth images along with the hazy-clear and dehazed-clear SSIM scores.

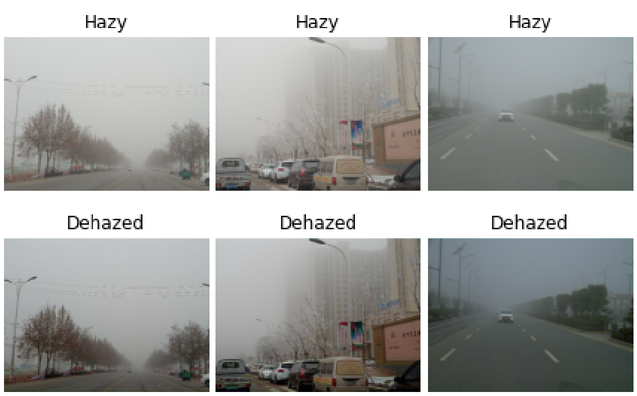
Figure 4 shows the dehazed images for a sample of test images provided specifically for testing. The images did not have their corresponding ground truth images and hence their SSIM were not included for these sets of images. Since the ground truth images were not provided, the comparison cannot be made but the results show that haze was removed from the scene.

4.2. Experimental results for Rain removal

The experimental setup for rain removal from images is conducted similarly to the haze removal process as well. A sep-

Fig. 3. 3 different examples of images with haze, haze-removed images and the corresponding ground truth images each with their SSIM scores with themselves and their corresponding clear images.

Fig. 4. 3 different examples of images with haze, haze-removed images and the corresponding ground truth images each with their SSIM scores with themselves and their corresponding clear images.



arate test set of images was created which is separate from the training samples that were used to train the rain removal model. These test images are passed through the pipeline and the output should be same as the input images with rain streaks removed. Once this happens for the test sample, it can be inferred that the model has generalized well and can operate on various rainy images to remove the rain droplets. Figure 5 shows the rainy, derained, and the corresponding ground truth images along with the rainy-clear and derained-clear SSIM scores.

Figure 6 shows the derained images for a sample of test images provided specifically for testing. The images did not have their corresponding ground truth images and hence their SSIM were not included for these sets of images. Since the ground truth images were not provided, the comparison cannot be made but the results show that rain droplets were removed from the scene.

Fig. 5. 3 different examples of images with rain droplets, rain-removed images and the corresponding ground truth images each with their SSIM scores with themselves and their corresponding clear images.

Fig. 6. 3 different examples of images with rain droplets, and their corresponding derained images.

5. DISCUSSION

In this section, we are going to discuss why a particular methodology works for a set of images and include some of the failure cases obtained from various experiments and provide a possible explanation of why this method did not work.

We experimented on how the images would look like if the hazy image is passed through the dehazing pipeline twice instead of just once and check how well or worse it performs. Figure 7 shows how the image looks like after passing the input image through the pipeline once and when it passed through twice. It can be seen that, when the input image is passed through it once, it removes the hazy components to an extent but retains the color of the scene properly. But when the image is sent through it twice, the method removes a lot more haze but the image loses some of its features. For example, in Figure 7, the image after being dehazed twice became darker and saturated and even though it has all the content of the input image, color information is sometimes lost. This method can be considered if the image's color of the scene is not very important and if dehazing the image is the most important criterion to increase visibility. When the image is passed through the network once, it removes the hazy components corresponding to the atmospheric scattering model. But when the same image is passed through twice, the image looks saturated. This is because a lot of images in the training dataset look saturated. It can be seen that, when the initial dehazed image is obtained, it has a lesser degree of haze and similar images in the training dataset looked a lot more saturated. This is probably the reason why when the images are passed through the pipeline twice, it tends to get saturated.

Fig. 7. Input dehazed image and the image when passed through the dehazing pipeline once and when the image is passed through the pipeline twice.

In the test set of rain removal, a lot of images had really small droplets which almost looks similar to haze in images. So, we conducted an experiment where we passed the input rainy image through the dehazing pipeline after sending it through the deraining pipeline. Figure 8 summarizes the result for this experiment. It can be seen that for the images which have very small droplets that look like haze, it was able to remove those noises. It can be seen in Figure 8 that when the two pipelines are cascaded together, they can remove far more noises in the scene than when they are operating alone. The rainy images in many of the real world rainy images had a lot of haze and very thin rain streaks. In such cases, the cascaded model seemed to perform well since the derained image still had traces of haze in it.

Fig. 8. Input rainy image and the image when passed through the deraining pipeline alone and when the image is passed through both the pipelines.

6. REFERENCES

[1] Ren et al., "Single image dehazing via multi-scale convolutional neural networks," Springer International Publishing, 2016.

[2] Cai et al., "Dehazenet: An end-to-end system for single image haze removal," *IEEE Transactions on Image Processing*., vol.25, pp.5187-5198, Nov.2016.

[3] Li et al., "A Cascaded Convolutional Neural Network for Single Image Dehazing," *IEEE Access*., Mar.2018

[4] He et al., "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*., Dec.2015

[5] Li et al., "Single Image Rain Streak Decomposition Using Layer Priors" *IEEE Explore*., 2017

[6] Luo et al., "Removing rain from a single image via discriminative sparse coding" *IEEE Explore*., 2015

[7] Li et al., "AOD-Net: All-in-One Dehazing Network", *2017 IEEE International Conference on Computer Vision (ICCV)*., pp.4780-4788, Oct.2017

[8] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding and J. Paisley, "Removing Rain from Single Images via a Deep Detail Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

A. APPENDIX

1. Link to image dehazing Colab notebook
2. Link to image deraining Colab notebook
3. This link contains the input rainy images given as the test set, the derained images and the images which are both derained and dehazed.
4. This link contains the input hazy images given as the test set, the dehazed images and the images which are dehazed twice.