Generic Color Theme Generation Based on Image Reference

I Gede Govindabhakta 13519139 (*Author*)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13519139@std.stei.itb.ac.id

***Abstract*—Many applications now provide extensive customizability options in terms of functionality and aesthetics. Color customizations are widely available in a large assortment of programs from text editors such as Visual Studio Code to customizations on desktop environments such as GNOME Desktop. In this paper, the author proposes an approach for automatic generic theme generation consisting of main and accent color candidates based on image references using image processing techniques such as edge detection, object segmentation, and histogram manipulation.**

Keywords—theme generation, color, customization

I. INTRODUCTION (*HEADING 1*)

Customizability in programs has been increasing in popularity with the emergence of open source programs being widely available and proprietary software providers providing extensible customization options to meet customer demands. This customizability can be seen in the emergence of customer maintained and populated marketplaces such as theme stores, plugin sharing pages, and other sources. Programs such as Visual Studio Code, among others, utilizes user editable files such as Javascript Object Notation (JSON) files to expose application configurations to tech savvy users intent to personalize their applications to their personal likings.

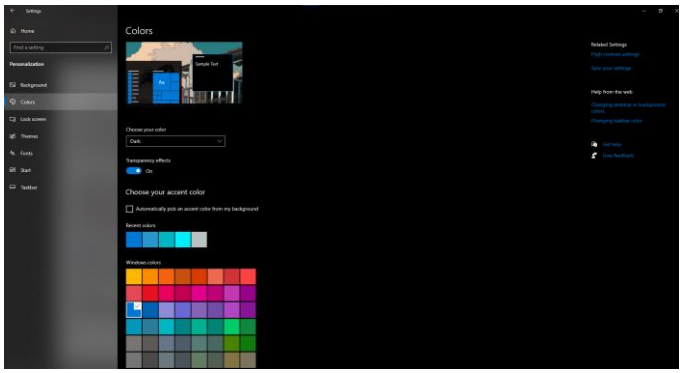

Image 1. Color customization options available in Windows 10

Among the most basic of customization options is the option to customize the user interface color of many application and programs, particularly dark mode options being the most widely used. Many color customization systems mainly consist of a foreground or main color theme, accompanied by an assortment of accent color customizations. Examples include Windows 10 and 11 now allowing the user to change from light to dark mode changing the foreground color for most programs and accent color options to show different coloring when highlighting actions or selected options in menus.

Image processing includes the processes and tools used to manipulate images, which are static collections of colors commonly displayed in a 2 dimensional fashion. These processes may include basic operations such as filters such as blurring to advanced manipulations such as object detection and segmentation which does not exclude the possibility of incorporating other technologies such as artificial intelligence and deep learning to further enhance the capabilities of image processing.

Image processing has proven to be useful for interpreting and finding key features in images such as recognizing people in a crowd, finding and recognizing characters in a digital image which can the be interpreted into text, and grouping segments of an image together based on similarity between neighboring regions. Tools such as histograms give insight into the composition and distribution of colors in an image which can give a rough estimation into how an image is well defined between objects, the general brightness of the scene, and possible cutoff values for removing unnecessary parts of an image.

This paper will discuss an approach that utilizes image processing methods that include color manipulation, edge detection, object segmentation, and histogram manipulation, to create a program capable of generating candidate colors, categorized under foreground or main color and accent colors, based off a provided reference image. The program is limited to only providing a list of candidates and not applying the colors to a customization scheme for any programs nor deterministically deciding or grading the generated color candidates

II. THEORY

A. Digital Image Processing

A digital image is a 2-dimensional collection of points named pixels which contain values representing colors in some form, possibly as a result of sampling from a continuous image source or as a computer generated image. Digital images can be traced from a the spatial and time domain, where only observing the spatial domain results in a static image which is often referred to as a picture, and including the time domain results in a dynamic digital image which is often referred to as a motion picture or video. This paper will not discuss motion pictures.

Digital images are represented as a function of x and y resulting in a matrix M x N where M and N denote the horizontal and vertical dimensions of the image in pixels known as an image's resolution. A visualization of a digital image represented as a function is presented in figure 2.

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Figure 2. Function representation of a digital image

The matrix presented is a collection of values generated from function f(x, y). The function returns a value, commonly representing a gray value or level. In computers, this gray value is often represented in 8 bits with values ranging from 0 to 255. Images with gray values represented with 1 bit, with values ranging from 0 to 1 are referred to as binary images.

These matrices may also include other dimensions, such as adding multiple layers to have a wider representation of colors such as RGB, with each layer's gray level representing a certain value of either red, green, or blue in the specified pixel. Images which only include one layer, which represent only on type of gray level, often the color gray, is referred to as a grayscale image.

B. Edge Detection

Edges are parts of objects which define a clear end to a region or specify a segmentation. In images, edges are areas of the image which show the boundaries of certain objects, often but not always recognizable by steep color changes in neighboring pixels. Image processing typically registers edges by their respective axis, thus generating a vertical edge and horizontal edge. Some methods use a diagonal edge in place of the typical horizontal and vertical edge definitions.

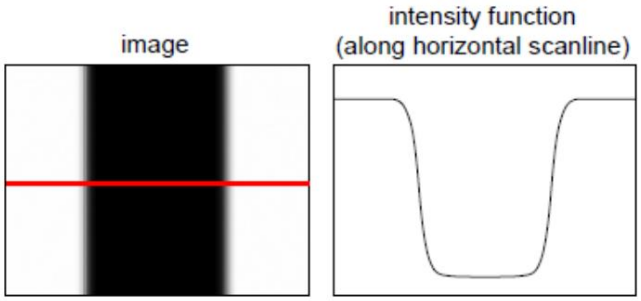

Figure 3. Visualization of an edge in an image

Steep edges are edges which show a very sudden change in gray levels. Around the defined edge, the gray levels are typically more consistent, thus further defining the edge even when used with a wider kernel function. Both humans and computers often have no issue identifying these type of edges as they are often clearly defined. Examples of matrices which denote a steep edge are included in figure 4.

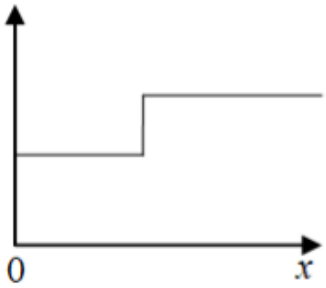

Figure 4. Change in gray level that occurs in a steep edge

Sloping edges show a more gradual change in gray levels. Defining whether a certain rate of change is defined as an edge is often accompanied by a certain threshold. Sloping edges are often the type of edges easily identifiable by humans, but not by computers. Examples of sloping edges are included in figure 5.

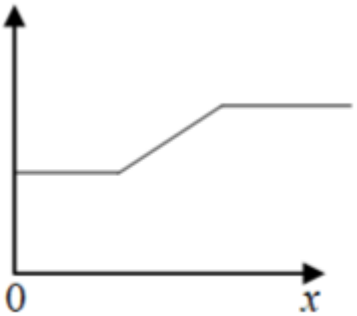

Figure 5. Change in gray level in a sloping edge

Noisy edges, which often are present when taking image samples from photos taken in real life instead of computer generated images such as vector graphics, are edges which are accompanied by noise such that identifying them often requires preprocessing in forms such as smoothing or

thresholding. Examples of noisy edges are included in figure 5.

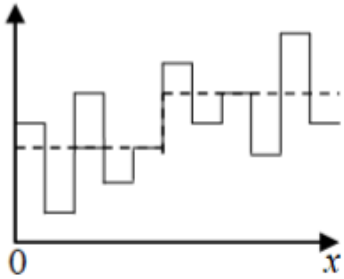

Figure 6. Change in gray level of a noisy edge

When detecting edges, a variety of methods from first degree derivatives such as the Sobel operator and second degree derivatives like a Laplacian are available with different trade offs. First degree derivatives define an edge by the rate of change of pixels, inline with common intuition, and is thus commonly accompanied by a threshold. Second degree derivatives, or Laplacians, define an edge by the rate of sudden change in pixel gray levels. An edge defined by a Laplacian operator is defined by zero-crossing, where second degree derivatives cross the zero value when traversing through the matrix or by interpolation when such occurrences are infrequent.

C. Image Smoothing

Image smoothing is a process of reduces harder defined edges in an image, often with the intent of reducing noise present. Image smoothing works by selecting values that reduce the maximum intensity of a pixel, relative to pixels around it. A result of gaussian filtering is included in figure 7.

Figure 7. Result of a Gaussian filter being applied to an image

Kernels such as the Gaussian filter are often used among others. The matrix or kernel used for convolution in a Gaussian filter is included in figure 8.

1	2	1
2	4	2
1	2	1

Figure 8. Gaussian Mask

D. Object segmentation

Object segmentation is the process of separating or segmenting an object from the larger foreground of an image. This process may take several approaches such as identifying the objects using various algorithms such as but not limited to contour identification, edge detection, and machine learning approaches such as convolutional neural networks.

A common approach used is to apply thresholding to a gray scale image based on a low and upper bound, defining a window where an object would be separated from the foreground which is particularly effective when the object is well defined against the foreground. The thresholded image is then used as a mask to later segment the object away from the foreground by applying a bitwise AND operation on the mask and the original image. An example of the masking and segmentation process is included in figure 9.

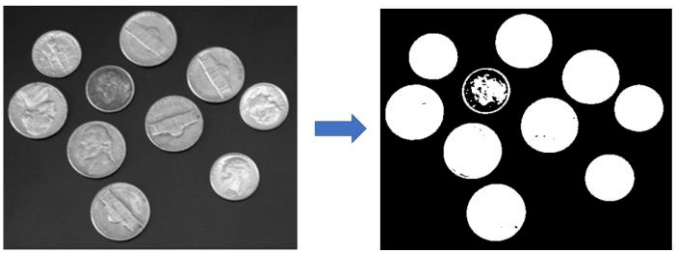

Figure 9. Masking process for object segmentation in image processing

E. Color

Color is a spectrum, mostly the visible region, which resides inside a perfect light (white light). This spectrum consists of other colors with the ranges of wavelengths being visible as different colors to the human eye.

Images contain information, previously mentioned as gray levels, that show the intensity of a color. Multiple systems are available for use when representing a color which may include different approaches to defining the hue of a color, the inclusion of transparency, and others.

The RGBA system (Red, Green, Blue, Alpha) represents colors as a vector of 4 components, with RGB being primary colors which can be used to reconstruct any other color available and alpha being a value to represent the transparency of the color. A visualization of the RGB spectrum is included in figure 10.

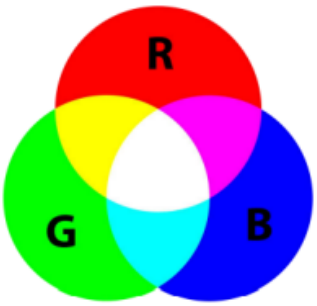

Figure 10. The RGB spectrum

III. Implementation

The program is implemented in python and utilizes multiple common libraries used in image processing such as OpenCV2 and PIL (pillow). The main steps taken during the generation of the generic customization themes is described below:

1. Segmenting the object away from the foreground

Object segmentation is used to separate the main object away from the foreground. This is to enable the creation of main colors from the foreground image and accent colors from the main object. The segmentation is initiated by converting the image to grayscale, followed by thresholding to generate the appropriate masks. The mask and its inverse are used under a bitwise AND operation to generate and write the object and main images. The code used to implement this behavior is included below.

```
def get_object(img):
    img_gray = gray_image(img)
    th, im_th = cv2.threshold(img_gray, 220, 255,
cv2.THRESH_BINARY_INV)
    im_floodfill = im_th.copy()
    h, w = im_th.shape[:2]
    mask = np.zeros((h+2, w+2), np.uint8)
    cv2.floodFill(im_floodfill, mask, (0, 0), 255)
    im_floodfill_inv = cv2.bitwise_not(im_floodfill)
    mask = im_th | im_floodfill_inv

    inv_mask = cv2.bitwise_not(mask)

    mask = cv2.merge((mask.copy(), mask.copy(),
mask.copy(), mask.copy()))
    inv_mask = cv2.merge((inv_mask.copy(),
inv_mask.copy(), inv_mask.copy(), inv_mask.copy()))

    imga = cv2.cvtColor(img, cv2.COLOR_BGR2BGRA)

    masked = cv2.bitwise_and(mask, imga)
    unmasked = cv2.bitwise_and(inv_mask, imga)
    cv2.imwrite(OBJECT_FILENAME, masked)
    cv2.imwrite(FOREGROUND_FILENAME, unmasked)

    return masked
```

The resulting generated from the test samples are included below.

Figure 11. Foreground image

Figure 12. Main object image

2. Generation of candidate colors

Generation of candidate colors for the generic customization schema is done by selecting the most prominent colors from the histogram in each image and taking the best 10, excluding transparent colors. The colors are then converted from RGBA to HEX to be more appropriate for use in most customization schemaas. The implemented code and generated color palette from the test image is included below.

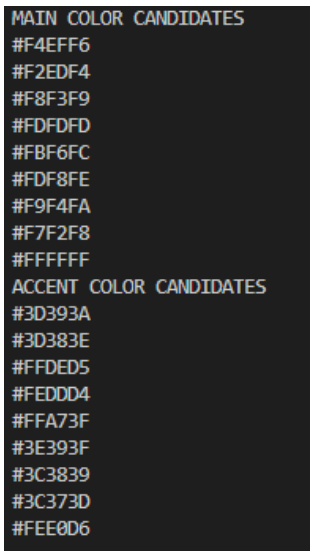

Figure 12. Generated candidate colors

Figure 13. Generated main color candidates

Figure 14. Generated accent color candidates

IV. Analysis

The generated color palettes based on the test image were sufficient and as expected. The program has issues when separating the object from images with less well defined boundaries between foreground and main object. This may be related to the arbitrary value used as the threshold low and upper bound. Further improvements can be made by using algorithms such as the Otsu operator to determine an appropriate threshold.

V. Conclusion

Using image processing techniques such as object segmentation, masking, and histogram analysis, the program was able to successfully generate an appropriate main and accent color palette suitable for generic customizations. Further improvements can be made to the program to better segmentate main objects and to integrate into existing customization platforms.

Source Code

https://github.com/Govindabhakta/pengcit-makalah

References

[1] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/18-Pendeteksian-Tepi-Bagian1-2022.pdf accessed on 19 December 2022

[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2021-2022/22-Segmentasi-Citra-Bagian1-2022.pdf accessed on 19 December 2022

https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2020-2021/07-Konvolusi.pdf accessed on 19 December 2022

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2022
Ttd

I Gede Govindabhakta 13519139