CSE527 Literature Review
Deep Learning Dehazing Techniques

Alex Cuba
acuba

Rebecca Hassett
rhassett

1 Introduction

Dehazing images is essential for computer vision tasks such as object detection, image registration, and image segmentation. Neural networks have existed for many years, but before they gained popularity in dehazing, earlier techniques relied on the availability of scene depth information or multiple images of the same scene. Oakley and Satherley [21] assumed scene depth is known when developing a physics-based scene contrast restoration model. Kopf et al. [9] uses georeferenced images encoding scene depth and texture information. Narashimhan et al. [19] has a scattering model that expects at least two or more images of the same scene. As research pivoted towards single image dehazing techniques, techniques started focusing on statisical priors such as dark channel prior, contrast maximization, and color attenuation. Machine learning techniques such as support vector machines and random forests were also commonly used. Support vector machines work well with high dimensional data that has limited training samples which are properties of high resolution satellite imagery [17]. Around 2014, deep learning gained popularity leading to models such as DehazeNet and AOD-Net. Modern dehazing techniques are being applied to video object detection and scene depth estimation as well.

2 Dehazing Overview

Haze is an atmospheric condition where dust, fog, and other particles obscure visibility. The atmospheric scattering and absorption caused by haze prevents all of the scene reflected light from reaching the imaging equipment [13] Algorithms suffer from degraded scene radiance, as well as loss of contrast and color fidelity. This results in poor performance on tasks such as object detection and segmentation [13]. The next section will discuss statistical prior-based techniques used to combat these issues.

3 Prior-Based Dehazing Techniques

Before deep learning methods with large and varied data sets were more available and common, computer scientists were more in favor of developing methods of training networks on a single image, in a method known as Deep Image Prior. A prior distribution, as the name states, is a distribution which represents what a natural image should look like. This distribution as well as other features are traditionally hand-crafted, and can be fitted to account for many different image related problems including denoising, superresolution, flash-no flash reconstruction, and more. While in the realm of dehzing, prior-based techniques have fallen out of favor for deep learning techniques, prior-based methods are still used in computer vision, and as shown by Ulyanov et all. in [28], it preforms as well if not better than deep learning networks in the previously mentioned problems of denoising, superresolution, et cetera, due to not having issues of overfitting.

3.1 Contrast Maximization

The concept of maximizing the contrast of an image was one of the first single image dehazing methods derived that did not require any input from the user, or geometric information about the image. The way that Tan et al. were able to come to this conclusion stems from two main observations: that images with high visibility tend to have a higher contrast than those with low visibilty, and airlight that depends on its distance from the viewer tends to be smooth. [25] These observations led to the construction of a method which determines light chromaticity of the image to removes the light color, which in return allows us to determine the distance between the labels of neighboring pixels in order to determine the smoothness of the image. We then take these smoothness values and plug them into Markov random fields, or MRFs, to get the estimated airlight of the image and finally use the estimated airlight to compute the attenuation and dehaze the image. This whole process is done in several steps, starting with calculating the chromaticity of the image, which uses

the following formula:

$$I(x) = D(x)e^{-\beta d(x)}\gamma(x) + A(x)\alpha \quad (1)$$

I = image intensity,
x = 2D spatial location,
β = atmospheric attenuation coefficient,
d = distance between an object in the image and the observer,
α = light chromaticity.
While α and γ are both color vectors, D and A are both scalar values with the following formulas:

$$D(x) = L_{\infty r}\rho_r(x) + L_{\infty g}\rho_g(x) + L_{\infty b}\rho_b(x) \quad (2)$$

$$A(x) = (L_{\infty r} + L_{\infty g} + L_{\infty b})(1 - e^{-\beta d(x)})) \quad (3)$$

In these formulas, L_∞ is meant to represent the atmospheric light, and ρ is the reflectence of the object. Both equations divide these values into their respective rgb color channels. The result of these calculations are shown by Tan et al. in their paper by using the image shown on the next page.

Figure 3. *Left: Input image. Right: the result of normalizing the environmental light,*

After this normalization has been made, we can now calculate the data and smoothness of the image through an MRF using the following formulas centered on a square patch p_x, with values A_x and η, which represent A(x) and the strength of the smoothness term respectively:

$$E(A_x|p_x) = \sum_x \phi(p_x|A_x) + \eta \sum_{x,y \in N_x} \psi(A_x, A_y) \quad (4)$$

$$\phi(p_x|A_x) = \frac{C_{edges}([D\gamma']_x^*)}{m} \quad (5)$$

$$\phi(p_x|A_x) = 1 - \frac{|A_x - A_y|}{\Sigma_c L_{\infty c}} \quad (6)$$

In these equations $C_{edges}([D\gamma'$ is our cost function, and once we find this value, which is equivalent to our estimated airlight, we can use it to determine our attenuation and dehaze the image. The final results of these equations are shown to the right.

Figure 11. *Top: input image. Second from top: the direct attenuation. Third from the top: the ground truth. Bottom: the airlight. Note that, we increase the intensity of the direct attenuation, since the input image is considerably darker than the ground truth.*

3.2 Dark Channel Prior

The use of a dark channel prior to dehaze an image was developed after the concept of maximizing the contrast. Different from maximizing the contrast, this method is based on the observation that in haze-free images, non-sky patches have at least one channel that has very low intensity in some pixels. This means that the minimum intensity in a patch has a very low value. [6] To calculate this observation, we first need to estimate the transmission of the atmospheric light using this equation:

$$\bar{t}(x) = 1 - \min_{c}(\min_{y \in \Omega(x)}(\frac{I^c(y)}{A^c})) \tag{7}$$

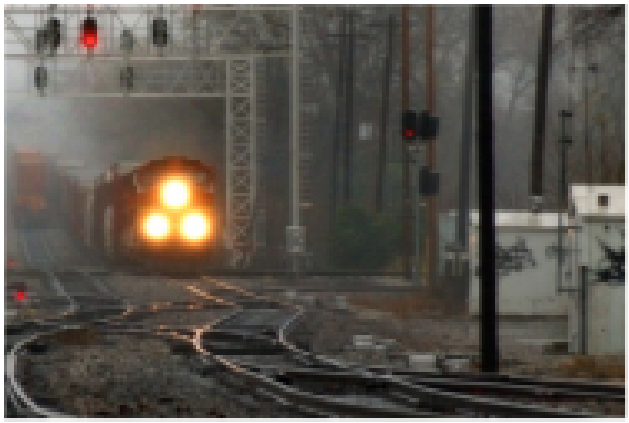
In this equation, $\frac{I^c(y)}{A^c}$ is the normalized haze image, which we minimize to directly give us the estimation directly. After this, we apply a soft matting algorithm to to refine our transmission, as our transmission mapping has a similar form to our imaging equation. [6] We take the vector forms of our map t(x) and $\bar{t}$(x), and create the following loss function, which will give us transmission E, based on the Matting Laplacian matrix L, with the λ is the regularization parameter:

$$E(t) = t^T L t + \lambda (t - \bar{t})^T (t - \bar{t}) \tag{8}$$

In this equation, the first term is the smooth term, and the second is the data term. After the matting has been completed, we compute the scene radiance and and estimate the atmospheric light, using this equation:

$$J(x) = \frac{I(x) - A}{\max(t(x), t(0))} \tag{9}$$

where I is the image and t(x) is the transmission matrix. An example of the outputs of this entire process can be seen to the side. The top image is the input image, the middle is the output, and the bottom on is the dark channel. The red rectangles show the patches that are used to calculate the atmospheric light.



3.3 Color Attenuation Prior

The color attenuation prior method, developed by Zhu et al. in [33], builds off of the previous prior methods, and like them, builds its prior off of a series of assumptions. In this case, the color attenuation prior makes the assumption that the more hazy an image is, the greater the difference between the saturation and the brightness of the given image patch. However, this can vary greatly over an image depending on

its depth. To avert this, we also need to make an assumption that the haze of the image is positively correlated to the depth, based on the following equation, where d is our scene depth, c is the concentration of the haze, v is the brightness, and s is the saturation:

$$d(x) \propto c(x) \propto v(x) - s(x) \tag{10}$$

With this set of assumptions, after our generation of hazy images, we can estimate our coefficients for scene depth, saturation, and brightness, from the following equation:

$$d(x) \sim p(d(x)|x,\theta_0,\theta_1,\theta_2,\sigma^2) = N(\theta_0 + \theta_1 v + \theta_2 s, \sigma^2). \tag{11}$$

x = location within image,
$\theta_0, \theta_1, \theta_2$ = linear coefficients.

Once these coefficients are estimated, we can the calculate the the estimation of the depth information, and therefore restore the depth map of the hazy image. However, this may fail in certain situations, such as when there are large white objects. This is due to high brightness and low saturation, which the model interprets as being very distant in the image. To prevent these issues from arising, we make a raw depth map with consideration for each pixel in a neighborhood, as well as make the assumption that the scene depth is locally constant. To put this into motion, we must use the following equation, where $\Omega_r(x)$ is the r by r neighborhood that is centered at x:

$$d_r(x) = \min_{y \in \Omega_r(x)} d(y) \tag{12}$$

This equation, depending on the size of the neighborhood, can lead to large amounts of blocking artifacts on the image, however this can be resolved with some simple image guiding filtering. The image below shows this process in action.

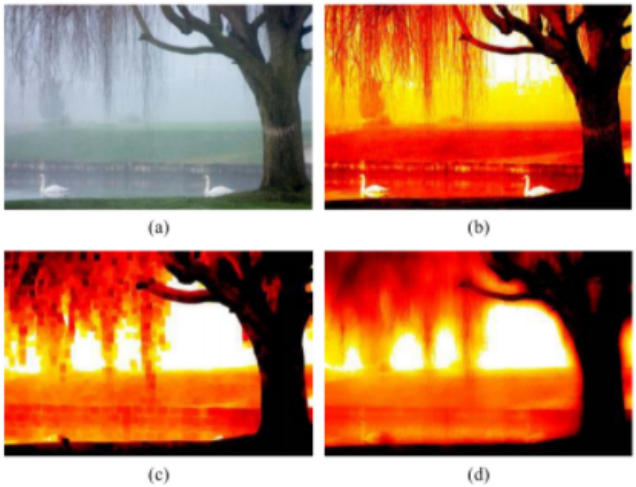


Fig. 8. Refinement of the depth map. (a) The hazy image. (b) The raw depth map. (c) The depth map with scale $r = 15$. (d) The refined depth map.

Now that our depth map has been calculated, we can move onto the final step of this model, which is to recover the final estimated scene radiance. To do this we first calculate atmospheric light A for our hazy image I by selecting the top 0.1 percent of pixels in the depth map. This calculation is shown in this equation:

$$A = I(x), x \in \{x | \forall y : d(y) \leq d(x)\} \tag{13}$$

Once we calculate the atmospheric light using this equation, we then estimate our transmission t, with our scattering coefficient β:

$$t(x) = e^{-\beta(x)} \tag{14}$$

We then plug this transmission value into our final scene radiance equation:

$$J(x) = \frac{I(x) - A}{t(x)} + A = \frac{I(x) - A}{e - \beta d(x)} + A. \tag{15}$$

However, the model actually limits our transmission values between 0.1 and 0.9 to prevent the generation of too much noise, so the actual equation looks like this:

$$J(x) = \frac{I(x) - A}{\min\{\max\{e^{-\beta d(x)}, 0.1\}, 0.9\}} + A \tag{16}$$

In both of these equations, the solution J is our haze-free image.

3.4 Comparison of Prior Methods

Unfortunately, the source code created for the prior based methods is not publicly available, and as a result we cannot test on it to get definitive qualitative results. However, the authors of these papers compared their code to these other models for us, allowing us to state the broad similarities and differences between them. While these methods are similar in their approaches of using a prior, the assumptions that each method makes leads to one method struggling to correctly dehaze an image in one instance, while in another the same method preforms greater than all others. For example, since Tan et al.'s method looks for high intensity pixels in an image for its atmospheric light calculation,this can lead to errors if the brightest pixel does not belong to the sky, but a white object such as a car or house, as mentioned by Tang et al. in [26]. This issue does not arise in the color attenuation method by Zhu et al., as they account for this and use the previously mentioned raw depth map with considerations to pixels in the neighborhood. Tan et al.'s equations also underestimate the transmission due to not being physically based. As a result, images produced by following the equations in Tan et al.'s paper tend to be over saturated. [6] This allows the dark channel prior to also work better on black and white methods than the maximization of the contrast. In turn, the dark channel prior method fails in situations where the background is a similar color to the atmospheric light. As we can see in the example below, the

haze thickness of the person and the marble behind him are about the same, but when seen through the dark channel, the person is noticeably less hazy. [26] We can see the contrast feature on the right gives a much better source of information, which is provided by Tan et al.'s method.

(a) hazy image (b) dark channel D_{10} (c) contrast C_{10}

The dark channel prior method also tends to over-dehaze images in comparison to other methods, due to it taking a lower bound of its t(x) transmission matrix, which it achieves by using a "min" filer, which was shown in equation 7. This over-dehazing often results in He et al.'s results to be darker than they should be, as well as the colors in the image becoming distorted. This especially becomes relevant when compared to Zhu et al.'s color attenuation prior, which more accurately removes the haze without distorting colors. A comparison of these two methods side by side is provided below. The left column is the original hazy image, the middle column uses the dark channel prior, and the final column uses the color attenuation prior.

4 Dehazing Deep Learning Techniques

Dehazing techniques prior to deep learning relied on handcrafted features including maximum contrast, dark channel, and color disparity that involved complicated fusion methods. The assumptions made for these visual image properties is not valid for all cases of hazy images. He et al. assumes the dark channel prior is near zero for haze-free images, yet this is not true when the atmospheric light is similar to the scene objects. Tan et al. maximizes local contrast to improve visibility, but this leads to cases of dehazed images with halos and color distortion. Convolutional neural networks can be utilized to learn the weights mapping hazy image inputs to their transmission maps. These mappings model intrinsic haze formation properties without statistical priors. [24] The atmospheric scattering model is used by many of the CNNs discussed below to model hazy image formation:

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{17}$$

where I(x) is the output hazy image, J(x) is the dehazed image. The two parameters estimated by many of the dehazing deep learning networks are A for global atmospheric light and t(x) for the transmission matrix. The transmission matrix is denoted by:

$$t(x) = e^{-\beta d(x)} \tag{18}$$

where β is the atmospheric scattering coefficient and d(x) is the distance between scene objects and the camera [11].

4.1 Multi-Scale CNN

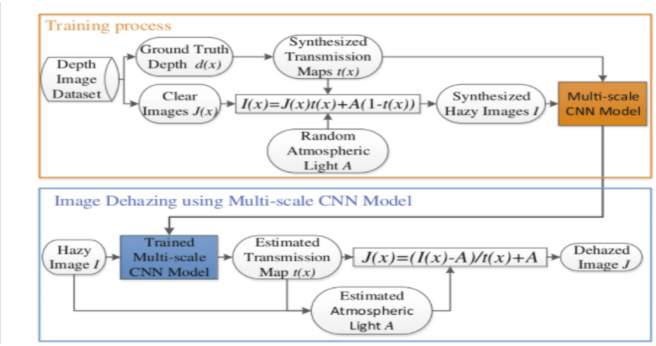


Figure 1: Multi-Scale CNN [24]

In 2016, Ren et al. proposed a multi-scale CNN for single-image dehazing. This network is trained with a popular dataset for dehazing, the NYU Depth dataset [20], which contains the clear images J(x) and depth maps d(x) used to generate the synthetic hazy images for training. The CNN architecture includes a coarse scale network for a holistic prediction of the transmission map. The coarse scale network design is a linear combination of convolution, max-pooling and upsampling operations followed by ReLU activations. This is followed by a fine scale network to refine the transmission map. Without the fine-scale network, output images lack fine details and contain more halo artifacts. This multi-scale feature mapping approach using different convolutional filter sizes to retrieve features at different scales is common in future architectures. The atmospheric light A is separately

estimated using the highest intensity pixel from the 0.1% darkest pixels in the estimated transmission map. The estimated transmission map and atmospheric light are then used in the atmospheric scattering model to dehaze the images [24]. This technique individually estimates the transmission map and atmospheric light. The disadvantage to individually estimating these parameters is the accumulation of errors resolved by reformulated architectures such as AOD-Net [11].

4.2 DehazeNet

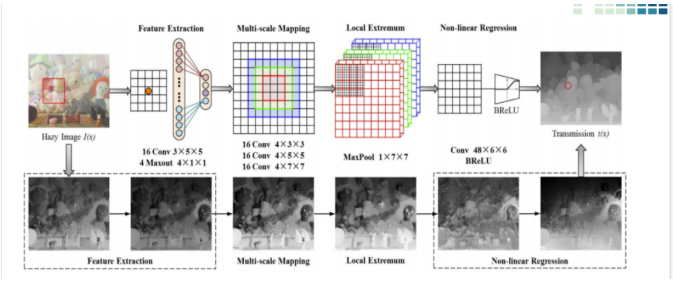


Figure 2: DehazeNet Architecture [1]

Another one of the first end-to-end deep learning systems for dehazing is DehazeNet. DehazeNet has multiple cascading convolutional layers and pooling layers followed by nonlinear activations after some layers. DehazeNet performs four separate operations: feature extraction, multi-scale mapping, local extremum, and non-linear regression. The maxout activation function is used after feature extraction, and the BReLU (Bilateral Rectified Linear Unit) activation function is used for nonlinear regression. Common activation functions are Sigmoid which suffers from vanishing gradient and slow convergence, and ReLU which is designed to work optimally on classification problems rather than regression problems such as image restoration. Ren et al.'s CNN architecture uses linear ReLU activation layers. DeHazeNet lists using BReLU as a major contribution due to its improvements in convergence and reducing search space in dehazing images [1]. AOD-Net and DCPDN switch back to using ReLU activations as it was found to be more effective in their specific settings [11] [31]. Activation functions are an important area to investigate when setting up dehazing neural network infrastructure.

Another major contribution of DehazeNet emphasized by Cai et al. is how DehazeNet directly learns assumptions/priors used in previous methods. DehazeNet extracts features based on the dark channel prior method [6] using an opposite filter weight that sets -1 at the center of one channel. Convolving with this filter has maximum outputs coinciding with the minimum values of the color channels similar to the dark channel prior method. When a round filter weight is used for the convolution, features with maximum contrast are found similar to the maximum contrast prior method [25]. Convolving with

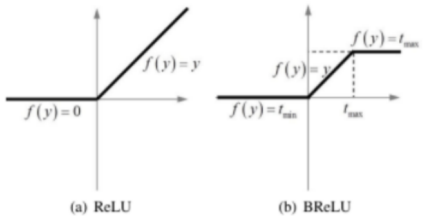


Figure 3: DehazeNet Architecture [1]

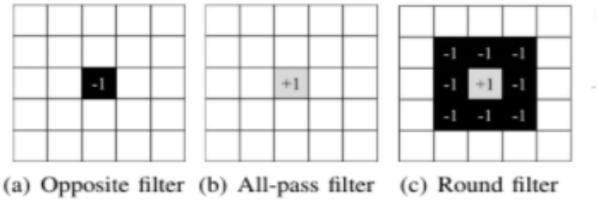


Figure 4: DehazeNet Convolutional Filters [1]

both all-pass filters and opposite filter weights, then the feature outputs are similar to maximum and minimum feature maps which can be used in the color space transformation from RGB to HSV to utilize the color attenuation and hue disparity priors [33], [26]. DehazeNet improves upon this prior methods by better handling white color and sky regions. It is difficult to distinguish between sky and haze since they both use the same atmopsheric scattering model. Contrast magnification methods such as [25] and boundary constraint methods [18] both suffered from sky regions that were oversaturated and much darker than expected. Color attenuation prior [33] and random forest methods avoid distorting the sky color, but non-sky regions are poorly enhanced by the non-content regression models. DehazeNet avoids affecting the sky color while having good dehazing effects in non-sky regions. Based on metrics such as MSE, SSIM, and PSNR, DehazeNet performed much better than techniques based on priors such as color attenuation (CAP), contrast maximization (FVR), dark channel prior (DCP), and using ensemble classifiers such as random forests (RF) [1]. SSIM is used in image restoration to measures the structural similarity between images to capture important local and detailed image information [10].

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{19}$$

4.3 AOD-Net

DehazeNet and Ren et al. multi-scale convolutional neural network accumulate errors by estimating the transmission map in

Metric	ATM [39]	BCCR [11]	FVR [38]	DCP [9]	CAP[2][18]	RF [17]	DehazeNet
MSE	0.0689	0.0243	0.0155	0.0172	0.0075 (0.0068)	0.0070	**0.0062**
SSIM	0.9890	0.9963	0.9973	0.9981	0.9991 (0.9990)	0.9989	**0.9993**
PSNR	60.8612	65.2794	66.5450	66.7392	70.0029 (70.6581)	70.0099	**70.9767**
WSNR	7.8492	12.6230	13.7236	13.8508	16.9873 (17.7839)	17.1180	**18.0996**

Figure 5: MSE, SSIM, and PSNR Results [1]

an intermediate step before estimating the atmospheric light and clear image. In 2018, AOD-Net (All-in-One Dehazing Network) builds on this by estimating both the transmission map and atmospheric light in the same step to directly estimate the clear image. The atmospheric scattering model equation in (17) is reformulated to include both the transmission map and atmospheric light:

$$J(x) = K(x)I(x) - K(x) + b \tag{20}$$

where:

$$K(x) = \frac{\frac{1}{t(x)}(I(x) - A) + (A - b)}{I(x) - 1} \tag{21}$$

AOD-Net consists of a K-estimation module with five convolutional layers. The K-estimation module generates multi-scale features by using parallel convolutions with different filter sizes [11]. It is inspired by the architecture in Ren et al. which concatenates coarse scale features with intermediate fine scale features. DehazeNet also utilizes multi-scale feature extraction with parallel convolutions of varying filter sizes (3x3,5x5,7x7) to achieve scale invariance [1]. AOD-Net concatenates various intermediate convolutional output layers as depicted below:

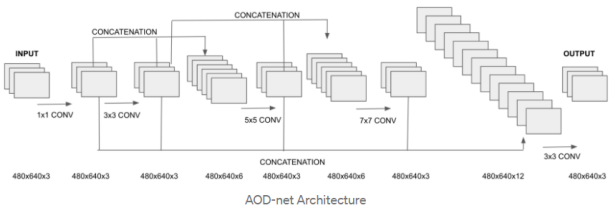


Figure 6: AOD-Net Architecture [11]

The average mean-squared error of AOD-Net is significantly lower than the prior-based methods, and still much lower than DehazeNet and MSCNN.

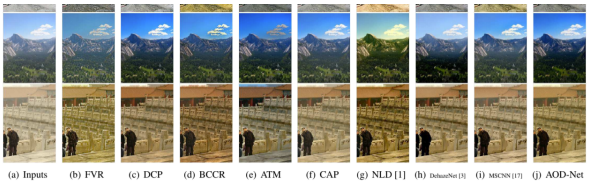


Figure 6. Challenging natural images results compared with the state-of-art methods.

Metrics	ATM [22]	BCCR [12]	FVR [25]	NLD [1]	DCP [8]	MSCNN [17]	DehazeNet [3]	CAP [32]	**AOD-Net**
MSE	4794.40	917.20	849.23	2130.60	664.30	329.97	424.90	356.68	**260.12**

Table 3. Average MSE between the mean images of the dehazed image and the groundtruth image, on TestSet B.

Qualitative comparisons reveal that the prior-based methods results in unrealistic color tones or oversaturation. DehazeNet darkens regions such as the region in the center of the mountain, and CAP blurs textures. The sky region in DehazeNet appears most similar to the input images, whereas MSCNN and FVR are oversaturating the sky region [11].

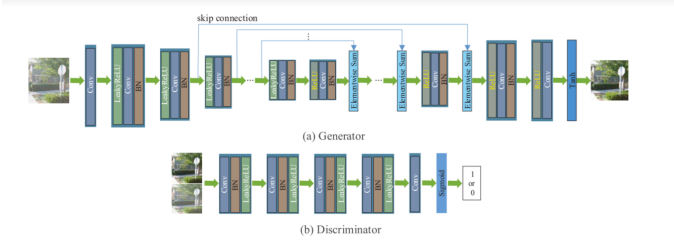


Figure 2: The network structure of the proposed method. The generator network contains an encoder and decoder process. The same color used in the rectangles denotes the same operation. "Conv" and "BN" denote the convolution and the batch normalization operations.

Figure 7: Dehaze-cGAN Architecture [14]

4.4 Dehaze-cGAN

Another architecture developed in 2018 to estimate the transmission map and atmospheric light together uses a cGAN with an encoder-decoder structure for the generator. Inspired by ResNet [7], U-Net [22] architectures and similar to pix2pix [8], skip connections between mirrored layers is added to break through the bottleneck in the decoder. Another similarity to pix2pix is the discriminator architecture using convolutional, batch normalization, and LeakyReLU activation. The final layer is a sigmoid function for normalizing the data to [0,1]. Unlike other architectures using MSE Loss, the cGAN loss includes the adversarial loss, the perceptual loss, L1-regularized gradient prior and content-based pixel-wise loss:

Adversarial Loss where I represents the hazy images and J represents the clear images output by the generator, and D represents the discriminator. Color distortions appear when using just this loss term.

$$L_A = \frac{1}{N}\sum_{i=1}^{N} log(1 - D(I_i, \tilde{J}_i)) \tag{22}$$

Perceptual Loss where F_i represents feature maps from the pre-trained VGG network to constrain the network. Adding

this extra term maintains image details, but adds artifacts that degrades the image quality.

$$L_P = \frac{1}{N}\sum_{i=1}^{N} ||F_i(G(I_i)) - F_i(J_i)||_2^2 \tag{23}$$

L1-Regularized Gradient Prior on Generator Output and Pixel-Wise Loss which removes artifacts

$$L_T = \frac{1}{N}\sum_{i=1}^{N} (||G(I_i) - J_i||_1 + \lambda||\delta G(I_i)||_1) \tag{24}$$

Dehaze-cGAN Generator Loss Function:

$$L = \alpha L_A + \beta L_P + \gamma L_T \tag{25}$$

Dehaze-cGAN Discriminator Loss Function:

$$max_D \frac{1}{N}\sum_{i=1}^{N} (log(1 - D(I_i, \tilde{J}_i)) + log(D(I_i, J_i))) \tag{26}$$

PSNR and SSIM results are higher for Dehaze-cGAN compared to DehazeNet, MSCNN, and AOD-Net which could be attributed to the joint estimation of the transmission map and atmospheric light. A major contribution by Dehaze-cGAN is introducing a discriminator to improve image quality as well as a modified loss function based on pre-trained VGG features and an L1-regularized gradient prior. This could account for less hazy residuals in the Dehaze-cGAN outputs compared to AOD-Net [14].

4.5 Densely Connected Pyramid Dehazing

DCPDN, Densely Connected Pyramid Dehazing Network, is an architecture developed around the same time as Dehaze-cGAN that also uses GANs to estimate the transmission map and clear image jointly. Similar to Dehaze-cGAN, there is an encoder-decoder structure to analyze features from multiple layers of the CNN. This is inspired by MSCNN and DehazeNet's multi-scale feature extraction for scale invariance. DCPDN builds on this with a dense block as the basic structure to maximize information flow along these features and guarantee convergence. There is a multi-level pyramid pooling module to refine the learned features based on the global structural information similar to the fine-scale network in MSCNN. DCPDN also concatenates intermediate layers to prevent bottleneck in the decoder just like Dehaze-cGAN. DCPDN improves on prior methods such as MSCNN and DehazeNet that use MSE loss by using a new edge preserving loss. MSE loss tends to blur and add halo artifacts to the output images. The edge preserving loss is composed of L2 loss, two-directional gradient loss, and feature edge loss:

$$L_E = \lambda_{E,l2} L_{E,l2} + \lambda_{E,g} L_{E,g} + \lambda_{E,f} L_{E,f} \tag{27}$$

The overall loss is:

$$L = L_E + L_A + L_D + \lambda_j L_j \tag{28}$$

where L_E is the edge preserving loss, L_A is the L2 loss for estimated atmospheric light, L_D is the dehazing loss, and L_j is the joint discriminator loss. The figure below illustrates the DCPDN architecture. [31].

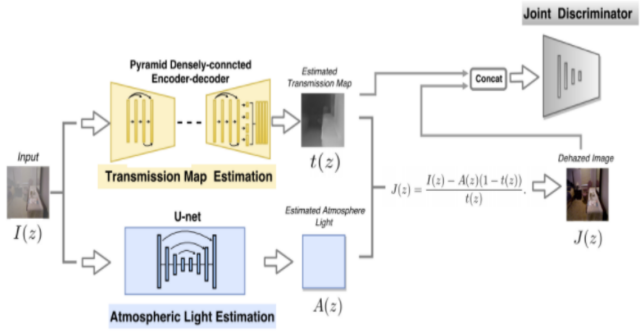


Figure 8: DCPDN Architecture [31]

4.6 Modern Dehazing Approaches

Some of the recent approaches for dehazing have been moving away from using a physical model such as the atmospheric scattering model. GFN, Gated Fusion Network, and GMAN, Generic Model Agnostic Network, emphasize that the atmospheric scattering model is a simplification of haze formation, and may not extend to natural images as well as the tested synthetic images. [15]. Natural images can often be much denser than most synthetic training datasets [5]. GFN identifies and removes haze using image fusion techniques on white balanced, contrast enhanced, and gamma corrected images derived from the original image. These images are derived due to the observation that hazy images have color changes due to atmospheric light and introduced attenuation problems. Both GMAN and GFN use an encoder-decoder network structure with skip connections. GMAN uses MSE Loss and Perceptual Loss where perceptual loss captures more high level feature differences. GMAN calculates the perceptual loss by using both the ground truth and output image in a pre-trained VGG network, similar to dehaze-cGAN. Also similar to dehaze-cGAN, GFN builds a discriminator so that an adversarial loss can be added to the MSE loss to prevent halo artifacts in output images. Another method for removing fog that does not rely on the atmospheric scattering model was DID-MDN. Zhang and Patel developed a Multi-Stream Dense CNN to classify the density of raindrops or fog in the image so that tasks such as dehazing can be efficiently performed [32].

Another recent area of improvement to dehazing has been simultaneous image dehazing and depth estimation. Depth estimation is an important challenge with applications in robotics and augmented reality. Existing depth sensors have limited capabilities inspiring researchers to find low-cost and

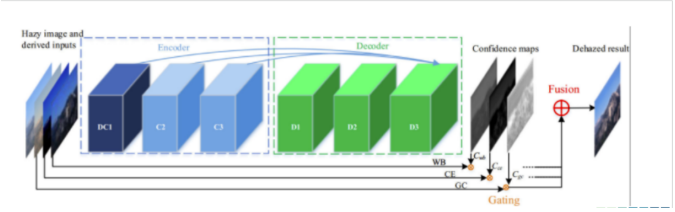


Figure 9: Gated Fusion Network Architecture [23]

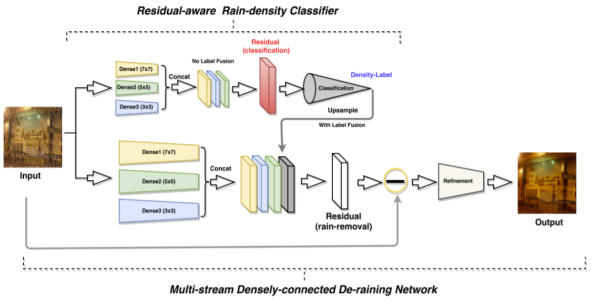


Figure 10: Multi-Stream Dense CNN [32]

efficient deep learning techniques for depth estimation. LiDARs are depth sensors that are expensive and are restricted by only providing sparse measurements for distant objects. Stereo cameras are difficult to use because they require large baselines and computationally expensive calibration. Structured light depth sensors such as the Kinect are sensitive to sunlight and have high power consumption. Ma et al. finds better results in computer vision tasks when sparse depth information is included as an additional channel to an RGB image as input into a ResNet based depth predictor. However, it is not common to have the depth information available in hazy scenes [16] Lee et al. developed a method to handle this by simultaneously estimating the dehazed image and the fully scaled depth map. The architecture includes a single dense encoder and four separate decoders. The decoders individually predict the scene radiance, the atmospheric light and the transmission map for dehazing and the last decoder estimates the depth map. Motivated by DPCDN [31], the decoders have additional refinement blocks with a dense pyramid-like structure with varying spatial scales. This architecture also benefits from a pretrained DenseNet2001 model on ImageNet. Its loss function is a linear combination of reconstruction loss, atmospheric light loss, transmission map loss, and depth transmission consistency loss. This model was tested with the synthetic haze NYU dataset instead of a real-haze dataset [10].

Finding an accurate, representative dataset with haze and ground truth image pairs is another challenge that has gained attention recently. Past techniques such as MSCNN [24] and DehazeNet [1], used the NYU depth dataset where they added synthetic haze. Models trained on synthetic haze had worse performance on naturally hazy images. Training on naturally

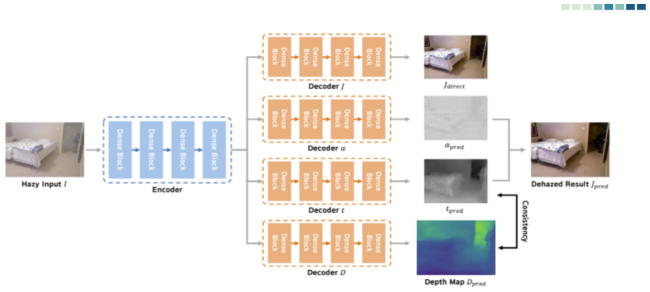


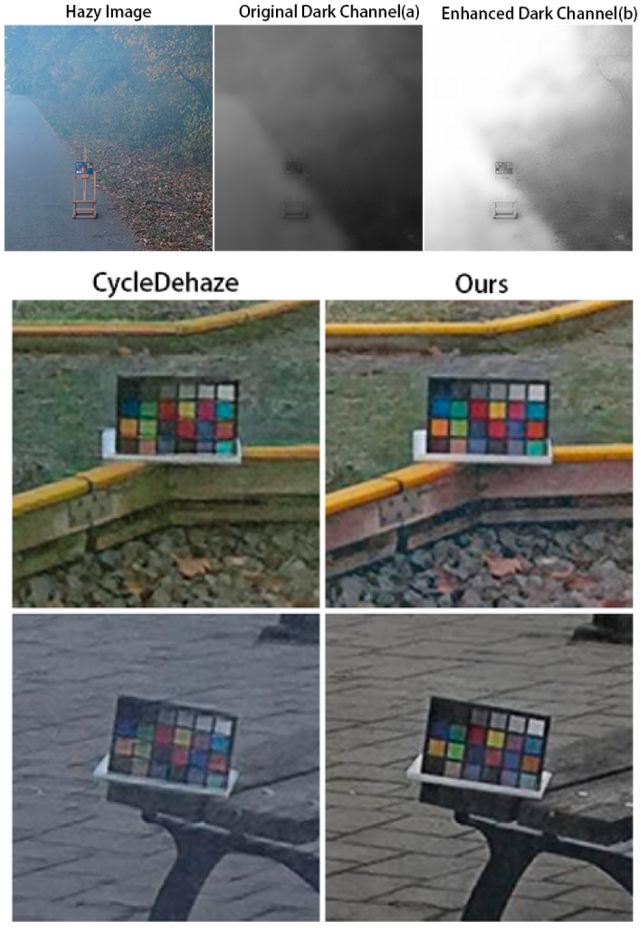
Figure 11: Simultaneous Depth Estimation and Dehazing Architecture [10]

hazy images is a challenge because of the lack of ground truth samples. Codruta et al. created 35 naturally hazy and ground truth image pairs using different scenes and a professional haze machine, the LSM1500 PRO 1500 W. These images were captured in an indoor environment for the I-Haze dataset. These haze generators used cast-type aluminum heat exchange to evaporate water-based fog fluid. These generated particles were all 1-10 microns distributed with a fan [2]. It is crucial to have varying fog particle sizes, shapes and densities in the training set [32]. Another major improvement to the datasets made by Codruta et al. was the use of a Macbeth color checker on the objects in the scene to use in the color restoration CIEDE2000 metric [2]. Another naturally hazy dataset using a fog machine was the Color and Multispectral Image Databases, CHIC, with nine images of varying fog densities and two different scenes [4]. Codruta et al. developed the O-haze dataset that captured outdoor foggy images with 45 different scenes. These images were taken over the span of 8 weeks to capture different outdoor weather environments (sunny, cloudy, etc.). Images were not taken in windy conditions with speeds more than 3 km/hr [3]. I-Haze and O-Haze improve on CHIC by providing greater scene, object, and color variety. The I-Haze and O-Haze datasets have also been used for the NITRE workshop image dehazing challenges.

4.6.1 Attention Mechanism

Originally developed for use in sequence-to-sequence models for language translation, attention mechanisms have only in the past several years become pertinent to the field of computer science, and even more recently for use in dehazing models. Attention mechanisms, as the name suggests, revolve around the concept of 'attention'. This operates similarly to how we as humans give our attention to something. In other words, we direct our focus to what we are trying to pay attention to. When we use attention mechanisms in our deep learning models, it allows the model to pay attention to and put a greater focus on certain factors while processing the data. This became relevant to computer science because it allows our models to take a more human-like approach in analyzing the sets of images we give it. Specifically, we

would allow our model to simply take 'glimpses' at selected regions in an image, rather than using a 'sliding window' approach over the entirety of the image. [34] This allows our models to be less computationally expensive due to us not scaling our computations with the number of pixels in the image. As a result, our models should produce better results in a shorter time frame due to attention mechanism. In regards to dehazing, an attention mechanism approach the problem was not done until this year, by Chen et al. in [35]. In their implementation, they used a version of CycleDehaze, a GAN which generates dehazed images over two generators and discriminators. The attention mechanism is used in collaboration with a dark channel inspired by He et al.'s dark channel prior discussed earlier. The attention mechanism is used to both label the hazy areas as well as the quantification of haze concentration. The attention mechanism achieves this by using a coefficient v to enhance the dark channel. This allows it to more easily tell high concentrations of haze as they appear whiter. An example of this process using the attention mechanism is shown below.



The attention mechanism, as predicted, did have a positive output on Chen et al.'s output when compared to vanilla CycleDehaze. As they reported, their images came out with a more accurate color representation, as well as producing a higher quality image overall. This distinction is shown in the below image. Note that the column labelled "Ours" refers to the improved CycleDehaze with the attention mechanism on the dark channel prior as developed by Chen et al.

4.6.2 Video Dehazing

Li et al. were among the first to explore CNN architectures for video dehazing combined with video detection. The main problem that EVD-Net tackles is temporal fusion for consecutive frames. This technique is promising since it uses multi-frame coherence since transmission map and atmospheric light should not change much over consecutive frames. EVD-Net works with batches of 5 images at a time where t is the current frame, (t-1), (t-2) are previous frames, and (t+1), (t+2) are future frames. Li et al. investigate three different strategies to fuse consecutive frames known as I-Level fusion, K-level fusion, J-level fusion. I-Level Fusion fuses images at the input level. All five input frames are concatenated along their first dimension before the first convolutional layer is applied. The images are then fused at the pixel level before performing single image dehazing on the fused image. K-Level fusion is handled during the K-estimation. J-level fusion fuses images at the output level. K-level fusion had superior results compared to I-Level fusion and J-Level fusion [12].

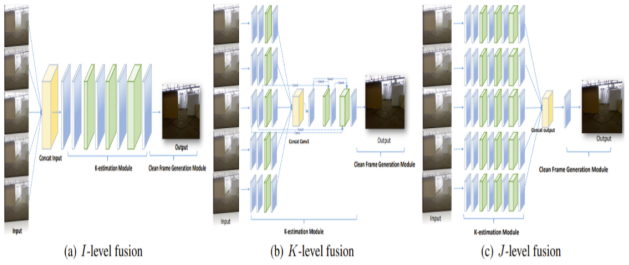


Figure 12: EVD-Net Fusion Techniques [12]

5 Comparison

Most of the earlier deep learning models with trained on synthetic haze datasets only. This section discusses comparing their performances when training on the NYU depth dataset synthetic hazy images and testing on the synthetic hazy images and naturally hazy images separately. For the naturally hazy images, we tested 27 different outdoor and indoor scenes. Original code is attached in the github repository [39].

5.1 AOD-Net Setup

The AOD-Net original code is in a tensorflow ipynb file that can be run in Google Colab. To run this model, we had to disable v2 behavior to prevent errors with v1 tensorflow code.

We added code for loading results and ground truth images, and analyzing the images with PSNR and SSIM.

5.2 DehazeNet Setup

The DehazeNet original code is in Matlab. To run this model, setup Matlab 2020 and Image Processing and Computer Vision Toolbox. DehazeNet also had errors with its convConst.cpp file that needed to have updates made to its variable types. Refer to [36] for the updated file. We added code for testing model on naturally hazy and synthetically hazy images as well as for PSNR and SSIM analysis.

5.3 MSCNN Setup

The MSCNN original code is in Matlab, and we also added code for testing model on naturally hazy and synthetically hazy images as well as for PSNR and SSIM analysis.

5.4 GFN Setup

The GFN original code is in Matlab and uses Matcaffe pretrained models. We needed to setup a Ubuntu virtual machine, install a cpp compiler and install Matlab 2020. The tutorials followed to setup Matcaffe in Ubuntu includes [37] and [38]. Caffe has to be compiled from source files because the precompiled caffe in Ubuntu does not include Matcaffe. Even though we were running on Ubuntu 18, we had to still install the dependencies listed for Ubuntu versions less than 17 according to the documentation. After installing all of the dependencies, we used make to compile caffe and then matcaffe. We could not run this model because there were errors related to solver deletion listed in Appendix 1.

5.5 Results

Most of the earlier deep learning models were only tested on synthetic haze datasets. We decided to compare their performances on naturally hazy images with outdoor and indoor scenes. Even though AOD-Net performed the best on the synthetic haze dataset, it had the lowest PSNR and SSIM scores for the naturally hazy dataset. Listed below are the PSNR and SSIM values:

MSCNN Naturally Hazy Results:

Average PSNR: 16.2820
Average SSIM: 0.4693

MSCNN Synthetically Hazy Results:

Average PSNR: 12.1023
Average SSIM: 0.4833

DehazeNet Naturally Hazy Results:

Average PSNR: 15.8094
Average SSIM: 0.5983

DehazeNet Synthetically Hazy Results:

Average PSNR: 10.8694
Average SSIM: 0.4345

AOD-Net Results:

Average PSNR: 10.9589
Average SSIM: 0.0003193

5.6 Naturally Hazy Dataset

AOD-Net Dehazed Indoor Image:

Dehaze-Net Dehazed Indoor Image:

MSCNN Dehazed Indoor Image:

Dehaze-Net Dehazed Outdoor Image:

AOD-Net Dehazed Outdoor Image:

MSCNN Dehazed Outdoor Image:

5.7 Synthetically Hazy Dataset

MSCNN NYU Depth Dataset:

DehazeNet NYU Depth Dataset:

5.8 Comparison

Surprisingly, PSNR and SSIM for naturally hazy images with MSCNN and DehazeNet were slightly higher. However, a qualitative comparison shows that there is still more haze remaining for the naturally hazy images.

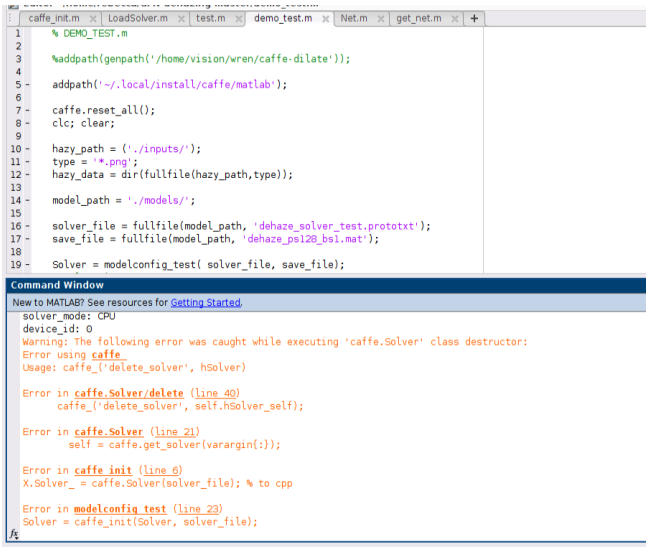
5.9 Future Analysis

Improvements that could be made to this project in the future includes testing on tif images, performing video dehazing and running the remaining models such as dehaze-cGAN, GFN, and DCPDN.Another improvement would be training on the naturally hazy dataset as well.

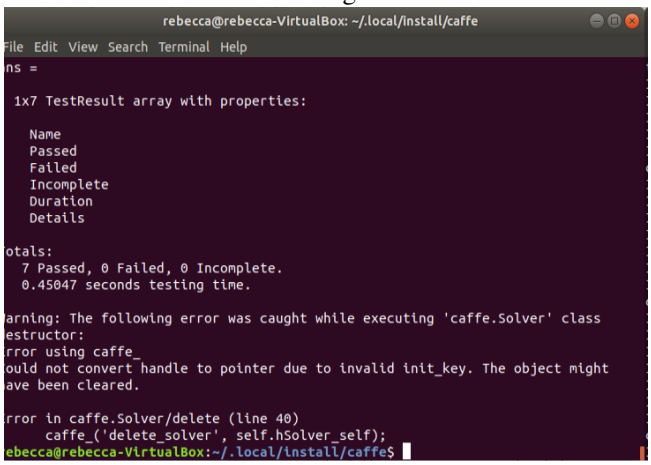
5.10 Comments

This project was interesting because of how dehazing techniques have transformed over the past decade, and how each technique built on the prior technique.

5.11 Appendix 1: Matcaffe Issues



Error seen running matcaffe test:



We saw this was resolved in a prior pull request here, but the error still persists [41].

References

[1] Cai, B., Xu, X., Jia, K., Qing, C., Tao, D. (2016). DehazeNet: An End-to-End System for Single Image Haze Removal. IEEE Transactions on Image Processing, 25, 5187-5198.

[2] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, Christophe De Vleeschouwer (2018). I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images. In arXiv:1804.05091v1.

[3] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, Christophe De Vleeschouwer (2018). O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images. In IEEE Conference on Computer Vision and Pattern Recognition, NTIRE Workshop .

[4] El Khoury, Jessica, Jean-Baptiste Thomas, and Alamin Mansouri. "A database with reference for image dehazing evaluation." Journal of Imaging Science and Technology 62.1 (2018): 10503-1.

[5] Guo, T., Li, X., Cherukuri, V., Monga, V. (2019). Dense scene information estimation network for dehazing. In Proceedings - 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW 2019 (pp. 2122-2130). [9025389] (IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops; Vol. 2019-June). IEEE Computer Society. https://doi.org/10.1109/CVPRW.2019.00265

[6] He, K., Sun, J., Tang, X. (2009). Single image haze removal using dark channel prior. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 1956-1963.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

[8] Isola, P., Zhu, J., Zhou, T., Efros, A.A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5967-5976.

[9] Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D. (2008). Deep photo: model-based photograph enhancement and viewing. ACM Trans. Graph., 27, 116.

[10] Lee, B., Lee, K., Oh, J., Kweon, I.S. (2020). CNN-Based Simultaneous Dehazing and Depth Estimation. 2020 IEEE International Conference on Robotics and Automation (ICRA), 9722-9728.

[11] Li, B., Peng, X., Wang, Z., Xu, J., Feng, D. (2017). AOD-Net: All-in-One Dehazing Network. 2017 IEEE International Conference on Computer Vision (ICCV), 4780-4788.

[12] Li, B., Peng, X., Wang, Z., Xu, J., Feng, D. (2018). End-to-End United Video Dehazing and Detection. AAAI.

[13] Li, C., Guo, J., Porikli, F., Fu, H., Pang, Y. (2018). A Cascaded Convolutional Neural Network for Single Image Dehazing. IEEE Access, 6, 24877-24887.

[14] Li, R., Pan, J., Li, Z., Tang, J. (2018). Single Image Dehazing via Conditional Generative Adversarial Network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8202-8211.

[15] Liu, Z., Xiao, B., Alrabeiah, M., Wang, K., Chen, J. (2018). Generic Model-Agnostic Convolutional Neural Network for Single Image Dehazing. ArXiv, abs/1810.02862.

[16] Ma, F., Karaman, S. (2018). Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. 2018 IEEE International Conference on Robotics and Automation (ICRA), 1-8.

[17] Ma, L., Liu, Y., Zhang, X., Ye, Y., Lin, G., Johnson, B. (2019). Deep learning in remote sensing applications: A meta-analysis and review. Isprs Journal of Photogrammetry and Remote Sensing, 152, 166-177.

[18] Meng, G., Wang, Y., Duan, J., Xiang, S., Pan, C. (2013). Efficient Image Dehazing with Boundary Constraint and Contextual Regularization. 2013 IEEE International Conference on Computer Vision, 617-624.

[19] Narasimhan, S., Nayar, S. (2008). Contrast restoration of weather degraded images. SIGGRAPH 2008.

[20] N. Silberman, P., Rob Fergus (2012). Indoor Segmentation and Support Inference from RGBD Images. In ECCV.

[21] J. P. Oakley, B. L. Satherley (1998). Improving image quality in poor visibility conditions using a physical model for contrast degradationIEEE Transactions on Image Processing, 7(2), 167-179.

[22] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), pages 234–241, 2015.

[23] Ren, W., Ma, L., Zhang, J., Pan, J., Cao, X., Liu, W., Yang, M. (2018). Gated Fusion Network for Single Image Dehazing. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3253-3261.

[24] Ren, W., Pan, J., Zhang, H., Cao, X., Yang, M. (2019). Single Image Dehazing via Multi-scale Convolutional Neural Networks with Holistic Edges. International Journal of Computer Vision, 128, 240-259.

[25] Tan, R. (2008). Visibility in bad weather from a single image. 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1-8.

[26] Tang, K., Yang, J., Wang, J. (2014). Investigating Haze-Relevant Features in a Learning Framework for Image Dehazing. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2995-3002.

[27] Tarel, J., Hautière, N. (2009). Fast visibility restoration from a single color or gray level image. 2009 IEEE 12th International Conference on Computer Vision, 2201-2208.

[28] Ulyanov, D., Vedaldi, A. Lempitsky, V. Deep Image Prior. Int J Comput Vis 128, 1867–1888 (2020).

[29] Yang, H., Pan, J., Yan, Q., Sun, W., Ren, J., Tai, Y. (2017). Image Dehazing using Bilinear Composition Loss Function. ArXiv, abs/1710.00279.

[30] Yue Feng (2020). A Survey on Video Dehazing Using Deep Learning Journal of Physics: Conference Series, 1487, 012018.

[31] Zhang, H., Patel, V. (2018). Densely Connected Pyramid Dehazing Network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3194-3203.

[32] Zhang, H., Patel, V. (2018). Density-Aware Single Image De-raining Using a Multi-stream Dense Network. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 695-704.

[33] Zhu, Q., Mai, J., Shao, L. (2015). A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. IEEE Transactions on Image Processing, 24, 3522-3533.

[34] Mnih, V., Heess, N., Graves, A., et al. 2014. Recurrent Models of Visual Attention. In Proc. NIPS, 2204–2212.

[35] Chen, J.; Wu, C.; Chen, H.; Cheng, P. Unsupervised Dark-Channel Attention-Guided CycleGAN for Single-Image Dehazing. Sensors 2020, 20, 6000.

[36] https://github.com/jmbuena/toolbox.badacost.public/blob/db650c29a3

[37] https://caffe.berkeleyvision.org/installation.htmlprerequisites

[38] https://www.youtube.com/watch?v=DnIs4DRjNL4

[39] https://github.com/rebeccaHassett/ImageDehazingModels

[40] https://towardsdatascience.com/preparing-tiff-images-for-image-translation-with-pix2pix-f56fa1e937cb

[41] https://github.com/BVLC/caffe/pull/5588

Alex Cuba contributed to this project by analyzing and detailing the information about the prior based methods, as well as their comparison, in this report. He also analyzed attention mechanisms, as well as adjusted and corrected the formatting the entire document.

Rebecca Hassett contributed to this project by finding references for all topics (prior-based methods, deep learning architectures, etc.) except for the attention mechanism resources and setting up the bibliography and latex format. I wrote the introduction and dehazing overview, deep learning dehazing, modern dehazing except for attention mechanism, and comparison sections. I found, ran and analyzed the code samples.