Exhibition and Seminar on Science and Creative Technology, University of Al-Azhar Indonesia (EXSACT-A 2024) Proceeding

DOI: http://dx.doi.org/10.36722/exc.v2i1.2774

Real-Time Video Filtering Using Python: A Study on Dynamic Video Enhancement Techniques

Fahri Ramadhan[1], Aini Stirman[1], Fauzan Dharmawan[1], Dwi Astharini[1]

[1]Department of Electrical Engineering, Faculty of Science and Technology, Universitas Al Azhar Indonesia South Jakarta, 12110, Indonesia

Corresponding author/E-mail: ainistirman@gmail.com

***Abstract*—This paper presents a basic exploration of digital image processing method in representing a color spectrum using Python. Digital Image Processing is a science that moves to create, analyze, and produce information in the form of images with various coloring. The filters used in this paper are Red and Black Frame, Edge Detection, Negative and HSV in Python. The results of this filter show the difference in the resulting color spectrum. By separating the color spectrum, you will get a color output that varies according to the command.**

***Keywords*—** *Image Processing, Filter, Intensity, Color Spectrum.*

INTRODUCTION

The multimedia industry and computer technology have experienced rapid progress in the last few decades, especially in the field of Digital Image Processing. Nowadays, images and videos have become very important mediums in various fields of communication, entertainment, and even education. Recently, digital image processing has become important because it provides a fast, non-invasive, low-cost analysis of metallic materials corrosion [1]. At the outset, prior to digitalization, an image must be captured by a sensor. The sensor (or sensors) may be one of any number of modalities. Typically, one thinks of optical devices, but images can be generated in many other ways, such as nuclear magnetic resonance (NMR), infrared (IR), sonar, radar, ultrasound, synthetic aperture radar (SAR), scanning tunneling microscopy (STM), or atomic force microscopy (AFM) [2].

This paper has the goal of presenting a real-time discussion of video filtering in representing a color spectrum using the Python programming language. Python is a popular and versatile programming language, and has become the top choice for many image and video processing applications because of its powerful capabilities and various available libraries.

The main aspects described in this paper are the use of filters such as Red and Black Frame, Negative Filter, Edge Detection, and HSV. These filters have an important role in manipulating the color spectrum and creating interesting visual effects. The use of Python in implementing this filter enables efficient and effective image processing. Negative filter is an image where its lightest areas appear as darkest and the darkest areas appear as lightest [3]. Edge detection is primarily used to retrieve crucial data from an image, such as the object's shape and reflection. Because each pixel in an image has a variable intensity value, edge detection algorithms use this characteristic of the image to find the borders [4]. While HSV can segment images based on color by utilizing the upper and lower values of the HSV value so that objects and backgrounds are separated and can segment noisy color images. HSV accuracy is also better in segmenting color images [5].

The results of implementing these filters will be explained in detail in this paper. The difference in the color spectrum produced by each filter will be analyzed and presented to understand the impact of using filters in video processing. Through this research, it is hoped that readers will understand the importance of Digital Image Processing in representing the color spectrum and how the Python programming language can be used to achieve this goal. This research can also provide valuable insights into Digital Image Processing applications and coloring techniques.

METHOD

Real Time Video filtering is a transaction with parameters using electronic hardware and software components, namely laptops and webcams and indicators contained in Python. The electronic software component that we use is, to ensure image processing in this research using opencv in Python. In this research will use Red and Black Frame, Edge Detection and Negative and HSV in Python.

OpenCV

OpenCV (Open Source Computer Vision Library) is a free and open source library created by Intel with the goal of streamlining image-related programming. Face tracking, face detection, facial recognition, Kalman filtering, and various AI (Artificial Intelligence) techniques are just some of the features that OpenCV already has. and offers a variety of live computer vision algorithms for low-level APIs [6]. OpenCV is a free and open source computer vision library for C/C++ which has also been ported to Python, Java and Matlab. OpenCV has various features that can be used. The following are some of the most important OpenCV features, including:

Image and video I/O

With this interface we can read image data from files, or from a live video feed. It can also create image and video files.

Computer Vision in general and digital image processing (for low and mid level APIs)

With this interface we can perform experiments with various standard computer vision algorithms. This includes detection of lines, edges, tops, elliptical projections, image pyramid for multi-scale image processing, template matching, and various transforms (Fourier, discrete cosine, distance transform) and others.

High-level computer vision module

OpenCV also includes "high level" capabilities, such as additional capabilities for face detection, face recognition, including optical flow.

Methods for AI and machine learning

Computer vision applications often require machine learning or other AI methods, some of which are available in the OpenCV machine learning package

.

Image sampling and transformation

OpenCV includes interfaces for subregion substraction of images, random sampling, rotating, and more.

Methods for creating and analyzing binary images

Methods to account for 3D modeling

Simulation Algorithm

Outline:

- This research uses digital image procesing method
- Using laptop hardware + webcam-using visual code studio software
- With python opencv library to make the video filter
- Using Matplotlib library to make color distribution histogram/graphic.

Further steps

- Make sure the webcam is working properly
- Input : real time video image- in visual study code import library cv2 for image processing operations
- Webcam reads video image and connects directly to image processing in python
- Python processes a number of color filters for display
- Display results are compared with original and filtered
- Import library matplotlib.pyplot for histogram visualization

RESULTS AND DISCUSSION

Based on the four types of filters used with python, it will display differences in the color spectrum with the primary color rule red, green, blue (RGB). The filtered view is combined with the actual image to see how the intensity-to-frequency ratio in pixels of the color spectrum compares. Each filter produces a different intensity and frequency of RGB coloring. The color intensity scale used to assess the dark and light in images and RGB image processing ranges from a minimum limit of 0 to a maximum of 255.

Red and Black Frame Filter

In the red and black filter, the original video image is changed with image processing techniques in greyscale form. Then perform binary thresholding with a lower threshold of 50 being black and an upper threshold of 255 being white. then converted back into RGB components and removed the Blue and Green components. So this filter only shows the red color with the highest intensity.

Figure 1 shows the difference between the two images. The first image is the original video image and the second is the video image after being filtered red and black. This shows that in the original video image there are red, blue and green primary color components. After image processing, the video image only displays red

and black colors. The black color is obtained from the separation of pixel frequencies in the grayscale form into two categories, namely black and white based on the threshold and red by removing the blue and green components in the RGB rule.

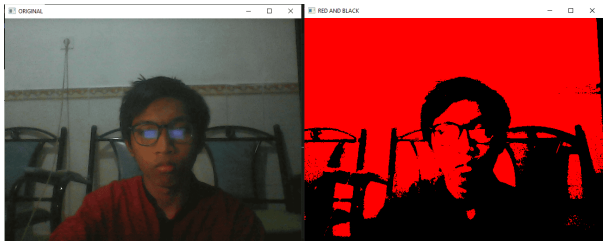

Figure 1. Original vs Red Black Filter

Table 1. Red and Black Filter Indicator

No.	Indicator	Value
	RGB to Greyscale	
1	Thresholding	Lower : 50 Upper : 255
2	Blue Component	0
3	Green Component	0

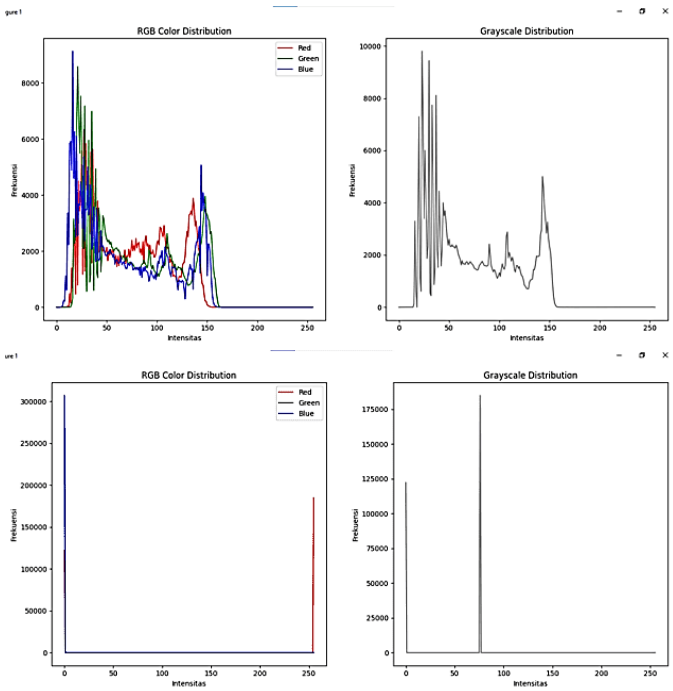

Figure 2. Histogram Original vs Red and Black Filter

The filter indicator used greatly influences the filter results to be produced for this study. Table 1 shows the values used in the red and black filters. With this value, an image with the intensity of the desired color spectrum will be produced. If the value of this indicator changes, it will also produce a different video image filter.

In Figure 2 the histogram displays the color distribution with a comparison of the intensity from a scale of 0 ~ 255 and the frequency range in unlimited pixels based on the resulting video image. The top image shows the histogram on the original video image and the bottom image shows the histogram for the red and black filter. The histogram of the original video image shows an even distribution of RGB colors at a low intensity scale which shows the original image in dark conditions. Meanwhile, the histogram on the red and black filter shows the number of occurrences of red at high intensity with an intensity of 0 for blue and green at the highest frequency, which means that the color has been changed to black. This difference in color distribution clearly illustrates the condition of the RGB/primary color spectrum in the video image before and after filtering with digital image processing techniques for red and black filters.

Negative Filter

Negative filters in python use the bitwise_not() function. This function applies the NOT operation to each color intensity pixel and converts the pixel value to its opposite. So that the color spectrum produced on the negative filter will produce a color spectrum that is inversely proportional to the original.

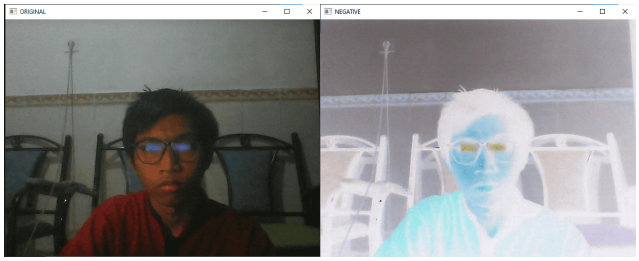

Figure 3. Original vs Negative Filter

Figure 3 shows the images for the original video and the negative filter video. In the original video image, the color spectrum is clearly visible which is based on mixing the RGB primary color components with intensity and contrast in dark conditions. In the video image negative filter, it can be seen that the intensity of the RGB primary color component changes in the opposite direction and the contrast becomes brighter. This indicates a change in the color spectrum pattern between the original video image and the video image filter.

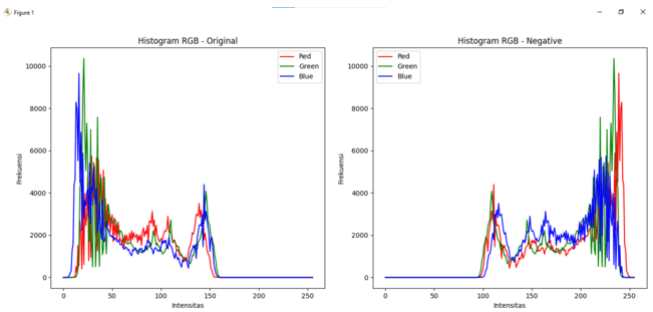

Figure 4. Histogram Original vs Negative Filter

Figure 4 is a histogram showing the distribution of colors with intensity from a scale of 0 - 255 and a frequency range in unlimited pixels based on the resulting video image. The first image displays the histogram for the original video image and the second image is the histogram for the inverse filter. In the first histogram, it can be seen that the distribution of the RGB primary colors is at the lower middle intensity level, which means that the color distribution is close to the black point. Meanwhile, in the second histogram, the resulting RGB color distribution is at the middle to high intensity level, which means that the RGB color distribution is close to the white point. The distribution of the color spectrum produced by the histogram shows a change in contrast and reverses the intensity values.

Edge Detection Filter

Edge detection is an image processing technique by identifying the edge of an object in the pixel intensity of the image. Edge detection works by comparing changes in color intensity that suddenly change so that they can be identified as boundaries between objects. In this study, edge detection was carried out using the canny method in python.

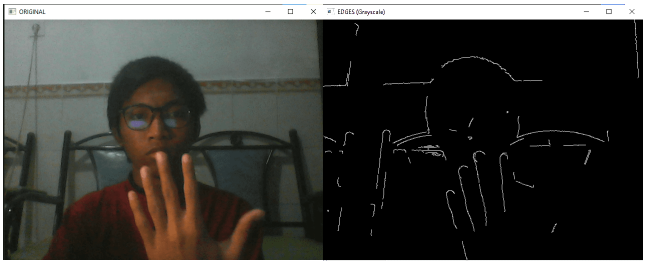

Figure 5. Original vs Edge Detection Filter

Figure 5 shows a comparison between the original video image and the video image edge detection. In the video, the image edge selection displays the image with the detected borders and the black color is the background. With the canny method, image processing is done by taking the highest pixel intensity and separating it from the lowest pixel intensity which is then ignored.

Table 2. Edge Detection Indicator

No	Grayscale form		
	Indicator	Lower	Upper
1	Threshold	100	200

The indicator in image processing in this study is by converting it into grayscale form and then thresholding is carried out to determine the minimum and maximum limits for object intensity detection as shown in table 2. The pixel gradient that is above the lower threshold number is considered a potential edge, which means it may become an edge if it occurs sudden changes, while the upper threshold value is the minimum intensity detection for edges and below that number is ignored.

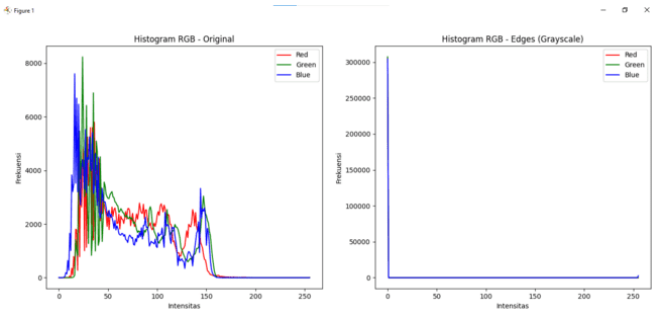

Figure 6. Histogram Original vs Edge Detection Filter

Figure 6 shows the histogram of the original video image color distribution against edge detection. In the edge detection histogram, the intensity of the RGB primary color has a very low frequency. This is because in the edge detection filter, the color spectrum that passes through the filter has intensity limit conditions so that the color spectrum that does not meet the intensity limit will be changed to 0 intensity.

HSV Filter

HSV filter is an image processing technique that filters the original image in a color space that separates the color components into three dimensions, namely Hue, Saturation, Value. Hue is the color component that appears in the image. Saturation describes the extent to which the color mixes with white. and Value is the light-dark value of the color. HSV filter in Python is made using the masking technique and the bitwise_AND function which is useful for filtering the color spectrum that has been set at the masking stage to be displayed then changing the color spectrum outside the mask boundary to black.

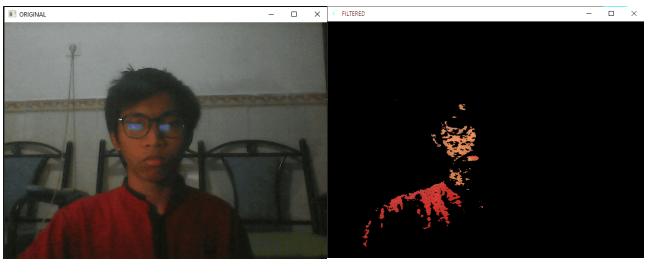

Figure 7. Original vs HSV Filter

Figure 7 shows a comparison between the original video image and the HSV filter video image. In the HSV filter, image processing is carried out on the original video by filtering the desired hue, saturation and value levels to be displayed. Indicator on the HSV filter as follows:

Table 3 shows the indicators used in the HSV filter image processing technique. The limit on Hue scales 0 - 10, Saturation 100 - 255, and Value 100 - 255. Based

on the value of this indicator, values outside the indicator will be filtered to black and will only display values that are within the limit.

Table 3. HSV Indicator

No.	Indicator	Lower	Upper
1	Hue	0	10
2	Saturation	100	255
3	Value	100	255

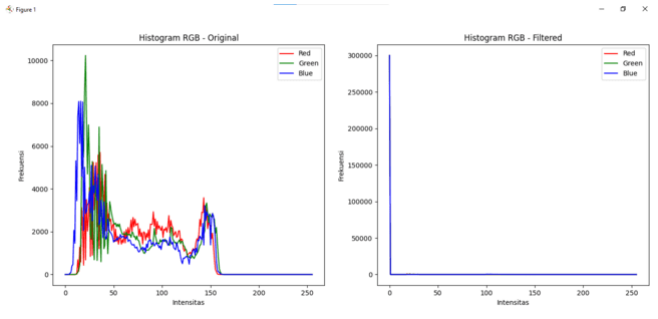

Figure 8. Histogram Original vs HSV Filter

Figure 8 is a histogram of the color distribution comparison of the original video image and the HSV filter. The first image is the original video image histogram in which the color distribution in the RGB primary color rule is evenly distributed. The second image is the histogram of the HSV filter, in which the color distribution of the RGB base color that passes through the filter has very little intensity and is dominated by 0 or black intensity. This is because the HSV processing of the resulting image is changed from the primary color rule to a three-dimensional component, thereby limiting the filtered color spectrum.

CONCLUSION

This study is designed to study digital image processing working with the use of color filters through the Python algorithm (OpenCV). During the study, the use of filters for digital image processing works by separating by limiting the intensity of a spectrum of colors that it passes through to then leaving the specified color spectrum. Light contrast and noise can affect the frequency of color intensity in the image to be filtered. Comparison of the number of colors based on the RGB primary colors displayed by the filter in the histogram with a comparison of the frequency in pixels and intensity on a scale of 0-255 produces results that vary from the original video image. Therefore, digital image processing with the Python algorithm (OpenCV) is very helpful in filtering the color spectrum and in the future further improvements can be made so that the tool is more effective.

REFERENCES

[1] D. Xia, S. Song, L. Tao, Z. Qin, Z. Wu, Z. Gao, J. Wang, et al. "Review-material degredation assessed by digital image processing: Fundamentals, progresses, and challenges," *Journal of Material Science and Technology*, vol. 53, pp. 146-162. 2020.

[2] E. R. Dougherty. "Digital Image Processing Methods." New York: Marcel Dekker, Inc, 2020.

[3] S. Khanal. "Negative Image". Medium, 2020. Available at: https://samirkhanal35.medium.com/negative-image-6cb7d5edcb54 (Accessed: 17/07/2023).

[4] R. Ranjan & V. Avasthi. 2021. "Enhanced Edge Detection Techinque in Digital Images Using Optimised Fuzzy Operation," *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*, pp. 215-219. 2022.

[5] T. Sadhukhan, S. Chatterjee, R. K. Das, S. Ghosh, & D. Das. "Efficient Removal of Noise from Image using HSV Filtering," *2019 Global Conference for Advancement in Technology (GCAT),* pp. 1-4. 2019.

[6] Bobsis. "Cara Melakukan Instalasi OpenCV yang Terintegrasi dengan Python 3.7 di Windows 10." Binus University, 2020. Available at: https://binus.ac.id/bandung/2020/01/cara-melakukan-instalasi-opencv-yang-terintegrasi-dengan-python-3-7-di-windows-10/ (Accessed: 17/07/2023).