Color histogram equalization with weighted distribution applied to hazy images

Alejandro Rico Espinosa
Electrical and Computer Engineer
University of Victoria
Victoria, BC, Canada
V00972959

***Abstract*—Image enhancement has been an important task to achieved in computer vision, since it might improve performance of other algorithms such as object recognition. It had been implemented in multiple areas trying to recover diverse factors. This paper mix tow of them by applying a fuzzy color histogram equalization with weighted distribution [1] into dehazing reconstructed images which presents non-homogeneous blurriness. This was perform by constructing the fuzzy dissimilarity histogram from neighborhood pixels intensities to improve contrast. Then, gamma correction to enhance dark regions was apply and the maximum saturation was modified to avoid fading effect. Finally, metrics of PSNR and SSIM indexes were computed and compared with the indexes of previous images to determine if the algorithm helped the reconstructed dehazing task results to be similar to the groundtruth images. Additionally, a qualitative evaluation is performed and discussed inside this work. Overall, method do not seems to help improving scores in this specific scenario, but enhance image details trhought sharpening.**

***Index Terms*—histogram, color enhancement, hazy images, equalization, sharpness**

I. Introduction

Haze is a natural phenomena that affects image quality by reducing their visibility. In the last decade recovering visual information of this type had plenty of interest since it is important in applications such as aerial or ground surveillance, automatic driving and automatic traffic control [2]. One of the main problems with dehazing techniques is that there is not any standardize metric of evaluation. Also, the lack of datasets that contains real images and the few amount of samples had been one of the main issues while training models or testing algorithms. This, due to the difficulty on recording images with and without haze in the same lighting conditions.

Actual methods using complex structures of Convolutional Neural Networks (CNN) could achieved high recover of hazy images. However, some problems as blurriness and color variations are still found in the results of this task. Therefore, it is important to address these issues by combining other enhancement techniques previously proposed.

Color enhancement for example, is ideal to improve image quality for applications such as medical research, and underwater image analysis. During image acquisition, poor lighting and adverse environmental conditions might disturb the output result [1]. Moreover, for some applications of computer vision such as skin segmentation, contrast and brightness are an important factor to consider [8].

Approaches such as histogram equalization and gamma correction had been suggested to enhance low light images. However, some of these techniques have limitations due to abrupt changes in hue, maximum saturation shifting and luminance contrast. [1]

This work intents to applied a fuzzy dissimilarity histogram algorithm, with gamma correction, and saturation maximization into the reconstructed dehazed images to increase their quality aspect in color and get them to look more similar to the groundtruths. We expect this to works as a post-processing of the images, to make the dehazing results more accurate for future applications like the autonomous driving system.

The main contributions of this paper are:

1) We implemented the code in python of the original algorithm. Please refer to the GitHub repository for details.
2) We validate the effectiveness of the algorithm in terms of PSNR and SSIM on non-homogeneous reconstructed dehazed images.

II. Literary Review

Model that describes a hazing image can be formulated as presented in equation 1, where I is the intensity, J is the scene radiance and A is the global atmospheric light and t is the medium transmission.

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

The global goal of haze removal is to recover A, J and t from I. Classical dehazing approaches formulates mathematical models based on priors. The well know *Dark Channel Prior* approach [3] argues that in a patch of a haze-free image at least one channel (of the RGB) has some pixels whose intensity are very low and close to zero. Then, it is possible to build a dark channel where it´s intensity is a rough approximation of thickness of haze. They used the brighter pixels as reference to build this channel, since the color of sky for example is usually very similar to the atmospheric light. Therefore, it is possible to estimate the atmospheric light through the pixels with higher intensity of the image. This work used patches in

University of Victoria - Computer Vision ECE536.
https://github.com/alejorico98/fuzzy-color-histogram-equalization

the image, and demonstrate that bigger sizes of patches lead to higher performances in the haze removal task.

However, in the non-homogeneous haze images is hard to estimate atmospheric light with these classical approaches, since haze density is not strongly correlated with the image depth [4]. Therefore, many of the modern methods that try to plainly recover a hazed image, use convolutional neural networks instead. In the last year a lot of variations had been formulated and presented. Some of them use Res2Net architectures combined with transfer learning to boost up the response, others use generative adversarial networks to reconstruct the image, and some others try to set a stage of prepossessing techniques like channel splitting and dark channel prior approach [2].

Nevertheless, non of these approaches suggested a post-processing algorithm to enhance any of the aspects that the convolutional networks did not completely solve. After image reconstruction with high score models, we still noticed color degradation and blurriness problems.

Color enhancement had been also widely studied and discussed in computer vision since it is essential for better visual perception [1]. As an example, histogram equalization is well known for it´s simplicity and efficiency. Unfortunately, some of the proposed methods to enhance contrast failed to improve image quality when images are exposed to poor illumination. Several histogram equalization approaches had been formulated: ESIHE, MMSICHE, GEM, CLAHE. Despite the big amount of alternatives found for color restoration, currently approaches still might fails to improve contrast of poorly illuminated color images or might create unnatural look [1].

A weighted fuzzy color histogram equalization had been proposed to preserve the brightness and to enrich the contrast in the input images. Weighted distributions of gamma corrections (AGCWD) had been shown to achieve better results compared with the histogram equalization. [9] Due to the nature of this algorithm, we think that it is worth to apply it to the non-homogeneous haze removal task. Since this work intend to formulated the approach as a post-processing step after using a CNN, we want to avoid at maximum the introduction of complexity into the model. Hence, we would like to avoid more networks architectures and backpropagation's techniques. For this reason, this work proposes to use a novel fuzzy color histogram equalization to balance colors in the reconstructed images.

III. Proposed Approach

A general overview of the implemented approach is presented on this section. This algorithm was mainly formulated by other authors [1], but a couple of sightly modifications where made to adapt the proposed work in our images. Also, one additional step (values normalization between 0 and 255) that was not specified in the original paper was added in order to successfully complete the color equalization enhancement. The methodology will be described in detail.

Let the input image we are working with has 3 channels in the color spaces RGB (Red, Green and Blue) with intensity values (I) between 0 and 255 in each channel. Here (x,y) represents the pixel coordinate in the matrix.

$$I(x, y) = \{R(x, y), G(x, y), B(x, y)\} \tag{2}$$

Fist, we perform an stretching of the color channels to enrich the color details of the image. This is basically a type of normalization per channel based on each maximum (R_{max}) and minimum (R_{min}) intensity. Equation 3 defines the exact procedure for the case of the red channel that returns values of intensity between 0 and 1. Note that since we are working with the minimum and maximum value, non of the intensities can be negative, and due to the fraction either none of them could be greater than 1.

$$R_new(x, y) = \frac{R(x, y) - R_{min}}{R_{max} - R_{min}} \tag{3}$$

After this, we transform the RGB stretched image to the HSI (Hue, Saturation, Intensity) color space. Equations 4, 5, and 6 detailed this process. In similar way, the inverse procedure can be done to transform color space from HSI to RGB. During the coding, we use an already implemented Python script for this part [5]. Note that the algorithm intends to preserve the original color of the image, so hue is assumed to be correct and it will not be modified. Our intensity matrix will have values between 0 and 1 since our input is already normalized.

$$H = \begin{cases} \cos^{-1} \frac{\frac{1}{2}*[(R-G)+(R-B)]}{(R-G)^2+(R-B)*(G-B)^{\frac{1}{2}}} & \mathit{if}\ B \leq G \\ 360 - \cos^{-1} \frac{\frac{1}{2}*[(R-G)+(R-B)]}{(R-G)^2+(R-B)*(G-B)^{\frac{1}{2}}} & \mathit{if}\ B > G \end{cases} \tag{4}$$

$$S = 1 - \frac{3}{R + G + B} * [min(R, G, B)] \tag{5}$$

$$I = \frac{R + G + B}{3} \tag{6}$$

Then, we will like to compute the fuzzy dissimilarity histogram. We will assume each pixel coordinate of the intensity matrix I(x,y) with size MxN acn be represented by: x that is in the range [0,M-1] and y that is in the range [0,N-1].

We start by computing a membership function to map each intensity value to respect with their neighborhoods in a grid of 3x3. This function can be define by equation 7. Original paper define approach as a max function to always get value as positive or add zero in the other case. Instead of this, we use the absolute value of the number. This change was done since in the next step we should calculate the mean value of the memberships intensities relation. If we add zero instead of the absolute value, we would get lower mean. Since we are applying the algorithm to images previously reconstructed with GAN´s (generative adversarial network), we could have no correlation at all between neighborhood pixels intensities, so the original algorithm would reduce values in this area. To avoid intensity patches and possibly enhance some edges in these regions, we just set the absolute value instead of the max function.

$$\mu(u,v) = abs(1 - I(x,y) - I(u,v))/SD \quad (7)$$

Where u=x+i, v=y+j for i and j belonging to $[-1, 0, 1]$. SD denotes the standard deviation of the original image. Then we define $P_{mf}(x,y)$ as the mean value of μ in the 3x3 grid neighborhood. The final membership function is no more than the complement of $P_{mf}(x,y)$. Since values are between 0 and 1, and the complement is not specified in the original paper, we set equation 8 to represent the final formulation.

$$\mu_d(x,y) = (1 - P_{mf}(x,y)) \quad (8)$$

Then, we proceed to compute the histogram of this correlation function with a pixel count of possible intensities in the range [0,255], just as a normal histogram construction. For each possible intensity in this range, we sum up all the $\mu_d(x,y)$ for which I(x,y) is equal to the intensity we are evaluating. Equation 9 describes this step properly.

Recall, that original work do not specified any details for this part, however we should define a transformation of intensities to get the values between 0 and 255 instead to the ones that we have which are between [0,1] before computing the histogram. This transformation can be done straightforward by multiplying the value for the maximum intensity (i.e. 255) and then round it to the closest integer. We use a mathematical approach before rounding the value in Python to have 4 decimals of floating point accuracy and perform better the transformation. Details are presented in equation 10.

$$h_{fd}(p_i) = \sum\sum \mu_d(x,y) \quad \textit{for I(x,y)=}p_i \quad (9)$$

$$r(x,y) = int(\frac{int(I(x,y) * 1000)}{1000}) * 255 \quad (10)$$

p_i is each possible intensities in the range [0,255], $\mu_d(x,y)$ is the value of the final membership function and I(x,y) is the intensity of the original image matrix formulated at the beginning of this section. Moreover, r(x,y) is the result intensity transformation for coordinate x,j of the original matrix and int is the Python function to transform float into integer. Since $\mu_d(x,y)$ has value in the ranges [0,1], the maximum possible value we can get in the result histogram $h(p_i)$ will be the scalar size of the image M*N. This histogram interprets for the contextual data presented in the neighborhood of a pixel, providing the average dissimilarity measure in each intensity level of the input image [1].

After this, we apply a gamma correction for enhance the contrast and improve dark areas of the image. Original work proposed a histogram clipping before performing the correction, but we did not implement this step due to the type of images we have as input. Haze reconstructed images might still contain non-homogeneous haze areas which are not strongly correlated with deepness. Therefore, clipping the histogram could become into an unnatural looks of the image.

For the gamma correction we first compute the probability density function (PDF) and its corresponding cumulative density function. Equations 11 and 12 formulate these processes for i equal to each possible intensity in the range [0,255].

$$pdf(i) = \frac{h_{fd}(p_i)}{\sum_{i=0}^{255} h_{fd}(p_i)} \quad (11)$$

$$cdf(i) = \sum_{j=0}^{i} pdf(j) \quad (12)$$

Original work do not use these functions directly to perform the gamma correction, but aboard a weighed histogram distribution and the cumulative weighted distribution function as well, which can be define with equations 13 and 14 respectively.

$$pdf_w(i) = pdf_{max} * (\frac{pdf(i) - pdf_{min}}{pdf_{max} - pdf_{min}})^{\alpha} \quad (13)$$

$$cdf_w(i) = \sum \frac{pdf_w(j)}{\sum pdf_w(j)} \quad (14)$$

Where, pdf_{max} and pdf_{min} represents the maximum and minimum value of the already defined pdf function and $\alpha = cdf(i)$. Finally, we compute γ of each intensity using equation 15. With this we can set correction of each intensity in our image as defined by equation 16.

$$\gamma(i) = 1 - cdf_w(i) \quad (15)$$

$$I_e(x,y) = i * (\frac{i}{255})^{\gamma(i)} \quad (16)$$

After performing the gamma correction into the image intensity we should try to improve the saturation of the image. Original work demonstrates that de-saturation is fundamental part of the algorithm. For this, is ideal to set the maximum saturation between the original image and the image with intensity values enhanced. Therefore we applied the following procedure:

1) Save original saturation matrix of the image that we have in the HSI space into a new variable
2) Replace the intensity matrix of the HSI image with the corrected intensity matrix that we build through equation 16
3) Transform from HSI to RGB
4) Transform the result back from RGB to HSI
5) Compare each value in the (x,y) coordinates of original saturation matrix, with the resulting matrix of transformation done in the previous step. Take the maximum between these two values and build the final S_{max} that represents the maximum saturation matrix.
6) Set the final corrected image as the original Hue matrix that we assume is correct from the begginning, the

Maximum saturation matrix and the corrected intensity matrix.
7) If it is desire, transform your final result back into the RGB space

If we perform correctly all these steps, the enhancement algorithm must be complete.

IV. Evaluation

A. Dataset and Mterics

Due to the nature of the paper, no specific dataset is need for training since we intend to apply an algorithm. For testing we used the 2021 hazed dataset found in the original NTIRE challenge [7]. We previously trained a CNN architecture to dehaze these images, and use the reconstructed images as input to the algorithm (to recreate the post-processing step mentioned at the beginning of this document). To accurately compare the results and performance, we only apply this approach into the last 5 samples of the original dataset. **Full results, dataset, code, and figures can be find in the GitHub repository linked to this paper**.

As metric of evaluation we calculate the SSIM (Structural Similarity) and the PSNR (peak signal to noise ratio) of the testing dataset (the last 5 samples of the original dataset).

B. One Example

First we will present some results in each step when we apply the described approach into an example image. Figure 1 shows the input image to the algorithm and figure 2 displays the original histogram distribution. Recall that for histogram plotting purposes an already implemented code was used [6].

Fig. 1. Input Example

Note that input image includes blurry areas and some edges are not completely defined. Additionally, the histogram shows great amount of pixels in the high intensity areas in each of the color channels. This let us interpret that colors are not balanced and that image tends to be brighter.

Now, figure 3 shows the fuzzy histogram and the following figures 4, 5, 6, 7 presents the probability distribution function,

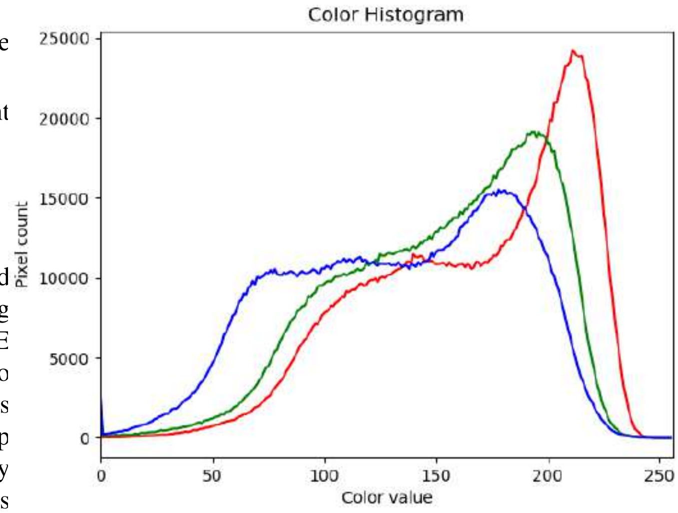

Fig. 2. Histogram of original image

the cumulative distribution function, the weighted probability distribution function and the weighted cumulative distribution function respectively.

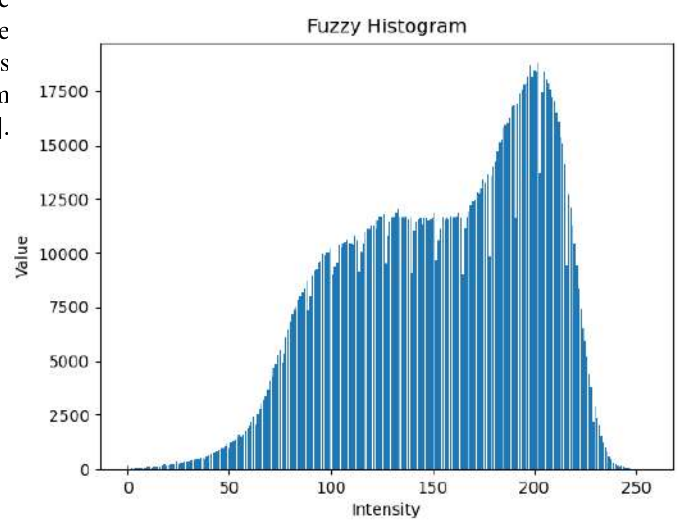

Fig. 3. Fuzzy Histogram

From these figures we appreciate the evolution of histogram in each step through the algorithm. First, we can notice that the fuzzy histogram shape distribution is similar to the original histogram. Recall that the fuzzy histogram provides the dissimilarity measure in each intensity level of the 3x3 grid neighborhoods. A result similar to the one presented in figure 3 indicate us that in average, there is a big difference between the high intensity pixels. Therefore, we might state again that the image could have an excess of brightness and that this phenomenal is usually non-uniformly around the neighborhood pixels.

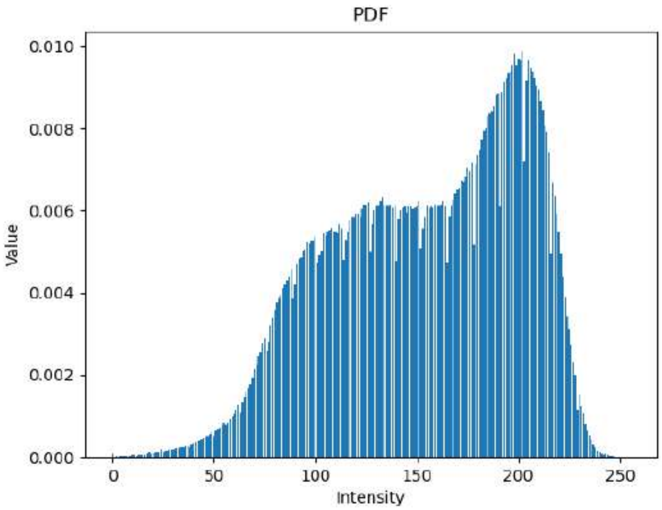

Fig. 4. Probability density function

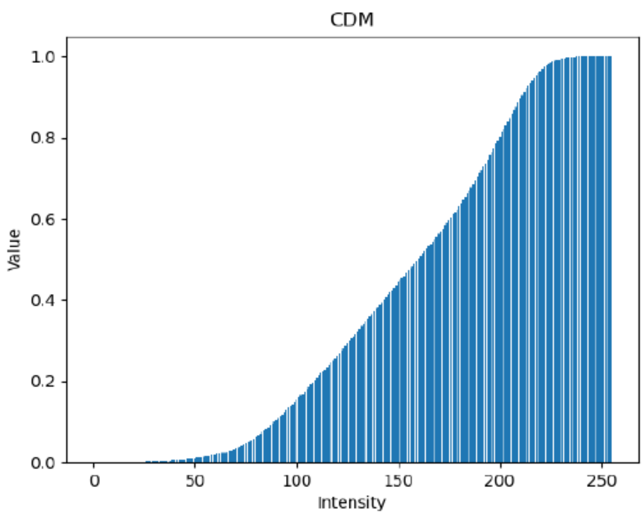

Fig. 5. Cumulative probability density function

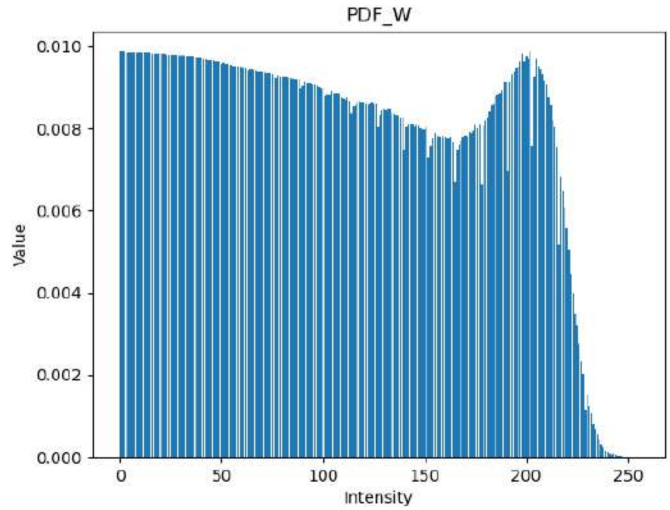

Fig. 6. Weighted probability density function

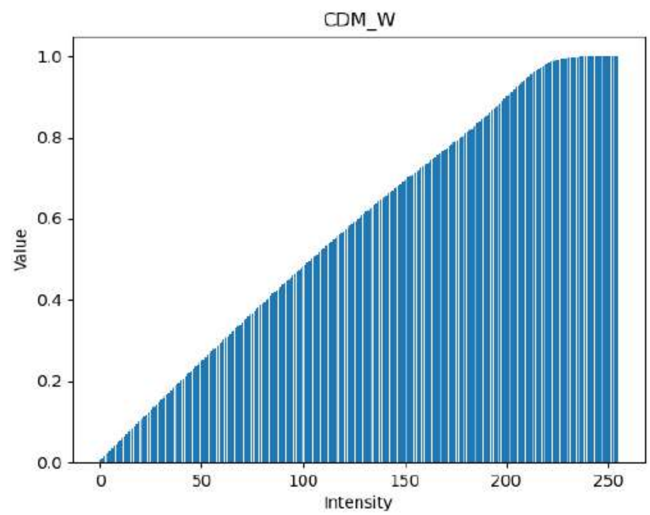

Fig. 7. Weighted cumulative probability density function

Moreover, it is expected that the PDF has the same shape than the fuzzy histogram since we are only trying to assign a probability (weight) to each mapped intensity. The cumulative histogram also confirms that most of the pixel distribution dissimilarities are found in high intensity values, since it not start to rise more than the 20% after the intensity value of 100.

The PDF with weighed distributions intends to preserve brightness and enrich the contrast. Since our input PDF has low amount of probability in low intensities pixels, the weighed distribution assign a higher score to them and lower to the high intensities. This let us realize that weighted PDF is taking into account the exceed of brightness that we initially has and correcting it by adapting the probability.Additionally, this diagram let us understand why the process of clipping (that we skip in our modified algorithm) is not completely necessary. At the end, we will be scoring each of the intensities depending on the amount of dissimilarity that we have and penalizing the high density distributions.

This idea can be extended to figure 7 which presents a 1 to 1 linear relation between intensity values and the probability of get them. This is the ideal expected behaviour and let us interpret that the gamma correction is going to accurately be performed.

Finally, figures 8 and 9 presents the resulted image and it´s respective histogram per channel.

Visually we see that the edges and some image details are more sharp. This is logical since we maximize the saturation and enhance contrast. Also, notice that the intensity of pixels was corrected. We get rid of the brightness regions along our

Fig. 8. Image result

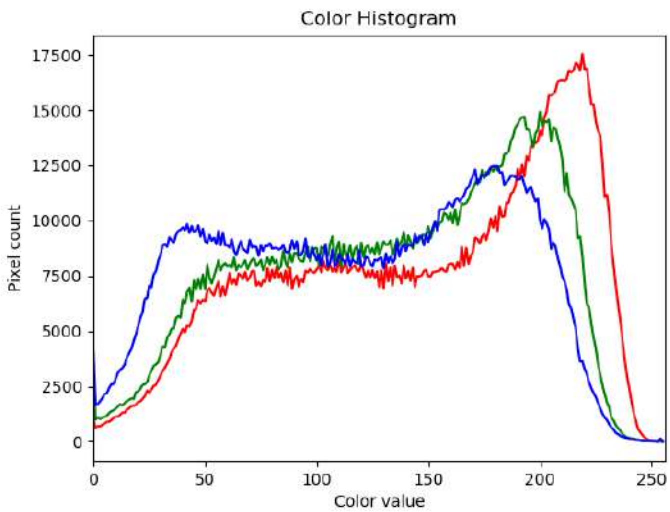

Fig. 9. Histogram of resulted image

image, even if they were not homogeneous. Now purity of colors is better defined overall image regions.

By the other side, the histogram of the resulted image tends to maintain original image color distribution. However, we notice that the resulting histogram is somehow more balance compared with the initial one. We have now low intensity pixels in each of the 3 colors and the concentration of high intensity values decreases, which means the brightness of initial input is corrected. We could argue that the result is similar to a type of histogram equalization, where we adjust the pixels in each of the respective color channels.

C. Quantitative Results

Table I summarize the results after evaluating the 5 images of the testing dataset and computing the metrics previously described (PSNR and SSIM).

TABLE I
RESULTS

Image	Befor Color Enhancement algorithm		After Color Enhancement algorithm	
	PSNR	SSIM	PSNR	SSIM
1	18,9203	0,7864	16,2009	0,7696
2	21,1364	0,8437	21,9402	0,8382
3	21,9621	0,8652	21,5319	0,8425
4	16,4898	0,7561	13,7941	0,6916
5	16,1929	0,7061	13,1507	0,6267
Average	**18,9403**	**0,7915**	**17,3235**	**0,7525**

Overall, based on the quantitative scores, we can state that the algorithm do not help to improve the similarity index score and either the peak to noise ratio index. This means that the image is less similar to the groundtruth after post-processing it using our code. One of the possible causes for this can be related with the sharper effect that we have in the output. Sharpening the image will pronounced more the edges and details, but at the same time it will make the defects more visual as well. In the reconstruction task in the dehazing problem (previous step of the algorithm) we already have errors in our image which might include fake edges/corners, pixels distortions, and unwanted borders highlights. Figure 10 clearly illustrates this effect.

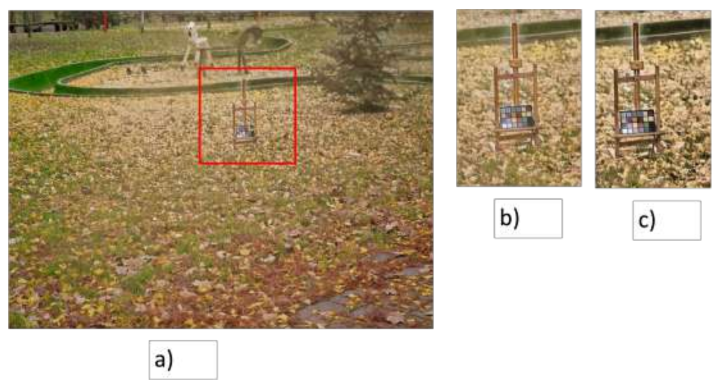

Fig. 10. **a)** input image, **b)** augmented patch of input image, **c)** augmented patch of output image

The metric indexes we are using to compare performance are pixel wised, which means that any minimum change can increase the distance between real pixel value and the one we are getting. The algorithm implemented in this work enrich the color regions, that might lead into making the defects more visible and hence the image is less similar to the groundtruth.

By the other side, we would not discard the algorithm since the weighted contrast enhancement using the fuzzy histogram helps to recover edges and reducing blurry patches. Figure 11 illustrates this result, in where the areas inside the red boxes got more details after applying the algorithm.

V. Conclusion

To conclude, we saw how the weighed fuzzy histogram algorithm was applied into dehazed reconstructed images. We interpreter that the weighted approach for histograms is recommended and will help to avoid some steps such as

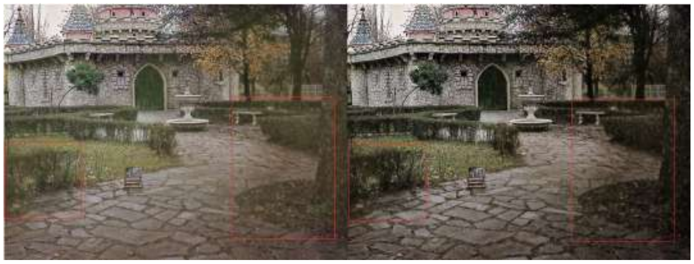

Fig. 11. Input original image in the left and the output result image after color enhancement algorithm in the right

clipping. It is basically trying to balance intensities with lower values distribution and penalize the ones that exceed, so algorithm help us to reduce brightness intensity across the dehazed images, regardless that problem is not homogeneous over all areas.

In the quantitative evaluation algorithm did not perform well, since peak to noise ratio score and similarity index score always decreases. However, as it was stated at the introduction of this paper, there has not been set any standard measure to determinate image quality in the dehazing task. Therefore, it is possible that we are increasing other different indexes apart from the ones evaluated during this paper. Additionally, we should take into account that our input images already have noise or not perfect reconstruction. Then, it is possible that this algorithm also increases these errors, lowering the scores values.

Overall algorithm makes images sharper, enriching the details such as edges. Additionally, it helps to reduce blurriness patches improving the visual perspective. Since the technique can be directly related with histogram equalization, and the haze problem is defined as an intensity recover, we would want to use the proposed algorithm to enhance blurry regions. Future work sets to try with different dataset (other reconstructions) and different metrics of evaluation. For example computing the the cumulative probability of blur detection (CPBD) before and after applying the fuzzy histogram algorithm.

REFERENCES

[1] Mayathevar, Krishnamurthy Veluchamy, Magudeeswaran Subramani, Bharath, "Fuzzy Color Histogram Equalization with Weighted Distribution for Image Enhancement", 2020, Optik. 216. 164927. 10.1016/j.ijleo.2020.164927.

[2] C. O. Ancuti et al., "NTIRE 2021 NonHomogeneous Dehazing Challenge Report," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 627-646, doi: 10.1109/CVPRW53098.2021.00074.

[3] Kaiming He, Jian Sun and Xiaoou Tang, "Single image haze removal using dark channel prior," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1956-1963, doi: 10.1109/CVPR.2009.5206515.

[4] M. Fu, H. Liu, Y. Yu, J. Chen and K. Wang, "DW-GAN: A Discrete Wavelet Transform GAN for NonHomogeneous Dehazing," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021, pp. 203-212, doi: 10.1109/CVPRW53098.2021.00029.

[5] D. Pu Wei, "Python realizes the exchange of RGB and HSI color space", PyCharm Community Edition, 2017. https://blog.actorsfit.com/a?ID=00600-7603c319-89a9-468d-a2fd-a491d010e4e5

[6] M. Meysenburg, "Creating histograms," Creating Histograms – Image Processing with Python. https://datacarpentry.org/image-processing/05-creating-histograms/

[7] NTIRE 2021 NonHomogeneous Dehazing Challenge, 2021, (https://competitions.codalab.org/competitions/28032)

[8] G. Schaefer, M. Rajab, M. Emre Celebi, H. Iyatomi, "Colour and contrast enhancement for improved skin lesion segmentation". Computerized Medical Imaging and Graphics, vol. 35, pg. 99-104, 2011.

[9] S. Huang, F. Cheng, Y. Chiu, Efficient contrast enhancement using adaptive gamma correction with weighting distribution, IEEE Trans. Image Process. 22 (3) (2013) 1032–1041.