Price Tag Recognition using HSV Color Space

M. N. A. Hussin, A. H. Ahmad and M. A. Abdul Razak
Department of Electronics & Computer Engineering, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Malaysia.
mohdazhar@utm.my

***Abstract*— Object and character recognition from images are widely used for example plate car number recognition, scanner and even in the autonomous car. Researchers had developed lot of algorithms and image processing techniques to extract the features in the image such as hue, saturation and value (HSV) color filtering. However, the application of optical character recognition and image processing is not limited to car plate recognition but it can be implemented to recognize the other application such as price tag recognition. This project employed HSV color model for filtering the color and several image processing techniques to get the Region of Interest (ROI) and Optical Character Recognition (OCR) to recognize the product name and price in the price tags. Three price tags from shopping malls namely A, G and T are selected and analyzed in Android system. The image of price tags is filtered by color and applied rectangle contour detection to find the bounded area of the price tags with background image. Then, the process continues with masking the specific part in the price tags to recognize the character. In this method, color and contour detection able to separate the ROI with background and almost get the desired result with detection accuracy of 92.3% for price tag G, 90.63% for price tag T and 59.4% for price tag A.**

***Index Terms*— HSV Color Space; Optical Character Recognition; Region of Interest.**

I. Introduction

Shopping is a monthly routine for some people and sometime become a weekly and even daily routine. Also, most people are concerned with price of the products that they want to buy. They also compare the price from another store to find the lowest price. This is to avoid over spending during shopping.

Besides that, some shopping malls do not update the price tag displayed at the shelves, thus causes the prices are unmatched during payment. Sometimes customers have to pay more due to these mistakes. Using a calculator during shopping is a best way to avoid overspending by calculating each item's price. Moreover, people have limited memory to remember everything such as the name of the products and prices. Also, calculator only shows the number and make difficult to compare with receipt after payment.

This paper would discuss the development of Android application on a smart phone that can detect the price tag such as price and product name by employing image processing technique.

II. Previous Works

A. Color Detection

Color filtering is an effective method to distinguish the objects by selecting the specific color. There are several color models that have been developed such as hue based color space (HSI, HSV and HSL), luminance based color space (YCbCr, YIQ, YUV), RGB based color space (RGB, normalized RGB) and perceptually uniform color space (CIEXYZ, CIELAB, CIELUV).

Guerrero *et al.* [1] used support vector machine with RGB color space to identify the crop with greenish color as green channel give more useful information. However, this method is unable to separate plants pixel from the background and in low or bright lighting condition.

The most popular color detection algorithm is called CAMSHAFT algorithm proposed by Bradski in 1998 [2] by using HSV color space. HSV stands for Hue, Saturation and Value where hue is described as pure color, saturation refer as dilution of pure color with white light and value refer to brightness of the color. Author only relies on color probability distribution of color histogram in hue channel to detect the skin color. However, there is a consideration need to acknowledge before use the proposed method. The target object color must not same as background color otherwise this method failed to recognize the object.

Hamuda *et al.* [3] used HSV color space and applied same as Bradski's method [2] to discriminate cauliflower plant among weeds and soil. Authors use HSV decision tree to detect the ROI which is mainly contains one color. Authors use all HSV channels to identify cauliflower region and varies illumination and removing background. After detecting using HSV, they applied a method called finding contour to detect the shape of the cauliflower and feed into decision tree to verify the output.

According to Shaik *et al.* [4], RGB color space is not suitable for color based detection and analysis due to mixing in color (chrominance) and intensity (luminance) information. Authors proposed a skin detection by comparing YCbCr and HSV based color on various lighting and background condition. They conclude that YCbCr color space can be applied for complex color images with uneven illumination while HSV based detection is suited for simple images with uniform background.

B. Detection of Region of Interest

A technique to detect ROI is using contour detection. Arbelaez *et al.* [5] proposed a new approach to contour detection and image segmentation. By using spectral clustering, the brightness, color and texture are cues. The cues are computed on every location on the image by applying oriented gradient operators. Eigenvalue and eigenvectors of the matrix are derived to encode the contour information. The image is segmented by constructing a set of initial regions from the contours. The region is then formed the hierarchy.

P. Kapsalas *et al.* [6] proposed a technique to detect an object in image by clustering the point of interest and hierarchically forming candidate region according to similarity and spatial proximity predicates. He used feature extraction from keypoint derived by Harris corner detector to find the point that has large gradient in all directions. The feature points are grouped via an unsupervised clustering approach. Then check the combinations of ROI at all spatial configuration using boosted classifiers.

Another approach to detect the ROI is using the classifier. Forczmański and Markiewicz [7] proposed a technique to automatic detection of important object such as stamp and logo in the paper using AdaBoost cascade classifier and Haar-Like features. In this method, the data are trained iteratively and down scale an input image by 10%. The test images undergo two steps to detect the ROI. The steps are detection performed by AdaBoost cascade of weak classifiers of sample image and verification to increase the detection accuracy. Low-level features are used to verify detected candidate images. According to them, it is not necessary to reduce the dimensionality or features as it does not improve the classification accuracy.

III. Object Detection Method

In this section, author would develop an image processing pipeline to detect the price tag based on color filtering and contour detection. The image processing engine is separated into two parts which are image processing and optical character recognition. Author only focusing on image processing as the main process is to detect the price tag. Moreover, author used open source library for optical character recognition from Google called OCR Tesseract [8].

A. *Image Processing Engine*

Image processing engine is adopted form various techniques such as HSV color filtering, adaptive threshold, image segmentation, morphology and median filter to detect the desired objects. Author would discuss on development of image processing engine to detect the region of price tag and using optical character recognition to recognize the text such as price and product name. Figure 1 shows the block diagram of image processing engine.

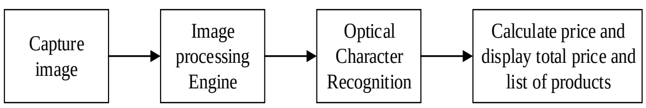


Figure 1: Block diagram of image processing engine

The image processing engine started by selecting the location of purchase to decide the range of hue values of color to be detected. Then, the captured image that contains price tag is converted to HSV color space and filter by the selected color. Next, the image is converted to binary image and applied rectangle contour detection by finding the largest rectangle contour. When the price tag is detected, the image is cropped and mask of price tag is applied to the cropped image. Finally, OCR is employed to detect the text such as price and product name and the results are displayed. Figure 2 shows the process of image processing engine.

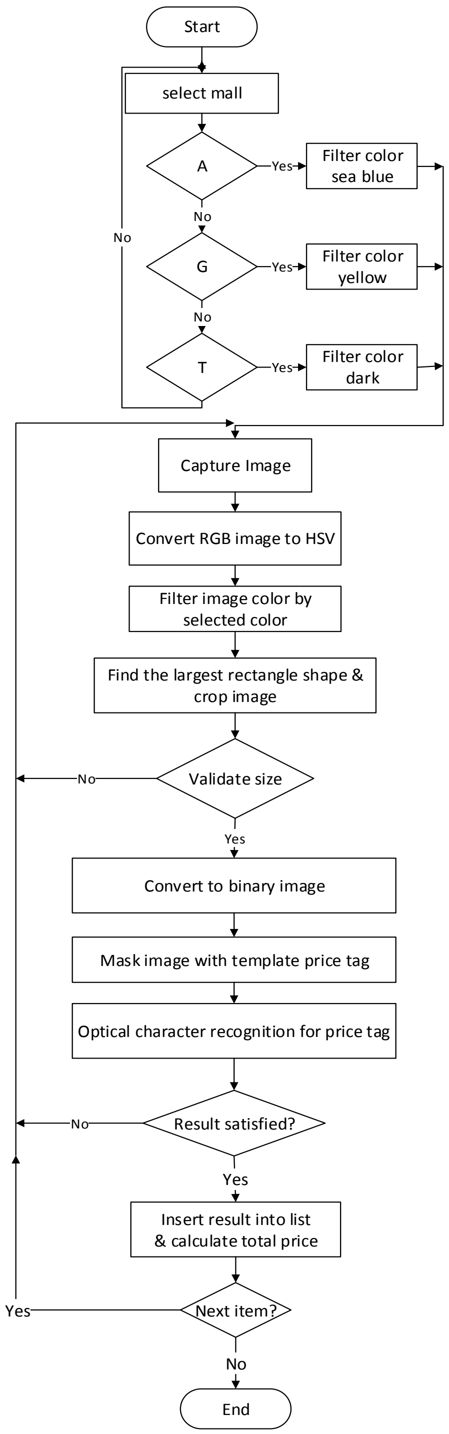


Figure 2: Image processing engine pipeline

B. *Price Tags Features*

Every shopping mall has different price tag design, color and font size. Besides that, most price tags are standardized within a mall except for sale season where the price tag is different. Author only deals with standard price tag from three different shopping malls namely as A, G and T.

There are some common attributes among the price tags for example, each price tag contains product name, price and barcode. The most important part are product name and price. The barcode and other codes would be ignored as it is not necessary for this project. The standard price tag has standard location for price and product name. However, it is different between malls. By taking the advantage of these features, mask image is used to get the location of the price and product name by using predefined mask. Another feature that exist in the price tag is color and rectangle shape. Color and rectangle shape can be used to detect the location of price tag in the image. Figure 3 to Figure 5 show three difference types of price tags from shopping malls A, G and T respectively.

Figure 3: Price tag of shopping mall A

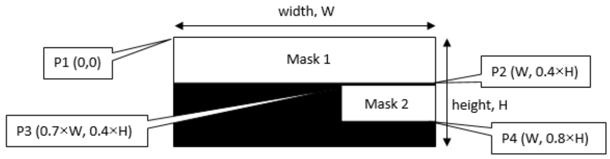
Figure 4: Price tag of shopping mall G

Figure 5: Price tag of shopping mall T

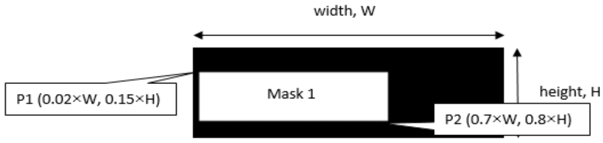
C. Price Tag Mask

The price tag masks are designed to get the desired part of the image which are price and product name. Since the image taken would give different size, ratio is used to scale the image. Ratio width to height is used to get the size of the price tag and set the point of mask.

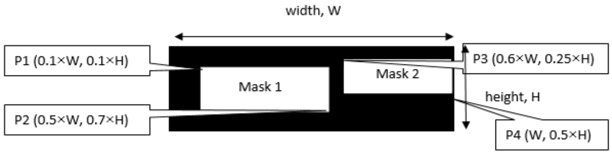
Price tag A has ratio width to height of 2:1. There are two rectangle masks would be drawn namely Mask 1 and Mask 2 as shown in Figure 6(a). Mask 1 has point P1 at the top-left and point P2 at bottom-right while Mask 2 has point P3 at top-left and point P4 at bottom-right. To get the point P1, P2, P3 and P4 from Figure 6(a), a mathematical method is used to calculate the points based on ratio. In the computer vision, point P1 is located at (0,0) in Cartesian plane. Point P2 and P4 are calculated by measuring the height of the price tag to the height of region to mask. Point P3 is calculated by measuring the width of price tag to the width of mask region. It is approximate 40% or 0.4 of height in y-direction for point P2. Thus, point P2 is located at (W, 0.4×H). Point P3 is located approximate 70% or 0.7 of width and 40% or 0.4 of height. Thus, point P3 is located at (0.7×W, 0.4×H). Point P4 is located approximate 80% or 0.8 of height, which is (W, 0.8×H). Price tag G has a ratio of width to height of 3:1 as shown in Figure 6(b). Price tag G only has one mask area with two points which is point P1 at top-left and P2 at bottom-right. Point P1 is approximate 2% or 0.02 from the width and 15% or 0.15 from the height of price tag. Thus, the point P1 is located at (0.02×W, 0.15×H). Point P2 is located at 70% or 0.7 from the width and 80% or 0.8 of the height. So, the point P2 is located at (0.7×W, 0.8×H). Figure 6(a) to 6(c) shows the example of price tag mask from shopping mall A, G and T respectively. Table 1 shows the summary of points for price tag masks.



(a) Price tag A



(b) Price tag G



(c) Price tag T

Figure 6: Price tag mask for shopping mall A, G and T

Table 1
Summary of Price Tag Masks

Price Tag	Ratio (W:H)	Mask 1		Mask 2	
		P1	P2	P3	P4
A	2:1	(0,0)	(W, 0.8 × H)	(0.7× W, 0.4×H)	(W, 0.8×H)
G	3:1	(0.02×W, 0.15×H)	(0.7×W, 0.8×H)	-	-
T	3:1	(0.1×W, 0.1×H)	(0.5×W, 0.7×H)	(0.6×W, 0.25×H)	(W, 0.5×H)

IV. Result And Discussion

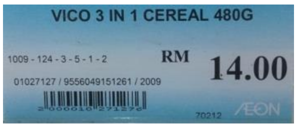
Several images of price tags have been used to select the best criteria of HSV values to detect the color of price tag for shopping mall A, G and T. By using these criteria of the filter, images were tested from various conditions such as in bright and dark lighting.

A. HSV Color Filter Criteria

Both price tags from shopping mall A, G and T were used to get the range of hue (color), the minimum value of saturation and brightness. These sample images were manually cropped and converted to HSV color space to get the HSV plot.

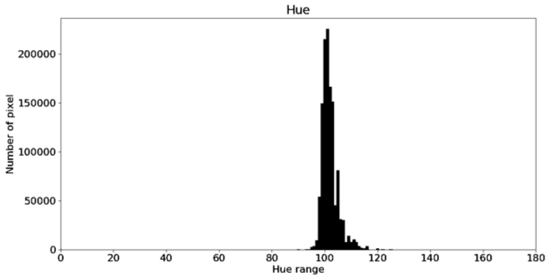
1) Criteria of Price Tag for Shopping Mall A

Five images of price tag from shopping mall A were used to get the range and minimum values of HSV from the plot. Price tag A consist of two color which is blue at top and bottom, and fade blue and almost white in the middle. The middle region of price tag A is a very critical part as it would give problem for further process. The middle area would give false results to the contour detector as white area would be the same color for the background. Figure 7 (a), (b), (c), (d) and (e) shows the examples of image and HSV plotted for price tag A. Table 2 shows the summary of the HSV range from price tag A.

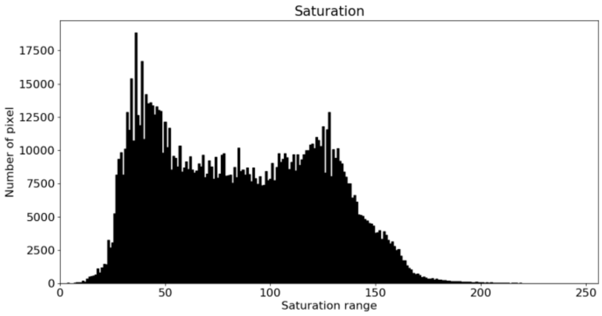


(a) Original image

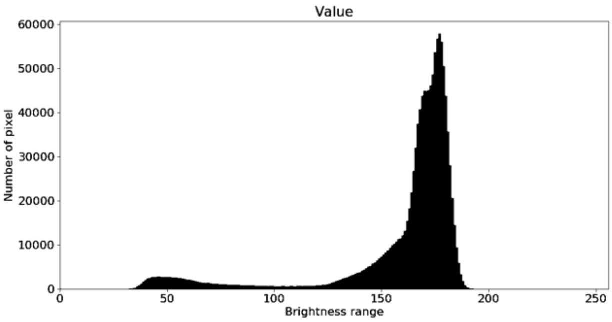
(b) HSV image



(c) Hue range of blue color



(d) Saturation range of gray level



(e) Brightness range

Figure 7: HSV plots for price tag A.

Table 2
HSV Values for Price Tag A

# pic	Min Hue	Max Hue	Min Saturation	Min Value (brightness)
1	87	118	10	40
2	90	118	15	35
3	92	120	15	40
4	90	120	5	50
5	90	118	10	45
HSV filter criteria	**89.8**	**118.8**	**5**	**35**

The characteristics of a blue color filter are described in the Table 2. The average minimum and maximum values for hue were 89.8 and 118.8 respectively. Maximum value for saturation and value (brightness) were fixed to 255 to get the maximum brightness and saturated blue color in bright condition.

However, minimum saturation is needed as lighting condition changes slightly. The minimum saturation and brightness were 5 and 35 accordingly. Minimum saturation cannot set to 0 as the filter would filter unnecessary pixels such as white noise. Minimum saturation was used to fill the white region in the middle of the price tag to cover the price tag area. Otherwise, the price tag region would split into two parts. Thus, further analysis cannot be done as a price tag ratio does not match with price tag mask.

2) Criteria of Price Tag for Shopping Mall G

Four images from difference illumination were used form price tag G to get the range of hue and minimum values of saturation and brightness. The background color for price tag G is yellow. So, there is not much variation in color. Figure 8 (a), (b), (c), (d) and (e) show the examples of image and HSV plotted for price tag G.

Table 3 shows the summary of the average range for hue color and minimum values of saturation and brightness. The average of the hue range for yellow is between 22 and 31.25 respectively. The minimum saturation is 7 and minimum brightness is 32 based on four sample images. These criteria were used to filter yellow color with difference illumination.

Table 3
HSV Values for Price Tag G

# pic	Min Hue	Max Hue	Min Saturation	Min Value (brightness)
1	21	33	59	50
2	22	30	60	52
3	23	32	7	32
4	22	30	50	50
HSV filter criteria	**22**	**31.25**	**7**	**32**

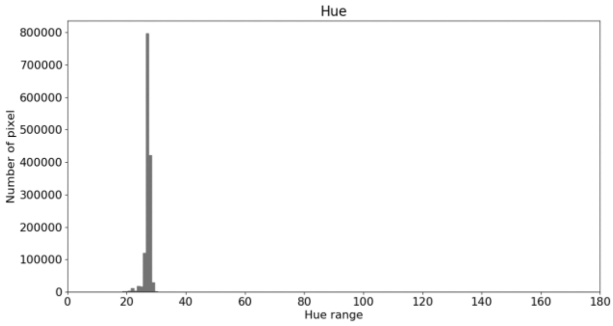
1) Criteria of Price Tag for Shopping Mall T

Four images of price tag T were used to obtain the range of hue, saturation and brightness criteria for HSV filter. The images were manually cropped so that the interested color range can be verified. In this case, dark blue color which is the border of the price tag is considering as ROI. However, the blue border only covers the price region while the product name with white as shown in the original image in Figure 9. To get the product name, author need to extend the points at top-right and bottom-right by 2 times the width of the border. Figure 9 shows the plots of HSV range for price tag T. Table 4 shows the summary of HSV criteria for price tag T.

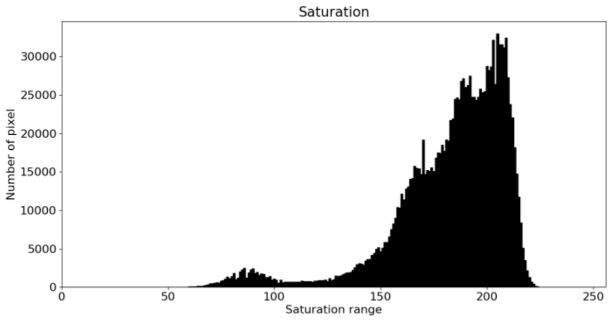
Table 4 shows the summary of the average range for hue color and minimum values of saturation and brightness for price tag T. The average range of hue is around 98.25 to 120.5 while minimum saturation is about 5 and minimum brightness is 45. This is acriteria HSV filter for price tag T.

(a) Original image

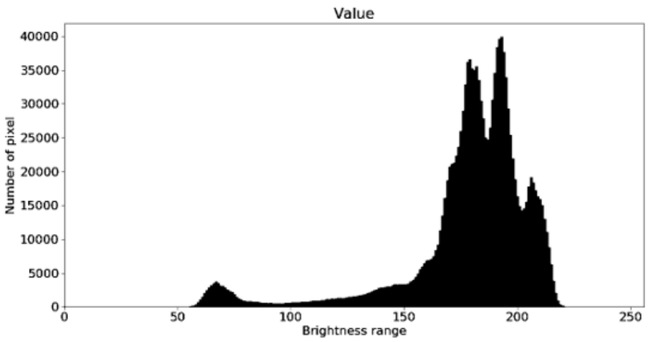
(b) HSV image



(c) Hue range for yellow color



(d) Saturation range of gray level

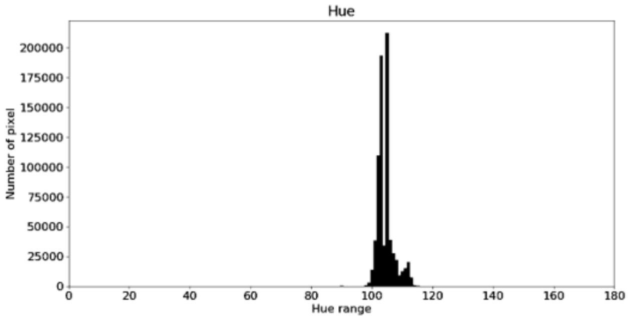


(e) Brightness range

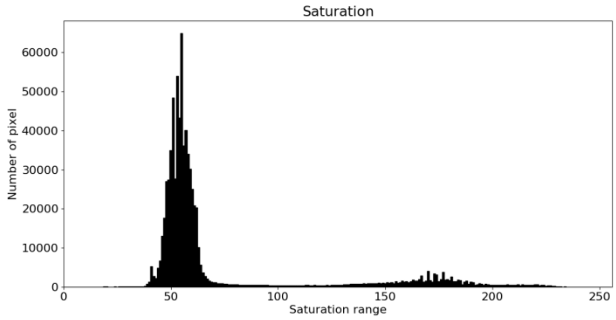
Figure 8: HSV plots for price tag G

(a) Original image

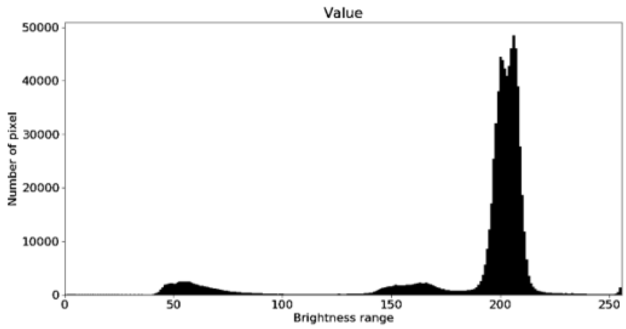
(b) HSV image



(c) Hue range for yellow color



(d) Saturation range of gray level



(e) Brightness range

Figure 9: HSV plots for price tag T

Table 4
HSV Values for Price Tag T

# pic	Min Hue	Max Hue	Min Saturation	Min Value
1	100	130	5	45
2	100	120	5	46
3	95	115	45	47
4	98	117	44	45
HSV Filter Criteria	**98.25**	**120.5**	**5**	**45**

B. *Result of Color Filter*

In this experiment, author separate blue color based on result from Table 2 where minimum and maximum hue is 89.8 and 118.8 accordingly. Minimum saturation and brightness were set to 5 and 35 accordingly so that the detector can cover the white area at the middle of price tag. The image was converted to HSV color space and filtered as shown in Figure 10 (a), (b) and (c).

(a) (b) (c)

Figure 10: (a) Original image. (b) HSV image. (c) Mask of the color filter

After the color filtering, author applies find largest rectangle contour that exists in the picture by using a mask image as shown in Figure 10 (c). The white area is a blue color that is filtered previously and consist some white noise. The white noise would affect the result of the contour detection and recognition of price tag. In order to reduce the noise, median filter with a kernel size of 9 by 9 was applied to the mask image.

When the largest rectangle contour is detected, the detection area on the original image is cropped and the result is shown in the Figure 11. The cropped image is converted to binary image by applying an adaptive threshold. The result is shown in the Figure 12.

Figure 11: Cropped price tag

Figure 12: Binary image of cropped price tag

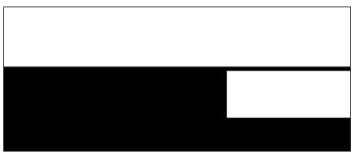


Figure 13: Price tag mask

Then, price tag mask in the Figure 13 is applied to the binary image from Figure 11. The result is shown in Figure 14 for product name and Figure 15 for the price of the price tag. The OCR was applied two times, which was for the product name (Figure 14) and price (Figure 15). This was due to the accuracy of the OCR drop slightly as both regions have a different font size. The result of OCR is shown in Figure 16.

Figure 14: Cropped product name from price tag.

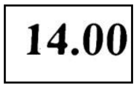


Figure 15: Cropped price from price tag.

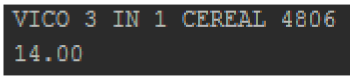


Figure 16: Result of OCR

Figure 17 (a) to (h) show the operation of image processing engine and optical character recognition on price tag G and T respectively.

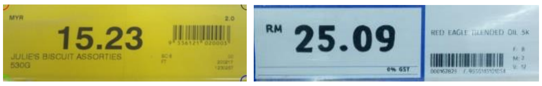
C. *Accuracy of Detection*

Accuracy of detection for the price tag is a percentage of pass over the total sample test to detect the region of price tag. This test is not included accuracy of OCR as only applied OCR Tesseract Library in this project. Price tag A has a percentage pass rate of 59.4% where only 19 sample pictures of price tag out of 32 able to recognize the region of the price tag. While 40.6% or 13 out of 32 were failed to recognize the price tag as the detector do not get the largest rectangle in the image. The failed image also has the problem such as region of interest are split into two parts.

Price tag G and T show the pass rate above 90% with an accuracy of detection of 92.3% or 12 and 90.63% or 29 sample pictures passed. However, only one sample picture was failed for price tag G due to price tag do not follow the standard size. There were three sample images of price tag T failed to recognize. This was due to white noise covers the edge of the mask image, flash appear during capture, background same color as price tag and ratio of mask image was out of range. Figure 18 shows the bar chart of percentage pass and failed for 3 difference price tag. Table 5 shows the summary of the accuracy of price tag detector based on three types of price tag.

(a) Original images, HSV and mask of color filter images of mall G

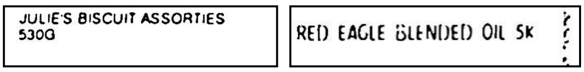
(b) Original images, HSV and mask of color filter images of mall T



(c) Cropped image of price tag G and T

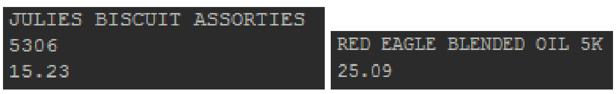
(d) Binary images of price tag G an T

(e) Price tag masks for G and T



(f) Cropped product name form price tag G and T

(g) Cropped price form price tag G and T



(h) Result of OCR

Figure 17: Operation of image processing engine and optical character recognition on price tag G and T

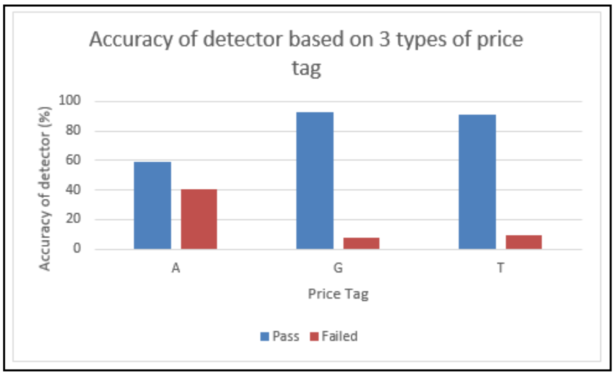


Figure 18: Accuracy of detection based on 3 types of price tags

Table 5
Accuracy of Price Tag Detector

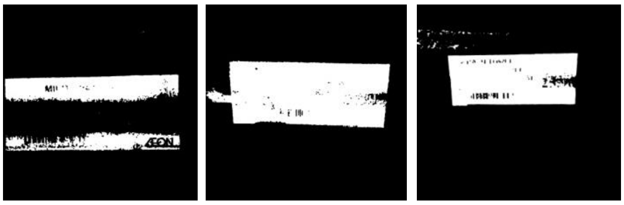
Price Tag	Total Sample	Pass Sample	Failed Sample	Pass Rate	Failure Rate
A	32	19	13	59.4%	40.6%
G	13	12	1	92.3%	7.7%
T	32	29	3	90.63%	9.7%

The accuracy of detection for price tag A was very low which was about 59.4%. This was due to several problems occur in the image such as price tag color at the middle was almost white instead of fade blue and reflection of the light at the price tag region. Even though the HSV filter criteria were already set to minimum values, the detector failed to find the largest rectangle in the image as the mask image after color filter was split into two parts as shown in Figure 19 (a) and the white noise present around the price tag region as shown in Figure 19 (b) and (c). Another problem was the detection area was not perfectly rectangle and the budge around the rectangle as shown in Figure 19 (b) and (c). These images condition cannot be avoided as the original images were already having a problem. The rectangle contour approximation was unable to find the approximate rectangle as the image consists lots of white noise. Even though the median filtered was already applied before finding the contour.

Author used a small number of samples for price tag G as the background color only consists of one color. So, there was not much variation in color range. Only one sample of price tag failed to detect as the position of price and product name were not followed the standard price tag. Thus, the masking image produces the false result and the character cannot be recognized. So, the sample image failed. Price tag T also consists of defect images during capture the image. The defect was due to excessive white noise around the price tag mask and existing reflection of light on the price tag as shown in Figure 20 (a), (b) and result of the price tag mask in Figure 20 (c) respectively.

(a) (b) (c)

Figure 19: Example of failed sample image.



(a) (b) (c)

Figure 20: Example of mask for failed sample image.

V. CONCLUSION

In this paper, two types of image processing techniques were used to detect price tag which was color and contour detection. Color detection was able to find the ROI which was a price tag by using the HSV color space. HSV color space can be used to filter color with different variation such as blue and fade blue in price tag A by setting the range of hue, minimum brightness and saturation value approach 9 and 35 respectively. Contour detection also able to find the largest rectangle that assumes as price tag and extract the four vertices points for further process. However, contour detection was not immune to white and black noises as it would give different results of the detection. Thus, median filter was needed to remove the noises. Moreover, OCR also able to recognize the text of price and product name to keep track item purchased by performing in pipeline such as recognize the text of price first and then followed by product name. The accuracy of OCR drops significantly when there were two fonts in different size and images taken more than 20cm in the distance. The accuracy of the detector for price tag G was almost perfect which was 92.3% and price tag T with 90.63% while price tag A was 59.4%.

ACKNOWLEDGMENT

This work is supported by the Universiti Teknologi Malaysia (UTM) under the Research University Grant (GUP) (Q.J130000.2623.12J51).

REFERENCES

[1] J. M. Guerrero, G. Pajares M. Montalvo, J. Romeo and M. Guijarro, "Support Vector ,Machines for crop/weeds identification in maize fields," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11149-11155, 2012.

[2] G. R. Bradski., "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, vol. 2, pp. 12-21, 1998.

[3] E. Hamuda, B. Mc Ginley, M. Glavin and E. Jones "Automatic crop detection under field conditions using the HSV colour space and morphological operations," *Computers and Electronics in Agriculture*, vol. 133, pp. 97-107, 2017.

[4] K. B. Shaik, P. Ganesan, V. Kalist, B. S. Sathish and J. M. M. Jenitha, "Comparative study of skin color detection and segmentation in HSV and YCbCr color space," *Procedia Computer Science*, vol. 57, pp. 41-48, 2015.

[5] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898-916, 2011.

[6] P. Kapsalas, K. Rapantzikos, A. Sofou and Y. Avrithis, "Region of interest for accurate object detection," in *2008 International Workshop on Content-Based Multimedia Indexing*, London, 2008, pp. 147-154.

[7] P. Forczmański and A. Markiewicz, "Two-stage approach to extracting visual objects from paper documents," *Machine Vision and Applications*, vol 27, no. 8, pp. 1243-1257, 2016.

[8] R. Smith and Z. Podobny, Tesseract Open Source OCR Engine (main repository), 2017.