# Image Dehazing by Combined Network

SHABEEBA N
*Department of Computer Science and Engineering ,*
*MES College of Engineering,Kuttippuram,india*
shabeebanpuliyavu@gmail.com

Mr.SREEKANTH E S
Assistant Professor
*Department of Computer Science and Engineering ,*
*MES College of Engineering,Kuttippuram,india*
Sreekanthes@mesce.ac.in

***Abstract*—Haze is an everyday weather phenomenon. It is primarily caused by numerous tiny particles in the atmosphere; these particles scatter and absorb light, thereby producing the haze weather effect. Images collected under hazy weather conditions exhibit low-picture-contrast and color-saturation problems. Therefore, haze can be seriously detrimental to the performance of a viual system. Based on designing a fully convolutional neural network to recognize haze structures in input images and restore clear, haze-free images**

**Depth information is the most important information in the image dehazing process. Method proposes an end-to-end generative method for image dehazing. Method use a Generic Model-Agnostic Convolutional Neural Network also known as GMAN for our network.**

***Keywords—Image dehazing, Convolutional Neural Network, GMAN Network, Parallel Network, TensorFlow,***

## I. INTRODUCTION

Haze and smog are among the most common environmental factors impacting image quality and, therefore, image analysis. This paper proposes an end-to-end generative method for image dehazing. Generic data models are speculations of traditional data models. They characterize normalized general connection types, along with the sorts of things that might be connected by such a connection type. Clarifying the hazy images and ensuring the effectiveness of the vision system has been an active topic for researchers and has implications in the fields of computer vision and imaging sciences.

## RELATED WORKS

### A. PHYSICAL MODEL

The atmospheric scattering model has been the classical description for the hazy image generation process

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

where I (x) is observed hazy image, J (x) is the scene radiance ("clean image") to be recovered. There are two critical parameters: A denotes the global atmospheric light and t(x) is the transmission matrix defined as:

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

where β is the scattering coefficient of the atmosphere, and d (x) is the distance between the object and the camera. Traditional Methods: coped with haze removal by maximizing the local contrast. proposed a physically grounded method by estimating the albedo of the scene. discovered the effective dark channel prior (DCP) to more reliably calculate the transmission matrix. Further enforced the boundary constraint and contextual regularization for sharper restored images. An accelerated method for the automatic recovery of the atmospheric light was presented in developed a color attenuation prior and created a linear model of scene depth for the hazy image,
and then learned the model parameters in a supervised way.

### B. DEEP LEARNING METHODS

CNNs have witnessed prevailing success in computer vision tasks, and are recently introduced to haze removal. exploited a multi-scale CNN (MSCNN), that first generated a coarse-scale transmission matrix and later refined it. proposed a trainable end-to-end model for medium transmission estimation, called DehazeNet. It takes a hazy image as input, and outputs its transmission matrix. Combined with the global atmospheric light estimated by empirical rules, a haze-free image is recovered via the atmospheric scattering model. All above methods share the same belief, that in order to recover a clean scene from haze, it is the key to estimate an accurate medium transmission map. The atmospheric light is calculated separately, and the clean image is recovered based on. Albeit being intuitive and physically grounded, such a procedure does not directly measure or minimize the reconstruction distortions. As a result, it will undoubtedly give rise to the sub-optimal image restoration quality. The errors in each separate estimation step will accumulate and potentially amplify each other. In contrast, AOD-Net is built with our different belief, that the physical model could be formulated in a "more end-to-end" fashion, with all its parameters estimated in one unified model. AOD-Net will output the dehazed clean image directly, without any intermediate step to estimate parameters. Different from that performs end-to-end learning from the hazy image to the transmission matrix, the fully end to-end formulation of AOD-Net bridges the ultimate target gap, between the hazy image and the clean image.
using the COCO Dataset which is a large-scale object detection, segmentation, and captioning dataset.

## PROPOSED SYSTEM

Method proposes an end-to-end generative method for image dehazing. It is based on designing a fully convolutional neural network to recognize haze structures in input images and restore clear, haze-free images.

Generic data models are speculations of traditional data models. They characterize normalized general connection types, along with the sorts of things that might be connected by such a connection type. Proposed Method is actually a comparison study of the proposed depth information network Our work basically can create a python application where we train a model with the proposed depth info from different datasets. We can achieve this in python using tensorflow
Our model doesn't use the physical atmosphere scattering model which is widely used in most of the cases of image dehazing.

Here use a Generic Model-Agnostic Convolutional Neural Network also known as GMAN for our network model
GMAN network is agnostic in the sense that without using the atmosphere scattering model it produced better results than all previous models.

### A. *Generic Model-Agnostic Convolutional Neural Network*

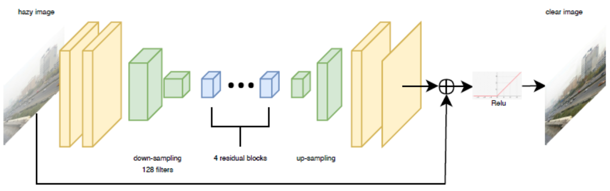


Fig 1; GMAN Architecture(without perceptual loss)

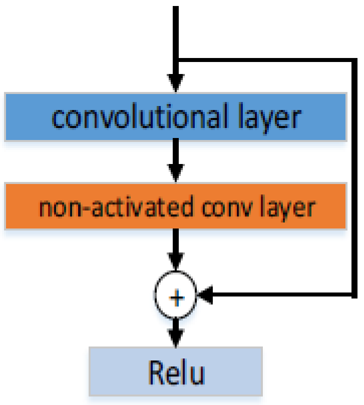


Fig 2; CNN with shortcut connection.

Functionally addressing, this architecture is composed of an end-to-end generative method which uses an encoder-decoder approach to solve the dehazing problem.
The first two layers which input image encounters are convolution blocks with 64 channels. There are 2 down sampling blocks(encoder) with stride 2. The encoded image is then fed to a layer build with 4 residual blocks. Each block contained a shortcut connection(same as ResNets). This residual layer helps to understand the haze structures. After these comes the up sampling or deconvolution (decoder) layer which reconstructs the output of residual layers.

The last two layers are convolutional blocks that transform the up sampled feature map into a 3 channel RGB image which finally added to the input image(global residual layer) and then ReLU to give dehazed output.
This global residual layer helps in capturing the boundary details of objects with different depths in the scene.
The encoder part of architecture helps to decrease the dimension of the image and then fed the down sampled image to the residual layer to extract the image features and the decoder part is expected to learn and regenerate the missing data of the haze-free image.
parallel network ,to improve performance along with mean-squared loss, the perceptual loss was also used.

### TensorFlow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks

To begin coding, follow the steps below:

1. Pre-processing and loading data: -

The images are resized into a standard resolution. Then normalized and returns the image in the form of n. This function takes the path to the image and read it and decode it into a uint8 tensor. It resize to 412x548 image size in the dataset is 413x550 ,but the network function is creating issues because of odd input values. Finally normalize it and returns the image in the form of normalized tensors nnormalized tensors

2. Workout the network function: -

We then construct the functions of the network

First is the GMAN network in which all the layers have several filters as 64 except encoding layers having 128 and the final output layer has 3 channels(RGB)

After GMAN, Parallel Network(PN) has dilated convolution layers with all layers having 64 filters except the last one with 3 channels(RGB).

3. Training: -

We then write the training function of the model.
In each epoch, have a training loop and a validation loop.
In the training loop, we take the training data and compute the gradients and apply them to calculate training loss.
In the validation loop, here take those gradients computed in

that epoch and apply them on validation data to check the output and validation loss. Finally, save the model of that epoch.

## B. *Parallel Model*

PN(Parallel network) architecture is shallower than GMAN but has the same encoder-decoder structure.
But what difference it makes is it uses dilated convolutions(or convolution with holes) instead of traditional convolutions. The convolutional layer with a larger reception field is not learnable to generate a generalizable feature map. Dilation allows the exponential increase of the reception field without spatial dimensions loss in pixel level. In short, this parallel network helps to capture the details overlooked by the GMAN network.

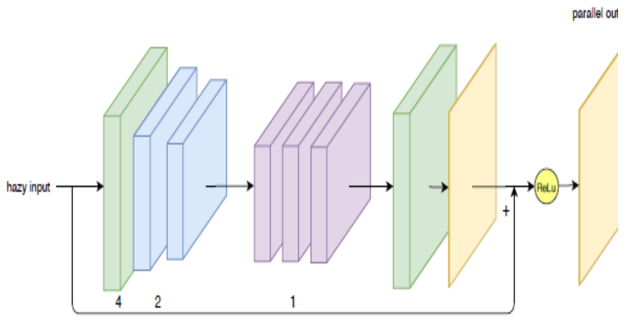


Fig 3 :Parallel Network, The dilation rate of these blocks are green: 4, blue: 2, and purple: 1

The First 3 layers are encoder blocks with a dilation rate of 4, 2, 2 respectively .Then coming three layers have a dilation rate of 1. Followed by a deconvolution block with dilation rate 4 to transfer the feature maps to their original size. After this comes to the final convolution layer which transforms the image to a 3 channel RGB image. Throughout now all layers have 64 channels except the last one. After this same as GMAN output is added with the original input and passed through the ReLU unit.

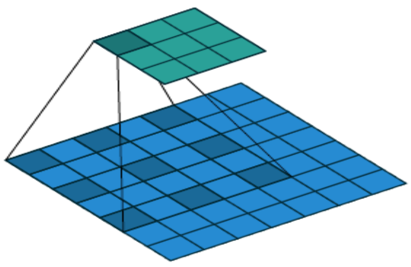


Fig 4: Dilated Convolution with kernel-size 3,

dilation rate 2.

## C. *Combined Network*

Inside our parallel network, whose detail is shown in Fig., almost all convolutional layers (except the last one that generate RGB image) employ the dilated convolution to extract features using larger reception fields and reconstruct a RGB image that is complementary to the output of GMAN. The first layer of PN has the dilation rate 4 and this reduce the size of the input image. Following are two layers with dilation rate 2, and then they are connected with three layers with dilation rate 1. These 3 different rates lead to different effective kernel size as: 9 × 9, 5 × 5 and 3 × 3 since input kernel size of dilated convolution layers are unified to 3 × 3. To transfer the feature maps back to original size, a dilated deconvolution layer which has dilation rate 4 is employed and the output size is 224 × 224. After deconvolution layer, I also use a normal convolutional layers to obtain the RGB image and also regard this layer as refinement operation. Although the PN is a shallow network with less than 10 layers, I still drag the input image to the last layer and do addition operation before ReLu function in order to calculate residual image which makes the network easier to be trained and optimized (same as global residual learning in GMAN). Until now, all layers except the last convolutional layer have 64 channels (number of filters), and along with GMAN's output, two haze-free images are generated from the two streams. Since these two outputs are very similar, I set two dynamic weights $\alpha1$ and $\alpha2$ to those images (see Equation (3) for balancing the contribution from two streams of our entire network. Then final estimation of the original haze-free image is the combination of two globally weighted map, and this process can be seem as a fusion module where shortcomings of GMAN are corrected and refined by PN's complementary output.

$$J\hat{}\ overall = \alpha 1 J\hat{}GMAN + \alpha 2 J\hat{}\ P\ N \qquad (3)$$

And through training GMAN and PN by minimizing the similarity between $J\hat{}$ *overall* and ground truth image, $\alpha1$ and $\alpha2$ can learn to reach the optimal value automatically

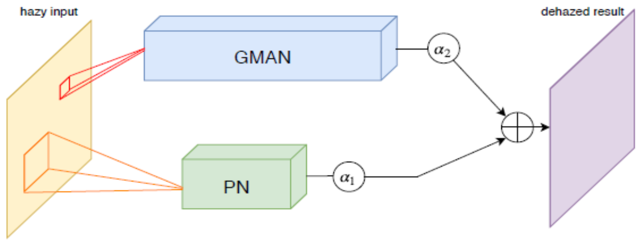


Fig 5: Combined Network.

## D. *Functions used to output*

def load_image() function takes path to the original and hazy image then make a split dictionary with keys: 'train', 'val' with 90% training data and 10% validation data.
from_tensor_slices() function on the original and hazy image paths of the training set followed by a mapping function that loads the image(load_image).
display_img() function to display the output of validation data after the training of the individual epoch. It takes a model, hazy, and original image as an argument.

Network function
Conv2D() Transpose to construct this function.

train_model()
train_model(epochs, train, val, net, train_loss_tracker, val_loss_tracker, optimizer)

In each epoch, we have a training loop and a validation loop.

In the training loop, we take the training data and compute the gradients and apply them to calculate training loss.

In the validation loop, we take those gradients computed in that epoch and apply them on validation data to check the output(using display_img function) and validation loss.

Finally, save the model(weights, variables) of that epoch and reset the loss metrics.

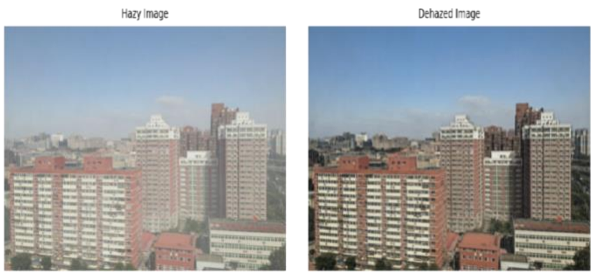
evaluate()

evaluate(net, test_img_path)

some random foggy (naturally hazed) images from kaggle datasets.9U

## CONCLUSION

Depth information is the most important information in the image dehazing process. Method proposes an end-to-end generative method for image dehazing. Method use a Generic Model-Agnostic Convolutional Neural Network also known as GMAN for our network model. GMAN network is agnostic in the sense that without using the atmosphere scattering model it produced better results than all previous models. Work successfully designed an algorithm to dehaze(restore)a hazy image. Key points learned here are GMAN architecture and research behind the dehazing domain. How to custom train a model in TensorFlow How to deal with dataset in which features and labels both are images



## REFERENCES

[1] M. Zheng, G. Qi, Z. Zhu, Y. Li, H. Wei and Y. Liu, "Image Dehazing by an Artificial Image Fusion Method Based on Adaptive Structure Decomposition," in IEEE Sensors Journal, vol. 20, no. 14, pp. 8062-8072, July15, 2020, doi: 10.1109/JSEN.2020.2981719

[2] Dong, Hang, et al. "Multi-scale boosted dehazing network with dense feature fusion." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.

[3] Zhang, Xiaoqin, et al. "Multi-level fusion and attention-guided CNN for image dehazing." IEEE Transactions on Circuits and Systems for Video Technology 31.11 (2020): 4162-4173.

[4] Jiao, Wenjiang, et al. "Single image mixed dehazing method based on numerical iterative model and DehazeNet." Plos one 16.7 (2021): e0254664.

[5] Tan, Mingming, et al. "Image-dehazing method based on the fusion coding of contours and colors." IEEE Access 7 (2019): 147857-147871.