PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

UE21EC342AC1 - DIP 1

Report on

"Dehazing of an Image using Dark Channel Prior"

Submitted by

Team number: 23

Srinivasa Puranika Bhatta (PES1UG21EC295)

Sumukh Shadakshari (PES1UG21EC302)

Shrihari Narayan P.B (PES1UG21EC275)

Aug - Dec 2023

COURSE INSTRUCTOR: DR. SHIKHA TRIPATHI

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROGRAM - B.TECH

Abstract:

The main aim of the project is to implement the Dark Channel Prior algorithm for haze removal. We have compared it with basic histogram equalization of the Y channel of input images in the YCbCr color space, and assessed if our method gives a better result
.

Introduction:

Under bad weather conditions, such as fog and haze, the quality of images degrades severely due to the influence of particles in the atmosphere. Suspended particles scatter light and result in attenuation of reflected light from the scene and the scattered atmospheric light mixes with the light received by the camera and changes the image contrast and color. The amount of scattering depends on the distances of the scene points from the camera. The local transmission which depends on the scene depth has to be estimated. To solve this, we use Dark Channel Prior which gives us an effective method to estimate the local transmissions for hazy images.

Theory and Algorithm:

Mathematical Model

The formation of a haze image can be represented by the equation:
$I(x) = J(x)t(x) + A(1 - t(x))$

where I is the observed hazy image, J is the scene radiance(undegraded image), A is the global atmospheric light, and t is the medium transmission describing the portion of the light that is not scattered and reaches the camera. The goal here is to recover J, A, and t from I.

The Dark Channel

The dark channel prior is based on the observation that in haze-free outdoor images, in most of the non-sky patches, at least one colour channel has very low intensity at some pixels. The dark channel of the image basically contains the minimum intensity value in a patch across all the channels. The low intensities in the dark channel are mainly due to:

- Dark objects, shadows of trees and rocks.
- Colourful objects having bright colors of green, red, yellow and blue.
- Objects with greyish color (i.e., having equal intensity across the three channels) are rarely found outdoors.

For an image J, we define the dark channel as:

$$J^{dark}(\mathbf{x}) = \min_{c \in \{r,g,b\}} \left(\min_{\mathbf{y} \in \Omega(\mathbf{x})} (J^c(\mathbf{y})) \right)$$

where J c is a colour channel of J and Ω(x) is a local patch centred at x.

Estimating the Atmospheric Light

The dark channel of a haze image approximates the haze denseness. We can use it to estimate the atmospheric light. We pick the top 0.1% brightest pixels in the dark channel. These pixels are the most haze opaque. Among these pixels the pixel with the highest intensity in the Image is selected as the atmospheric light. This method is better than picking the brightest pixel, since, in images, the brightest pixel could be on white objects.

Estimating the Transmission

We assume that the transmission in a local patch Ω(x) is constant and denote the patch's transmission by t~(x). Taking min operation in the local patch on the haze imaging equation,

$$\min_{\mathbf{y}\in\Omega(\mathbf{x})}(I^c(\mathbf{y})) = \tilde{t}(\mathbf{x}) \min_{\mathbf{y}\in\Omega(\mathbf{x})}(J^c(\mathbf{y})) + (1-\tilde{t}(\mathbf{x}))A^c$$

This reduces to:

$$\min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\frac{I^c(\mathbf{y})}{A^c})) = \tilde{t}(\mathbf{x}) \min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\frac{J^c(\mathbf{y})}{A^c})) + (1 - \tilde{t}(\mathbf{x})).$$

According to dark channel prior, the dark channel of the haze-free J tends to 0,

$$J^{dark}(\mathbf{x}) = \min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(J^c(\mathbf{y}))) = 0$$

Substituting J to be 0, we get

$$\tilde{t}(\mathbf{x}) = 1 - \min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\frac{I^c(\mathbf{y})}{A^c}))$$

Here, the second term on the right hand side can be calculated as the dark channel of (I/A). In general, the atmosphere has particles and there seems to be little haze when we look at far off objects. Presence of haze helps to perceive some depth. Hence, we keep a small amount of haze for distant objects using ω (0<ω<1) as a parameter.

$$\tilde{t}(\mathbf{x}) = 1 - \omega \min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\frac{I^c(\mathbf{y})}{A^c}))$$

Guided Filter

There are some block effects since the transmission is not always constant in a patch. We refine the transmission map using guided filter. Guided filter is a type of edge-preserving smoothing operator, which filters the input image (Estimated transmission map) under the guidance of another image (Original image).

Compared to the bilateral filter, the guided image filter has two advantages: bilateral filters have high computational complexity, while the guided image filter uses simpler calculations with linear computational complexity. Bilateral filters sometimes include unwanted gradient reversal artifacts and cause image distortion. The guided image filter is based on linear combination, making the output image consistent with the gradient direction of the guidance image, preventing gradient reversal.

The filtered output q, as a function of the input image p and guidance image I is calculated as:

$$a_k = \frac{\frac{1}{|w|}\sum_{i \in w_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} \quad \text{and} \quad b_k = \bar{p}_k - a_k \mu_k$$

Here, σ_k and μ_k are the variance and mean of I in window w_k, p_k is the mean of pixels in p and |w| is the number of pixels in w_k.
The final filtering output is given by:

$$q_i = \frac{1}{|w|} \sum_{k:i \in w_k} (a_k I_i + b_k) = \bar{a}_i I_i + \bar{b}_i$$

We assume that the relation between I and q is linear.

Depth Map

Using the transmission map, a depth map can also be constructed for the scene. When the atmosphere is homogenous, the transmission t can be expressed as:

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

where β is the scattering coefficient of the atmosphere. This equation indicates that the scene radiance is attenuated exponentially with the scene depth d.

Recovering the Scene

Radiance With the refined transmission map, we can recover the scene radiance using the haze equation. But, the direct attenuation term J(x)t(x) can be very close to zero when the transmission t(x) is close to zero for extremely thick haze. The directly recovered scene radiance J will then be prone to noise. Therefore, we restrict the transmission t(x) to a lower bound t_0, which means that a small amount of haze is preserved in very dense regions.
The final scene radiance J(x) is recovered by:

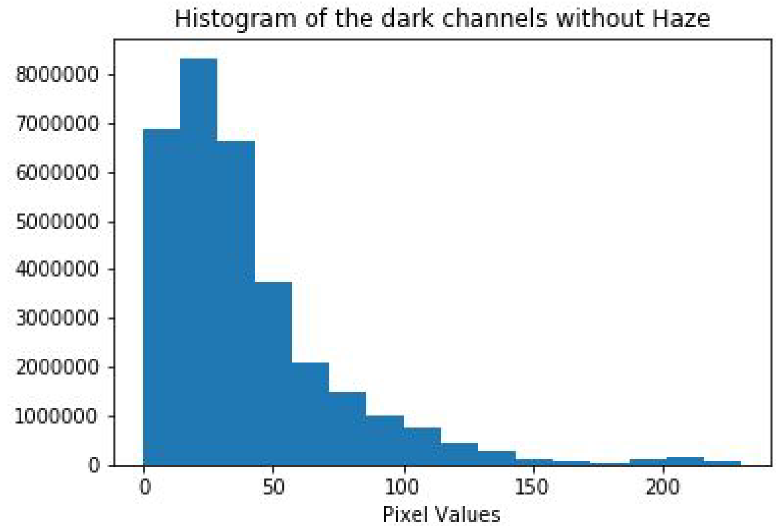
$$\mathbf{J}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) - \mathbf{A}}{\max(t(\mathbf{x}), t_0)} + \mathbf{A}$$

Colour Correction

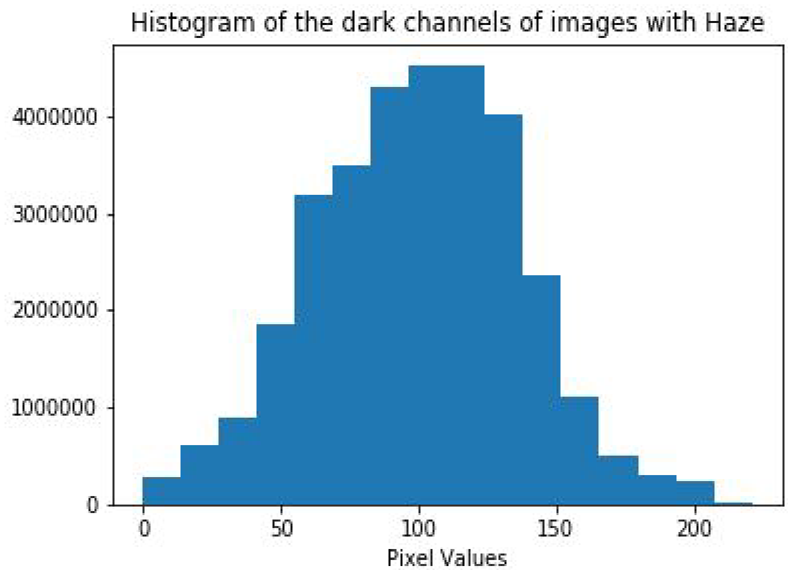
Since the scene radiance is usually not as bright as the atmospheric light, the image after haze removal looks dim. So, we increase the exposure of J(x) for display. Colour balance helps in colour enhancement and contrast enhancement. Colour balance stretches the values of the three channels R, G and B so that they occupy the maximal possible range [0,255] and provide us with some contrast. However, there might be some pixels already having a value of 0 or 255. In such cases, we clip the
pixels in such a way that a small percentage of the pixels closer to 0 are clipped to a higher value and pixels closer to 255 are clipped to a lower value. Then we stretch the channels to obtain the desired contrast. If there was coloured light with Red and Green dominating, the colour balancing would enhance the Blue channel making the ambient light lose the yellowish hue.

Verification of the Dark Channel

Since the basic idea of the haze removal technique is based on the dark channel observation, it is crucial to confirm the idea with a dataset of images. To verify the dark channel prior, we ran an experiment on the dataset of about 45 outdoor images, known as O-Haze. We built a histogram for the dark channels on the collection of images with and without haze.

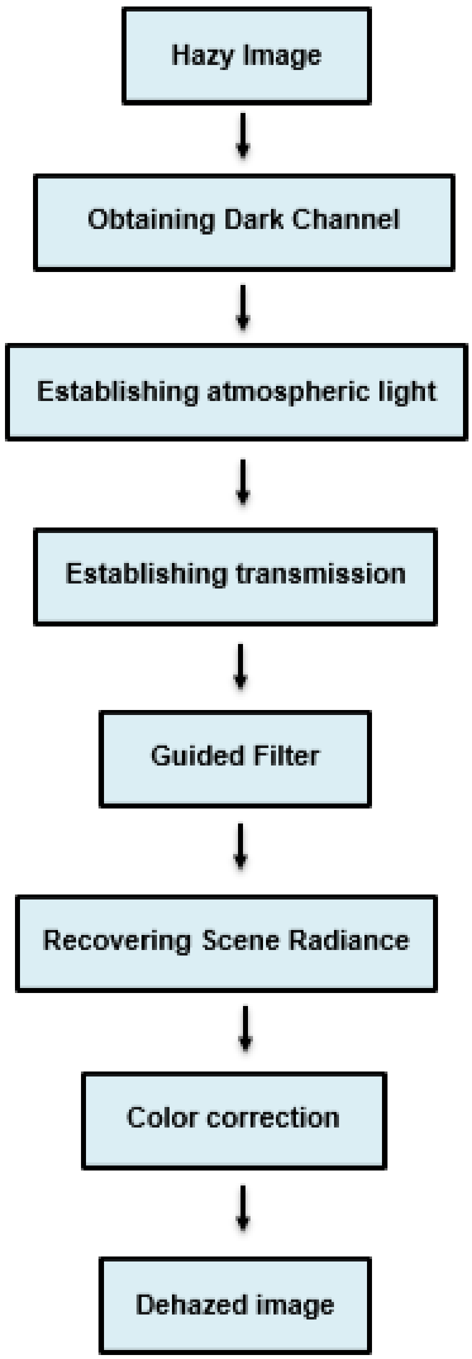


As observed from the histogram, it is evident that majority of the values in the dark channel are having low values. The values that are above 200 are mostly due to the presence of sky in the image.



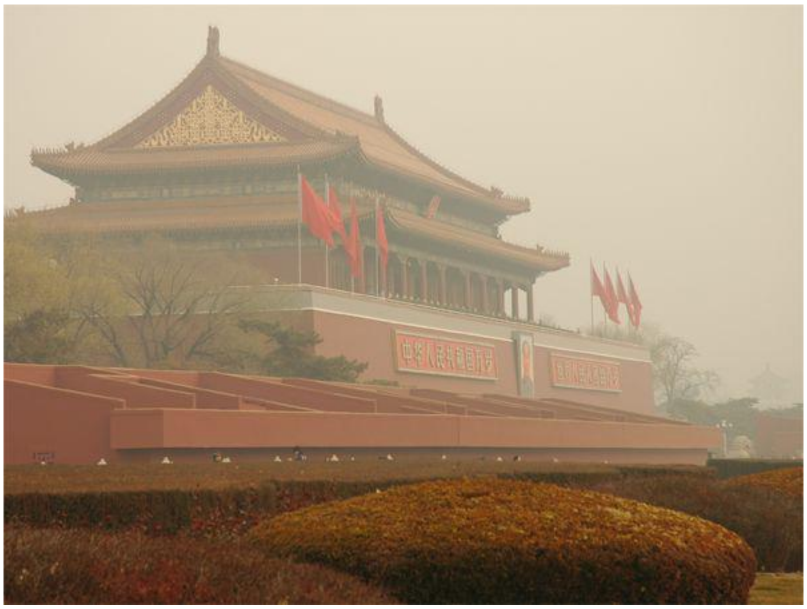
As seen from this histogram, the observation of dark channel values being low does not conform with the histogram in case of image affected by haze. This is because the dark channel of the haze image will have higher intensity in regions covered with dense haze. Hence, visually we could say that the intensity of the dark channel is a rough approximation of the thickness of the haze in that region.

Flowchart/ Diagram:



Results and Comparison

Original Hazy Image



Dark Channel

Atmospheric Light

The above image shows the bounding box around the region containing the top 0.1% of the brightest pixels in the dark channel, used to obtain the value of the atmospheric light.

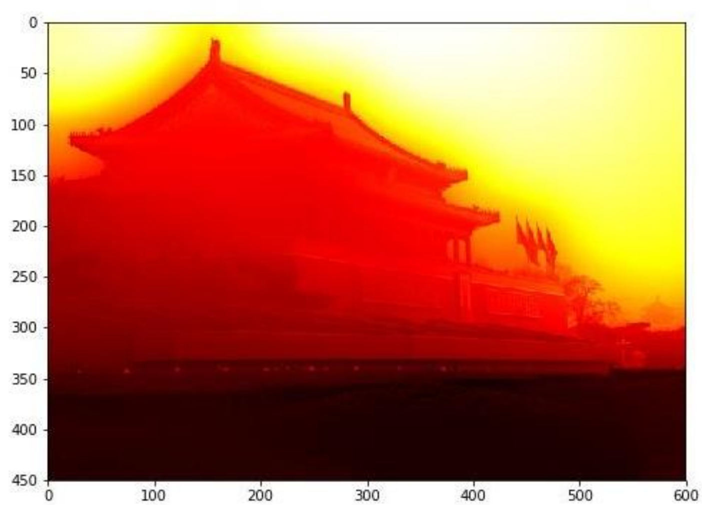
Estimated Transmission Map

Since the sky and the temple were the most hazy areas, their transmission (amount of light reaching the camera) must be less, which is confirmed by the transmission map above.

Transmission map after refinement with Guided Filter

Clearly, the transmission map has improved considerably by using the guided filter. The transmission before refinement was blocky and inaccurate. However, this transmission fits the shapes in the image perfectly.

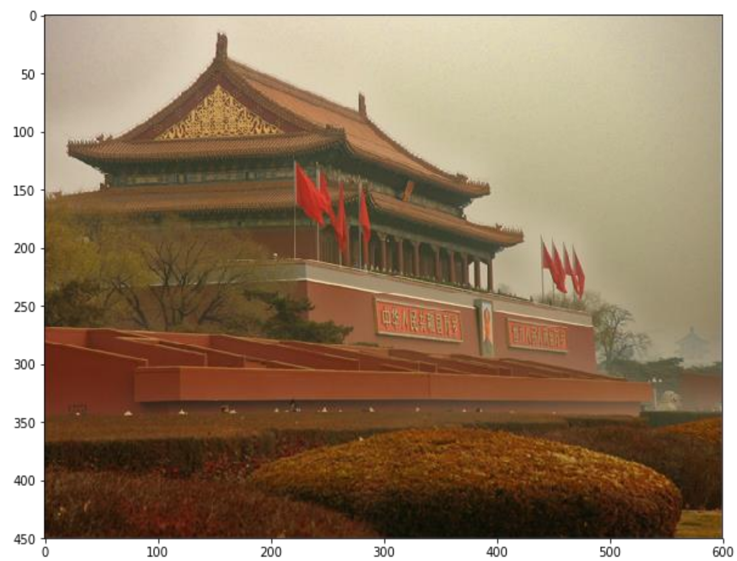
Depth Map



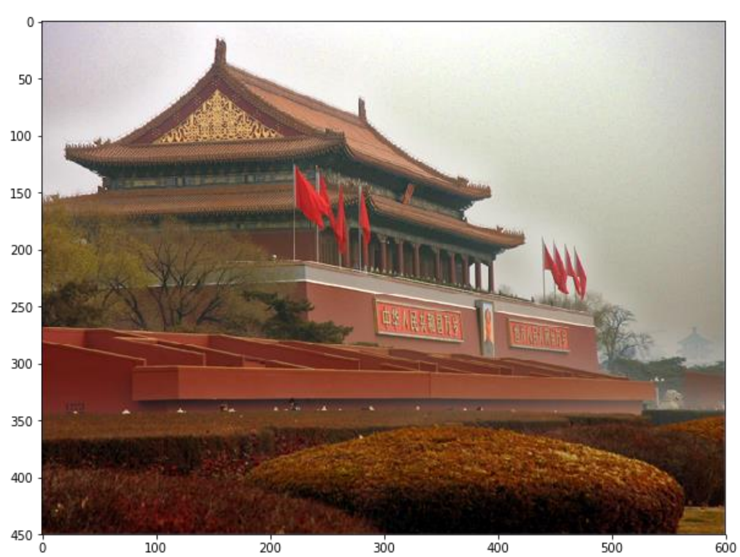
The depth map obtained is highly accurate. The sky appears the farthest, the temple in the middle and the grass closest to the camera, which is accurate.

This proves our point made previously, that the local transmission which depends on the scene depth

Restored Image



After Colour Correction



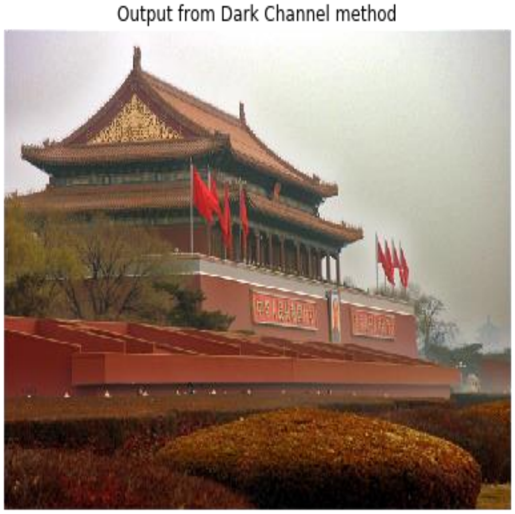
Comparison with Histogram Equalization

We compared the output from the Dark Channel method with basic histogram equalization performed on the Y channel of the images in the YCbCr colour space.

Output from Dark Channel method

Output from Histogram Equalisation

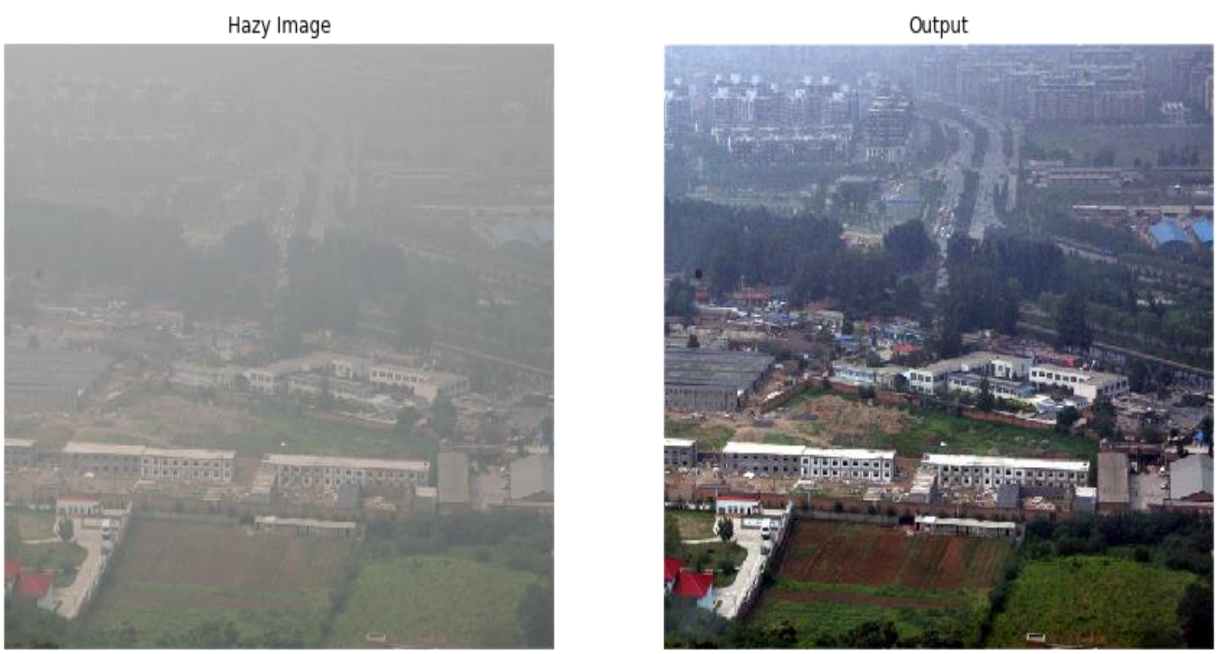
Output from Dark Channel method



Output from Histogram Equalisation

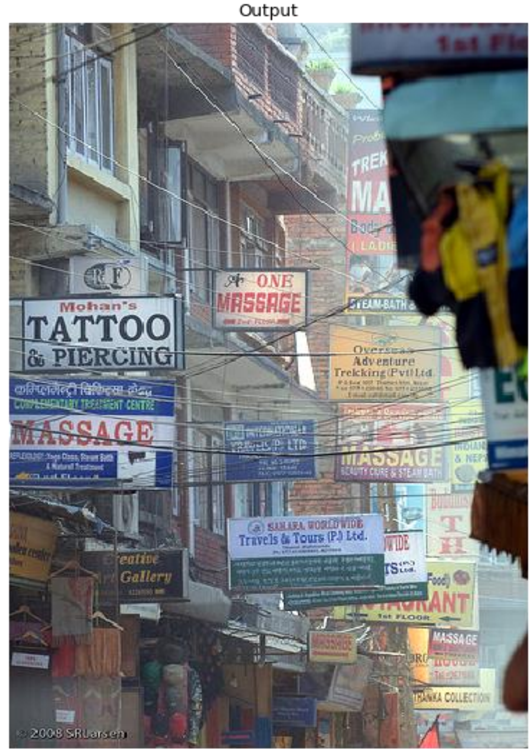
Clearly, histogram equalisation fails to remove haze from the images and instead, messes up the colours of the objects.

- Success results



Hazy Image

Output



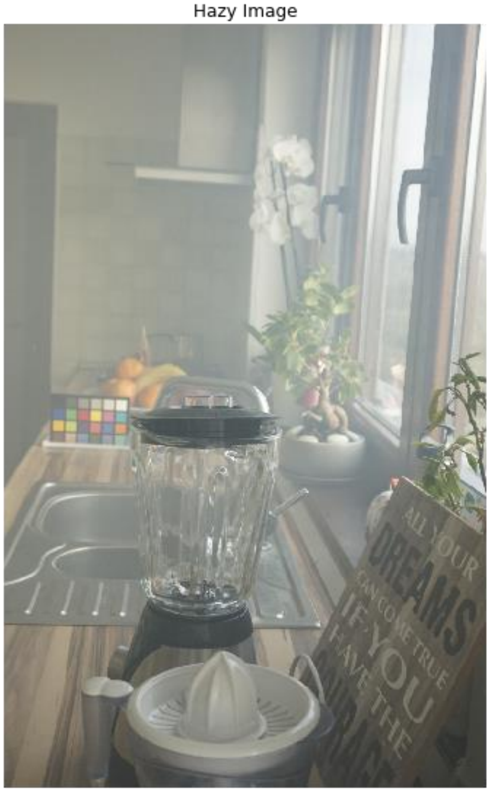
Hazy Image

Output

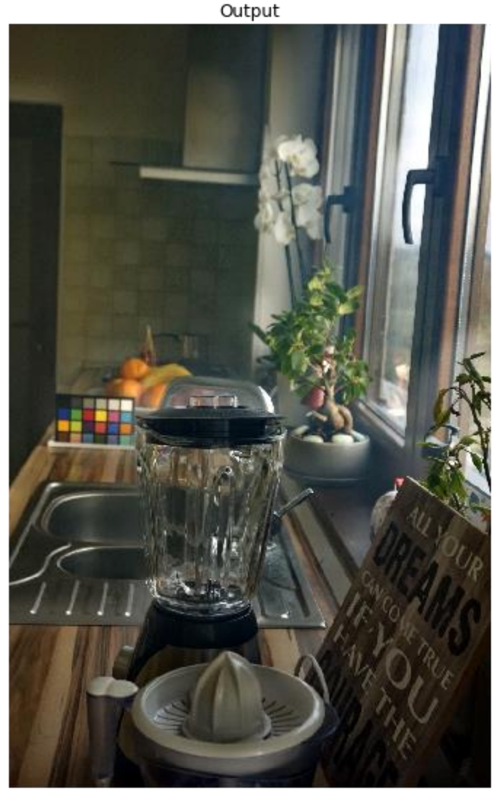
Hazy Image

Output

Hazy Image



Output



- ## Failure Cases

In this image, the 0.1% brightest pixels were found in the light and clothes, instead of the haze and smoke.

Here, since the haze is tinted yellow, its dark channel value is very low, which is why the atmospheric light is chosen from the clouds, and the sunlight induced haze is not removed.

Code:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import os

def get_dark_channel(img, wind_size):
    dark_channel = np.zeros((img.shape[0], img.shape[1]))
    img =
cv2.copyMakeBorder(img,wind_size//2,wind_size//2,wind_size//2,wind_size//2,cv2.B
ORDER_CONSTANT,value=[255,255,255])
    no_rows = img.shape[0]
    no_cols = img.shape[1]
    min_channel = np.zeros((no_rows, no_cols))
    for row in range(no_rows):
        for col in range(no_cols):
            min_channel[row-wind_size//2][col-wind_size//2] =
np.min(img[row,col,:])
    for row in range(wind_size//2, no_rows-wind_size//2):
        for col in range(wind_size//2, no_cols-wind_size//2):
            dark_channel[row-wind_size//2][col-wind_size//2] =
np.min(min_channel[row-wind_size//2:row+wind_size//2,col-
wind_size//2:col+wind_size//2])
    return dark_channel

def get_atm_light(im, dark_channel_img):
    img = im.copy()
    num_pixels = dark_channel_img.size
    num_brightest = num_pixels//1000
    haze_density_sort_idx = np.argsort(dark_channel_img,axis=None)[::-1]
    brightest = haze_density_sort_idx[0:num_brightest]
    brightest = np.unravel_index(brightest,dark_channel_img.shape)
    brightest_pixels = img[brightest]
    top_intensities = np.average(brightest_pixels, axis=1)
    max_intensity = np.argmax(top_intensities)
    A = brightest_pixels[max_intensity]
    img[brightest]=[255,0,0]
    row_min = np.min(brightest[0])
    row_max = np.max(brightest[0])
    col_min = np.min(brightest[1])
    col_max = np.max(brightest[1])
    cv2.rectangle(img,
(col_min,row_min),(col_max,row_max),(0,0,255),thickness=2)
    return A
```

```python
def refine(img, transmission, radius, epsilon):
    mean_guide = cv2.blur(img,(radius,radius))
    mean_trans = cv2.blur(transmission,(radius,radius))
    mean_gt = cv2.blur(img * transmission, (radius,radius))

    a = mean_gt - mean_guide * mean_trans
    var_guide = cv2.blur(img * img,(radius,radius)) - (mean_guide * mean_guide)
    a = a/(var_guide + epsilon)
    b = mean_trans - a * mean_guide

    q = cv2.blur(a,(radius,radius)) * img + cv2.blur(b,(radius,radius))
    return q

def recover_image(img, trans_bar, atm_light, t0):
    trans_recover = np.copy(trans_bar)
    trans_recover[trans_recover < t0] = t0
    J = np.zeros((img.shape))
    J[:,:,0] = ((img[:,:,0] - atm_light[0])/trans_recover) + atm_light[0]
    J[:,:,1] = ((img[:,:,1] - atm_light[1])/trans_recover) + atm_light[1]
    J[:,:,2] = ((img[:,:,2] - atm_light[2])/trans_recover) + atm_light[2]
    return J

def color_balance(img, s):
    out = np.copy(img)
    hist = np.zeros((256,1))
    no_of_pixels = img.shape[0] * img.shape[1]
    for i in range(3):
        channel_vals = img[:,:,i]
        for pixel_val in range(256):
            hist[pixel_val] = np.sum((channel_vals == pixel_val))
        for pixel_val in range(256):
            hist[pixel_val] = hist[pixel_val-1] + hist[pixel_val]
        Vmin = 0
        while (Vmin < 255 and hist[Vmin] <= no_of_pixels*s):
            Vmin += 1
        Vmax = 255
        while (Vmax > 0 and hist[Vmax] > no_of_pixels*(1-s)):
            Vmax -= 1
        channel_vals[channel_vals < Vmin] = Vmin
        channel_vals[channel_vals > Vmax] = Vmax
        out[:,:,i] = c
   return out
```

```
def run(img, omega, t0, radius, dark_rad):
    dark_channel_img = get_dark_channel(img, dark_rad)
    dark_channel_img = dark_channel_img.astype('uint8')
    atm_light = get_atm_light(img,dark_channel_img)
    t_bar = get_dark_channel(img/atm_light,dark_rad)
    trans_bar = 1-(omega * t_bar)
    i=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)/255
    t_refine = refine(i, trans_bar, radius, 0.0001)
    im = img.astype("double")
    J = recover_image(im, t_refine, atm_light, t0)
    J = ((J-np.min(J))/(np.max(J)-np.min(J)))*255
    cb_J = color_balance(np.uint8(J),0.005)
    return cb_J


clear_img = run(im, 0.85, 0.1, 30, 15)
cv2.imwrite(output_path, clear_img)
```

Observations and Conclusions:

The algorithms seem to perform really well on most outdoor images. It removes most of the haze from the images, only leaving out some when the haze is extremely thick.

Since the dark channel prior assumption only works when haze is the brightest object in the dark channel, the algorithm fails if there is a prominent white object/light in the picture that is brighter than the haze.

It also fails to perform when the haze is tinted, for example, from a beam of sunlight. This is because the colour makes the haze part black in the dark channel prior.

References:

[1] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior"

[2] J. Pang, O. C. Au and Z. Guo, "Improved Single Image Dehazing Using Guided Filter"

[3] C. O. Ancuti, C. Ancuti, R. Timofte and C. D. Vleeschouwer, "O-HAZE: a dehazing benchmark with real hazy and haze-free outdoor images"

[4] C. O. Ancuti, C. Ancuti, R. Timofte and C. D. Vleeschouwer, "I-HAZE: a dehazing benchmark with real hazy and haze-free indoor images"

[5] N. Limare, J. L. Lisani, J. M. Morel, A. B. Petro, C. Sbert, "Simplest Color Balance"