May 6th, 2023

CS445 Final Project Report - Dehazing Land and Underwater Images with Neural Network Classifier

Team members: Daniel Zhuang (dzhuang6), Han Chen (hanc7), Eric Jin (ericjin2)

Overview:

We created a single-image dezaher based on the approaches of "Single Image Haze Removal Using Dark Channel Prior" by He et al. (referred to as "paper 1" in this report) for land dehazing and "A Retinex-Based Enhancing Approach for Single Underwater Image" by Fu et al. ("paper 2") for underwater dehazing. We trained a neural network to classify if an input image is on land or underwater. Based on its classification, we would run the appropriate algorithm implementation to reduce the fog of the image. We also used methods described in "Blind Contrast Enhancement Assessment by Gradient Ratioing at Visible Edges" by Hautiér et al. ("paper 3") to evaluate the effectiveness of our dehazing method.

Motivation and impact:

We wanted to explore how to enhance images under different foggy conditions. Hazy land images and underwater images present different challenges due to different natures of the environments. Land haze could be due to air pollution and fog while underwater haze could be due to ocean particles or lack of sunlight. Hence, we wanted to create a single program to dehaze images of both types. Our program automatically detects the environment of the hazy image and chooses the optimal dehazing algorithm, which could be used in photo-editing programs to dehaze various images with a single tool. Another possible impact is to help pre-process hazy images for object-detection AI so that the image features are clearer, leading to better accuracy.

Approach:

Neural Network:

The resnet18 model was used to train the neural network. (We toyed around with our own architecture but it was not efficient). A batch size of 32 was used with 4 epochs and a cross entropy loss function, 686 hazy non-underwater images and 890 underwater images are used as the dataset. Of this data, 80% was used for training. The output of the classifier then determines which dehazing method to use. Link to data folder (access through Illinois account): https://drive.google.com/drive/folders/1eGwCdjslzm0a5SKiIQMhgKlYHT_65cIb?usp=share_link

Paper 1 Implementation:

We use the haze image model $\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1 - t(\mathbf{x}))$ referenced in paper 1, where $\mathbf{I}$ is the input intensity, t is the medium transmission, $\mathbf{A}$ is the atmospheric light, and $\mathbf{J}$ is the scene radiance. $\mathbf{I}$, t, and $\mathbf{A}$ are used to find the dehaze image $\mathbf{J}$. To estimate $\mathbf{A}$, we take the brightest (parameterized, with default threshold being 0.1%) pixels of the input image in what is known as the "dark channel prior". A pixel's dark channel is defined as the minimum intensity across channels in a given patch centered at that pixel. This more accurately reflects atmospheric light compared to methods that simply choose the brightest pixel.

To estimate t, we take normalize the dark channel prior of the image with atmospheric light and also multiply it by some constant in [0, 1] (parameterized to 0.95) to create a small amount of natural haze. Then we subtract 1 by this aforementioned term. This produces a rough transmission map that reflects the non-scattered portion of light that reaches the camera. It is rough because the dark channel prior groups the image into patches. This is later refined using soft matting.

Now, we rearrange the haze image model we can solve for **J** given **I** and the estimates of *t* and **A**. Additionally, we add a restriction of *t* to have some small lower bound (parameterized to be 0.01) since for *t* close to 0, **J**(**x**)*t*(**x**) is close to 0, leading to noise. Thus, some haze is preserved in high-haze portions of the image. However, ultimately, it greatly dehazes images, makes edges more clear, and colors more vibrant.

Paper 2 Implementation:

$\mu, \alpha, \beta, \gamma, \lambda$ are hyper-parameters. As the underwater tends to be more blue or green owing the light refraction in the water. The first part is color correction. Calculating the mean and standard deviation of each channel and using parameter mu decides the maximum and minimum value of each channel separately. $S^c_{max} = S^c_{mean} + \mu\sigma^c, S^c_{min} = S^c_{mean} - \mu\sigma^c$. Then, based on the maximum and minimum, we tune the original RGB value by $S^c_{CR} = \min(255, \max(0, \frac{S^c - S^c_{min}}{S^c_{max} - S^c_{min}}))$.

After the color correction, we decompose the original value $S = RI$, where S is the observed image, R is the reflectance, and I is illumination. $R \in [0,1]$, so, we have $I \geq S$. We initialize the I and I_0 as the Gaussian low-pass filtered image of L, R to be 0. L is the luminance layer of color corrected image.

Next, we are going to tune value R and I in sequence. Use matrix $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$ as the filters with R to calculate D_xR, D_yR. $shrink(x, \epsilon) = \frac{x}{|x|}\max(|x| - \epsilon, 0)$. Update derivatives $d_x = shrink\left(D_xR, \frac{1}{2\lambda}\right), d_y = shrink\left(D_yR, \frac{1}{2\lambda}\right)$. Given updated d_x, d_y *and* I, update R, I by

$$R = \mathcal{F}^{-1}\left(\frac{(1+\beta\lambda)\mathcal{F}(L/I)}{\mathcal{F}(1) + \beta\lambda\left(\mathcal{F}(d_x)^*\mathcal{F}(d_x) + \mathcal{F}(d_y)^*\mathcal{F}(d_y)\right)}\right),$$

$$I = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\gamma I_0 + L/R)}{\mathcal{F}(1+\gamma) + \alpha\left(\mathcal{F}(d_x)^*\mathcal{F}(d_x) + \mathcal{F}(d_y)^*\mathcal{F}(d_y)\right)}\right),$$

where $\mathcal{F}$ is Fast Fourier Transform operator, * is the complex conjugate. All calculations are component-wise operators. In order to clip $L \leq I, I = \max(I, L)$.

To address the problem of under-exposure, a slight improved histogram specification is worked on the illumination I. We Cumulative Density Functions (CDF) $C(z)$ of the pixel value as the formula in the paper. We define another CDF $Cf(z)$, a distribution as the paper explains. To lighten dark regions and preserve naturalness to avoid over-enhancement, $I_{enhanced} = Cf^{-1}(C(I))$.

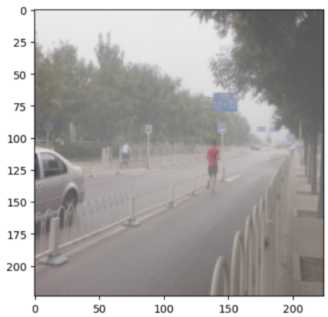
Finally, we get $L_{enhanced} = I_{enhanced}R_{enhanced}$. Then the new Lab color space is transformed into RGB to acquire the final enhanced color image. We use the same parameter as the write used, $\mu = 2.3, \alpha = 100, \beta = 0.1, \gamma = 1, \lambda = 10$.

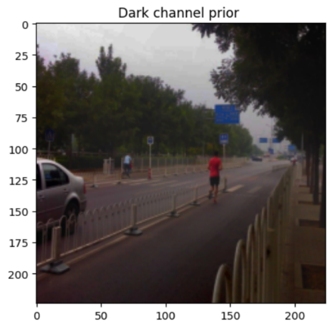
Evaluation Method:

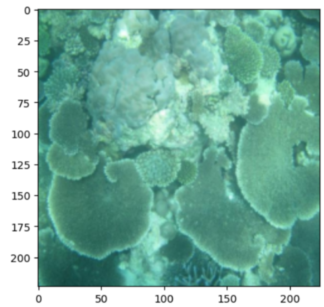
Ratio e: n_o and n_r denote respectively the cardinal numbers of the set of visible edges in the original image I_o and in the contrast-restored image I_r. According to paper 3, we propose to compute *e*, the rate of new visible edges in I_r: $e=(n_r-n_o)/\ n_o$ The value of *e* evaluates the ability of the method to restore edges which were not visible in I_o but are in I_r.

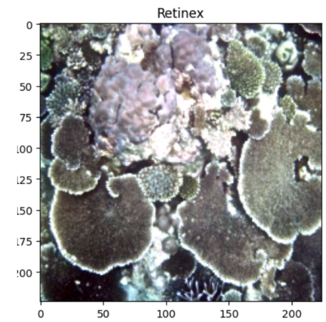
Ratio σ: we propose to compute the number n_s of pixels which are saturated (black or white) after applying the contrast restoration but were not before. We normalize this value by the size of the image, which gives the σ indicator: $\sigma = n_s / N$, where N denotes the number of pixels in one channel.

Results:



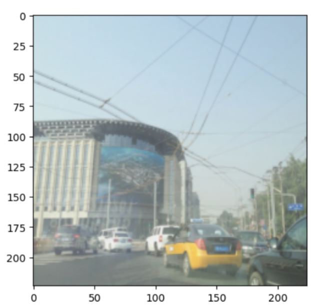


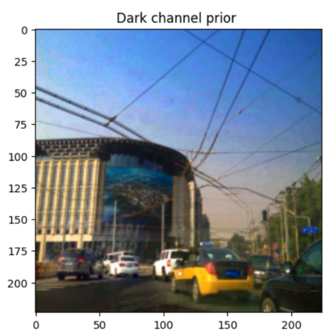


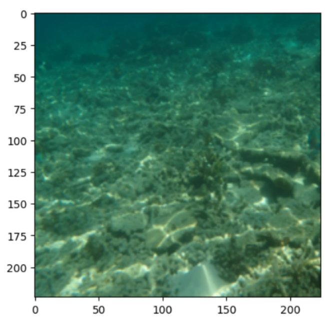


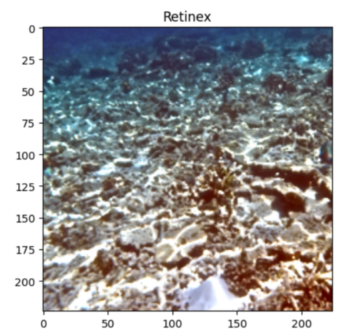
e = 0.15361574668378264, sigma = 0.005620216836734694 (left), e = 0.016346040363507612, sigma = 0.08111447704081633 (right)

See figures 1a, 1b, and 1c for the dark channel priors and transmission maps for the road image (left).
Figures 2a and 2b show that there were more edges in the dehazed image (left), meaning the image was clearer.









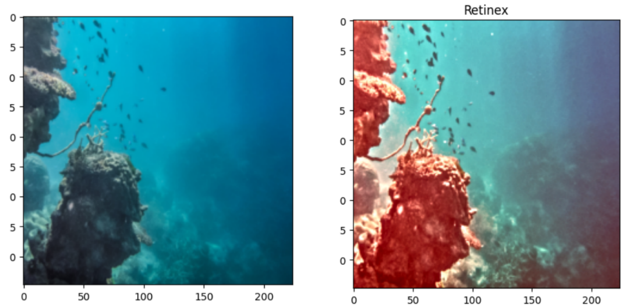
e = 0.58012471543106, Sigma = 0.002272002551020408 (left), e = 0.008338017612891137, Sigma = 0.06549426020408163 (right)

<u>Significance:</u>

These results show the original image on the left and the image on the right after dehazing. Using our classifier, we were able to consistently and correctly detect whether the image is underwater or not and use the retinex or dark channel prior method respectively. For land images, the image becomes more vibrant with edges more clearly shown. For underwater images, there is a clear color shift from the ocean green to an estimate of the objects' true colors. The edges also become clearer. Sigma tends to be low for most images we tested indicating less saturation. e tends to be somewhat high for some images indicating more edges visible but also low for some images, potentially due to the input image already having clear edges. Overall, our method is accurate in classification and does well to dehaze most images.

May 6th, 2023

Results Limitations:



e = 0.6820285871357889, sigma = 0.0093071238520408163

This result looks reddish owing to the limitation of the color correction part. The distribution of original image RGB values differs from image to image, so one fixed mu value is not enough to tune all images. After printing out the range, we find that the red channel is pretty saturated which means mu is small for this red channel. The better way to fix this is using a neural network to find the appropriate value for each single separately, but in our implementation we only find a parameter that works for most of them.

Accuracy of the classifier for detecting land versus underwater images:

```
Accuracy of the network on the test images: 99%
```

Implementation details:

- Utilized Python on Google Colab
- We used Pytorch to build the neural network classifier
- Pymatting library implementation of the Laplacian matrix is used for refining the paper 1 transmission map
- Images for the classifier are from: Reside dataset, kaggle, UIEB dataset, O-Haze dataset
- Other packages used: numpy, matplotlib, os, math, scipy, cv2, and sklearn

Challenge and Innovation:

The real innovation in this project is that we have a way of blending together multiple dehazing methods to choose which one is best for the dehazing task. We use a neural network to classify an image based on the type of hazing that is present in the image so that we can choose the optimal de-hazer. Right now it can only tell the difference between underwater dehazing and non-underwater dehazing, but this can be expanded to dehazing of sunlight haze, or any specific type of image noise. By doing this, we combine the two papers into one project so that they can both be used automatically when they are needed.

We believe that by seamlessly blending multiple methods of dehazing together, we earn the full 20 points for innovation. A lot of the effort for this innovation is that we need a large amount of training data that clearly shows the difference between the two types of dehazing cases. Specifically, there was a lot of data for underwater images, but not a whole lot of images for hazy non-underwater images. Multiple datasets had to be combined to get enough data for training purposes. Another takeaway from our findings is that we clearly needed to segment the data to something more than underwater/not underwater. The amount of haze, and the RGB value distribution require different parameters to dehaze each image. For example, some images are not very hazy and don't need larger parameter values that could distort the image. There were also cases where retinex worked better for non-underwater images and dark channel prior worked better for underwater images which proves that there is not a clear black and white distinction on which method goes with which scenario. If this project is to be continued, we would need a way to further classify the image into more classes and find ways to train the parameters differently (possibly with another neural net).

May 6th, 2023

Paper References:

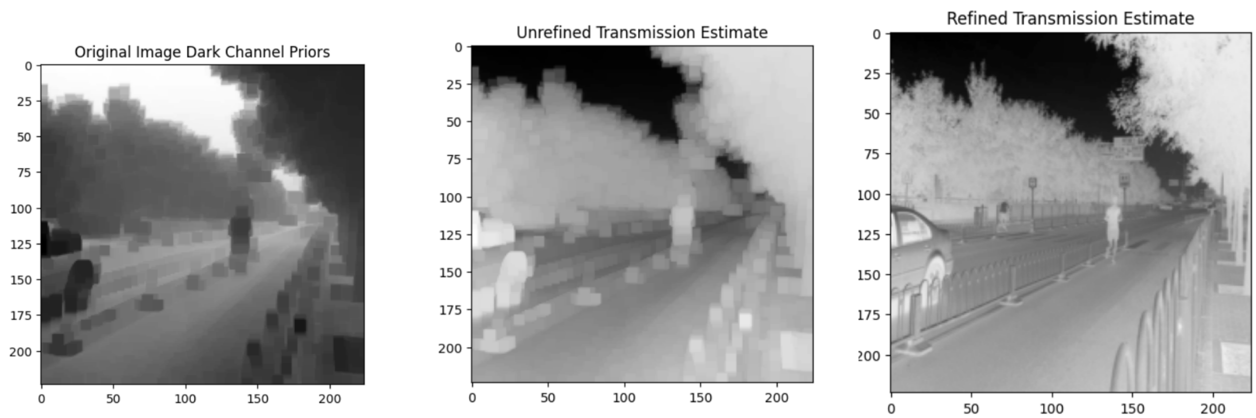
- "Single Image Haze Removal Using Dark Channel Prior" by He et al.: https://projectsweb.cs.washington.edu/research/insects/CVPR2009/award/hazeremv_drkchnl.pdf (We refer to this paper as "**paper 1**")
- "A Retinex-Based Enhancing Approach for Single Underwater Image" by Fu et al.: https://xmu-smartdsp.github.io/news/newspdf/A%20RETINEX-BASED%20ENHANCING%20APPROACH%20FOR%20SINGLE%20UNDERWATER%20IMAGE.pdf ("**paper 2**")
- "Blind Contrast Enhancement Assessment by Gradient Ratioing at Visible Edges" by Hautiér et al.: https://www.researchgate.net/publication/26518907_Blind_Contrast_Enhancement_Assessment_by_Gradient_Ratioing_at_Visible_Edges ("**paper 3**")

Image sources:

- https://li-chongyi.github.io/proj_benchmark.html
- https://sites.google.com/view/reside-dehaze-datasets/reside-standard?authuser=0
- https://www.kaggle.com/datasets/rajat95gupta/hazing-images-dataset-cvpr-2019/code
- https://www.kaggle.com/datasets/arnaud58/landscape-pictures
- https://data.vision.ee.ethz.ch/cvl/ntire18//o-haze/
- Input image of a train on tracks (used as the sample image for the paper 1 implementation) from paper 1

Figures:

1a, 1b, 1c (left, middle, right) Road - with Dark Channel Priors and Transmission Maps:



2a (left) - Edge detection of original image of road, 2b (right) Edge detection of new image of road dehazed: