Table of Contents

Acknowledgement

I would like to express my warmest thanks to my lecturer, **Dr. Vazeerudeen Hameed**, for his patient delivery of the knowledge throughout this module, Imaging and Special Effects. He always ensures that not a single student is left behind in this module. His contribution is sincerely appreciated and acknowledged. Besides, I would also like to express my gratitude to my family and friends for their continuous support and understanding when undertaking my assignment.

1.0 Objectives

The primary objective of the assignment is to propose an algorithm for recognizing objects with their color characteristics. By detecting different colors in the figure, it will segment a specific color from the background based on the user's choice to identify the objects. The other objectives will be the real-life implementation of color-based object detection, such as identifying withered leaves or sun-scorched grass, detecting vehicle plates, recognizing people with different skin colors, and others.

2.0 Problem Statement

Recognizing an object requires the knowledge of image processing, pattern recognition, and computer vision based on the concepts of Image and Special Effect (ISE). Most of the time, image processing is the first step of developing the algorithm for an automatic object recognition system. It is mainly used to improve the image quality by reducing the background noise in order to provide a clearer vision for the objects' characteristics in that image. Noise disruption will significantly influence the data signal in a connected network during the image collection or transmission process that may destroy the detailed feature information hidden in the image. Hence, the image processing technique is a must to ensure the accuracy of the image data (Huang & Zhu, 2021).

The object identification methods can be according to the object's color, contours, size, and others. Color acting as one of the object's characteristics can be used to differentiate the objects in an image. For example, in a landscape photo, the blue color may represent the sky, the white color may represent the cloud, the green color may represent the leaves or grasses, and the brown color may represent the tree trunk. Color-based object detection will be helpful for filtering the desired color for further research and analysis of the image. It provides a simple and effective method to outline the objects while matching the images based on the color histograms. Hence, in this assignment, a color model will be proposed to solve the object detection problem.

3.0 Literature Review

Object detection technique has been widely exposed to multiple applications: from image annotation to vehicle license plate localization, from object tracking to face recognition system (Kim et al., 2002; Duan et al., 2017; Cheng et al., 2018; Singh & Prasad, 2018). They all have a similarity that is to identify the differences among the scanned objects in an image. After that, the algorithm will screen out the target object tallied with the user's requirements. For example, the face recognition technique only focuses on the captured faces in an image. Therefore, the designed algorithm needs to accept the facial characteristics, such as the face shape, skin color, contour lines, and others, as parameters to filter only the faces for further analysis (Singh & Prasad, 2018).

Color-based object detection technique requires a color model representing different ranges of color from the perspective of human vision. As color is an abstract concept, it is hard to describe the color details using written words. A color model will fill the gap to represent a color with a set of numbers. Nowadays, several color models have been introduced, such as the RGB color model, the HSI color model, the HSV color model, and others.

The research by Zhao et al. (2020) contributes to RGB-D salient object detection in real time. RGB-D image is the traditional red-green-blue (RGB) image combined with the corresponding depth image to gain the stereoscopic viewing angles of the objects in an image. They use single stream encoder to build advantage of early fusion instead of two-stream encoder. After that, they applied the depth-enhanced dual attention module (DEDA) to optimize the decoder during the middle fusion performance.

Suksaengjun et al. (2015) used the HSI color model (hue-saturation-intensity) in the development of an object detection-based smart cage. This color model is implemented to detect the dogs' excrement as HSI thresholding will be outstanding in the same tone of excrement's color. Hue (H) value is a color attribute describing a pure color that can be represented from 0 to 360 degrees. For instance, 0 degree represents red, 120 degrees is green, and 360 degrees mean magenta. Saturation (S) value is pure color's vividness that is diluted by white light. It can be described in the range between 0 and 1. Intensity (I) value is the gray level of the image. Same with saturation value, the representing range is from 0 to 1 (0 is black, 1 is white). The collected image of a webcam will be RGB in its original format. A conversion for the original image into an HSI color model is a must to thresholding the HSI image.

HSV color model (hue-saturation-value) is similar with HSI color model. The only difference is the value (V) which represents the color brightness. 0 value results in the pure black color of the picture, while 1 value results in no black color mixed into the picture and reveals the most color. HSV color model was applied for improving the shadow suppression of a moving object. As the shadows are similar in hue, the saturation and brightness of shadows can be used to distinguish them with a more accurate description than the traditional RGB color model. Hence, a shadow detection based on the HSV model can achieve a better performance of moving object localization with satisfactory results.

4.0 Color Space Description and Justification

HSV Color Space

A color space is a mathematical model interpreting the existing components of color profiling according to the color model. The chosen color space for the proposed color-based object detection algorithm will be HSV color space established by Alvy Ray Smith in 1978. It consists of three parameters, which are hue, saturation, and value. Hue (H) is the pure color's components given a weighting on a scale of 0 to 360 degree. Saturation (S) is the colorfulness of the color. It has a range between 0 and 1, where 0 represents no color in the picture and 1 displays the most color in the picture. Value (V) also called brightness is the luminance and intensity of the color. It is also given a range from 0 to 1, where 0 shows only dark color in the picture and 1 gives the most bright-colored image (Tarek Mamdouh, 2020). The following data is the distribution of colors in hue:

- Red color: 0 - 60 degrees
- Yellow color: 61 - 120 degrees
- Green color: 121 - 180 degrees
- Cyan color: 181 - 240 degrees
- Blue color: 241 - 300 degrees
- Magenta color: 301 - 360 degrees

Figure 1 Hue Components in HSV color space (Tarek Mamdouh, 2020)

In OpenCV, the range of hue, saturation, and value is different from the actual HSV color space. Hue (H) ranges from 0 to 180, saturation (S) ranges from 0 to 255, and value (V) ranges from 0 to 255.

The reason of choosing HSV color space is because it can express a clearer perception color contact compared to RGB color space. HSV model acting as non-linear transformation of

RGB model mimics the ways of interpreting colors from the human’s perception. In computer vision, HSV has better performances on the changing external lighting conditions of a picture, such as pale shadow, even a minor change in the saturation or brightness. Hence, it can clearly outline the boundary of an object by removing the lighting effects. In contrast, RGB is harder to be used for determining the specific color of an object (Shuhua & Gaizhi, 2010; Tarek Mamdouh, 2020).

5.0 Critical Comments and Analysis

To implement the color-based object detection, an algorithm based on HSV color space was introduced. We tested the algorithm with a picture of balloons with different colors (red, orange, yellow, green, blue, magenta). The following image will be used as the input for testing the proposed algorithm:

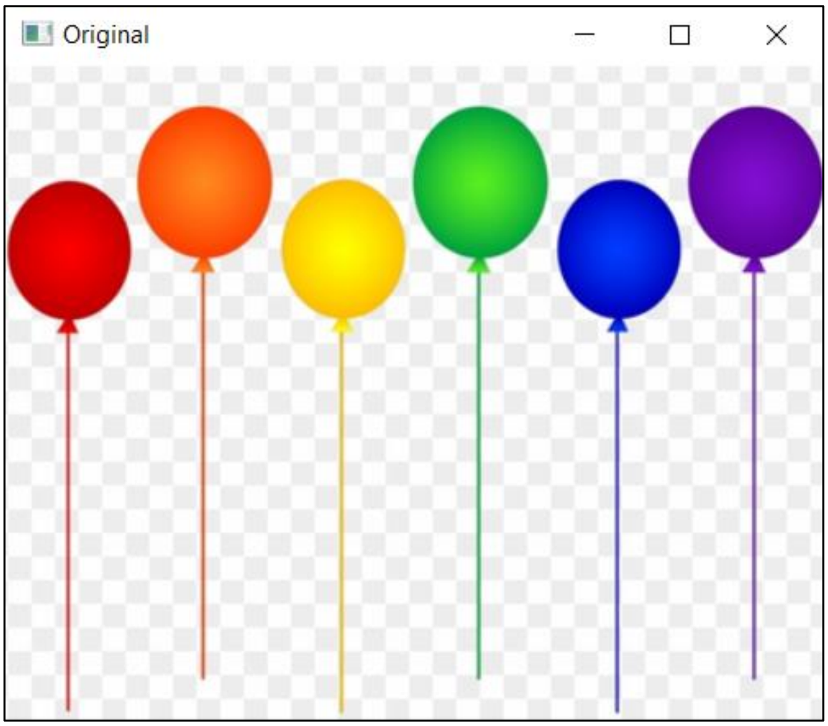
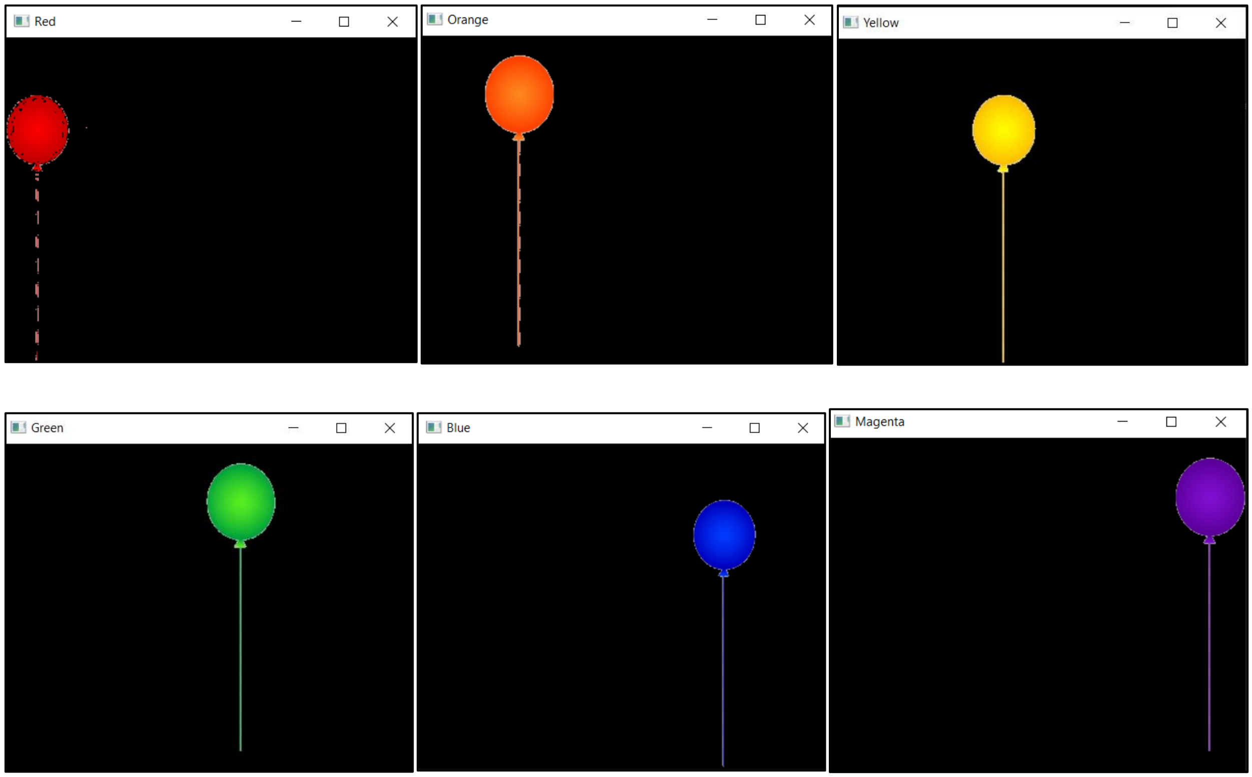


Figure 2 Original Image

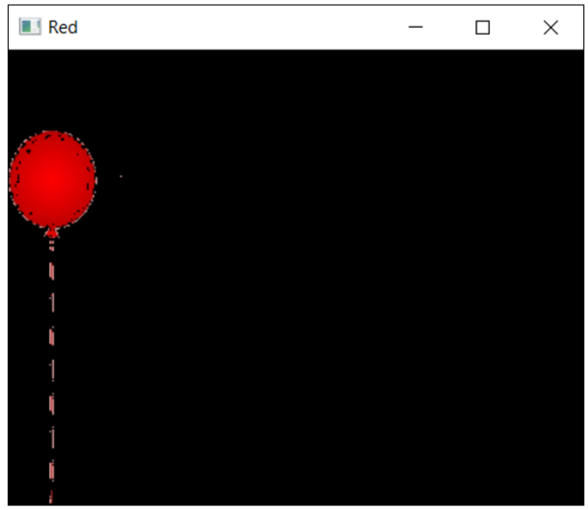
The program will perform image segmentation based on the balloon's color. Each color will be separated for different output. Firstly, the algorithm is designed to convert the BGR image to HSV image. Before thresholding the HSV image, the lower and upper boundary of the three HSV components (hue, saturation, value) are defined. For example, red color was set 0 – 5 for the hue boundary, 50 – 255 for the saturation boundary, and 70 – 255 for the value boundary. The following images will show the gained results based on the proposed algorithm:



When performing color identification, there were difficulties regarding the range of HSV components. For the hue aspect, the actual value for each color range is different with the expected distribution of color in previous section (color space description and justification). For example, the expected range of hue for red color is 0 – 30 degrees. However, within 0-30 degrees hue, red, orange, and yellow balloon will be shown for the output. Hence, it is recommended to adjust the lower and upper boundary for each hue according to the desired colors. For red color, the adjusted range of hue is between 0 – 5 degrees.

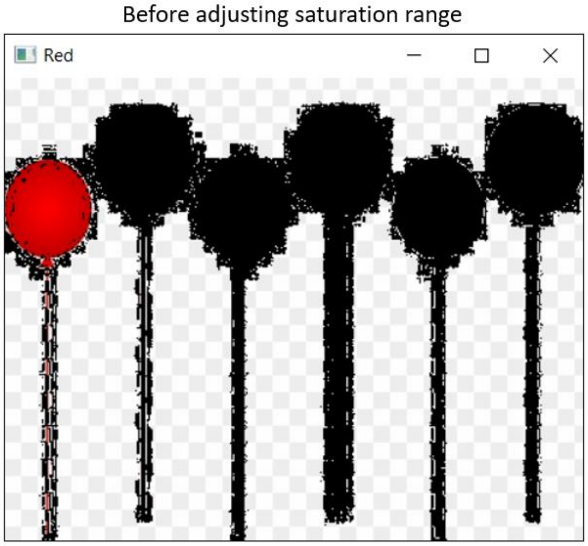
Before adjusting hue range

After adjusting hue range

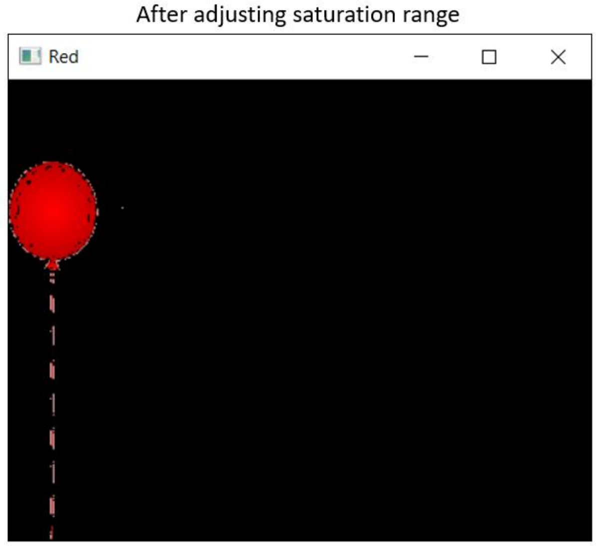


For the saturation aspect, there will be a big noise which interrupts the finding of red colors when performing image segmentation. The more tolerant range for saturation range, the higher possibility of false result that may including the white and grey color for the output. The original range for the saturation is between 0 and 255 degrees. Hence, the saturation range is adjusted to accept only 50 to 255 degrees to avoid the unnecessary color, such as white and grey color.

Before adjusting saturation range



After adjusting saturation range



6.0 Conclusion

In short, the proposed algorithm for this assignment is built based on the HSV color space to perform image segmentation for color-based object detection. The applied programming language is python with the OpenCV library. Overall, the proposed algorithm can only segment single color instead of multiple combination of colors in a picture. This assignment is completed with the basic image processing techniques learnt during the lecturers and online sources. A further investigation regarding the real-life implementation of color-based object detection can be completed by considering the interferences of colorful background.

7.0 References

Cheng, Q., Zhang, Q., Fu, P., Tu, C., & Li, S. (2018). A survey and analysis on automatic image annotation. *Pattern Recognition*, *79*, 242-259. https://doi.org/10.1016/j.patcog.2018.02.017

Cucchiara, R., Crana, C., Piccardi, M., Prati, A., & Sirotti, S. (2001). Improving shadow suppression in moving object detection with HSV color information. *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*. https://doi.org/10.1109/itsc.2001.948679

Duan, C., Rao, X., Yang, L., & Liu, Y. (2017). Fusing RFID and computer vision for fine-grained object tracking. *IEEE INFOCOM 2017 - IEEE Conference On Computer Communications*. https://doi.org/10.1109/infocom.2017.8057161

Huang, K., & Zhu, H. (2021). Image Noise Removal Method Based on Improved Nonlocal Mean Algorithm. *Complexity*, *2021*, 1-10. https://doi.org/10.1155/2021/5578788

Kim, K., Jung, K., & Kim, J. (2002). Color Texture-Based Object Detection: An Application to License Plate Localization. *Pattern Recognition With Support Vector Machines*, 293-309. https://doi.org/10.1007/3-540-45665-1_23

Shuhua, L., & Gaizhi, G. (2010). The application of improved HSV color space model in image processing. *2010 2Nd International Conference On Future Computer And Communication*. https://doi.org/10.1109/icfcc.2010.5497299

Singh, S., & Prasad, S. (2018). Techniques and Challenges of Face Recognition: A Critical Review. *Procedia Computer Science*, *143*, 536-543. https://doi.org/10.1016/j.procs.2018.10.427

Suksaengjun, P., Thanapatay, D., Chinrungrueng, J., & Nobuhiko, S. (2015). Smart cage using object detection. *2015 6Th International Conference Of Information And Communication Technology For Embedded Systems (IC-ICTES)*. https://doi.org/10.1109/ictemsys.2015.7110814

Tarek Mamdouh. (2020). *Color spaces (RGB vs HSV) - Which one you should use?*. HubPages. Retrieved 28 November 2021, from https://discover.hubpages.com/technology/Color-spaces-RGB-vs-HSV-Which-one-to-use.

Zhao, X., Zhang, L., Pang, Y., Lu, H., & Zhang, L. (2020). A Single Stream Network for Robust and Real-Time RGB-D Salient Object Detection. *Computer Vision – ECCV 2020*, 646-662. https://doi.org/10.1007/978-3-030-58542-6_39

8.0 Appendix

Code

```
import cv2
import numpy as np

# Input
orig = cv2.imread("test.png")
cv2.imshow("Original", orig)
# Convert BGR to HSV
hsv = cv2.cvtColor(orig, cv2.COLOR_BGR2HSV)
```

```
# Define ranges of color in HSV color space
lower_red = np.array([0, 50, 70])
upper_red = np.array([5, 255, 255])

lower_orange = np.array([6, 50, 70])
upper_orange = np.array([15, 255, 255])

lower_yellow = np.array([16, 50, 70])
upper_yellow = np.array([35, 255, 255])

lower_green = np.array([36, 50, 70])
upper_green = np.array([89, 255, 255])

lower_blue = np.array([90, 50, 70])
upper_blue = np.array([125, 255, 255])

lower_magenta = np.array([126, 50, 70])
upper_magenta = np.array([160, 255, 255])
```

```
# Threshold the HSV image
red_mask = cv2.inRange(hsv, lower_red, upper_red)
red_res = cv2.bitwise_and(orig,orig, mask= red_mask)

orange_mask = cv2.inRange(hsv, lower_orange, upper_orange)
orange_res = cv2.bitwise_and(orig,orig, mask= orange_mask)

yellow_mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
yellow_res = cv2.bitwise_and(orig,orig, mask= yellow_mask)

green_mask = cv2.inRange(hsv, lower_green, upper_green)
green_res = cv2.bitwise_and(orig,orig, mask= green_mask)

blue_mask = cv2.inRange(hsv, lower_blue, upper_blue)
blue_res = cv2.bitwise_and(orig,orig, mask= blue_mask)

magenta_mask = cv2.inRange(hsv, lower_magenta, upper_magenta)
magenta_res = cv2.bitwise_and(orig,orig, mask= magenta_mask)
```

```
# Show output
cv2.imshow("Red", red_res)
cv2.imshow("Orange", orange_res)
cv2.imshow("Yellow", yellow_res)
cv2.imshow("Green", green_res)
cv2.imshow("Blue", blue_res)
cv2.imshow("Magenta", magenta_res)

cv2.waitKey()
```