Journal of Engineering Technology and Applied Physics

Fuzzy Logic Weighted Averaging Algorithm for Malaysian Banknotes Reader Featuring Counterfeit Detection

Turki Khaled Al-Hila, Wai Kit Wong*, Thu Soe Min and Eng Kiong Wong
Faculty of Engineering and Technology, Multimedia University, 75450 Jln Ayer Keroh Lama, Malaysia.
**Corresponding author:* wkwong@mmu.edu.my, *ORCiD:* 0000-0003-1477-8449
https://doi.org/10.33093/jetap.2023.5.2.3
Manuscript Received: 27 February 2023, Accepted: 28 March 2023, Published: 15 September 2023

***Abstract* — This paper proposed a novel fuzzy logic weighted averaging (FLWA) algorithm in image processing techniques to detect counterfeit Malaysian banknotes. Image acquisition techniques on banknote position detection and re-adjustment, image pre-processing techniques, feature extraction methods on Malaysian banknotes' watermarks are also covered in the paper. The FLWA Algorithm has the advantage of a much simpler model since it is a human guidance learning algorithm that does not require enrolment process to get the specific weights for each security feature. Each security feature is treated with equal weight. The experimental results also shown that FLWA model also outperform the MobileNet model and VGG16 model in Malaysian banknotes' counterfeit detection. It has a distinct advantage over earlier or current banknote counterfeit detection techniques in that it adopted the known watermarks features, with known machine learning techniques to identify real Malaysian banknotes and detect those counterfeit Malaysian banknotes.**

Keywords— Image Processing, Fuzzy Logic, Malaysian Banknotes, Banknote Reader, Counterfeit Detection.

I. INTRODUCTION

Due to the acceptance and quick adaption by forgers with highly advanced technology, counterfeit banknotes are a severe issue worldwide. Therefore, one of the most efficient ways to eliminate banknote forging is to employ a lot of the readily available, accurate, affordable, and reliable counterfeit detecting equipment and software.

Banknotes, especially the Malaysian banknotes are made available with watermarks to indicate the validity of a legal banknote. However, majority of these banknotes' watermarks are created for the users' visual judgement. Visually impaired person is facing difficulty in reading them, especially on identifying the real and fake for a banknote they received in hand. Due to their low eyesight or lack of vision when reading the receiving banknotes, visually impaired ethnic groups are the most often victims of receiving counterfeit currency. As a result, image processing-based counterfeit banknote detection readers are required to assist visually impaired individuals handling banknotes in reducing their losses.

As a rapidly expanding technology, image processing has had an impact on numerous enterprises and industries. Image processing techniques had shown competent applications in banknotes counterfeit detection too [1 - 5]. Image processing can extract important information from an image, and it is a type of signal processing that takes an image as input and outputs the image or its features as results. Practically, the image features from the watermarks that captured from the banknotes can also be extracted as information to allow the banknote reader to decide the real/fake of a checked banknote.

Since the features extracted from the watermarks are uncertain information, fuzzy logic [6, 7] is the best technique for representing and manipulating them. This paper investigated fuzzy logic and image processing techniques used in Malaysian banknote

Published by MMU PRESS. URL: https://journals.mmupress.com/index.php/jetap/index

counterfeit detection algorithms. It proposed a novel Fuzzy Logic based Weighted Averaging algorithm (FLWA) for detecting counterfeit Malaysian banknotes. The FLWA algorithm is compared to TWO cutting-edge parallel techniques (MobileNet model using RMSprop Loss Function [8] at TensorFlow's Keras API and VGG16 model using 2D Convolution Layer [9] at TensorFlow's Keras API). The experimental results demonstrate that FLWA outperforms the two parallel approaches in terms of accuracy, processing speed, and complexity. The remaining sections of the study are divided into four primary groups. The review of the relevant state of the art is presented in Section II, the proposed Fuzzy Logic Weighted Averaging Algorithm for Malaysian banknotes reader featuring counterfeit detection is shown in Section III, the performance analysis and results are presented in Section IV, and the research is concluded in Section V.

II. LITERATURE REVIEW & RELATED WORK

An efficient and quick image processing technique is required for the Malaysian banknotes counterfeit detector to function properly and allow the visually impaired person to read the real and the fake banknotes while engaging in business operations. According to the two subsections below, there are two types of image processing techniques currently used: supervised learning and human guided learning.

A. Supervised Learning Image Processing Techniques

Supervised learning image processing approaches are a sort of machine learning model for image processing problems when the available data consists of labelled examples. The labelled training image data, which consists of several training image instances, will be utilized by the model to infer a function [10]. A pair of input image objects (often vectors) and a desired output image value make up an example approach in the supervised learning image processing technology (also named the supervisory output). A supervised learning image-processing method analyses the training picture data to produce an inferred function that is then used to map fresh image occurrences. The following methods for related Supervised Learning image processing banknote counterfeit detection have been researched:

(i) Deep Learning Image Processing Based Banknote Counterfeit Detection

The study [11] suggests a method that may be implemented on accessible devices like smart phones and tabletsI in order for normal people to identify whether the banknotes they are in possession of is real or fake. Convolutional Neural Networks (CNN) were developed to discern between uploaded photographs that were genuine and those that were not. On a dataset that was created by the model developer, it was trained and tested. Utilizing a smartphone's camera, pictures are taken and sent to the CNN network. The outcomes are promising and can be improved with more study and adjustments to the Deep CNN model's design. The problem with this technique is that there are no regions of interest that incorporate security properties; instead, the classification of real and counterfeit banknotes depends on the dataset that has been trained.

In [12], the author suggests an end-to-end learning model built on Adapnet++ that controls feature extraction from the network (AdapNet++) and the eASPP model at various scale levels. The experiment was run utilising a camera-based dataset of entire images of Euro banknotes, allowing it to extract the most important elements at various scales. The network also receives basic input images; it doesn't need high-quality images, and it runs in uncontrolled circumstances. The network automatically learns which areas of interest best predict outcomes. The construction was sturdy, but since real banknotes include security characteristics and fraudulent ones don't, it is necessary to use a tool like a tilt or shooting light to observe the security features. Though image processing techniques have been extensively used in colour processing, the printing equipment has improved, and it will be challenging for that system to distinguish between genuine and false notes in the future.

The scientists of [8] observed that earlier scientists had developed banknote identification systems utilising deep learning AI techniques like CNN and R-CNN. Despite the very little dataset utilised for training in these systems, it has been demonstrated that the accuracy of banknote recognition is lower. The existing market systems also do not involve its implementation or development using embedded systems. To assist CNN's architects in identifying fake notes, they had acquired a variety of Ethiopian banknotes with varied ages and states and utilised several optimisation techniques. Experimental analysis has been validated with numerous CNN models, including InceptionV3, MobileNetV2, XceptionNet, and ResNet50. MobileNetV2, an effective and reliable Ethiopian banknote detector that used the RMSProp optimisation technique and a batch size of 32, outperformed other CNN models in terms of accuracy. The chosen model MobileNetV2 with RMSProp optimisation has been implemented using an embedded platform using a Raspberry Pi 3 B+ and other add-on devices. Real-time fraudulent note recognition in a web-based user interface (UI) has also been recommended by the work.

(ii) K-Nearest Neighbors (k-NN) Image Processing Based Banknote Counterfeit Detection

Hlaing and Gopalakrishnan [13] had also performed research on the KNN (K-Nearest Neighbor) method of detecting the value of Burmese banknotes using textural characteristics Gray-Level Co-Occurrence Matrix (GLCM) with a value of K=1. Rahmad and the researchers in [14] went one step further and used the KNN approach to check the authenticity of Indonesian banknotes (K-Nearest

Neighbor). In order to provide good accuracy and accurately extract the value of banknotes in rupiah, the completed research developed a model of banknote value extraction using colour features and the KNN approach, with a K value larger than that used in the previous research by Hlaing and Gopalakrishnan [13]. They have devised a KNN algorithm that is the core of their system for identifying counterfeit currency and relies on image processing. It is envisaged that the algorithm of this system would be able to handle the problem of detecting counterfeit money and be used for large financial transactions.

Utilizing the KNN approach, the colour feature of the banknotes is extracted and identified (K-Nearest Neighbor). The colours provide important information and values when describing an object. This phenomenon was seen in [15] as well. The Red, Green, and Blue (RGB) values are utilised for feature extraction in the procedure for determining the value of banknotes, and the output of feature extraction is then used to classify the value of banknotes using the K-Nearest Neighbor (KNN) methodology. The Naive Bayes approach is one of many classification algorithms used [16]. However, the Naive Bayes method is predicated on probabilistic or conditional premises. The implicit belief that all attributes are independent of one another is Naive Bayes' main flaw. In actual life, it is uncommon to encounter a set of predictors that are completely unconnected to one another.

A Malaysian currency denomination system was designed using Matlab software by researchers from Malaysia's UTM [17, 18]. Three features (RB, RG, and GB) were obtained from the RGB values of six different classes of banknotes (RM1, RM5, RM10, RM20, RM50, and RM100) for the k-Nearest Neighbors (k-NN) and Decision Tree Classifier (DTC) classification algorithms. The optimised k-NN and DTC were selected using ten-fold cross validation based on the smallest cross validation loss. A confusion matrix was then utilised to demonstrate how effectively the ideal DTC and k-NN models functioned. According to the results, the suggested k-NN and DTC model was able to reach 99.7% accuracy, with the performance of the RM50 class drastically decreasing. The simulated system, however, is only capable of detecting the denomination of Malaysian banknotes; counterfeit banknotes are not detectable.

(iii) Support Vector Machine (SVM) Image Processing Based Banknote Counterfeit Detection

An Indian counterfeit banknote detection system that can classify Indian currency denominations was proposed by Gopane and Kotecha [19]. Support Vector Machines and an image processing technique are combined in the proposed method to detect fraudulent Indian banknotes. The amount of training data currently available is insufficient, and the research lacks the rigors testing in a variety of real-world settings required for a comprehensive verification of the suggested approach (only 12 genuine and 12 fake banknotes).

A method for spotting counterfeit money based on multiple kernel support vector machines was proposed by Yeh *et al.* [20]. A support vector machine (SVM) is created to lower false rates. The luminance histograms of each partition on each banknote, which are divided into sections, serve as the system's input. There is a distinct set of kernels associated to each partition. A linearly weighted combination method is used to combine multiple kernels into a combined matrix. The best weights for kernel matrices can be discovered via semi-definite programming (SDP) learning. The employment of two strategies reduces the time and space requirements of the SDP method. In one method, the kernel weights are assumed to be positive, whereas in the other, the weights' sum is set to one. Experiments on Taiwanese banknotes show that the proposed approach outperforms single-kernel SVMs, conventional SVMs with SDP, and multiple-SVM classifiers. Only 50 genuine and 18 fraudulent banknotes are currently available for training, and the testing data is insufficient (only 20 genuine and 11 fake banknotes).

An algorithm for spotting counterfeit Peruvian currency was presented by Huavtalla *et al.* [21]. The method uses digital image processing and support vector machines to automatically diagnose problems (SVM). According to the researchers, earlier Peruvian systems for spotting fakes were made expressly to look at relevant features in dollars and euros. However, some counterfeiters in Peru managed to trick those anti-counterfeiting mechanisms. As part of the development of the detecting system, the intaglio marks etched on the banknotes were segmented and picture capture was enhanced. After segmentation, they applied embossing and Sobel filters, followed by an aperture morphological operation to produce distinctive features, which were then categorised by an SVM. The methodology was validated using a dataset of 240 samples provided by the Central Reserve Bank of Peru (BCRP), which had an accuracy of 96.5 %.

B. Human Guided Learning Image Processing Techniques

Subject matter experts can speed up the learning process by giving real-time instructions to the technology through a process called human-guided machine learning. For instance, if the machine learning model comes across any data that it is uncertain about, a person can be prompted to weigh in and offer comments. The model then combines the knowledge it has gained from this input to create a more accurate prediction for the future.

Humans do get involved when it is necessary for training or validation. In order to start from scratch, human-guided machine learning initially uses algorithms to do the time-consuming task of finding relationships within the data. This implies that the amount of time a human must spend carrying out a particular task will inevitably increase as machine learning accuracy grows. Following are some relevant humans led learning image processing-based banknote identification techniques that have been researched:

(i) Fuzzy Logic based Perceptual Image Hashing Algorithm

Wong *et al.* [22] proposed a Malaysian banknote recognition system employing image processing technologies and fuzzy logic algorithm for the blind and visually impaired. The Malaysian banknote reader will take a picture of the inserted banknote before sending it over Wi-Fi to the cloud server for image analysis. The cloud server is set up to accept banknote image transmissions from the banknote reader, process them using fuzzy logic and image-searching techniques based on perceptual hashing, and then send the value of the found banknotes back to the banknote reader. An audio message is played on a tiny speaker that is attached to the gadget to assist persons who are blind or visually impaired in understanding the results of the banknote reader.

The algorithm has been put through several experiment tests to find the optimal parameters to include and to evaluate the algorithm's effectiveness in that situation. The algorithm's performance is primarily influenced by the threshold setting, T. As a result, the environment influences the T-value setting, which is then controlled by people (background lighting). T must consequently be manually reset by a human operator each time the environment of the system changes. The reader will filter most of the database images during matching if T is set too high. If T is set too low, it will be hard to determine the banknote value because it will be discovered that numerous database photographs resemble the input image. This will cause a sharp increase in the banknote detection system failure rate since it will become increasingly difficult to determine the correct banknote value.

(ii) Comparison Between Color SIFT and Gray Scale SIFT Algorithms

Contrary to the prevalent current approaches, Doush and Btoush [15] presented the colour SIFT approach with the goal of merging colour and local object features descriptor (gray SIFT). The findings make it abundantly evident that there are more features discovered in the colour SIFT than in the grey SIFT. The human operator will update the algorithm's region of interests (ROIs) for the captured Jordanian banknotes in order to provide guidance for the best descriptor creation. The evaluation's findings indicate that colour SIFT descriptors perform very well when compared to grey SIFT descriptors in terms of processing speed and accuracy level.

(iii) Bangladeshi Banknotes Detection Using OCR etc. Methods

A fundamental piece of software that can identify fake Bangladeshi currency from photos was presented by Ahmed *et al.* [1]. The existing features of banknotes, such as micro-printing, optically variable ink (OVI), watermark, iridescent ink, security thread, and ultraviolet lines, are extracted using OpenCV's OCR (Optical Character Recognition), Contour Analysis, Face Recognition, Speeded UP Robust Features (SURF), and Canny Edge & Hough transformation algorithm. The algorithm's software tuning, which includes the Adaptive Threshold block size, Canny Threshold and Threshold Linking, Eigen Threshold, Brightness, Contrast, Sharpness, Zoom Scale, Input image DPI, and Input image format, among other factors, needs to be updated by a human operator to guide for various modes of banknotes counterfeit detection (UV, digital spectrum lighting etc.) Two measures are utilised to evaluate the success rate of this software: accuracy and speed. The study also discusses the benefits and cons of implementation elements that may have an impact on how well paper currency authentication systems using image processing function.

In summarising the literature review and related works, it is highlighted that the government actually creates the majority of banknotes' watermarks so that people with good vision may observe them and tell the difference between genuine and fake notes. Therefore, it is clear that visible light imaging technology can assist visually impaired people in capturing what a person with normal vision can see.

Since banknotes watermarks are different in kinds and alterable from time to time by the government during new series of banknotes be published with new security features being added, this research will consider the human guidance learning image processing technique and will also be based on the idea of obtaining components with a reasonable price and tolerable process since the Malaysian banknotes counterfeit reader must be able to react quickly and with a high or reasonable level of accuracy if visually impaired people are to use it for handling banknotes in their everyday business activities.

III. FUZZY LOGIC WEIGHTED AVERAGING ALGORITHM FOR MALAYSIAN BANKNOTES READER

This section discussed the research methodology on the proposed algorithm for Malaysian banknotes reader, including the feature extraction method for the eight targeted watermarks on Ringgit and the entire process of the Fuzzy Logic-based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection Algorithm.

A. Feature Extraction on Malaysian Banknotes

Based on the studies done on [23], there are eight security features being selected for the proposed Malaysian banknotes reader featuring counterfeit detection and their targeted Malaysian banknotes series are also listed in Table I. Image pre-processing techniques and feature extraction methods used in FLWA algorithm will be discussed in detail as below.

(i) Image Prepossesses:

To ease up the reader understanding, the Banknote image acquisition step will be explained in detail in Section III B for the Fuzzy Logic-based Weighted

Averaging Malaysian Banknote Counterfeit Detection Algorithm (FLWA). This subsection will discuss the detail image preprocesses that will be called by FLWA the later section. The image prepossesses divided into two steps:

Table I: Chosen Security Features on Malaysian Banknotes.

No.	Security Feature	Notes					
		RM1	RM5	RM10	RM20	RM50	RM100
1.	Perfect See-through Register	✓	✓	✓	✓	✓	✓
2.	Non-transparent Window	✓	✓				
3.	Shadow Image	✓	✓				
4.	Watermark Portrait			✓	✓	✓	✓
5.	Color Shifting security thread			✓	✓	✓	
6.	Colored Glossy Patch			✓	✓		✓
7.	Two Color Fluorescent Element	✓	✓	✓	✓	✓	✓
8.	Text and Logo	✓	✓	✓	✓	✓	✓

Step 1: **Resize image:** Certain images capture by the imaging tool and pass to the image processing tasks are in different sizes, these images be standardized in size. Resize all input images (Flat_FB_Flight, Flat_FB_Blight, Flat_FB_UVlight, Tilt_FB_Flight, Flat_BB_Flight, Flat_BB_Blight, Flat_BB_UVlight and Tilt_BB_Flight) to standard size images using the below equation:

$$input\ image = Resize(width, height, no.RGBchannel) \tag{1}$$

where FV_Flight_FP is the banknote's front view image captured with front light and placed in flat position.

FV_Blight_FP is the banknote front view image captured with back light and placed in flat position.

FV_UVlight_FP is the banknote's front view image captured with UV light and placed in flat position.

FV_ Flight_TP is the banknote's front view image captured with front light and placed in tilt position.

BV_Flight_FP is the banknote's back view image captured with front light and placed in flat position.

BV_Blight_FP is the banknote's back view image captured with back light and placed in flat position.

BV_UVlight_FP is the banknote's back view image captured with UV light and placed in flat position.

BV_ Flight_TP is the banknote's back view image captured with front light and placed in tilt position.

width is the dedicated reduced image width

height is the dedicated reduced image height

no.RGBchannel is number of Red, Green and Blue color channels, which is 3 in this case.

Step 2: **Remove image noise:** HSV color space detection is first used to detect the security features and noise will be removed with morphological process. the HSV value for each series of banknotes is fixed on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$, $Val_{(max)}$. These values will be determined in experimental results as shown in Section IV. By applying Morphological Transformations [24] based on closing for each banknote, the value of the defined kernel is fixed and will show in experimental results Section IV. Kernel is the structuring element used for image erosion [25].

(ii) Shadow Image Detection:

The Shadow Image watermark was used on RM1 and RM5. On crisp logic, first, the threshold values are fixed in HSV color space according to the banknote reader box internal environment and the backlight intensity. Second, the noise object will be removed Using Morphological Transformations method. After that, Search for the biggest and brightest/whitest bounded object, mark it as Shadow Image on image "Flat_FB_Blight". Noise object exclusion and check if the total pixels within the bounded area of the Shadow Image:

$$TP_{SI} \geq P_{SI} \times TR \tag{2}$$

where P_{SI} = Percentage of Shadow Image area in a Malaysian Banknote.

TR = Total Pixels in the Resized Image converted as shown in Eq. (1).

IF condition in Eq. (2) is NOT FULFILLED,

THEN the identified "Shadow Image" object is a noise object. Output: "Shadow Image is not detected."

ELSE IF condition in Eq. (2) is FULFILLED, THEN the identified “Shadow Image” object is possible watermark, proceed the following:

Bounding Box measurement: Assign H_{SI} as the height of the shadow image and W_{SI} as the width of the shadow image at bounding box as shown in Fig. 1. Measure shadow image bounding box’s height to width:

$$Ratio = H_{SI}/W_{SI} \quad (3)$$

IF $Th_{SI(min)} < H_{SI} / W_{SI} < Th_{SI(max)}$, For RM1
THEN Output: shadow image for RM1 is detected.
ELSE IF $Th_{SI(min)} < H_{SI} / W_{SI} < Th_{SI(max)}$, For RM5
THEN Output: shadow image for RM5 is detected.
ELSE Output: “shadow image is not detected.”

where $Th_{SI(min)}$ and $Th_{SI(max)}$ are the minimum and maximum threshold of “shadow image” height to width ratio, as shown in Fig. 1.

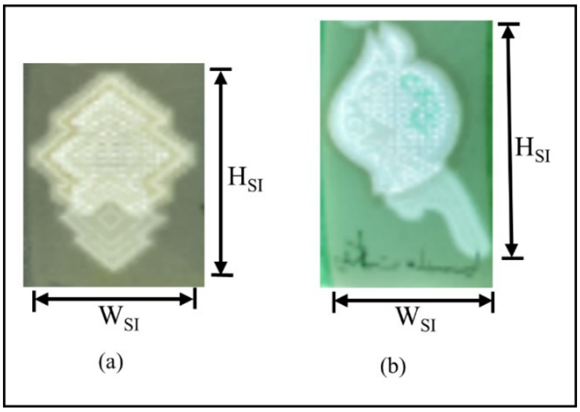


Fig. 1. Bounding Box (a) shadow image in RM1 and (b) shadow image in RM5.

Since ratio are uncertain information collected from different captured banknotes images, fuzzy logic can be used to handle them well. Fuzzification is a step to determine the degree to which an input data belongs to each of the appropriate fuzzy sets via the membership functions. The most common fuzzy membership functions are impulsive fuzzy membership function, triangular fuzzy membership function, trapezoidal fuzzy membership function and Gaussian fuzzy membership function [6].

Triangular fuzzy membership function [26, 27] is one of the most widely accepted and used membership function (MF) in fuzzy logic system design due to its *simplicity and linearity* in fuzzification approach. The triangle approach that fuzzifies the inputs can be defined with three significant parameters (minimum base, maximum base and height of the triangle). Figure 2 shows the Crisp set and Membership function of a fuzzy set. The fuzzy set approach normally extended to twice bandwidth of a crisp set. Based on this understanding, the watermarks feature membership functions formulation are done and shown below.

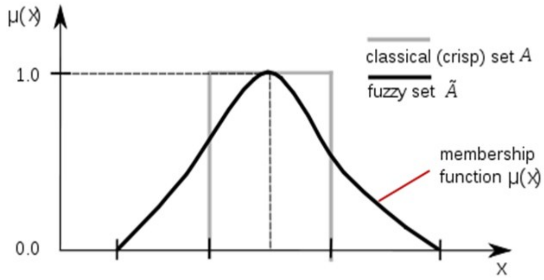


Fig. 2. Crisp set and Membership function of a fuzzy set.

A reasonable expression of the Shadow Image Detection fuzzy set in triangular membership function is required to formulate, SI_N: $Ratio \rightarrow [0,1]$:

$$SI_N(Ratio) = \begin{Bmatrix} 0 & if\ Ratio \leq a \\ \dfrac{Ratio - a}{b - a} & if\ a \leq Ratio \leq b \\ \dfrac{c - Ratio}{c - b} & if\ b \leq Ratio \leq c \\ 0 & if\ Ratio \geq c \end{Bmatrix} \quad (4)$$

where $Ratio = H_{SI}/W_{SI}$ is the shadow image bounding box’s height (H_{SI}) to width (W_{SI}).
N is the number of Ringgit value

$a = 2Th_{SI(min)} - \left(\frac{Th_{SI(max)} - Th_{SI(min)}}{2}\right)$

$b = \left(Th_{SI(min)} + \frac{Th_{SI(max)} - Th_{SI(min)}}{2}\right)$

$c = 2Th_{SI(max)} - \left(\frac{Th_{SI(max)} - Th_{SI(min)}}{2}\right)$

$Th_{SI(min)}$ and $Th_{SI(max)}$ are the minimum and maximum threshold of “shadow image” height to width ratio.

The corresponding triangular membership functions representation for Shadow Image Detection fuzzy set is shown in Fig. 3.

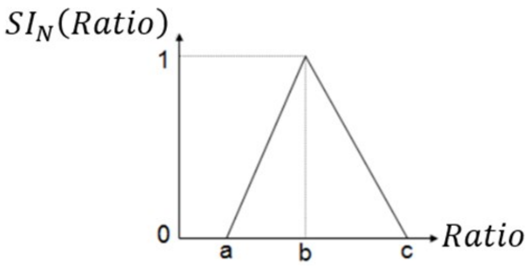


Fig. 3. Triangular membership functions representation for Shadow Image Detection fuzzy set.

(iii) Watermark Portrait Detection:

The Watermark Portrait was used on RM10, RM20, RM50 and RM100. On crisp logic, first, the noise object will be removed Using Morphological Transformations. Second, the threshold values are fixed in HSV color space according to the banknote reader box internal environment and the backlight intensity. The value of HSV will be in experimental results. After that, Search for the biggest and brightest/whitest bounded object, mark it as watermark portrait on image "Flat_FB_Blight". Noise object exclusion and check if the total pixels within the bounded area of the "watermark portrait and numeral". Lastly, detect the numeral in "watermark portrait" using PyTesseract on images "Flat_FB_Blight".

$$TP_{WP\_N} \geq P_{WP\_N} \times TR \quad (5)$$

where P_{WP_N} = Percentage of watermark portrait and numeral area in a Malaysian Banknote.
TR = Total Pixels in the Resized Image converted

IF condition in Eq. (5) is NOT FULFILLED,
THEN the identified "watermark portrait and numeral" object is a noise object. Output: watermark portrait and numeral are not detected.
ELSE IF condition in Eq. (5) is FULFILLED,
THEN the identified "watermark portrait and numeral" object is possible watermark, proceed to the following:

Bounding Box measurement: Assign H_{WP_N} as the height of the shadow image and W_{WP_N} as the width of the watermark portrait and numeral 10/ numeral 20 at bounding box as shown in Fig. 4. Measure watermark portrait and numeral bounding box's height to width:

$$\text{Ratio} = H_{WP\_N} / W_{WP\_N} \quad (6)$$

IF $Th_{WP\_N(min)} < H_{WP\_N} / W_{WP\_N} < Th_{WP\_N(max)}$,
IF numeral "1" AND numeral "0" for RM10 are detected
THEN Output: "watermark portrait is detected."
ELSE IF numeral "2" AND numeral "0" for RM20 are detected
THEN Output: "watermark portrait is detected."
ELSE IF numeral "5" AND numeral "0" for RM50 are detected
THEN Output: "watermark portrait is detected."
ELSE IF numeral "1" AND two numeral "0" for RM100 are detected.
THEN Output: "watermark portrait is detected."
ELSE Output: "watermark portrait is not detected."
(7)

where $Th_{WP\_N(min)}$ and $Th_{WP\_N(max)}$ are the minimum and maximum threshold of "watermark portrait and numeral" height to width ratio.

A reasonable expression of the Watermark Portrait Detection fuzzy set membership function is required, WP_N: *Ratio, x* $\rightarrow [0,1]$ and these relates multiple of fuzzy membership functions as below:

Bounding Box Ratio membership function:

$$\mu_N(Ratio) = \begin{cases} 0 & \text{if } Ratio \leq a \\ \dfrac{Ratio - a}{b - a} & \text{if } a \leq Ratio \leq b \\ \dfrac{c - Ratio}{c - b} & \text{if } b \leq Ratio \leq c \\ 0 & \text{if } Ratio \geq c \end{cases} \quad (8)$$

where $Ratio = H_{WP\_N} / W_{WP\_N}$ is the watermark portrait bounding box's height (H_{WP_N}) to width (W_{WP_N}).
N is the number of Ringgit value

$a = 2Th_{WP\_N(min)} - \left(\frac{Th_{WP\_N(max)} - Th_{WP\_N(min)}}{2}\right)$

$b = \left(Th_{WP\_N(min)} + \frac{Th_{WP\_N(max)} - Th_{WP\_N(min)}}{2}\right)$

$c = 2Th_{WP\_N(max)} - \left(\frac{Th_{WP\_N(max)} - Th_{WP\_N(min)}}{2}\right)$

$Th_{WP\_N(min)}$ and $Th_{WP\_N(max)}$ are the minimum and maximum threshold of "watermark portrait" height to width ratio.

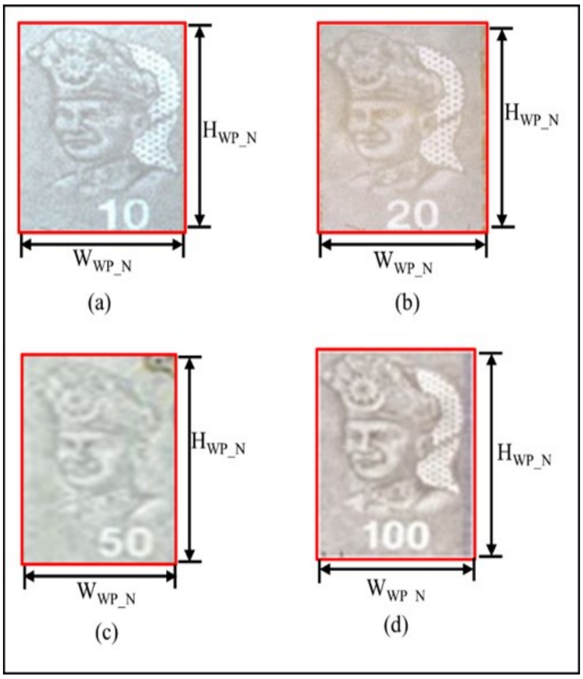


Fig. 4. Bounding Box (a) RM10, (b) RM20, (c) RM50 and (d) RM100.

Numeral "0" membership function:

$$\mu_{numeral0}(x) = \begin{cases} 1 & x = x_o \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_o is the numeral "0" being detected by PyTesseract.

Numeral "1" membership function:

$$\mu_{numeral1}(x) = \begin{cases} 1 & x = x_1 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_1 is the numeral "1" being detected by PyTesseract.

Numeral "2" membership function:

$$\mu_{numeral2}(x) = \begin{cases} 1 & x = x_2 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_2 is the numeral "2" being detected by PyTesseract.

Numeral "5" membership function:

$$\mu_{numeral5}(x) = \begin{cases} 1 & x = x_5 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_5 is the numeral "5" being detected by PyTesseract.

Two Numeral "0" membership function:

$$\mu_{numeral00}(x) = \begin{cases} 1 & x = x_{00} \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_{00} is the two numeral "0" being detected by PyTesseract.

$\mu_{numeral0}(x)$, $\mu_{numeral1}(x)$, $\mu_{numeral2}(x)$, $\mu_{numeral5}(x)$ and $\mu_{numeral00}(x)$ are impulsive fuzzy membership functions [28] because the detection of numeral relied on Pytesseract, hence it is either locate the particular numeral or otherwise.

Hence, based on the rules in Eq. (8), the Watermark Portrait Detection fuzzy set membership function is:

$$WP_N(Ratio, x) = \mu_N(Ratio)\big(\mu_{numeral1}(x) \times \mu_{numeral0}(x) + \mu_{numeral2}(x) \times \mu_{numeral0}(x) + \mu_{numeral5}(x) \times \mu_{numeral0}(x) + \mu_{numeral1}(x) \times \mu_{numeral00}(x)\big) \quad (9)$$

(iv) Perfect See-through Register Detection:

The Perfect See-through register watermark was used on RM1, RM5, RM10, RM20, RM50 and RM100. As shadow image was detected and watermark portrait was detected on image "Flat_FB_Blight". For RM1, RM5, RM10 and RM20, the portion numeric text of the perfect see-through register located at the top right side concerning the Y-axis symmetrical centerline by an area of ½ "Shadow Image" or "watermark portrait" horizontal length. For RM 100, the portion numeric text of the perfect see-through register located at the top side concerning the Y-axis symmetrical centerline "Shadow Image" or "watermark portrait" horizontal length as shown in Fig. 5. Next, detect the numeral for RM1, RM5, RM10, RM20 and RM100 in "Perfect See-through register using PyTesseract on images "Flat_FB_Blight" and "Flat_FB_Flight" which are the backlight and front light.

IF numeral "1" in Perfect See-through register for RM1 is detected at the backlight AND not detected at the front light, THEN Output: "Perfect See-through register" is detected.
IF numeral "5" in Perfect See-through register for RM5 is detected at the backlight AND not detected at the front light, THEN Output: "Perfect See-through register" is detected.
IF numeral "1" AND numeral "0" in Perfect See-through register for RM10 are detected at the backlight AND not detected at the front light, THEN Output: "Perfect See-through register" is detected.
IF numeral "2"AND numeral "0" in Perfect See-through register for RM20 are detected at the backlight AND not detected at the front light, THEN Output: "Perfect See-through register" is detected.
IF numeral "1"AND two numeral "0" in Perfect See-through register for RM100 is detected at the backlight AND not detected at the front light, THEN Output: "Perfect See-through register" is detected.
ELSE Output: "Perfect See-through register" is not detected.

(10)

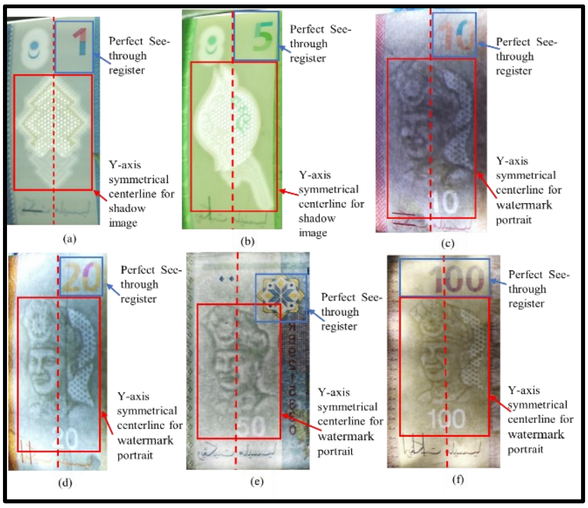


Fig. 5. Perfect See-through register (a) RM1, (b) RM5, (c) RM10, (d) RM20, (e) RM50 and (f) RM100.

A reasonable expression of the Perfect See-through register watermark detection fuzzy set membership function is required, $PS_N : Ratio, x \rightarrow [0,1]$ Since the Perfect See-through Register Detection fully count on PyTesseract software to detect the corresponding numerals, therefore impulsive membership functions are in used and these relates multiple of fuzzy membership functions as below:

Numeral "0" detected at backlight image's membership function:

$$\mu_{0,backlight}(x) = \begin{cases} 1 & x = x_0 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_0 is the numeral "0" being detected by PyTesseract.

Numeral "0" detected at front light image's membership function:

$$\mu_{0,frontlight}(x) = \begin{cases} 1 & x = x_0 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_1 is the numeral "0" being detected by PyTesseract.

Numeral "1" detected at backlight image's membership function:

$$\mu_{1,backlight}(x) = \begin{cases} 1 & x = x_1 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_1 is the numeral "1" being detected by PyTesseract.

Numeral "1" detected at front light image's membership function:

$$\mu_{1,frontlight}(x) = \begin{cases} 1 & x = x_1 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_1 is the numeral "1" being detected by PyTesseract.

Numeral "2" detected at backlight image's membership function:

$$\mu_{2,backlight}(x) = \begin{cases} 1 & x = x_2 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_2 is the numeral "2" being detected by PyTesseract.

Numeral "2" detected at front light image's membership function:

$$\mu_{2,frontlight}(x) = \begin{cases} 1 & x = x_2 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_2 is the numeral "2" being detected by PyTesseract.

Two numeral "0" detected at backlight image's membership function:

$$\mu_{00,backlight}(x) = \begin{cases} 1 & x = x_0 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_{00} is the two numeral "0" being detected by PyTesseract.

Two numeral "0" detected at front light image's membership function:

$$\mu_{00,frontlight}(x) = \begin{cases} 1 & x = x_0 \\ 0 & otherwise \end{cases}$$

where x is the element of numeral and x_{00} is the two numeral "0" being detected by PyTesseract.

Hence, based on the rules in (10), the Perfect See-through register detection fuzzy set membership function is:

$$\begin{aligned} PS_N(x) = {} & \mu_{1,backlight}(x) \times \overline{\mu_{1,frontlight}(x)} \\ & + \mu_{5,backlight}(x) \times \overline{\mu_{5,frontlight}(x)} \\ & + \mu_{1,backlight}(x) \times \mu_{0,backlight}(x) \\ & \times \overline{\mu_{1,frontlight}(x)} \times \overline{\mu_{0,frontlight}(x)} \\ & + \mu_{2,backlight}(x) \times \mu_{0,backlight}(x) \\ & \times \overline{\mu_{2,frontlight}(x)} \times \overline{\mu_{0,frontlight}(x)} \\ & + \mu_{1,backlight}(x) \times \mu_{00,backlight}(x) \\ & \times \overline{\mu_{1,frontlight}(x)} \times \overline{\mu_{00,frontlight}(x)} \end{aligned} \quad (11)$$

(v) Non-transparent Window Detection:

The Non-transparent Window watermark was used on RM1 and RM5. As Perfect See-through register was detected on image "Flat_FB_Blight". The Non-transparent window located at the top left side concerning the Y-axis symmetrical centerline by an area of ½ "Shadow Image" horizontal length. Then, the threshold values are fixed in HSV color space. After that, Search for the biggest and brightest/whitest bounded object, mark it as Non-transparent Window "Star" on image "Flat_FB_Blight". Compare the color intensity of the Star's pixels on images "Flat_FB_Blight" and "Flat_FB_Flight"(sample of RM1 and RM5 Star are shown in Fig. 6.

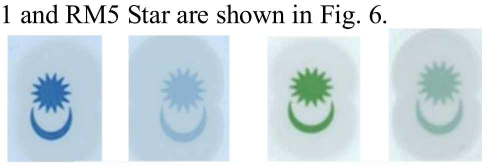


Fig. 6. Sample of RM1 and RM5 Star images captured with Backlight on (left side) and with Front light on (right side).

IF "Shadow Image for RM1 is detected" AND $Th_{NT(min)}$ <|Blue component for sampled pixel of Star on image "Flat_FB_Blight" - The same coordinate sampled pixel of Star in image "Flat_FB_Flight"|< $Th_{NT(max)}$,
THEN Output: Non-transparent Window for RM1 is detected.
ELSE IF "Shadow Image for RM5 is detected" AND $Th_{NT(min)}$ <|Green component for sampled pixel of Star on image "Flat_FB_Blight" - The same coordinate sampled pixel of Star in image "Flat_FB_Flight"|< $Th_{NT(max)}$,
THEN Output: Non-transparent Window for RM5 is detected.
ELSE Output: Non-transparent Window is not detected.

where $Th_{NT(min)}$ is the minimum threshold of the acceptable color intensity change of RM1/RM5 "Star" between images "Flat_FB_Blight" and "Flat_FB_Flight" which are the backlight and front light.
$Th_{NT(max)}$ is the maximum threshold of the acceptable color intensity change of RM1/RM5 "Star" between images "Flat_FB_Blight" and "Flat_FB_Flight" which are the backlight and front light.

A reasonable expression of the Non-transparent Window Detection fuzzy set in triangular membership function is required to formulate, NT_N: *Ratio* $\rightarrow [0,1]$:

$$NT_N(ratio, difference) = SI_N(ratio) \times \mu_N(difference) \quad (12)$$

where $SI_N(ratio)$ is the Shadow Image Detection fuzzy set membership function, as defined in Section (ii).
difference = |Blue or Green component for sampled pixel of Star on image "Flat_FB_Blight" - The same coordinate sampled pixel of Star in image "Flat_FB_Flight"|

Difference in Non-Transparent Window pixel's membership function:

$$\mu_N(Difference) = \begin{cases} 0 & if\ Difference \le a \\ \frac{Difference-a}{b-a} & if\ a \le Difference \le b \\ \frac{c-Difference}{c-b} & if\ b \le Difference \le c \\ 0 & if\ Difference \ge c \end{cases}$$

where N is the number of Ringgit value

$a = 2\mathrm{Th}_{NT(min)} - \left(\frac{\mathrm{Th}_{NT(max)}-\mathrm{Th}_{NT(min)}}{2}\right)$

$b = \left(\mathrm{Th}_{NT(min)} + \frac{\mathrm{Th}_{NT(max)}-\mathrm{Th}_{NT(min)}}{2}\right)$

$c = 2\mathrm{Th}_{NT(max)} - \left(\frac{\mathrm{Th}_{NT(max)}-\mathrm{Th}_{NT(min)}}{2}\right)$

$\mathrm{Th}_{NT(min)}$ is the minimum threshold of the acceptable color intensity change of RM1/RM5 "Star" between images "Flat_FB_Blight" and "Flat_FB_Flight" which are the backlight and front light.

$\mathrm{Th}_{NT(max)}$ is the maximum threshold of the acceptable color intensity change of RM1/RM5 "Star" between images "Flat_FB_Blight" and "Flat_FB_Flight" which are the backlight and front light.

(vi) Color Shifting security thread Detection:

The Color Shifting security thread was used on RM10, RM20 and RM50. On crisp logic, first, the threshold values are fixed in HSV color space. After that, Search for the regions of color shifting on image "Flat_FB_Flight" and "Flat_BB_Flight". Next, check if the total pixels within the bounded area of the Color Shifting Security. Last, Compare the color intensity of the regions of color shifting pixels on images "Flat_FB_Flight" and " Tilt_FB_Flight":

$$TP_{CSS} \ge P_{CSS} \times TR \qquad (13)$$

where P_{CSS} = Percentage of color shifting security area in a Malaysian Banknote.
TR = Total Pixels in the Resized Image converted

IF condition in Eq. (13) is NOT FULFILLED,
THEN the identified "Color Shifting Security" object is a noise object.
Output: "Color Shifting Security" is not detected.
ELSE IF condition in Eq. (13) is FULFILLED,
THEN the identified "Color Shifting Security" object is possible watermark, proceed to the following:

Bounding Box measurement: Assign H_{CSS} as the height of the color shifting security and W_{CSS} as the width of the color shifting security at bounding box as shown in Fig. 7. Measure color shifting security bounding box's height to width:

$$Ratio = H_{CSS}/W_{CSS} \qquad (14)$$

IF $\mathrm{Th}_{CSS(min)} < H_{CSS} / W_{CSS} < \mathrm{Th}_{CSS(max)}$,

IF "color shifting security" for RM10 (Three Region of color shifting == BLUE color) AND (Three Region of color shifting == RED color)
THEN Output: color shifting security is detected.
ELSE IF "color shifting security" for RM20 (Three Region of color shifting == GOLD color) AND (Three Region of color shifting == GREEN color)
THEN Output: color shifting security is detected.
ELSE IF "color shifting security" for RM50 (Five Region of color shifting == RED color) AND (Five Region of color shifting == GREEN color)
THEN Output: color shifting security is detected.
ELSE Output: "color shifting security" is not detected. (15)

where $\mathrm{Th}_{CSS(min)}$ and $\mathrm{Th}_{CSS(max)}$ are the minimum and maximum threshold of color shifting security height to width ratio.

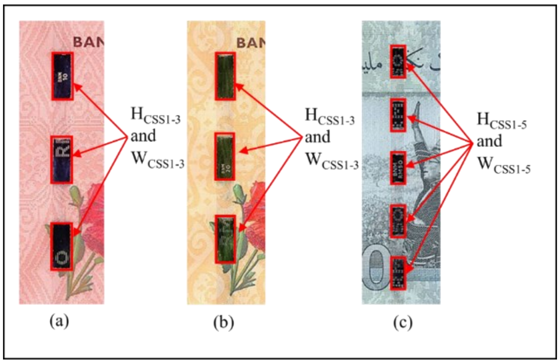


Fig. 7. The Color Shifting Security (a) RM10, (b) RM20 and (c) RM50.

A reasonable expression of the Color Shifting security thread detection fuzzy set membership function is required, $CSS_N : Ratio,\ y \rightarrow [0,1]$ and these relates multiple of fuzzy membership functions as below:

Color Shifting security thread Bounding Box Ratio membership function:

$$\mu_N(Ratio) = \begin{cases} 0 & if\ Ratio \le a \\ \frac{Ratio-a}{b-a} & if\ a \le Ratio \le b \\ \frac{c-Ratio}{c-b} & if\ b \le Ratio \le c \\ 0 & if\ Ratio \ge c \end{cases}$$

where $Ratio = H_{CSS} / W_{CSS}$ is the Color Shifting security thread bounding box's height (H_{CSS}) to width (W_{CSS}).
N is the number of Ringgit value

$a = 2\mathrm{Th}_{CSS(min)} - \left(\frac{\mathrm{Th}_{CSS(max)}-\mathrm{Th}_{CSS(min)}}{2}\right)$

$b = \left(\mathrm{Th}_{CSS(min)} + \frac{\mathrm{Th}_{CSS(max)}-\mathrm{Th}_{CSS(min)}}{2}\right)$

$c = 2\mathrm{Th}_{CSS(max)} - \left(\frac{\mathrm{Th}_{CSS(max)}-\mathrm{Th}_{CSS(min)}}{2}\right)$

$Th_{CSS(min)}$ and $Th_{CSS(max)}$ are the minimum and maximum threshold of color shifting security height to width ratio.

Color shifting of BLUE color membership function:

$$\mu_{BLUE}(y) = \begin{cases} 0 & if\ y \leq a \\ \dfrac{y-a}{b-a} & if\ \mathrm{a} \leq y \leq b \\ \dfrac{c-y}{c-b} & if\ b \leq y \leq c \\ 0 & if\ y \geq c \end{cases}$$

where y is the HSV color element of the detected point $[y_H \quad y_S \quad y_V]$

HSV value for blue is (241°, 1, 100%). H value for blue normally fall between 241° - 300°, S value for blue fall between 0 to 1, where 0 is gray, and 1 is a primary color, V value for blue fall between 0% to 100%, where 0 is completely black, and 100 is the brightest and reveals the most color.

Hence $a = [a_H \quad a_S \quad a_V]$ representing the minimum base value set for color shifting of BLUE and may set $a = [241° \quad 0.3 \quad 50\%]$.
$b = [b_H \quad b_S \quad b_V]$ representing the height of the triangle value set for color shifting of BLUE and may set $b = [300° \quad 1 \quad 100\%]$.
$c = [c_H \quad c_S \quad c_V]$ representing the maximum base value set for color shifting of BLUE and may set $c = [301° \quad 1.01 \quad 101\%]$.

This membership function is more tends to a right-angle triangle membership function to suit the BLUE color-space for more accurate detection.

Color shifting of RED color membership function:

$$\mu_{RED}(y) = \begin{cases} 0 & if\ y \leq a \\ \dfrac{y-a}{b-a} & if\ \mathrm{a} \leq y \leq b \\ \dfrac{c-y}{c-b} & if\ b \leq y \leq c \\ 0 & if\ y \geq c \end{cases}$$

where y is the HSV color element of the detected point $[y_H \quad y_S \quad y_V]$

HSV value for red is (0°, 1, 100%). H value for red normally fall between 0° - 60°, S value for red fall between 0 to 1, where 0 is gray, and 1 is a primary color, V value for red fall between 0% to 100%, where 0 is completely black, and 100 is the brightest and reveals the most color.

Hence $a = [a_H \quad a_S \quad a_V]$ representing the minimum base value set for color shifting of RED and may set $a = [0° \quad 0.3 \quad 50\%]$.
$b = [b_H \quad b_S \quad b_V]$ representing the height of the triangle value set for color shifting of RED and may set $b = [60° \quad 1 \quad 100\%]$.
$c = [c_H \quad c_S \quad c_V]$ representing the maximum base value set for color shifting of RED and may set $c = [61° \quad 1.01 \quad 101\%]$.

This membership function is more tends to a right-angle triangle membership function to suit the RED color space for more accurate detection.

Color shifting of GOLD color membership function:

$$\mu_{GOLD}(y) = \begin{cases} 0 & if\ y \leq a \\ \dfrac{y-a}{b-a} & if\ \mathrm{a} \leq y \leq b \\ \dfrac{c-y}{c-b} & if\ b \leq y \leq c \\ 0 & if\ y \geq c \end{cases}$$

where y is the HSV color element of the detected point $[y_H \quad y_S \quad y_V]$

HSV value for gold/yellow is (120°, 1, 100%). H value for yellow normally fall between 61° - 120°, S value for yellow fall between 0 to 1, where 0 is gray, and 1 is a primary color, V value for yellow fall between 0% to 100%, where 0 is completely black, and 100 is the brightest and reveals the most color.

Hence $a = [a_H \quad a_S \quad a_V]$ representing the minimum base value set for color shifting of GOLD and may set $a = [61° \quad 0.3 \quad 50\%]$.
$b = [b_H \quad b_S \quad b_V]$ representing the height of the triangle value set for color shifting of GOLD and may set $b = [120° \quad 1 \quad 100\%]$.
$c = [c_H \quad c_S \quad c_V]$ representing the maximum base value set for color shifting of GOLD and may set $c = [121° \quad 1.01 \quad 101\%]$.

This membership function is more tends to a right-angle triangle membership function to suit the GOLD color space for more accurate detection.

Color shifting of GREEN color membership function:

$$\mu_{GREEN}(y) = \begin{cases} 0 & if\ y \leq a \\ \dfrac{y-a}{b-a} & if\ \mathrm{a} \leq y \leq b \\ \dfrac{c-y}{c-b} & if\ b \leq y \leq c \\ 0 & if\ y \geq c \end{cases}$$

where y is the HSV color element of the detected point $[y_H \quad y_S \quad y_V]$

HSV value for green is (180°, 1, 100%). H value for green normally fall between 121° - 180°, S value for green fall between 0 to 1, where 0 is gray, and 1 is a primary color, V value for green fall between 0% to 100%, where 0 is completely black, and 100 is the brightest and reveals the most color.

Hence $a = [a_H \quad a_S \quad a_V]$ representing the minimum base value set for color shifting of GREEN and may set $a = [121° \quad 0.3 \quad 50\%]$.
$b = [b_H \quad b_S \quad b_V]$ representing the height of the triangle value set for color shifting of GREEN and may set $b = [180° \quad 1 \quad 100\%]$.
$c = [c_H \quad c_S \quad c_V]$ representing the maximum base value set for color shifting of GREEN and may set $c = [181° \quad 1.01 \quad 101\%]$.

This membership function is more tends to a right-angle triangle membership function to suit the GREEN color space for more accurate detection.
Hence, based on the rules in Eq. (15), the Color Shifting security thread Detection fuzzy set membership function is:

$$CSS_N(Ratio, y) = \mu_N(Ratio)(\mu_{BLUE}(y) \times \mu_{RED}(y) + \mu_{GOLD}(y) \times \mu_{GREEN}(y) + \mu_{RED}(x)\mu_{GREEN}(y)) \quad (16)$$

(vii) Colored Glossy Patch Detection:

The colored glossy patch was used on RM10, RM20 and RM100. On crisp logic, firstly, the threshold values are fixed in HSV color space. After that, Search for the regions of colored glossy patch on image "Flat_BB_Flight". Next, check if the total pixels are within the bounded area of the Color Shifting Security. Last, detect the numeral in colored glossy patch using PyTesseract on image "Flat_BB_Flight" and "Tilt_BB_Flight".

$$TP_{CGP} \geq P_{CGP} \times TR \quad (17)$$

where P_{CGP} = Percentage of colored glossy patch on the area in a Malaysian Banknote.
TR = Total Pixels in the Resized Image converted

IF condition in Eq. (17) are NOT FULFILLED,
THEN the identified "colored glossy patch" object is a noise object.
Output: "colored glossy patch" is not detected.
ELSE IF condition in Eq. (17) are FULFILLED,
THEN the identified "colored glossy patch" object is possible watermark, proceed to the following:

Bounding Box measurement: Assign H_{CGP} as the height of the colored glossy patch and W_{CGP} as the width of the colored glossy patch at bounding box as shown in Fig. 8. Measure colored glossy patch bounding box's height to width:

$$Ratio = H_{CGP}/W_{CGP} \quad (18)$$

IF ($Th_{CGP(min)} < H_{CGP} / W_{CGP} < Th_{CGP(max)}$)
THEN Output: "colored glossy patch" is detected.
ELSE Output: "colored glossy patch" is not detected.

where $Th_{CGP,top(min)}$ and $Th_{CGP,top(max)}$ are the minimum and maximum threshold of the colored glossy patch height to width ratio.

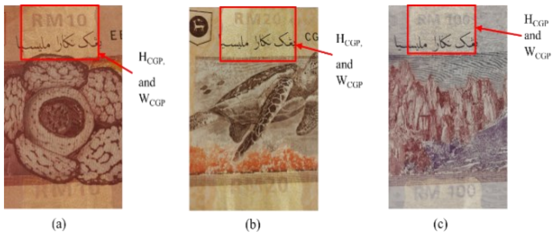


Fig.8. The Colored Glossy Patch (a) RM10, (b) RM20 and (c) RM100.

A reasonable expression of the Colored glossy patch Detection fuzzy set in triangular membership function is required to formulate, $CGP_N : Ratio \rightarrow [0,1]$:

$$CGP_N(Ratio) = \begin{cases} 0 & if\ Ratio \leq a \\ \frac{Ratio - a}{b - a} & if\ a \leq Ratio \leq b \\ \frac{c - Ratio}{c - b} & if\ b \leq Ratio \leq c \\ 0 & if\ Ratio \geq c \end{cases} \quad (19)$$

where $Ratio = H_{CGP}/W_{CGP}$ is the colored glossy patch bounding box's height (H_{CGP}) to width (W_{CGP}).
N is the number of Ringgit value
$a = 2Th_{CGP(min)} - \left(\frac{Th_{CGP(max)} - Th_{CGP(min)}}{2}\right)$
$b = \left(Th_{CGP(min)} + \frac{Th_{CGP(max)} - Th_{CGP(min)}}{2}\right)$
$c = 2Th_{CGP(max)} - \left(\frac{Th_{CGP(max)} - Th_{CGP(min)}}{2}\right)$
$Th_{CGP(min)}$ and $Th_{CGP(max)}$ are the minimum and maximum threshold of "colored glossy patch" height to width ratio.

(viii) Text and Logo Detection:

The Text and Logo were used on All Malaysian Notes. First, the threshold values are fixed in HSV color space. After that, Search for the Rectangle Text, and logo on image "Flat_BB_UVlight". Next, check if the total pixels within the bounded area of the Rectangle Text and logo:

$$TP_T \geq P_T \times TR \quad (20)$$
$$TP_L \geq P_L \times TR \quad (21)$$

where P_T = Percentage of Rectangle Text area in a Malaysian Banknote.
P_L = Percentage of Logo area in a Malaysian Banknote.
TR = Total Pixels in the Resized Image converted

IF condition in Eq. (20) is NOT FULFILLED,
THEN the identified "Text" object is a noise object.
Output: "Text" is not detected.
ELSE IF condition in Eq. (20) are FULFILLED,
THEN the identified "Text" object is possible watermark, proceed to the following:
IF condition in Eq. (21) are NOT FULFILLED,

THEN the identified "Logo" object is a noise object. Output: "Logo" is not detected.
ELSE IF condition in Eq. (21) are FULFILLED,
THEN the identified "Logo" object is possible watermark, proceed to the following:

Bounding Box measurement: Assign H_{RT} and H_L as the height of the Rectangle text and Logo, W_{RT} and W_L as the width of the Rectangle text and Logo at bounding box as shown in Fig. 9. Measure Rectangle text and Logo bounding box's height to width.

$$Ratio = H_{RT}/W_{RT} \quad (22)$$
$$Ratio = H_L/W_L \quad (23)$$

IF $Th_{RT(min)} < H_{RT} / W_{RT} < Th_{RT(max)}$
THEN Output: Text is detected.
ELSE Output: Text is not detected.

IF $Th_{L(min)} < H_L / W_L < Th_{L(max)}$,
THEN Output: Logo is detected.
ELSE Output: Logo is not detected.
where $Th_{RT(min)}$ and $Th_{RT(max)}$ are the minimum and maximum threshold of Rectangle Text.
$Th_{L(min)}$ and $Th_{L(max)}$ are the minimum and maximum threshold of Logo.

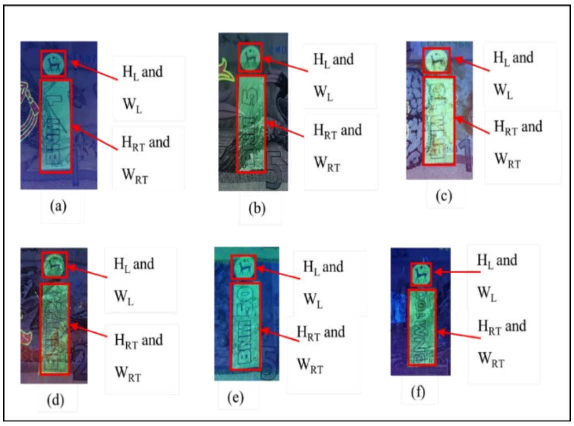


Fig. 9. The Text and Logo (a) RM1, (b) RM5, (c) RM10, (d) RM20, (e) RM50 and (f) RM100.

Two reasonable expression of the Text and Logo Detection fuzzy set in triangular membership functions are required to formulate, based on the rules in Eq. (23).

The Rectangle Text membership function:

$$T_N(Ratio) = \begin{cases} 0 & if\ Ratio \leq a \\ \dfrac{Ratio - a}{b - a} & if\ \mathrm{a} \leq Ratio \leq b \\ \dfrac{c - Ratio}{c - b} & if\ b \leq Ratio \leq c \\ 0 & if\ Ratio \geq c \end{cases} \quad (25)$$

$Ratio = H_{RT}/W_{RT}$ is the Rectangle Text bounding box's height (H_{RT}) to width (W_{RT}).

N is the number of Ringgit value

$a = 2Th_{RT(min)} - \left(\frac{Th_{RT(max)} - Th_{RT(min)}}{2}\right)$

$b = \left(Th_{RT(min)} + \frac{Th_{RT(max)} - Th_{RT(min)}}{2}\right)$

$c = 2Th_{RT(max)} - \left(\frac{Th_{RT(max)} - Th_{RT(min)}}{2}\right)$

$Th_{RT(min)}$ and $Th_{RT(max)}$ are the minimum and maximum threshold of "Rectangle Text" bounding box height to width ratio.

The Logo membership function:

$$L_N(Ratio) = \begin{cases} 0 & if\ Ratio \leq a \\ \dfrac{Ratio - a}{b - a} & if\ \mathrm{a} \leq Ratio \leq b \\ \dfrac{c - Ratio}{c - b} & if\ b \leq Ratio \leq c \\ 0 & if\ Ratio \geq c \end{cases} \quad (26)$$

$Ratio = H_L/W_L$ is the Logo bounding box's height (H_L) to width (W_L).
N is the number of Ringgit value

$a = 2Th_{L(min)} - \left(\frac{Th_{L(max)} - Th_{L(min)}}{2}\right)$

$b = \left(Th_{L(min)} + \frac{Th_{L(max)} - Th_{L(min)}}{2}\right)$

$c = 2Th_{L(max)} - \left(\frac{Th_{L(max)} - Th_{L(min)}}{2}\right)$

$Th_{L(min)}$ and $Th_{L(max)}$ are the minimum and maximum threshold of "Logo" bounding box height to width ratio.

(ix) Two-Color Fluorescent Element Detection:

The two-color fluorescent element was used on All Malaysian Notes. On crisp logic, first, the threshold values are fixed in HSV color space. After that, Search for the two-color fluorescent element on image "Flat_BB_UVlight". Next, check if the total pixels within the bounded area of the two-color fluorescent element. Last, detect the numeral in "two color fluorescent element" using PyTesseract on image "Flat_BB_UVlight":

$$TP_{TCFE} \geq P_{TCFE} \times TR \quad (27)$$

where P_{TCFE} = Percentage of two-color fluorescent element area in a Malaysian Banknote.
TR = Total Pixels in the Resized Image converted

IF condition in Eq. (27) is NOT FULFILLED,
THEN the identified "two color fluorescent element" object is a noise object. Output: "two color fluorescent element" is not detected.
ELSE IF condition in Eq. (27) is FULFILLED,
THEN the identified "two color fluorescent element" object is possible watermark, proceed to the following:

Bounding Box measurement: Assign H_{TCFE} as the height of the two-color fluorescent element and W_{TCFE}

as the width of the two-color fluorescent element at bounding box as shown in Fig. 10. Measure two color fluorescent element bounding box's height to width.

$$Ratio = \mathrm{H_{TCFE}}/\mathrm{W_{TCFE}} \tag{28}$$

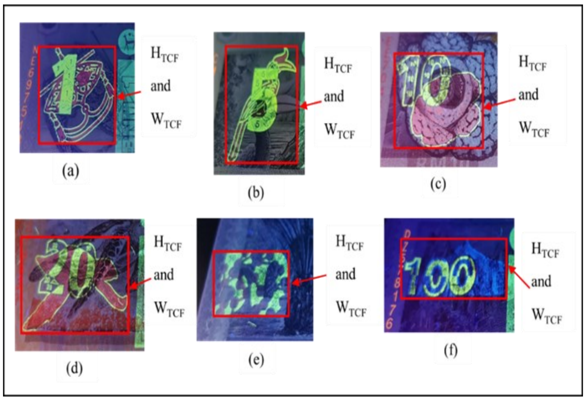


Fig. 10. The Two-Color Fluorescent Element (a) RM1, (b) RM5, (c) RM10, (d) RM20, (e) RM50 and (f) RM100.

IF $\mathrm{Th_{TCFE(min)}} < \mathrm{H_{TCFE}} / \mathrm{W_{TCFE}} < \mathrm{Th_{TCFE(max)}}$
THEN Output: two color fluorescent element is detected.
ELSE Output: "two color fluorescent element" is not detected.
where $\mathrm{Th_{TCFE(min)}}$ and $\mathrm{Th_{TCFE(max)}}$ are the minimum and maximum threshold of two-color fluorescent element height to width ratio.

A reasonable expression of the Two color fluorescent element Detection fuzzy set in triangular membership function is required to formulate, $TCFE_N : Ratio \rightarrow [0,1]$:

$$TCFE_N(Ratio) = \begin{cases} 0 & if\ Ratio \leq a \\ \dfrac{Ratio - a}{b - a} & if\ a \leq Ratio \leq b \\ \dfrac{c - Ratio}{c - b} & if\ b \leq Ratio \leq c \\ 0 & if\ Ratio \geq c \end{cases} \tag{29}$$

$Ratio = H_{TCFE}/W_{TCFE}$ is the shadow image bounding box's height ($\mathrm{H_{TCFE}}$) to width ($\mathrm{W_{TCFE}}$).
N is the number of Ringgit value

$a = 2\mathrm{Th_{TCFE(min)}} - \left(\frac{\mathrm{Th_{CFE(max)}} - \mathrm{Th_{CFE(min)}}}{2}\right)$

$b = \left(\mathrm{Th_{TCFE(min)}} + \frac{\mathrm{Th_{TCFE(max)}} - \mathrm{Th_{TCFE(min)}}}{2}\right)$

$c = 2\mathrm{Th_{TCFE(max)}} - \left(\frac{\mathrm{Th_{TCFE(max)}} - \mathrm{Th_{TCFE(min)}}}{2}\right)$

$\mathrm{Th_{TCFE(min)}}$ and $\mathrm{Th_{TCFE(max)}}$ are the minimum and maximum threshold of "Two-color flouresent element" bounding box height to width ratio.

B. *Fuzzy Logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection Algorithm*

The Fuzzy Logic based Weighted Averaging (FLWA) for Malaysian Banknote Counterfeit Detection Algorithm can be divided into Fifteen (15) essential steps, as shown below:

Step 1: Banknote Confirmation: A color sensor will be used to detect the inserted banknote's by received light intensity of notes:

$$RLI = [R, G, B] \tag{30}$$

where R = Red color saturation level, G = Green color saturation level, B = Blue color saturation level

$$CSL = [CSL_{max}, CSL_{min}, CSL_{diff}] \tag{31}$$

where $\mathrm{CSL_{max}}$ = maximum color saturation level, $\mathrm{CSL_{min}}$ = minimum color saturation level, $\mathrm{CSL_{diff}}$ = difference between maximum and minimum color saturation level

The hue property is the starting point for the HSV model, it is used to correctly identify or categorize colors. Hue, which is expressed as an angle on a color wheel, allows for the identification of colors by using single variable. Next calculate the hue value of the RGB color model.

$$H_R = (60 \times ((G - B) \div CSL_{diff}))$$

$$H_G = (60 \times \left((B - R) \div CSL_{diff}\right) + 120)$$

$$H_B = (60 \times \left((R - G) \div CSL_{diff}\right) + 240)$$

IF $(CSL_{max} == B)\ AND (H_B > V_B)$,
THEN the entered currency can be an RM1 note, set the color sensor output, CS=1, proceed to Step 2.
ELSE IF $(CSL_{max} == G)\ AND (H_G \leq V_G)$,
THEN the entered currency can be an RM5 note, set the color sensor output, CS=5, proceed to Step 2.
ELSE IF $(CSL_{max} == R)\ AND (H_R \leq V_R)$,
THEN the entered currency can be an RM10 note, set the color sensor output, CS=10, proceed to Step 2.
ELSE IF $(CSL_{max} == R)\ AND (H_R > V_R)$,
THEN the entered currency can be an RM20 note, set the color sensor output, CS=20, proceed to Step 2.
ELSE IF $(CSL_{max} == G)\ AND (H_G > V_G)$,
THEN the entered currency can be an RM50 note, set the color sensor output, CS=50, proceed to Step 2.
ELSE IF $(CSL_{max} == B)\ AND (H_B \leq V_B)$,
THEN the entered currency can be an RM100 note, set the color sensor output, CS=100, proceed to Step 2.
ELSE Output "The entered currency is NOT Malaysian Banknote".
where $\mathrm{V_R}$ is the value of red in the RGB color model.
$\mathrm{V_G}$ is the value of green in the RGB color model.
$\mathrm{V_B}$ is the value of blue in the RGB color model.

Step 2: Banknote Position Detection and Re-adjustment: Detect the inserted banknote position and request users to do re-adjustment if necessary. The front light of the camera is turned on, shooting on the banknote. Set the banknote in the flat position, the camera will capture the banknote image, store it as "Pos_B". Fig. 11 shows the top right-hand corner of the UPFRONT banknote position detection.

i. **Positioning RM1:** Identify the number of element "1" N_1 in "Pos_B" that is/are with size > 0.4n x 0.8n pixels using pytesseract [60, 61, 62], where n is the height of objects "1" (in pixel number) on the top right-hand corner of the UPFRONT RM1 banknote as shown in Fig. 11(a).

IF $N_1 == 2$

THEN the position of inserted currency is UPFRONT Position (as shown in Fig. 12(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_1 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 12(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_1 == 0$

THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig. 12(b)) or DOWNBACK Position (as shown in Fig. 12(d)). Proceed to Step 3.

ELSE **Not Fully Detected Case:** Request that the user remove the note, reverse it, and re-insert it into the Banknote Reader, Repeat Step 2(i).

Fig. 11. The n pixels at numeral "0", "1", "2" and "5" on banknote (a) RM1, (b) RM5, (c) RM10, (d) RM20, (e) RM50 and (f) RM100.

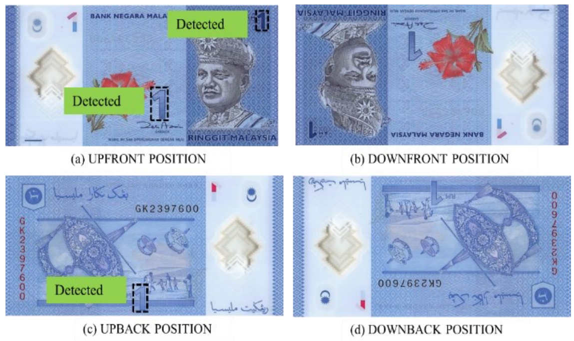


Fig. 12. The possible four inserted Position of RM1 into the banknote reader.

ii. **Positioning RM5:** Identify the number of element "5" N_5 in "Pos_B" that is/are with size > 0.5n x 0.8n pixels using pytesseract, where n is the height of objects "5" (in pixel number) on the top right-hand corner of the UPFRONT RM5 banknote as shown in Fig. 11(b).

IF $N_5 == 2$

THEN the position of inserted currency is UPFRONT Position (as shown in Fig.13(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_5 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 13(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_5 == 0$

THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig. 13(b)) or DOWNBACK Position (as shown in Fig. 13(d)). Proceed to Step 3.

ELSE **Not Fully Detected Case:** Request that the user remove the note, reverse it, and re-insert it into the Banknote Reader, Repeat Step 2(ii).

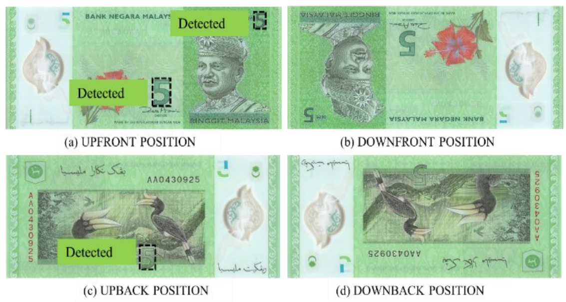


Fig. 13. The possible four inserted Position of RM5 into the banknote reader.

iii. **Positioning RM10:** Detect the number of objects "1", N_1 in "Pos_B" that is/are with size > 0.4n × 0.8n pixels using pytesseract, where n is the height of objects "1" (in pixel number). Also, detect the number of objects "0", N_0 in "Pos_B" with size > 0.5n × 0.8n pixels using pytesseract on the top right-hand corner of the UPFRONT RM10 banknote as shown in Fig. 11(c).

IF $N_1 == 2\ AND\ N_0 == 2$

THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 14(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_1 == 0\ AND\ N_0 == 2$

THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig. 14(b)). proceed to Step 3.

IF $N_1 == 1\ AND\ N_0 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 14(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_1 == 0\ AND\ N_0 == 1$

THEN the position of inserted banknote is DOWNBACK Position (as shown in Fig. 14(d)). Proceed to Step 3.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(iii).

iv. **Positioning RM20:** Detect the number of objects "2", N2 in "Pos_B" that is/are with size > 0.5n × 0.8n pixels using pytesseract, where n is the height of objects "2" (in pixel number). Also, detect the number of objects "0", N0 in "Pos_B" with size > 0.5n × 0.8n pixels using pytesseract on the top right-hand corner of the UPFRONT RM10 banknote as shown in Fig. 11(d).

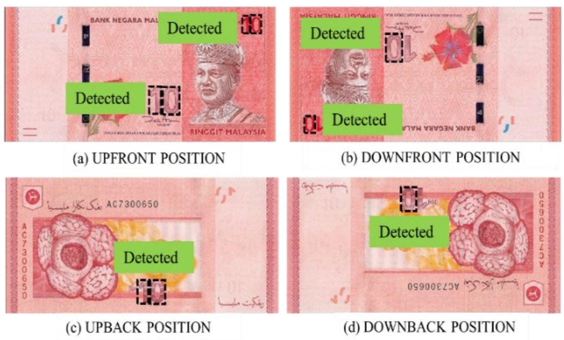


Fig. 14. The possible four inserted Position of RM10 into the banknote reader.

IF $N_2 == 2\ AND\ N_0 == 2$

THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 15(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_2 == 0\ AND\ N_0 == 2$

THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig. 15(b)). proceed to Step 3.

IF $N_2 == 1\ AND\ N_0 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 15(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_2 == 0\ AND\ N_0 == 1$

THEN the position of inserted banknote is DOWNBACK Position (as shown in Fig. 15(d)). proceed to Step 3.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2.

v. **Positioning RM50:** Identify the number of element "5", N_5 in "Pos_B" that is/are with size $> 0.5n \times 0.8n$ pixels using pytesseract, where *n* is the height of objects "5" (in pixel number). Also, identify the number of elements "0",N_0 in "Pos_B" with size $> 0.5n \times 0.8n$ pixels using pytesseract on the top right-hand corner of the UPFRONT RM50 banknote as shown in Fig. 11(e).

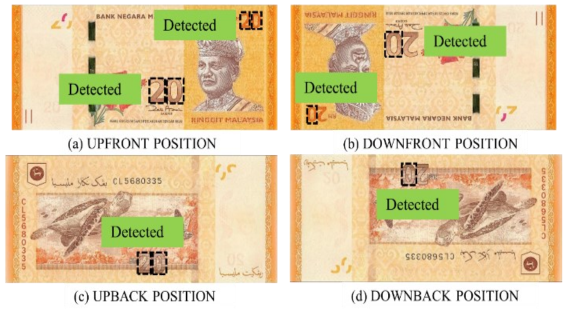


Fig. 15. The possible four inserted Position of RM20 into the banknote reader.

IF $N_5 == 2\ AND\ N_0 == 2$

THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 16(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_5 == 0\ AND\ N_0 == 2$

THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig. 16(b)). proceed to Step 3.

IF $N_5 == 1\ AND\ N_0 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 16(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.

IF $N_5 == 0\ AND\ N_0 == 1$

THEN the position of inserted banknote is DOWNBACK Position (as shown in Fig. 16(d)). proceed to Step 3.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(v).

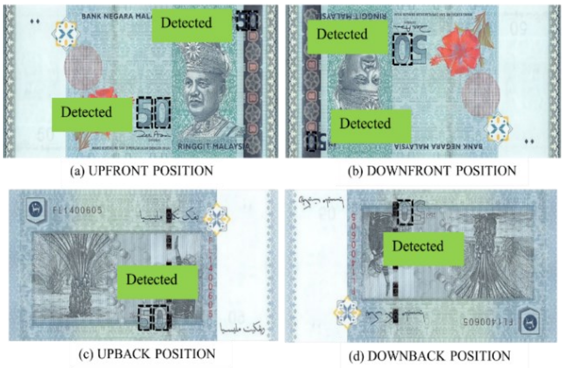


Fig. 16. The possible four inserted Position of RM50 into the banknote reader.

vi. **Positioning RM100:** Identify the number of element “1”, N_1 in “Pos_B” that is/are with size > $0.4n \times 0.8n$ pixels using pytesseract, where n is the height of objects “1” (in pixel number). Also, identify the number of elements “0”,N_0 in “Pos_B” with size > $0.5n \times 0.8n$ pixels using pytesseract on the top right-hand corner of the UPFRONT RM100 banknote as shown in Fig. 11(f).

IF $N_1 == 2\ AND\ N_0 == 4$
THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 17(a)). Set banknote position, FB=FRONT Banknote, Skip Step 3 and Proceed to Step 4.
IF $N_1 == 0\ AND\ N_0 == 4$
THEN the position of inserted banknote is DOWNFRONT Position (as shown in Fig.17(b)). proceed to Step 3.
IF $N_1 == 1\ AND\ N_0 == 2$
THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 17(c)). Set banknote position, BB=BACK Banknote, Skip Step 3 and Proceed to Step 4.
IF $N_1 == 0\ AND\ N_0 == 2$
THEN the position of inserted banknote is DOWNBACK Position (as shown in Fig. 17(d)). proceed to Step 3.
ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(vi).

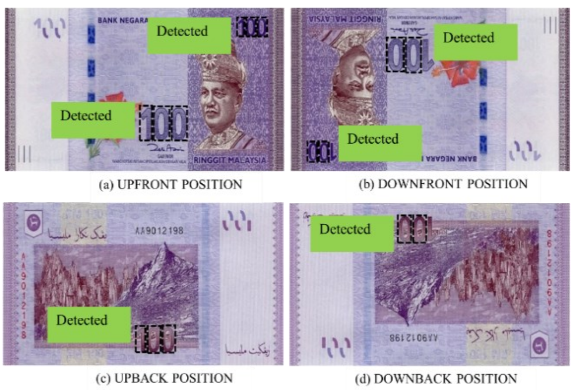


Fig. 17. The possible four inserted Position of RM100 into the banknote reader.

Step 3: Position Detection for Second Time: Rotates the image “Pos_B” by 180 degrees then detect the numeral again. In conclusions, the image should only be shown in the UPBACK or UPFRONT positions.

i. Position Detection in RM1

IF $N_1 == 2$, THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 12(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.
IF $N_1 == 1$, THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 12(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.
ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(i).

ii. Position Detection in RM5

IF $N_5 == 2$, THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 13(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.
IF $N_5 == 1$, THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 13(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.
ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(ii).

iii. Position Detection in RM10

IF $N_1 == 2\ AND\ N_0 == 2$
THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 14(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.
IF $N_1 == 1\ AND\ N_0 == 1$
THEN the position of inserted banknote is UPBACK Position (as shown Fig. 14(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.
ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(iii).

iv. Position Detection in RM20

IF $N_2 == 2\ AND\ N_0 == 2$
THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 15(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.
IF $N_2 == 1\ AND\ N_0 == 1$
THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 15(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(iv).

v. Position Detection in RM50

IF $N_5 == 2\ AND\ N_0 == 2$

THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 16(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.

IF $N_5 == 1\ AND\ N_0 == 1$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 16(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(v).

vi. Position Detection in RM100

IF $N_1 == 2\ AND\ N_0 == 4$

THEN the position of inserted banknote is UPFRONT Position (as shown in Fig. 17(a)). Set banknote position, FB=FRONT Banknote, Proceed to Step 4.

IF $N_1 == 1\ AND\ N_0 == 2$

THEN the position of inserted banknote is UPBACK Position (as shown in Fig. 17(c)). Set banknote position, BB=BACK Banknote, proceed to Step 4.

ELSE **Not Fully Detected Case:** Request the user to take out the banknote and reinsert back to the Banknote Reader, Repeat Step 2(vi).

Step 4: Capturing a Flat Currency notes Image: The image "Pos_B" is now in either the UPBACK or UPFRONT positions only. Furthermore, the image "Pos_B" should be saved as "Flat_FB_Flight" for UPFRONT Banknote positions or "Flat_BB_Flight" for UPBACK Banknote positions. Similarly capture "Flat_FB_Blight" for UPFRONT Banknote positions or "Flat_BB_Blight" for UPBACK Banknote positions. Ultraviolet light is turned on, captured and stored the "Flat_FB_UVlight" for UPFRONT Banknote positions or "Flat_BB_UVlight" for UPBACK Banknote positions.

Step 5: Tilt Banknote and Capture an Image: The tilt mechanism was used on RM10, RM20 and RM50. The tilt will be on both the UPFRONT and UPBACK positions of the RM10 and RM20, but only the UPBACK position of the RM50. Tilt the inserted banknote to $\emptyset_{optimum}$ the angle that will show the color shifting on the same banknote. Turn on the front light and the camera will capture in the tilted banknote, stored it as "Tilt_FB_Flight" for UPFRONT Banknote positions or "Tilt_BB_Flight" for UPBACK Banknote positions.

Step 6: Fully Capture the Banknote: However, to get the other banknote position, request the user to take out the banknote, reverse the banknote and reinsert back to the Banknote Reader as shown in Fig. 18. Repeated back those steps 1 to 5. Fully capture the banknote mean by having UPFRONT Banknote positions and UPBACK Banknote positions.

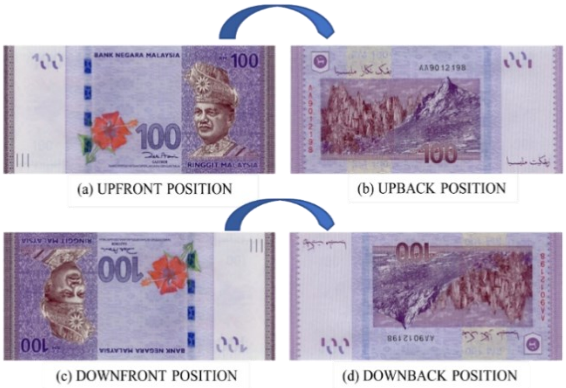


Fig. 18. Reverse the banknote RM100.

Step 7: Shadow Image Detection:

i) Detect Shadow Image in RM1:
Provoke Shadow Image Detection function in Section III A(ii) and calculate the Shadow Image fuzzy membership value, SI_1 using Eq. (4).

ii) Detect Shadow Image in RM5:
Provoke Shadow Image Detection function in Section III A(ii) and calculate the Shadow Image fuzzy membership value, SI_5 using Eq. (4).

Step 8: Watermark Portrait Detection:

i) Detect Watermark Portrait in RM10:
Provoke Watermark Portrait Detection function in Section III A(iii) and calculate the Watermark Portrait fuzzy membership value, WP_{10} using Eq. (9).

ii) Detect Watermark Portrait in RM20:
Provoke Watermark Portrait Detection function in Section III A(iii) and calculate the Watermark Portrait fuzzy membership value, WP_{20} using Eq. (9).

iii) Detect Watermark Portrait in RM50:
Provoke Watermark Portrait Detection function in Section III A(iii) and calculate the Watermark Portrait fuzzy membership value, WP_{50} using Eq. (9).

iv) Detect Watermark Portrait in RM100:
Provoke Watermark Portrait Detection function in Section III A(iii) and calculate the Watermark Portrait fuzzy membership value, WP_{50} using Eq. (9).

Step 9: Perfect See-through Register Detection:

i) **Detect Perfect See-through register in RM1:** Provoke Perfect See-through register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_1 using Eq. (11).

ii) **Detect Perfect See-through register in RM5:** Provoke Perfect See-through Register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_5 using Eq. (11).

iii) **Detect Perfect See-through register in RM10:** Provoke Perfect See-through Register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_{10} using Eq. (11).

iv) **Detect Perfect See-through register in RM20:** Provoke Perfect See-through Register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_{20} using Eq. (11).

v) **Detect Perfect See-through register in RM50:** Provoke Perfect See-through Register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_{50} using Eq. (11).

vi) **Detect Perfect See-through register in RM100:** Provoke Perfect See-through Register Detection function in Section III A(iv) and calculate the Perfect See-through register fuzzy membership value, PS_{100} using Eq. (11).

Step 10: Non-transparent Window Detection:

i. **Non-transparent Window watermark "Star" in RM1:** Provoke Non-transparent Window Detection function in Section III A(v) and calculate the Non-transparent Window fuzzy membership value, NT_1 using Eq. (12).

ii. **Non-transparent Window watermark "Star" in RM5:** Provoke Non-transparent Window Detection function in Section III A(v) and calculate the Non-transparent Window fuzzy membership value, NT_5 using Eq. (12).

Step 11: Color Shifting security thread Detection:

i. **Color Shifting Security in RM10:** Provoke Color Shifting security thread Detection function in Section III A(vi) and calculate the Color shifting security fuzzy membership value, CSS_{10} using Eq. (16).

ii. **Color Shifting Security in RM20:** Provoke Color Shifting security thread Detection function in Section III A(vi) and calculate the Color shifting security fuzzy membership value, CSS_{20} using Eq. (16).

iii. **Color Shifting Security in RM50:** Provoke Color Shifting security thread Detection function in Section III A(vi) and calculate the Color shifting security fuzzy membership value, CSS_{50} using Eq. (16).

Step 12: Colored Glossy Patch Detection:

i. **Colored Glossy Patch in RM10:** Provoke Colored Glossy Patch Detection function in Section III A(vii) and calculate the Color glossy patch fuzzy membership value, CGP_{10} using Eq. (19).

ii. **Colored Glossy Patch in RM20:** Provoke Colored Glossy Patch Detection function in Section III A(vii) and calculate the Color glossy patch fuzzy membership value, CGP_{20} using Eq. (19).

iii. **Colored Glossy Patch in RM100:** Provoke Colored Glossy Patch Detection function in Section III A(vii) and calculate the Color glossy patch fuzzy membership value, CGP_{100} using Eq. (19).

Step 13: Text and Logo Detection:

i. **Text and Logo in RM1:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_1using Eq. (25) and L_1 using Eq. (26)

ii. **Text and Logo in RM5:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_5using Eq. (25) and L_5 using Eq. (26)

iii. **Text and Logo in RM10:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_{10}using Eq. (25) and L_{10} using Eq. (26)

iv. **Text and Logo in RM20:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_{20}using Eq. (25) and L_{20} using Eq. (26)

v. **Text and Logo in RM50:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_{50}using Eq. (25) and L_{50} using Eq. (26)

vi. **Text and Logo in RM100:** Provoke Text and Logo Detection function in Section III A(viii) and calculate the Text and Logo fuzzy membership values, T_{100}using Eq. (25) and L_{100} using Eq. (26)

Step 14: Two Color Fluorescent Element Detection:

i. **Two Color Fluorescent Element in RM1:**

Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_1$ using Eq. (29).

ii. **Two Color Fluorescent Element in RM5:** Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_5$ using Eq. (29).

iii. **Two Color Fluorescent Element in RM10:** Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_{10}$ using Eq. (29).

iv. **Two Color Fluorescent Element in RM20:** Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_{20}$ using Eq. (29).

v. **Two Color Fluorescent Element in RM50:** Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_{50}$ using Eq. (29).

vi. **Two Color Fluorescent Element in RM100:** Provoke Two Color Fluorescent Element Detection function in Section III A(ix) and calculate the Two color fluorescent element membership value, $TCFE_{100}$ using Eq. (29).

Step 15: Decision making Apply fuzzy logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection algorithm below:

i. **Record the output of the inference security features** obtained from Step 7 – Step 14 in the form of x_y (SI_y, WP_y, PS_y, NT_y, CSS_y, CGP_y, $TCFE_y$, T_y), where x is the security feature for a particular Malaysian Banknote and y is the particular read Malaysian Banknote value.

ii. **Calculate the Assigned Weightage Averaging**, **WA** for each security feature for the particular Malaysian Banknote (WA_{SI}, WA_{WP}, WA_{PS}, WA_{NT}, WA_{CSS}, WA_{CGP}, WA_{TCFE}, WA_T and WA_L) by:

$$\mathrm{WA}_{x,\mathrm{RM}y} = \frac{1}{\mathrm{MNSF}_{\mathrm{RM}y}} \tag{30}$$

where x is the security feature for a particular Malaysian Banknote

y is the particular read Malaysian Banknote value

MNSF is the number of Security Features that selected for testing the particular y-value Malaysian banknotes.

The related suggested parameters for Fuzzy logic based Weighted Averaging (FLWA) are summarized in Table II to Table VII below, the security features are optional, and the Weighting Averaging (WA) are alterable according to user choices.

iii. **Calculate the de-fuzzified output, D_{FLWA}:**

$$D_{FLWA} = \frac{\sum_{n=1}^{MNSF_{RMy}} (x_y^n) \times \mathrm{WA}_{x,RMy}^n}{\sum_{n=1}^{MNSF_{RMy}} \mathrm{WA}_{x,RMy}^n} \tag{31}$$

iv. **Banknote counterfeit detection decision making** based on a set of rules:

IF CS = =1 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM1}}$,
THEN output “It is a real RM1 banknote”.
ELSE IF CS= =5 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM5}}$,
THEN output “It is a real RM5 banknote”.
ELSE IF CS= =10 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM10}}$,
THEN output “It is a real RM10 banknote”.
ELSE IF CS= =20 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM20}}$,
THEN output “It is a real RM20 banknote”.
ELSE IF CS= =50 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM50}}$,
THEN output “It is a real RM50 banknote”.
ELSE IF CS= =100 AND $D_{FLWA} \geq \mathrm{THRESHOLD_{RM100}}$,
THEN output “It is a real RM100 banknote”.
ELSE Output: “It is not Malaysian banknote”

where CS is color sensor output (1 =RM1, …., 100 = RM100)

$\mathrm{THRESHOLD_{RM1...RM100}}$ is the acceptable Threshold value set for deciding the real banknote. Normally set to at least half or up to two-third (meaning at least half or two-third of the selected Security Features successfully detected for the Malaysian banknote).

Table II: The suggested Security Features and Weighted Averaging of Security Features in RM1.

No.	Security Features	Assigned Weighting Averaging
1.	Shadow Image	SI → $WA_{SI,RM1}$
2.	Perfect See-through Register	PS → $WA_{PS,RM1}$
3.	Non-transparent Window	NT → $WA_{NT,RM1}$
4.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM1}$
5.	Text	TL → $WA_{TL,RM1}$
6.	Logo	L → $WA_{L,RM1}$
Total		$MNSF_{RM1}$ = 6

Table III: The suggested Security Features and Weighted Averaging of Security Features in RM5.

No.	Security Features	Assigned Weighting Averaging
1.	Shadow Image	SI → $WA_{SI,RM5}$
2.	Perfect See-through Register	PS → $WA_{PS,RM5}$
3.	Non-transparent Window	NT → $WA_{NT,RM5}$
4.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM5}$
5.	Text	T → $WA_{T,RM5}$
6.	Logo	L → $WA_{L,RM5}$
Total		$MNSF_{RM5}$=6

Table IV: The suggested Security Features and Weighted Averaging of Security Features in RM10.

No.	Security Features	Assigned Weighting Averaging
1.	Watermark Portrait	WP → $WA_{WP,RM10}$
2.	Perfect See-through Register	PS → $WA_{PS,RM10}$
3.	Color Shifting security thread	CSS → $WA_{CSS,RM10}$
4.	Colored Glossy Patch	CGP → $WA_{CGP,RM10}$
5.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM10}$
6.	Text	T → $WA_{T,RM10}$
7.	Logo	L → $WA_{L,RM10}$
Total		$MNSF_{RM10}$=7

Table V: The suggested Security Features and Weighted Averaging of Security Features in RM20.

No.	Security Features	Assigned Weighting Averaging
1.	Watermark Portrait	WP → $WA_{WP,RM20}$
2.	Perfect See-through Register	PS → $WA_{PS,RM20}$
3.	Color Shifting security thread	CSS → $WA_{CSS,RM20}$
4.	Colored Glossy Patch	CGP → $WA_{CGP,RM20}$
5.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM20}$
6.	Text	T → $WA_{T,RM20}$
7.	Logo	L → $WA_{L,RM20}$
Total		$MNSF_{RM20}$=7

Table VI: The suggested Security Features and Weighted Averaging of Security Features in RM50.

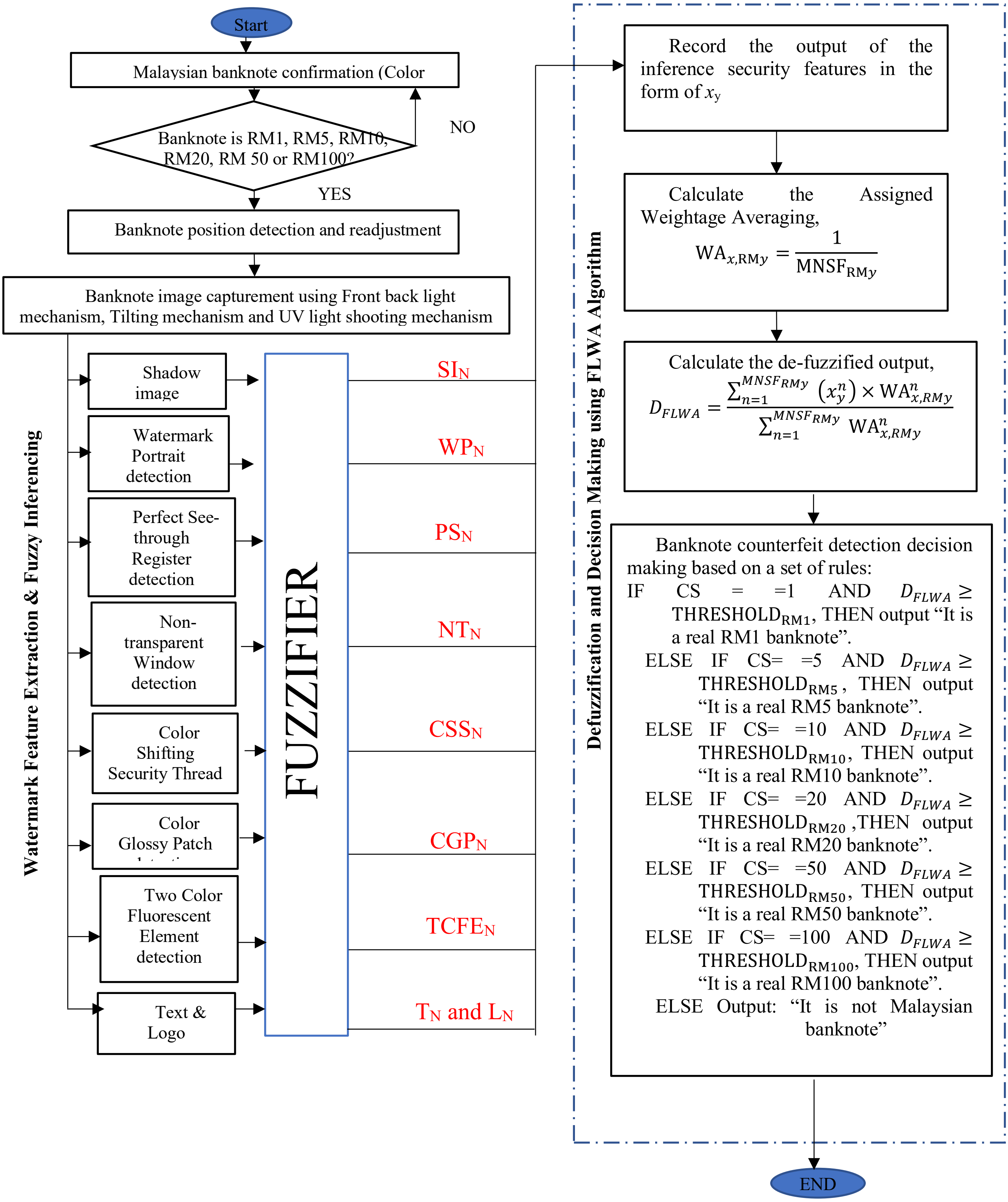
No.	Security Features	Assigned Weighting Averaging
1.	Watermark Portrait	WP → $WA_{WP,RM50}$
2.	Color Shifting security thread	CSS → $WA_{CSS,RM50}$
3.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM50}$
4.	Text	T → $WA_{T,RM50}$
5.	Logo	L → $WA_{L,RM50}$
Total		$MNSF_{RM50}$=5

Table VII: The suggested Security Features and Weighted Averaging of Security Features in RM100.

No.	Security Features	Assigned Weighting Averaging
1.	Watermark Portrait	WP → $WA_{WP,RM100}$
2.	Perfect See-through Register	PS → $WA_{PS,RM100}$
3.	Colored Glossy Patch	CGP → $WA_{CGP,RM100}$
4.	Two Color Fluorescent Element	TCFE → $WA_{TCFE,RM100}$
5.	Text	T → $WA_{T,RM100}$
6.	Logo	L → $WA_{L,RM100}$
Total		$MNSF_{RM100}$=6

The flow chart of the overall FLWA algorithm is shown on the next page.

Flow Chart for FLWA Malaysian Banknotes Detection Algorithm



IV. EXPERIMENTAL RESULTS

This section will discuss the parameters setup and experimental results for the proposed Fuzzy Logic Weighted Averaging (FLWA) Algorithm in Malaysian banknotes reader for counterfeit banknotes detection. Furthermore, the FLWA algorithm is compared with two state-of-the-art parallel methods namely the MobileNet model using RMSprop Loss Function (learning_rate=0.0001) and VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API.

A. Parameter Setup and Optimization for FLWA Algorithm in Malaysian Banknote Counterfeit Detection

For Image prepossesses as shown in Section III (A), *TR* is Total Pixels in the Resized Image converted in Section III (A) Eq. (2), TR = 250 (width) × 450 (height) = 112,500 pixels. Kernel is the structuring element used for erosion effect. For example, shadow image with 4 x 4 pixels. Such filter removed outlier 4x4 pixels that may be noise elements in the image. The reason of 4 x 4 pixels was chosen is according to the banknote reader box internal environment and the backlight intensity. If bigger matrix of pixels dimension (for e.g. 8 x 8 pixels) was chosen, then the filter shadow image will be out shape image. Also, if a smaller matrix of pixels dimension (for e.g. 1 x 1 pixels) was chosen, then the filter shadow image will be clipped image. For example, difference matrixes of pixels dimensions are shown in Fig. 19.

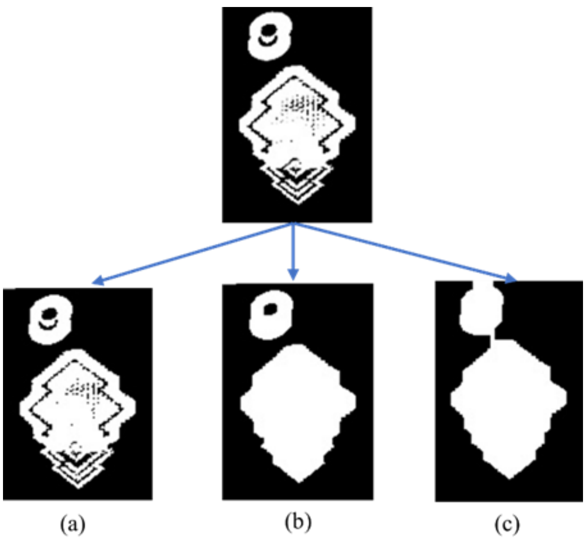


Fig. 19. (a) clipped (1x1) shadow image, (b) on shape(4x4) shadow image and (c) out shape (8x8) shadow image.

For **Shadow Image Detection,** as shown in Section III A(ii), the Shadow Image area in the real RM1 banknote is measured with dimension of 25 mm × 35 mm = 875 mm^2. Therefore, P_{SI} for RM1 is 0.1122 and the height to width ratio of Shadow Image pattern in RM1 banknote is 1.40. Shadow Image area in the real RM5 banknote is measured with dimension of 25 mm × 40 mm = 1,000 mm^2. Therefore, P_{SI} for RM5 is 0.1140 and the height to width ratio of Shadow Image pattern in RM5 banknote is 1.60. Since the banknote reader is shared between RM1 and RM5 detection, hence the minimum P_{SI} among the two is selected, and rounded to 0.11. Hence, Noise Object Exclusion part, any object with bounding box region smaller than 0.11×112,500 = 12,375 pixels will not be considered as Shadow Image. To better classify RM1 and RM5 from one another, for RM1, $Th_{SI(min)}$ is set to 1.35 and $Th_{SI(max)}$ is set to 1.45; whereas for RM5, $Th_{SI(min)}$ is set to1.55 and $Th_{SI(max)}$ is set to 1.65. Such setting is with the best tolerance gap to classify the two types of banknotes effectively.

For **Watermark Portrait Detection,** as shown in Section III A(iii), Watermark Portrait, and numeral area in the real RM10, RM20, RM50 and RM100 banknote is measured with a dimension of 25 mm × 43 mm = 1075 mm^2. Therefore, P_{WP_N} for RM10 is 0.1181. Therefore, P_{WP_N} for RM20 is 0.1141. Therefore, P_{WP_N} for RM50 is 0.1074. Therefore, P_{WP_N} for RM100 is 0.1039. However, the height to width ratio of the Watermark Portrait pattern in the RM10, RM20, RM50 and RM100 banknote is 1.72. Since the banknote reader is shared among RM10, RM20, RM50 and RM100 detection, hence the minimum P_{WP_N} among the four is selected and rounded to 0.10. Hence, Noise Object Exclusion part, any object with bounding box region smaller than 0.10×112,500 = 12,250 pixels will not be considered as Watermark Portrait. To properly categorize the Watermark Portrait in RM10, RM20, RM50 and RM100, $Th_{WP\_N(min)}$ is set to 1.67 and $Th_{WP\_N(max)}$ is set to 1.77.

For **Perfect See-through Register Detection,** as shown in Section III A(iv), there is no parameter need to identify. For **Non-transparent Window Detection,** as shown in Section III A(v), to get Th_{NT} and $Th_{NT,}$ 100 different real banknotes of RM1s'and RM5s images were captured for 100 pairs of images "Flat_FB_Blight" (backlight on) and image "Flat_FB_Flight" (Front light on). For RM1 the Blue color intensity value on the Star's sampled pixels were recorded and the difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" were calculated and tabulated in the plots of no. of attempts vs. |Blue color intensity difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" |as shown in Fig.20. From Fig.20, it is shown that most occurrence happened in between blue color intensity value of 112 to 131. Hence $Th_{NT(min)}$ is set to 112 and $Th_{NT(max)}$ is set to 131.

In addition, in RM5 the Green color intensity value on the Star's sampled pixels were recorded and the difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" were calculated and tabulated in the plots of no. of attempts vs. |Green color intensity difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" |as shown in Fig.21**Error! Reference source not found.**. From Fig. 21, it is shown that most occurrence happened in between green color intensity value of 114 to 135. Hence $Th_{NT(min)}$ is set to 114 and $Th_{NT(max)}$ is set to 135.

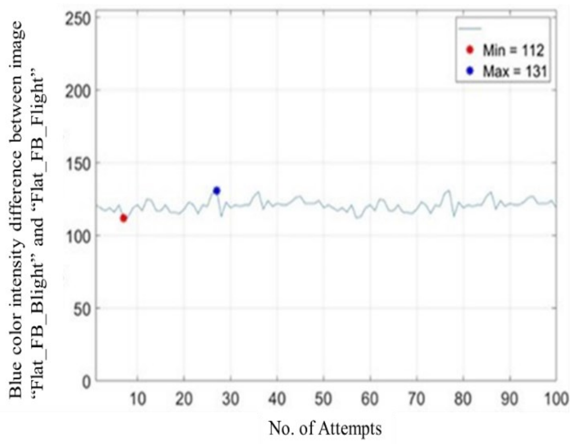


Fig. 20. Plots on (no. of attempts vs. |blue color intensity difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" |) for RM1.

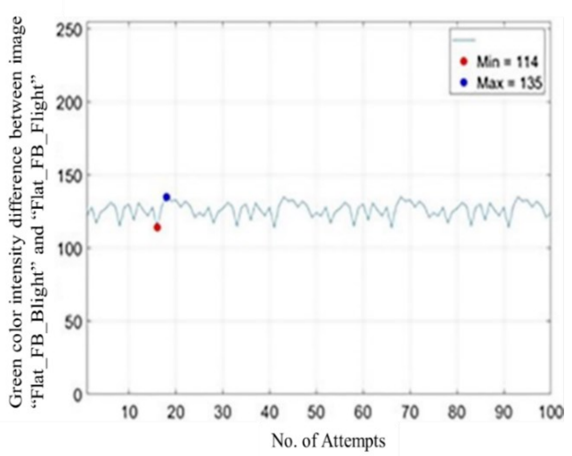


Fig. 21. Plots on (no. of attempts vs. |green color intensity difference between image "Flat_FB_Blight" and image "Flat_FB_Flight" |) for RM5.

For **Color Shifting security thread Detection,** as shown in Section III A(vi), Color Shifting security thread area in the real RM10 and RM20 banknote is measured with a dimension of 4mm × 10 mm = 40 mm^2. Therefore, P_{CSS} for RM10 and RM20 are (0.0044 and 0.0042 respectively) and the height to width ratio of Color Shifting security thread pattern in RM10 and RM20 banknote is 2.50. However, the Color Shifting security thread area in the real RM50 banknote is measured with a dimension of 3mm × 6mm = 18 mm^2. Therefore, P_{CSS} for RM50 is (0.0018) and the height to width ratio of Color Shifting security thread pattern in RM50 banknote is 2. Since the banknote reader is shared among RM10, RM20 and RM50 detection, hence the minimum P_{CSS} among the three is selected and rounded to 0.0018. Hence, Noise Object Exclusion part, any object with bounding box region smaller than 0.0018×112,500 = 202.5 pixels will not be considered as Color Shifting security. To properly categorize the Color Shifting security thread in RM10, RM20 and RM50 from one another, for RM10 and RM20, $Th_{CSS(min)}$ is set to 2.45 and $Th_{CSS(max)}$ is set to 2.55; whereas for RM50, $Th_{CSS(min)}$ is set to1.95 and $Th_{CSS(max)}$ is set to 2.05.

For **Colored Glossy Patch Detection,** as shown in Section III A(vii), the Colored Glossy Patch area in the real RM10 and RM100 banknote are the same measured. Moreover, the dimension of the Colored Glossy Patch area 18mm × 25 mm = 450 mm^2. Therefore, P_{CGP} for RM10 and RM100 are (0.0495 and 0.0435 respectively). In addition, the height to width ratio of Colored Glossy Patch pattern in RM10 and RM100 banknote are 1.39 and 2.5 respectively. The Colored Glossy Patch area in the real RM20 banknote is measured with a dimension of 19mm × 28mm = 532mm^2. Therefore, P_{CGP} for RM20 is (0.0564). Moreover, the height to width ratio of Colored Glossy Patch pattern in RM20 banknote is 1.47. Sine the banknote reader is shared among RM10, RM20 and RM100 detection, hence the minimum P_{CGP} among the three are selected 0.0435. Noise Object Exclusion part, any object with bounding box region smaller than, on top 0.0435×112,500 = 4893.75 pixels will not be considered as Colored Glossy Patch. To better classify RM10, RM20 and RM100 from one another, for RM10 and RM100, $Th_{CGP(min)}$ is set to 1.34 and $Th_{CGP(max)}$ is set to 1.44; whereas for RM20, $Th_{CGP(min)}$ is set to 1.42 and $Th_{CGP(max)}$ is set to 1.52.

For **Text and Logo Detection,** as shown in Section III A(viii), first with Text, the area of Rectangle Text in the real RM1, RM5, RM10, RM20, and RM50 banknote are the same measured. Moreover, the dimension of Rectangle Text area 12mm × 30mm = 360 mm^2. Therefore, P_{RT} on RM1, RM5, RM10, RM20, and RM50 are (0.0462, 0.0410, 0.0396, 0.0382 and 0.0359 respectively). In addition, the height to width ratio of Rectangle Text pattern in RM1, RM5, RM10, RM20, and RM50 banknote are 2.50. The Rectangle Text area in the real RM100 banknote is measured with a dimension of 12mm × 27 mm = 324mm^2. Therefore, P_{RT} for RM100 is 0.0313 and the height to width ratio of Rectangle Text pattern in RM100 banknote is 2.25. Second with Logo, the area of Logo in the real RM1, RM5, RM10, RM20, RM50 and RM100 banknote are the same measured. Moreover, the dimension of Logo area 10mm × 10mm = 100 mm^2. Therefore, P_L on RM1, RM5, RM10, RM20, RM50 and RM100 are (0.0128, 0.0114, 0.0109, 0.0382, 0.0106 and 0.0097 respectively). In addition, the height to width ratio of Logo pattern in RM1, RM5, RM10, RM20, RM50 and RM100 banknote are 1.

Since the banknote reader is shared among RM1, RM5, RM10, RM20, RM50 and RM100 detection, hence the minimum P_{RT} and P_L among the six is selected and rounded to 0.0313 and 0.0097. Hence, Noise Object Exclusion part, any object with bounding box region smaller than, on Text 0.0313×112,500 = 3521.25 pixels and on Logo 0.0097×112,500 = 1091.25 pixels will not be considered as Text and Logo. To better classify the Text and Logo on RM1, RM5, RM10, RM20, RM50 and RM100 from one another. First the Text, on RM1, RM5, RM10, RM20, and RM50, $Th_{RT(min)}$ is set to 2.45 and $Th_{RT(max)}$ is set to 2.55; whereas on RM100, $Th_{RT(min)}$ is set to 2.20 and $Th_{RT(max)}$ is set to 2.30. Second the Logo, on RM1, RM5, RM10, RM20, RM50 and RM100, $Th_{L(min)}$ is set to 0.95 and $Th_{L(max)}$ is set to 1.05.

For **Two-Color Fluorescent Element Detection,** as shown in Section III A(ix), Two-Color Fluorescent Element area in the real RM1 banknote is measured with a dimension of 31mm × 31 mm = 961 mm^2. Therefore, P_{TCFE} for RM1 is 0.1232 and the height to width ratio of Two-Color Fluorescent Element pattern in RM1 banknote is 1. The Two-Color Fluorescent Element area in the real RM5 banknote is measured with a dimension of 27mm × 36 mm = 972 mm^2. Therefore, P_{TCFE} for RM5 is 0.1108 and the height to width ratio of Two-Color Fluorescent Element pattern in RM5 banknote is 1.33. The Two-Color Fluorescent Element area in the real RM10 banknote is measured with a dimension of 33mm × 33 mm = 1089 mm^2. Therefore, P_{TCFE} for RM10 is 0.1197 and the height to width ratio of Two-Color Fluorescent Element pattern in RM10 banknote is 1. The Two-Color Fluorescent Element area in the real RM20 banknote is measured with a dimension of 24mm × 39 mm = 936 mm^2. Therefore, P_{TCFE} for RM20 is 0.0993 and the height to width ratio of Two-Color Fluorescent Element pattern in RM20 banknote is 1.63. The Two-Color Fluorescent Element area in the real RM50 banknote is measured with a dimension of 20mm × 21.5mm = 430mm^2. Therefore, P_{TCFE} for RM50 is 0.0429 and the height to width ratio of Two-Color Fluorescent Element pattern in RM50 banknote is 1.08. The Two-Color Fluorescent Element area in the real RM100 banknote is measured with a dimension of 13mm × 29 mm = 377mm^2. Therefore, P_{TCFE} for RM100 is 0.0364 and the height to width ratio of Two-Color Fluorescent Element pattern in RM100 banknote is 2.23.

Since the banknote reader is shared among RM1, RM5, RM10, RM20, RM50 and RM100 detection, hence the minimum P_{TCFE} among the six is selected and rounded to 0.0364. Hence, Noise Object Exclusion part, any object with bounding box region smaller than 0.0364×112,500 = 4095 pixels will not be considered as Two-Color Fluorescent Element. To properly categorize the Two-Color Fluorescent Element in RM1, RM5, RM10, RM20, RM50 and RM100 from one another. For RM1, $Th_{TCFE(min)}$ is set to 0.95 and $Th_{TCFE(max)}$ is set to 1.05. For RM5, $Th_{TCFE(min)}$ is set to 1.28 and $Th_{TCFE(max)}$ is set to 1.38. For RM10, $Th_{TCFE(min)}$ is set to 0.95 and $Th_{TCFE(max)}$ is set to 1.05. For RM20, $Th_{TCFE(min)}$ is set to 1.58 and $Th_{TCFE(max)}$ is set to 1.68. For RM50, $Th_{TCFE(min)}$ is set to 1.03 and $Th_{TCFE(max)}$ is set to 1.13. For RM100, $Th_{TCFE(min)}$ is set to 2.18 and $Th_{TCFE(max)}$ is set to 2.28.

B. Performance of FLWA Malaysian Banknote Counterfeit Detection Algorithm

For **RM1,** an experimental test was carried out with 1000 pieces of real RM1 banknotes and 1000 pieces of fake RM1 banknotes. Figure 22 show HSV value on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$ and $Val_{(max)}$ (a) Shadow Image, (b) Perfect See-through Register, (c) Non-transparent Window (d) Text, Logo, and Two-Color Fluorescent Element. Figure 23 to Fig. 28 show the detection of the security features correspondingly in RM1.

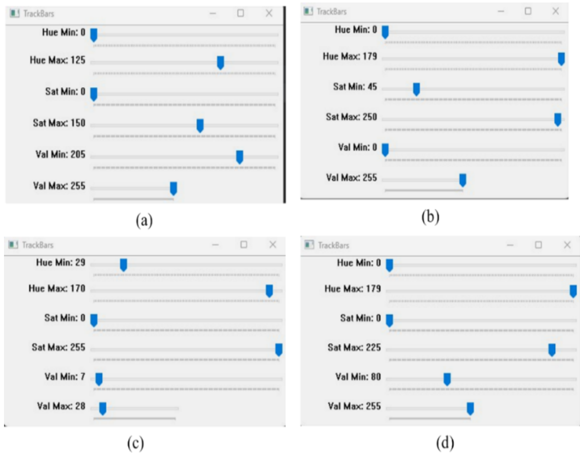


Fig. 22. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max) on RM1 (a) Shadow Image, (b) Perfect See-through Register, (c) Non-transparent Window and (d) Text, Logo, and Two Color Fluorescent Element.

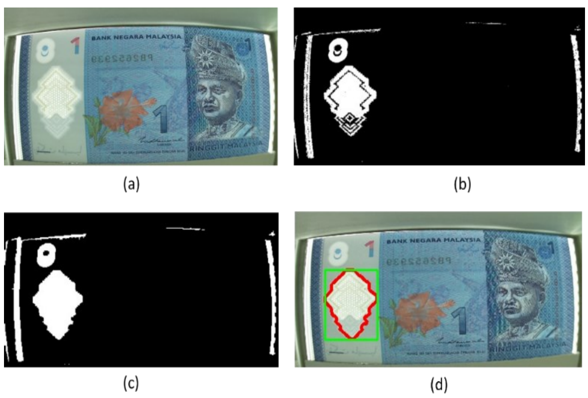


Fig. 23. Detecting Shadow Image on RM1 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect Shadow Image.

For **RM5,** an experimental test was carried out with 1000 pieces of real RM5 banknotes and 1000 pieces of fake RM5 banknotes. Figure 29 show HSV value on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$ and $Val_{(max)}$

(a) Shadow Image, (b) Perfect See-through Register, (c) Non-transparent Window (d) Text and Logo and (e) Two Color Fluorescent Element. Figure 30 to Fig. 35 show the detection of the security features respectively in RM5.

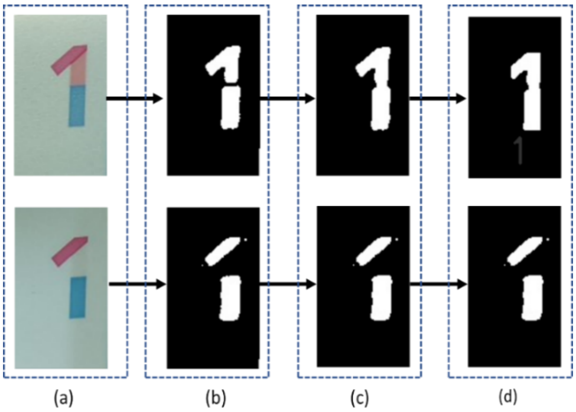


Fig. 24. Detecting Perfect See-through Register on RM1 (a) Original Image Backlight (up) and front light (down), (b) HSV value, (c) Morphological and (d) Detect the numeral "1".

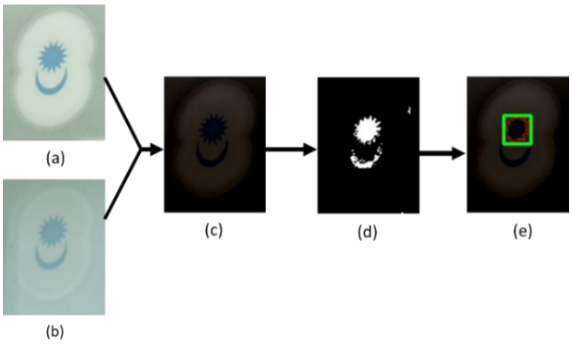


Fig. 25. Detecting Non-transparent Window on RM1 (a) Original Image Backlight light(up) and front light (down), (b) Subtract the Backlight light and front light Images, (c) HSV value and (d) Detect Non-transparent Window.

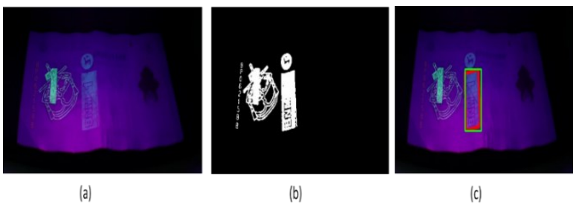


Fig. 26. Detecting Text on RM1 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

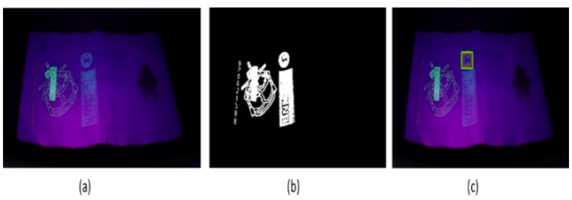


Fig. 27. Detecting Logo on RM1 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

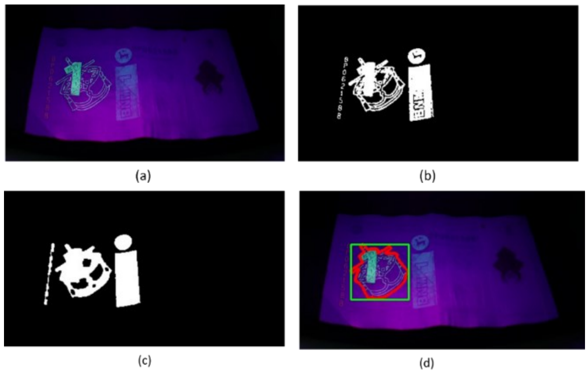


Fig. 28. Detecting Two Color Fluorescent Element on RM1 (a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two Color Fluorescent Element.

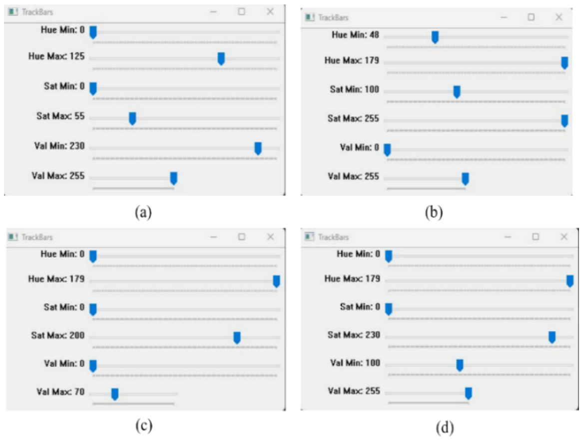


Fig. 29. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max) on RM5 (a) Shadow Image, (b) Perfect See-through Register, (c) Non-transparent Window and (d) Text, Logo, and Two-Color Fluorescent Element.

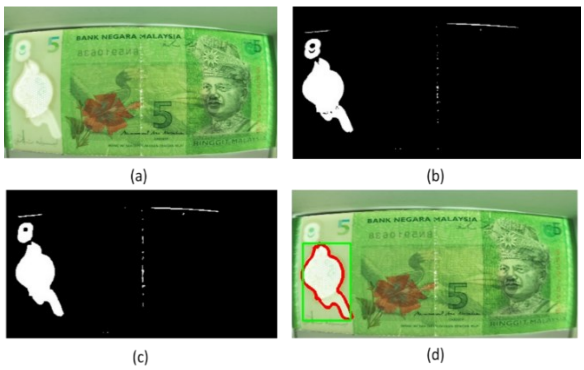


Fig. 30. Detecting Shadow Image on RM5 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect Shadow Image.

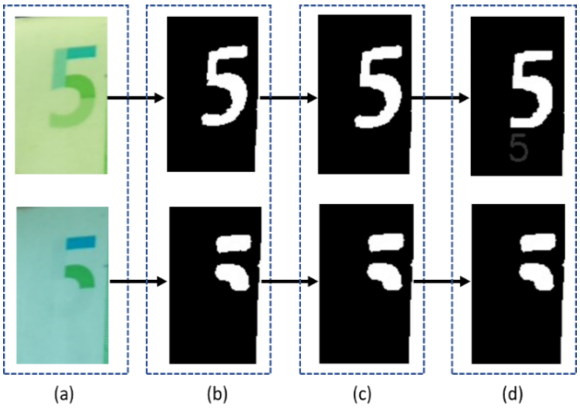


Fig. 31. Detecting Perfect See-through Register on RM5 (a) Original Image Backlight (up) and front light (down), (b) HSV value, (c) Morphological and (d) Detect the numeral "5".

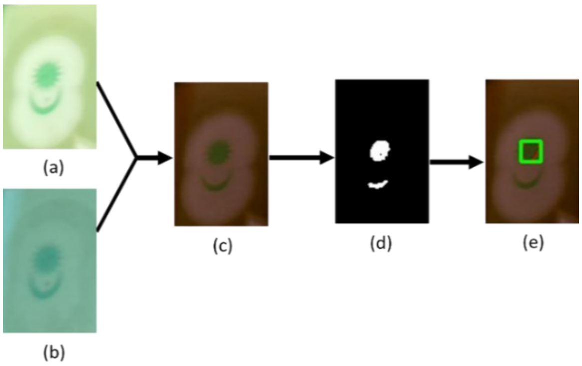


Fig. 32. Detecting Non-transparent Window on RM5 (a) Original Image Backlight light(up) and front light (down), (b) Subtract the Backlight light and front light Images, (c) HSV value and (d) Detect Non-transparent Window.

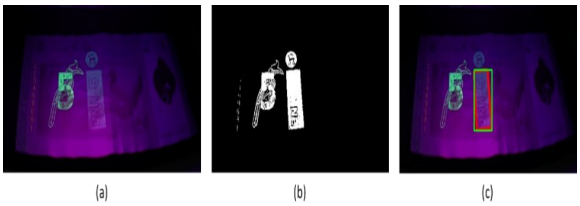


Fig. 33. Detecting Text on RM5 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

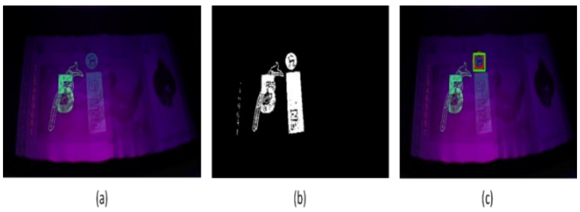


Fig. 34. Detecting Logo on RM5 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

For **RM10,** an experimental test was carried out with 1000 pieces of real RM10 banknotes and 1000 pieces of fake RM10 banknotes. Figure 36 show HSV value on $\mathrm{Hue}_{(min)}$, $\mathrm{Hue}_{(max)}$, $\mathrm{Sat}_{(min)}$, $\mathrm{Sat}_{(max)}$, $\mathrm{Val}_{(min)}$ and $\mathrm{Val}_{(max)}$ (a) Watermark Portrait, (b) Perfect See-through Register, (c) Color Shifting security flat position, (d) Color Shifting security tilt position (e) Colored Glossy Patch, and (f) Text, Logo, and Two Color Fluorescent Element. Figure 37 to Fig. 44 show the detection of the security features respectively in RM10.

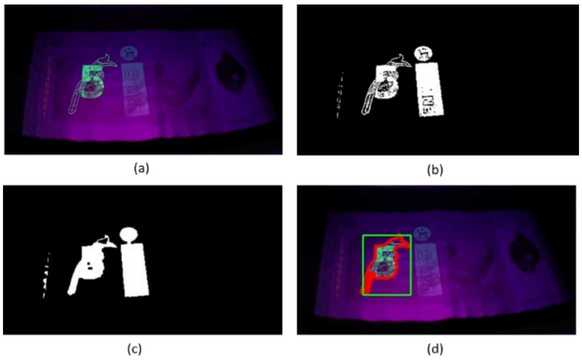


Fig. 35. Detecting Two Color Fluorescent Element on RM5 (a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two Color Fluorescent Element.

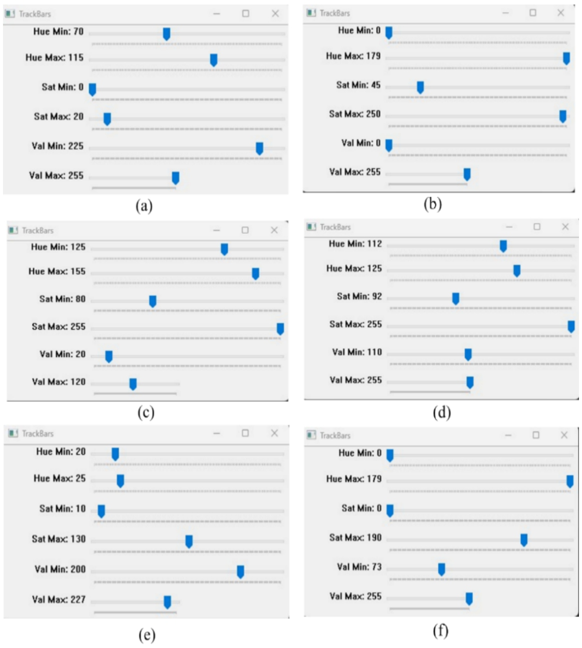


Fig. 36. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max) on RM10 (a) Watermark Portrait, (b) Perfect See-through Register, (c) Color Shifting security flat position, (d) Color Shifting security tilt position (e) Colored Glossy Patch and (f) Text , Logo, and Two Color Fluorescent Element.

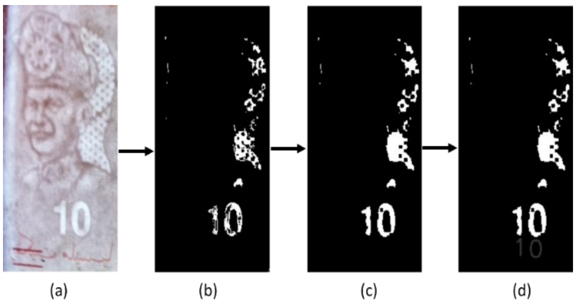


Fig. 37. Detecting Watermark Portrait on RM10 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the numeral "1"and "0".

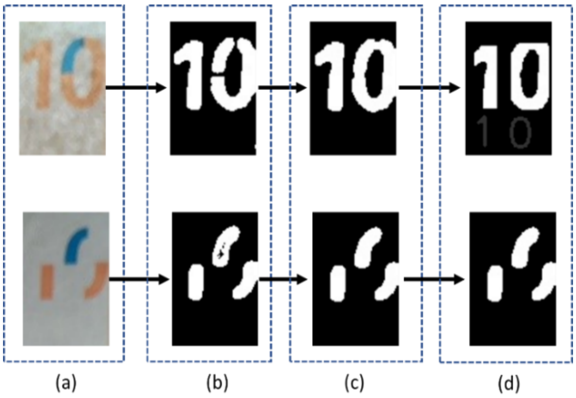


Fig. 38. Detecting Perfect See-through Register on RM10 (a) Original Image Backlight (up) and front light (down), (b) HSV value, (c) Morphological and (d) Detect the numeral “1”and “0”.

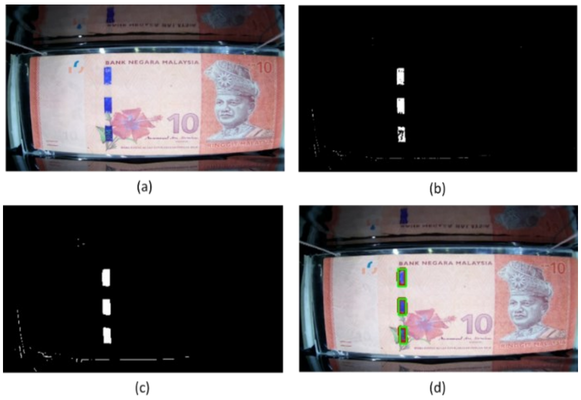


Fig. 39. Detecting Color Shifting security flat position on RM10 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security.

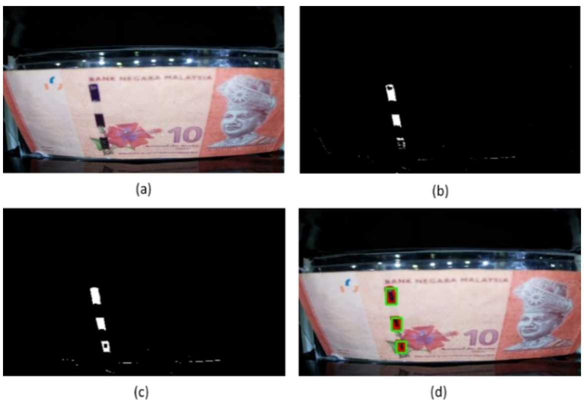


Fig. 40. Detecting Color Shifting security tilt position on RM10 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security.

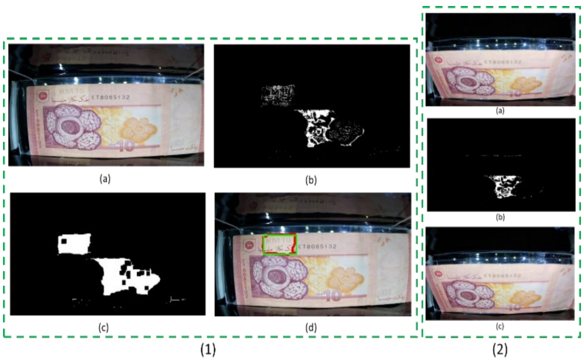


Fig. 41. Detecting Colored Glossy Patch on RM10 (1) (a)Original Image front light, (b) HSV value, (c) Morphological and (d) Detect Colored Glossy Patch. (2) (a) Tilt Original Image front light, (b) Morphological and (c) Non-Detect Colored Glossy Patch after tilt.

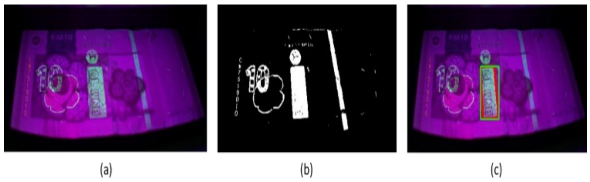


Fig. 42. Detecting Text on RM10 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

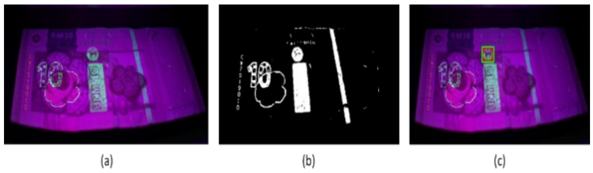


Fig. 43. Detecting Logo on RM10 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

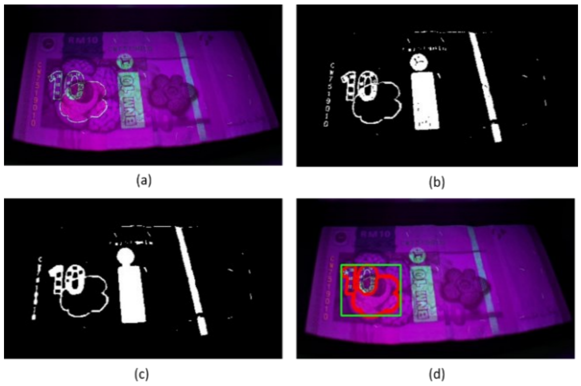


Fig. 44. Detecting Two Color Fluorescent Element on RM10(a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two Color Fluorescent Element.

Fig.45. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max) on RM20 (a) Watermark Portrait, (b) Perfect See-through Register, (c) Color Shifting security flat position, (d) Color Shifting security tilt position (e) Colored Glossy Patch and (f) Text , Logo, and Two Color Fluorescent Element.

For **RM20,** an experimental test was carried out with 1000 pieces of real RM20 banknotes and 1000 pieces of fake RM20 banknotes. Figure 45 show HSV value on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$ and $Val_{(max)}$ (a) Watermark Portrait, (b) Perfect See-through Register, (c) Color Shifting security flat position, (d) Color Shifting security tilt position (e) Colored Glossy Patch, and (f) Text, Logo, and Two Color Fluorescent Element. Figure 46 to Fig. 52 show the detection of the security features respectively in RM20.

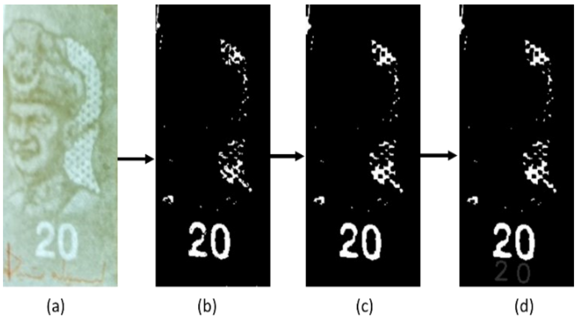


Fig. 46. Detecting Watermark Portrait on RM20 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the numeral "2"and "0".

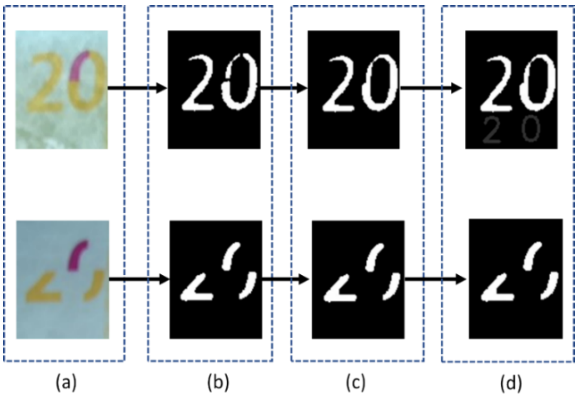


Fig. 47. Detecting Perfect See-through Register on RM20 (a) Original Image Backlight (up) and front light (down), (b) HSV value, (c) Morphological and (d) Detect the numeral "2"and "0".

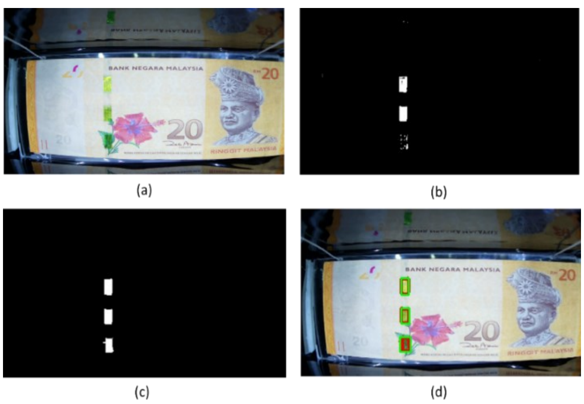


Fig. 48. Detecting Color Shifting security flat position on RM20 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security.

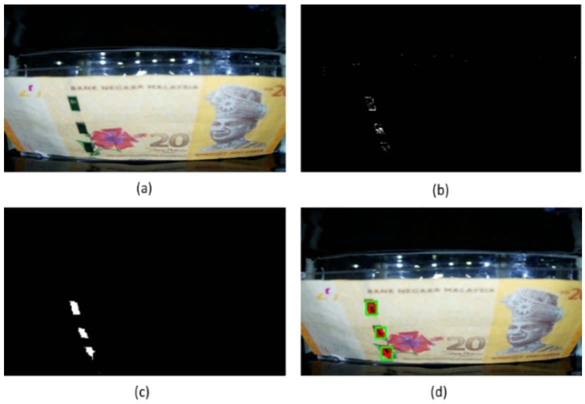


Fig. 49. Detecting Color Shifting security tilt position on RM20 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security

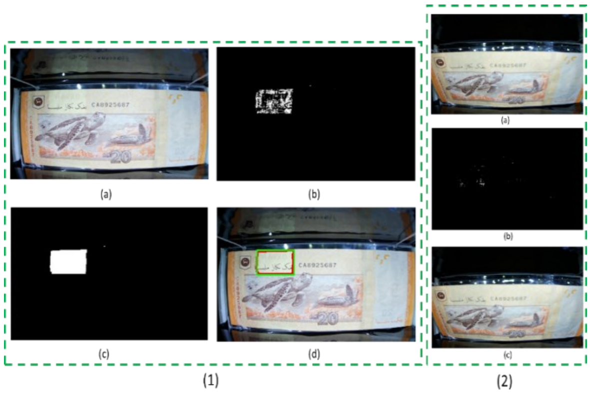


Fig. 50. Detecting Colored Glossy Patch on RM20 (1) (a)Original Image front light, (b) HSV value, (c) Morphological, (d) Detect Colored Glossy Patch. (2) (a)Original Image front light, (b) Morphological and (c) Detect Colored Glossy Patch.

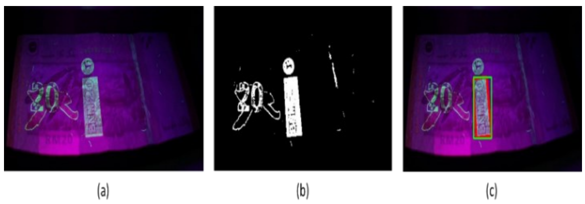


Fig. 51. Detecting Text on RM20 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

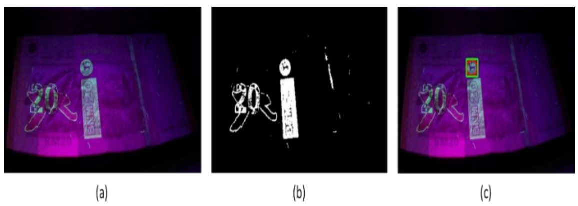


Fig. 52. Detecting Logo on RM20 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

For **RM50,** an experimental test was carried out with 1000 pieces of real RM50 banknotes and 1000 pieces of fake RM50 banknotes. Figure 54 shows HSV value on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$ and $Val_{(max)}$ (a) Watermark Portrait, (b) Color Shifting security flat position, (c) Color Shifting security tilt position and (d) Text, Logo, and Two-Color Fluorescent Element. Figure 55 to Fig. 60 show the detection of the security features respectively in RM50.

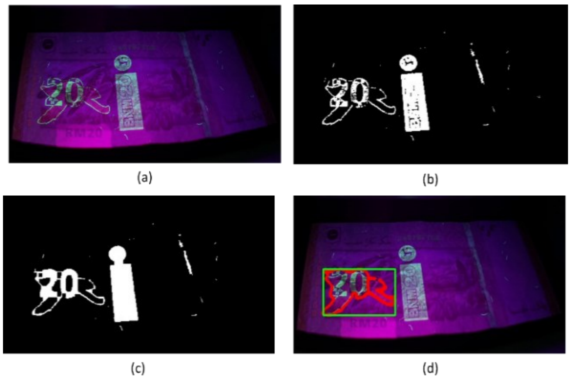


Fig. 53. Detecting Two Color Fluorescent Element on RM20 (a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two Color Fluorescent Element.

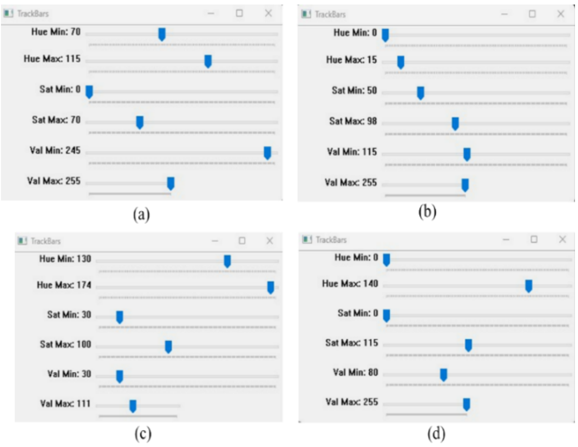


Fig. 54. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max)on RM50 (a) Watermark Portrait, (b) Color Shifting security flat position, (c) Color Shifting security tilt position and (d) Text, Logo, and Two-Color Fluorescent Element.

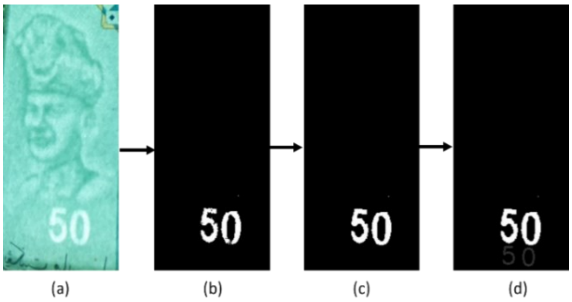


Fig. 55. Detecting Watermark Portrait on RM50 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the numeral “5”and “0”.

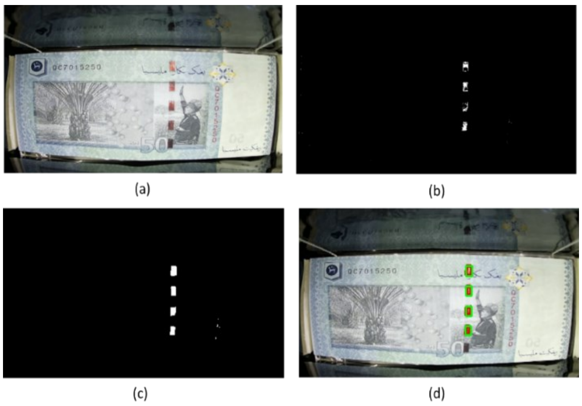


Fig. 56. Detecting Color Shifting security flat position on RM50 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security.

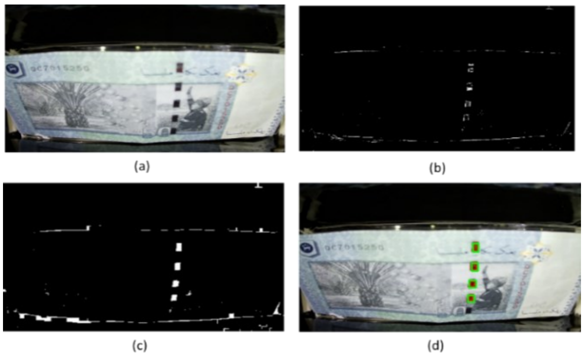


Fig. 57. Detecting Color Shifting security tilt position on RM50 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the three Color Shifting security.

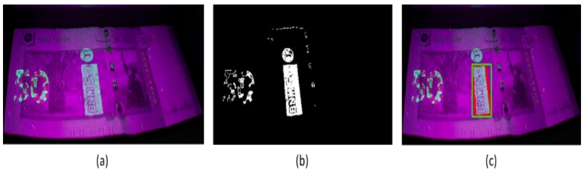


Fig. 58. Detecting Text on RM50 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

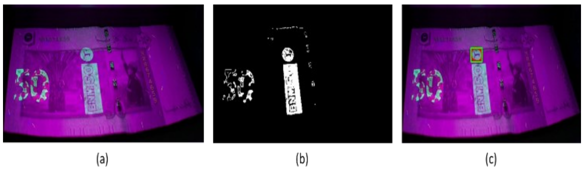


Fig. 59. Detecting Logo on RM50 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

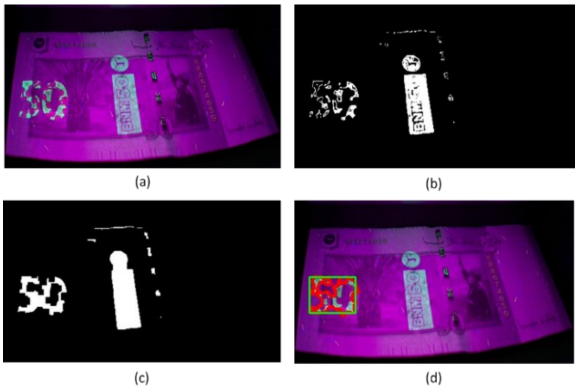


Fig. 60. Detecting Two Color Fluorescent Element on RM50 (a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two Color Fluorescent Element.

For **RM100,** an experimental test was carried out with 1000 pieces of real RM100 banknotes and 1000 pieces of fake RM100 banknotes. Figure 61 shows HSV value on $Hue_{(min)}$, $Hue_{(max)}$, $Sat_{(min)}$, $Sat_{(max)}$, $Val_{(min)}$ and $Val_{(max)}$ (a) Watermark Portrait, (b) Perfect See-through Register, (c) Colored Glossy Patch, and (d) Text, Logo, and Two-Color Fluorescent Element. Figure 62 to Fig. 67 show the detection of the security features respectively in RM100.

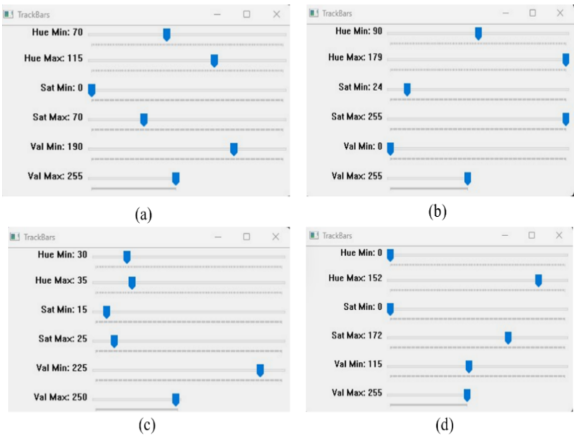


Fig. 61. HSV value on Hue(min), Hue(max), Sat(min), Sat(max), Val(min) and Val(max) on RM100 (a) Watermark Portrait, (b) Perfect See-through Register, (c) Colored Glossy Patch and (d) Text Logo, and Two Color Fluorescent Element.

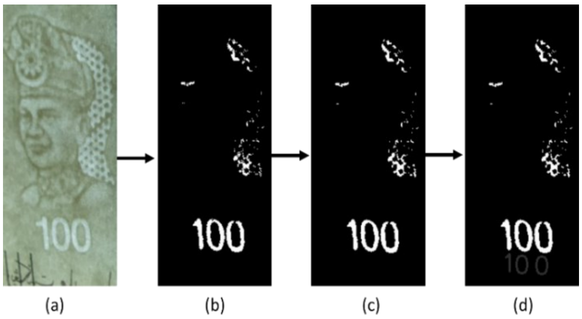


Fig. 62. Detecting Watermark Portrait on RM100 (a) Original Image backlight, (b) HSV value, (c) Morphological and (d) Detect the numeral “1”and “0”.

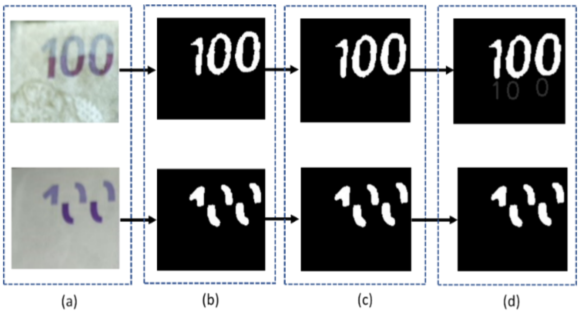


Fig. 63. Detecting Perfect See-through Register on R100 (a) Original Image Backlight (up) and front light (down), (b) HSV value (c) Morphological and (d) Detect the numeral “1”and “0”.

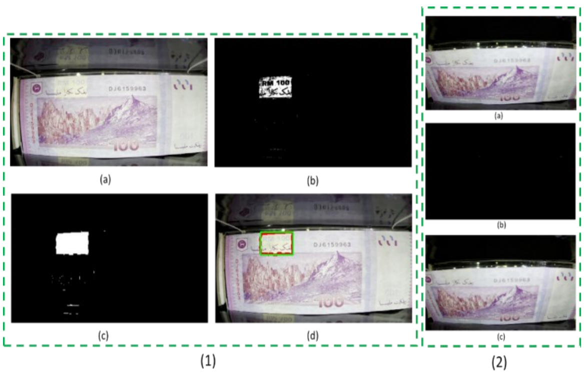


Fig. 64. Detecting Colored Glossy Patch on RM100 (1) (a)Original Image front light, (b) HSV value, (c) Morphological, (d) Detect Colored Glossy Patch. (2) (a)Original Image front light, (b) Morphological and (c) Detect .Colored Glossy Patch

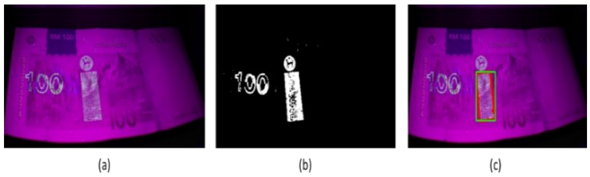


Fig. 65. Detecting Text on RM100 (a) Original Image UV light, (b) HSV value and (c) Detect Text.

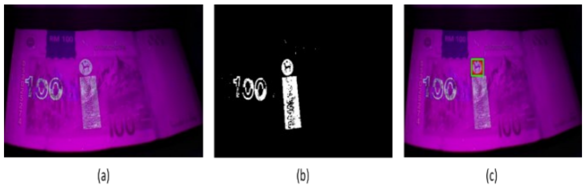


Fig. 66. Detecting Logo on RM100 (a) Original Image UV light, (b) HSV value and (c) Detect Logo.

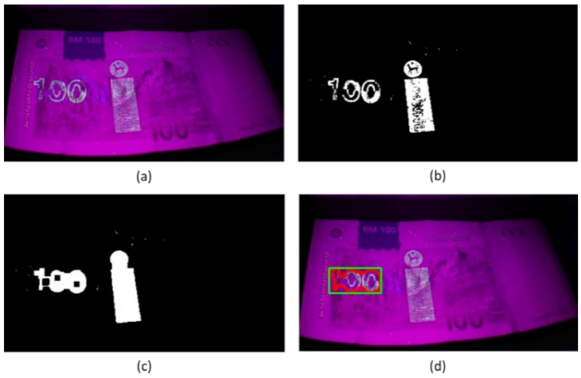


Fig. 67. Detecting Two-Color Fluorescent Element on RM100 (a) Original Image UV light, (b) HSV value, (c) Morphological and (d) Detect Two-Color Fluorescent Element.

For **Decision making,** as shown in Section III B step 15, the acceptable threshold value set for deciding the real banknote is set to at least half of the selected Security Features successfully detected for the Malaysian banknote. The four possible outcomes for the banknote identification scenario is list below as shown in Table VIII.

Table VIII: Confusion matrix for FLWA Malaysian Banknote.

N / Outcome	RM1	RM5	RM10	RM20	RM50	RM100
True positive (The predicted RM_N banknote is real, and it actually is a real RM_N banknote)	978	974	877	903	1000	964
True negative (The predicted RM_N banknote is fake, and it actually is a fake RM_N banknote)	1000	1000	1000	1000	1000	1000
False positive (The predicted RM_N banknote is real, and it actually is a fake RM_N banknote)	0	0	0	0	0	0
False negative (The predicted RM_N banknote is fake, and it actually is a real RM_N banknote)	22	26	123	97	0	36

For summary, the recommended banknote reader obtained around 98.9%, 98.7%, 93.9%, 95.10%, 100%, and 98.2%, accuracy for RM1, RM5, RM10, RM20, RM50, and RM100, detection respectively. Overall, the system has an accuracy of up to 97.46% in recognizing the correct banknote value and counterfeit detection. Noteworthy attention is placed on False Positive and False Negative cases, because these two cases may cause the visually impaired person to lose credits in their business.

For RM1 and RM5, they have 22 and 26 banknotes detection cases related to False Negative class and none cases related to False Positive class. Further analyzed on these False Negative cases each, it is found that those tested real RM1 and RM5 banknotes were not placed properly into the Malaysian banknote reader (center of the banknote slot) and the Malaysian banknotes reader had mistreated some other areas on the corresponding real banknote as the three Region of interest area, and this further identified the real RM1 or RM5 as the fake RM1 or RM5. To overcome this problem, normalized sizes were assigned on RM1 and RM5 at the Step 2 Algorithm (Banknote Position Detection and Re-adjustment) to better locking the three Region of interest area.

For RM10 and RM20, each has 123 and 97 banknotes detection cases related to False Negative class and no cases related to False Positive class. Further analysed on these 123 and 97 False Negative cases, it is found that the Colored Glossy Patch, Two-Color Fluorescent elements, Watermark Portrait and Color Shifting security thread watermarks were not detected in some of the tested real RM10 and RM20 banknotes. Investigation found that the Colored Glossy Patch, Two-Color Fluorescent elements, and Color Shifting security thread watermarks areas on these affected banknotes were faded. This further affected three of the seven measures in RM10 and RM20 security features. To overcome this problem, specific weights with less emphasis were assigned to these easily faded watermarks and more emphasis weight on other easily detected watermarks to identify RM10 and RM20 real banknotes.

.

C. Comparison with Parallel Work

The proposed FLWA algorithm for Malaysian banknote reader's detection accuracy and processing speed is compared with two state-of-the-art methods: 1) MobileNet model using RMSprop Loss Function (learning_rate = 0.0001) at TensorFlow's Keras API [8] and 2) VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API [9]. The computing specifications is Jupyter Notebook with Windows 11 Home Single Language, 11th Gen Intel (R) Core (TM) I7-11800H@ 2.3 GHz, NVIDIA GeForce RTX3060 Laptop GPU 16 GB Memory.

Experimental setup for method 1: It is understood that the model MobileNet with Loss Function RMSProp was selected with its best accuracy of about 96.80% in paper [8]. Convolutional Neural Networks using MobileNet model with Loss Function RMSProp (0.0001) optimization technique being trained with one-thousand Malaysian banknotes (RM1, RM5, RM10, RM20, RM50 and RM100) and tested with another one-thousand real (RM1, RM5, RM10, RM20, RM50 and RM100) and one-thousand fake (RM1, RM5, RM10, RM20, RM50 and RM100) banknotes. The average time to load the model and build up the interpreter objects (Training time) was 810 seconds with batch size = 32 and epochs = 20 and the average inference time while modeling detecting banknote (Testing time) was 1 second. The overall test Accuracy is 72.95%.

Experimental setup for method 2: Total of one thousand of Malaysian real banknotes (RM1, RM5, RM10, RM20, RM50 and RM100) are captured as the dataset for training the model. VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API [9] being trained and tested with 1000 real (RM1, RM5, RM10, RM20, RM50 and RM100) and 1000 fake (RM1, RM5, RM10, RM20, RM50 and RM100) banknotes. The average time to load the model and build up the interpreter objects (Training time) was 60 seconds with batch size = 32 and epochs = 20 and the average inference time while modelling detecting banknote (Testing time) was 1 second. The overall test Accuracy is 78.83%.

The accuracy and required processing time for the experimented methods and the proposed FLWA algorithm for Malaysian banknotes reader were summarized in Table IX to Table XI below.

Table IX: Confusion matrix for Method 1 MobileNet Model.

Accuracy						**Processing Time**	
		True positive	*True negative*	*False-positive*	*False-negative*	*Training Time (Per 1000 banknotes)*	*Banknote Detection Time (Per banknote)*
MobileNet model using RMSprop Loss	RM1	570	1000	0	430	810 Second	1 Second
	RM5	130	1000	0	870	810 Second	1 Second
	RM10	340	1000	0	660	810 Second	1 Second

Function (learning_rate =0.0001) at TensorFlow's	RM20	370	1000	0	630	810 Second	1 Second
	RM50	990	1000	0	100	810 Second	1 Second
	RM100	690	1000	0	310	810 Second	1 Second

Table X: Confusion matrix for Method 2 VGG16 Model.

Accuracy						**Processing Time**	
		True positive	*True negative*	*False-positive*	*False-negative*	*Training Time (Per 1000 banknotes)*	*Banknote Detection Time (Per banknote)*
VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API	RM1	1000	40	960	0	600 Second	1 Second
	RM5	1000	1000	0	0	600 Second	1 Second
	RM10	960	1000	0	40	600 Second	1 Second
	RM20	200	1000	0	800	600 Second	1 Second
	RM50	260	1000	0	740	600 Second	1 Second
	RM100	1000	1000	0	0	600 Second	1 Second

Table XI: Confusion matrix for Fuzzy Logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection.

Accuracy						**Processing Time**
		True positive	*True negative*	*False-positive*	*False-negative*	*Banknote Detection Time (Per banknote)*
Fuzzy Logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection Algorithm	RM1	978	1000	0	22	11.48 Second
	RM5	974	1000	0	26	11.48 Second
	RM10	877	1000	0	123	11.48 Second
	RM20	903	1000	0	97	11.48 Second
	RM50	1000	1000	0	0	11.48 Second
	RM100	964	1000	0	36	11.48 Second

Table XII: Computational Requirement for Training Stage in Each of the Banknote Detection Algorithm.

Method	**Add/sub**	**Multiply**	**Division**	**e^{-x}**	**< ,>**	**Pytesseract Cycle**
Fuzzy Logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection Algorithm	0	0	0	0	0	0
Method	**Optimizing Cycle**					
VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API	$TV \times TS^2 \times C_{In} \times L \times (K^2 \times C_{Out} \times 64 + 64 \times 64 \times TS/2 \times TS/2 + 64 \times 128 \times TS/4 \times TS/4 + 128 \times 128 \times TS/8 \times TS/8 + 128 \times 256 \times TS/16 \times TS/16 + 256 \times 256 \times TS/16 \times TS/16 + 256 \times 512 \times TS/32 \times TS/32 + 512 \times 512 \times TS/32 \times TS/32 + N \times M)$ $=TV\ TS^2\ C_{In}\ L\ (64 K^2\ C_{Out} + 1024 TS^2 + 512 TS^2 + 256 TS^2 + 128 TS^2 + 256 TS^2 + 128 TS^2 + 256 TS^2 + NM)$ **$=TV\ TS^2\ C_{In}\ L\ (64 K^2\ C_{Out} + 2560 TS^2 + NM)$ optimizing cycles.**					
MobileNet model using RMSprop Loss Function (learning_rate=0.0001) at TensorFlow's	$TS^2 \times C_{In} \times L \times F \times (K^2 \times C_{In} \times C_{Out})$ **$= TS^2 K^2 L\ F\ C_{In}^2\ C_{Out}$ optimizing cycles.**					

Table XIII: Computational Requirement for Detection Stage in Each of the Banknote Detection Algorithm.

Method	Add/sub	Multiply	Division	e^{-x}	< ,>	Pytesseract Cycle

Fuzzy Logic based Weighted Averaging (FLWA) Malaysian Banknote Counterfeit Detection Algorithm	$5HW$ $+7HW$ $+12HW$ $+10HW$ $+9HW$ $+10HW$ $+15\,HW$ $+10HW$ $= \mathbf{78}\,\boldsymbol{HW}$	$8\,HW$ $+8HW$ $+8\,HW$ $+8\,HW$ $+8\,HW$ $+8\,HW$ $+8\,HW$ $+8\,HW$ $= \mathbf{64}\,\boldsymbol{HW}$	$2\,HW$ $+2HW$ $+2\,HW$ $+2\,HW$ $+2\,HW$ $+2\,HW$ $+2\,HW$ $+2\,HW$ $= \mathbf{16}\,\boldsymbol{HW}$	$1\,HW$ $+1HW$ $+1\,HW$ $+1\,HW$ $+1HW$ $+1\,HW$ $+1\,HW$ $+1\,HW$ $= \mathbf{8}\,\boldsymbol{HW}$	$2\,HW$ $+1HW$ $+1HW$ $+2HW$ $+2HW$ $+2HW$ $+3HW$ $+2HW$ $= \mathbf{15}\,\boldsymbol{HW}$	**3 PC**
Method	**Computation Operations**					
VGG16 model using 2D Convolution Layer (32 neural) at TensorFlow's Keras API	***TS*** **x** ***N*** **x** ***M*** **x** ***P*** **Calculation steps**					
MobileNet model using RMSprop Loss Function (learning_rate=0.0001) at TensorFlow's	***TS*** **x** ***N*** **x** ***M*** **x** ***P*** **Calculation steps**					

The overall computation operations required for training stage and detection stage are also listed in Table XII and Table XIII respectively. The Computational complexity are counted based on the 3 categories:

1) **Computational** of **operations** that are used in the models (addition, subtraction, multiplication, division, exponential and comparative). This category has the lowest complexity among the three since it relates simple maths operations.
2) **Computational** of **features** used for the prediction in the models (pytesseract cycle). This category has the medium complexity among the three since it relates moderate feature/character extraction cycle.
3) **Computational** of parameters that the model (Neural Nets) is **optimizing** (Optimizing cycle). This category has the highest complexity among the three since it relates process function calculates correction as a result.

In the banknote detection algorithm, there are TWO (2) main parameters to determine the number of computation operations: number of pixels in 1 horizontal line per banknote image (width of input image), W and number of horizontal lines of pixels in one banknote image, H. There are ONE (1) main parameter to determine the computation feature: pytesseract cycle (PC).

There are TWELVE (12) main parameters to determine the number of computational optimizations: total number of Training and validation images, TV; the height and width of the input image, H and W respectively; Kernel Size (batch_size x epochs), K; Target size of resize input image, TS; Number of Input Channel, C_{In}; Number of Output Channel, C_{Out}; the number of filters in each layer, F; the number of layers in the network, L; the number of input neurons, hidden neurons and output neurons, N, M, P respectively. However, among these parameters, the computation complexity is dominant by TS, which are the resolution of the resized images. The TV, K, F, L and other parameters are normally small compared to TS.

From Table XII, it can be observed that VGG16 model is on the order of $O(TS^4)$ during training operations, while MobileNet model is on the order of $O(TS^2)$ during training operations. FLWA model has no training operations. Hence it may conclude that VGG16 model has higher computation complexity compared to MobileNet model in enrolment stage. On the other hand, FLWA model requires no database of reference image and no training operations at all.

The overall computation operations required for detection stage (input 1 image for banknote detection) using FLWA, VGG16 and MobileNet models are listed in Table 3.2. From Table XIII, it can be observed that VGG16 and MobileNet are on the order of $O(TS)$ during banknote detection operations, in terms of computation operation. These two algorithms are having equal big-Oh computation complexity in banknote detection. On the other hand, FLWA is on the order of $O(PC)$ during banknote detection operations, in terms of computation feature. VGG16 and MobileNet models supersede the FLWA model in terms of detection simplicity and has lower computational time consumption.

The MobileNet model and VGG16 model have the advantage of fast banknote recognition systems. They were initially banknote denomination detection techniques. They lack feature extraction methods to extract banknote watermarks, which leads to a lower accuracy in Malaysian banknote counterfeit detection. They also require a very large amount of enrolment data in order to get better accurate performance than other parallel techniques.

The proposed FLWA model has the advantage of a much simpler algorithm since it is a human guidance learning algorithm that does not require enrolment process to get the specific weights for each security feature. Each security feature is treated with equal weight. FLWA model also outperform the MobileNet model and VGG16 model in Malaysian banknotes counterfeit detection. It has a distinct advantage over earlier or current banknote counterfeit detection techniques in that it adopted the known watermarks features, with known machine learning techniques to identify real Malaysian banknotes and detect those counterfeit Malaysian banknotes.

V. CONCLUSION

In this paper, a fuzzy logic based weighted averaging (FLWA) method had been integrated with watermark features extraction method into Malaysian banknotes reader for Malaysian banknotes counterfeit detection. Overall, the FLWA model has an accuracy of up to 97.46% in recognizing the correct banknote value and counterfeit detection. Noteworthy attention is placed on False Positive and False Negative cases for the detection algorithm, because these two cases may cause the visually impaired person to lose credits in their business. It is found that those tested real RM1 and RM5 banknotes were not placed properly into the Malaysian banknote reader (center of the banknote slot) and the Malaysian banknotes reader had mistreated some other areas on the corresponding real banknote as the three Region of interest area, and this further identified the real RM1 or RM5 as the fake RM1 or RM5. To overcome this problem, normalized sizes were assigned on RM1 and RM5 at the Step 2 Algorithm (Banknote Position Detection and Re-adjustment) to better locking the three Region of interest area. It is also found that the Coloured Glossy Patch, Two Colour Fluorescent elements, Watermark Portrait and Colour Shifting security thread watermarks were not detected in some of the tested real RM10 and RM20 banknotes, due to faded banknotes presented in the test. This further affected three of the seven measures in RM10 and RM20 security features. To overcome this problem, specific weights with less emphasis were assigned to these easily faded watermarks and more emphasis weight on other easily detected watermarks to identify RM10 and RM20 real banknotes. The proposed FLWA algorithm detection accuracy and processing speed are also compared with two state-of-the-art methods: 1) MobileNet model and 2) VGG16 model. The MobileNet model and VGG16 model have the advantage of fast banknote recognition systems. They were initially banknote denomination detection techniques. They lack feature extraction methods to extract banknote watermarks, which leads to a lower accuracy in Malaysian banknote counterfeit detection compared to FLWA model. They also require a very large amount of enrolment data in order to get better accurate performance than other parallel techniques. The proposed FLWA model has the advantage of a much simpler algorithm, since it is a human guidance learning algorithm that does not require enrolment process to get the specific weights for each security feature. Each security feature is treated with equal weight. FLWA model also outperform the MobileNet model and VGG16 model in Malaysian banknotes counterfeit detection. Future studies may investigate several essential aspects to enhance the proposed algorithm, this includes covering more tests on worn and heavily faded banknotes to cater to those dynamic issues, and more security elements like security fibres and Intaglio watermarks. Security fibres are unable to cover in the current research due to the limitation of having a good resolution UV imaging tools to capture these micro-meters particles shown on the banknotes. Besides, Intaglio watermarks were also excluded from current research due to it is a raised printing effect by touch. Intaglio watermarks required a specially designed mechanism to extract. These two security elements will be added to make the banknote detection more resilient and reliable, allowing the Malaysian banknote counterfeit detection algorithm to be used in real-time with good counterfeit detection accuracy. The Malaysian banknote reader with counterfeit detection system can also be expanded to support additional foreign currencies reading in the future.

ACKNOWLEDGEMENT

The supporter of this research is the Fundamental Research Grant Scheme (FRGS) under the Ministry of Higher Education of Malaysia. (Grant no. MMUE/190246).

REFERENCES

[1] Z. Ahmed, S. Yasmin, M. Nahidul Islam and R. U. Ahmed, "Image Processing Based Feature Extraction of Bangladeshi Banknotes," *in The 8th Int. Conf. Software, Knowledge, Inform. Manage. and Appl.*, Dhaka, pp. 1-8, 2014.

[2] A. Tushar, B. Gajanan, W. Pratik and P. Chitra, "Fake Currency Detection Using Image Processing," *in IOP Conf. Series: Mater. Sci. and Eng.*, 263, 052047, 2017.

[3] S. Mrutunjay, O. Preetam and A. Konidena. "Image Processing Based Detection of Counterfeit Indian Bank Notes," *in 9th Int. Conf. Comput., Comm. and Network. Technol.*, pp. 1-5, 2018.

[4] S. M. Sabat and N. A. Twana, "Image-Based Processing of Paper Currency Recognition and Fake Identification: A Review," *Technium*, vol. 3, no. 7, pp. 46-63, 2021.

[5] J. M. R. Apoloni, S. D. G. Escueta and J. T. Sese, "Philippine Currency Counterfeit Detector using Image Processing," *in IEEE 18th Int. Colloq. Signal Process. & Appl.*, Selangor, pp. 436-441, 2022.

[6] A. Z. Lotfi and A. A. Rafik, *Fuzzy Logic Theory and Applications: Part I And Part II*, World Scientific, 2018.

[7] C. Jenny, C. Francisco, S. K. Arjab and C. Tianhua, *Fuzzy Logic: Recent Applications and Developments*, Springer Nature, 2021.

[8] D. T. Aseffa, H. Kalla and S. Mishra, "Ethiopian Banknote Recognition Using Convolutional Neural Network and Its Prototype Development Using Embedded Platform", J. Sensors, vol. 2022, 4505089, pp. 1-18, 2022.

[9] S. Nanda, M. Abbas, N. Momaya and K. A. Mahesh, "Indian Currency Detection for Blind People with VGG16," *Int. J. Innov. Res. in Technol.*, vol. 8, no. 2, pp. 541-544, 2021.

[10] M. Mohri, A. Rostamizadeh and A. Talwalkar, *Foundations of Machine Learning*, The MIT Press, 2012.

[11] S. Patel, R. Nargunde, C. Shah and S. Dholay, "Counterfeit Currency Detection using Deep Learning," *in 12th Int. Conf. Comput. Comm. and Network. Techn.*, Kharagpur, pp. 1-5, 2021.

[12] D. Saire and S. Tabbone, "Documents Counterfeit Detection Through A Deep Learning Approach," *in Proc. Int. Conf. Pattern Recogn.*, pp. 3915–3922, 2020.

[13] K. N. N. Hlaing and A. K. Gopalakrishnan, "Myanmar Paper Currency Recognition Using GLCM and k-NN,*" in 2nd Asian Conf. Def. Technol.*, 67–72, 2016.

[14] C. Rahmad, E. Rohadi and R. A. Lusiana,"Authenticity of Money Using the Method KNN (K-Nearest Neighbor) and CNN (Convolutional Neural Network)," *in IOP Conf. Series: Mater. Sci. and Eng.*, vol. 1073, 012029, 2021.

[15] I. A. Doush and A. B. Sahar, "Currency Recognition Using A Smartphone: Comparison Between Color SIFT and Gray Scale SIFT Algorithms," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29(4), pp. 484–492, 2017.

[16] G. Anahita, A. Jamilu and A. B. Azuraliza, "Comparative Analysis of Algorithms in Supervised Classification: A Case study of Bank Notes Dataset," *Int. J. Comput. Trends and Technol.*, vol. 17, no. 1, pp. 39-43, 2014.

[17] N.A. Jasmin Sufri, N.A. Rahmad, M.A. As'ari, N.A. Zakaria, M.N. Jamaludin, L.H. Ismail and N.H. Mahmood, "Image Based Ringgit Banknote Recognition for Visually Impaired*," J. Telecom. Electron. and Comput. Eng.*, vol. 9, no. 3–9, pp. 103-111, 2017.

[18] N. A. J. Sufri, N. A. Rahmad, N. F. Ghazali, N. Shahar and M. A. As'ari, "Vision Based System for Banknote Recognition Using Different Machine Learning and Deep Learning Approach," *in IEEE 10th Contr. and Sys. Graduate Res. Colloq.*, Shah Alam, pp. 5-8, 2019.

[19] S. Gopane and R. Kotecha, "Indian Counterfeit Banknote Detection Using Support Vector Machine," *SSRN Electron. J.*, 3568724, 2020.

[20] C. Yeh, W. Su and S. Lee, "Employing Multiple-kernel Support Vector Machines for Counterfeit Banknote Recognition," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 1439-1447, 2011

[21] B. Huaytalla, D. Humari and G. Kemper, "An algorithm for Peruvian counterfeit Banknote Detection Based on Digital Image Processing and SVM," *Advance. Sci., Technol. and Eng. Sys. J.*, vol. 6, no. 1, pp. 1171-1178, 2021.

[22] W. K. Wong, C. J. Tan, T. S. Min and E. K. Wong, "Fuzzy Logic Based Perceptual Image Hashing Algorithm in Malaysian Banknotes Detection System for the Visually Impaired," *Artif. Intellig. Advanc.*, vol. 3, no. 1, pp. 52–64, 2021.

[23] Bank Negara Malaysia, "Current Banknote Series," *Central Bank of Malaysia*. https://www.bnm.gov.my/current-banknote-series (accessed on Aug. 17, 2022).

[24] S. Bhutada, N. Yashwanth, P. Dheeraj and K. Shekar, "Opening and Closing in Morphological Image Processing," *World J. Advanc. Res. and Rev.*, vol. 14, no. 3, pp. 687–695, 2022.

[25] W. S. Alazawee, I. Abdel-Qader and J. Abdel-Qader, "Using Morphological Operations - Erosion Based Algorithm for Edge Detection," in *IEEE Int. Conf. Electron. Inform. Technol.*, vol. 2015-June, pp. 521–525, 2015.

[26] B. Y. Nesrine, B. B. Saoud and B. G. Henda, "Integrating Fuzzy Case-based Reasoning and Particle Swarm Optimization to Support Decision Making." *Int. J. Comput. Sci.*, no. 9(3), pp. 117-124, 2012.

[27] S. Ali, *Introductory Chapter: Which Membership Function is Appropriate in Fuzzy System?*, 79552, 2018.

[28] W. K. Ling (2007). *Chapter 2 – REVIEWS, Nonlinear Digital Filters*, Academic Press, pp. 8-31, 2007.