DEEP LEARNING-BASED FAKE-BANKNOTE DETECTION FOR THE VISUALLY IMPAIRED PEOPLE USING-LIGHT IMAGES CAPTURED BY SMARTPHONE CAMERAS

YONG NGEE MANG

B.ENG (HONS.) ELECTRICAL ENGINEERING (ELECTRONICS)

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : YONG NGEE MANG

Date of Birth : 17 MAY 1997

Title : 1E46 Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People Using Visible-Light Images Captured By Smartphone Cameras

Academic Session : 2021/2022

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
☑	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)	(Supervisor's Signature)
	IKHWAN HAFIZ BIN MUHAMAD
970517-13-6339	
New IC/Passport Number	Name of Supervisor
Date: 7 February 2022	Date: 21/06/2022

NOTE: * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis, and, in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Electrical Engineering (Electronics).

(Supervisor's Signature)

Full Name : IKHWAN HAFIZ BIN MUHAMAD

Position : LECTURER

Date : 21/06/2022

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at University Malaysia Pahang or any other institutions.

____

(Student's Signature)

Full Name : YONG NGEE MANG

ID Number : EA18172

Date : 17 June 2022

DEEP LEARNING-BASED FAKE-BANKNOTE DETECTION FOR THE VISUALLY IMPAIRED PEOPLE USING VISIBLE-LIGHT IMAGES CAPTURED BY SMARTPHONE CAMERAS

YONG NGEE MANG

Thesis submitted in fulfillment of the requirements
for the award of the
B.Eng (Hons.) Electrical Engineering (Electronics)

College of Engineering

UNIVERSITI MALAYSIA PAHANG

JUNE 2022

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Mr. Ikhwan Hafiz bin Muhamad for the continuous support of my final project and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this report. I could not have imagined having a better advisor and mentor for my final project.

Last but not the least, I would like to thank my family: my parents Ting King Ai and Yong Jin Hui for give me support and encouragement so that I could successfully complete this report.

ABSTRAK

Pengecaman automatik wang kertas palsu adalah tugas penting dalam pengendalian wang kertas praktikal. Penyelidikan mengenai tugas ini kebanyakannya melibatkan kaedah yang digunakan pada mesin pengisihan automatik dengan berbilang penderia pengimejan atau yang menggunakan penderia khusus untuk menangkap imej wang kertas dalam pelbagai panjang gelombang cahaya. Walau bagaimanapun, mereka memerlukan peranti khusus. Sementara itu, telefon pintar menjadi lebih popular dan boleh menjadi peranti pengimejan yang berguna. Projek ini akan menyiasat dan mencadangkan kaedah terbaik untuk mengklasifikasikan wang kertas palsu dan asli menggunakan imej cahaya nampak yang ditangkap oleh kamera telefon pintar berdasarkan rangkaian saraf konvolusi. Projek ini akan memberi tumpuan kepada wang kertas Malaysia sahaja. Akhir sekali, keputusan ketepatan, panggil balik dan kerugian bagi projek ini ialah 0.849, 0.971 dan 0.011586.

ABSTRACT

Automatic recognition of fake banknotes is an important task in practical banknote handling. Research on this task has mostly involved methods applied to automatic sorting machines with multiple imaging sensors or that use specialized sensors for capturing banknote images in various light wavelengths. However, they require specialized devices. Meanwhile, smartphones are becoming more popular and can be useful imaging devices. This project will investigate and propose the best method for classifying fake and genuine banknotes using visible-light images captured by smartphone cameras based on convolutional neural networks. This project will focus on Malaysia banknotes only. Finally, the result of precision, recall and loss for this project are 0.849, 0.971 and 0.011586.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Although electronic financial transactions are becoming more popular and the use of paper money has been decreasing recently, but the banknotes remain in recirculation owing to their reliability and simplicity in usage. Then the problem of fake banknote is still relevant. So, the purpose of this project is to recognise the banknote type and denomination, counterfeit detection, and fitness classification. Specialized device with multiple sensors for detecting anti-counterfeit features can easily recognize this kind of fake banknote. However, these devices are not readily available to some users, especially the visually impaired people. Meanwhile, smartphones with cameras are becoming more popular. Considering these issues, a fake banknote recognition method is proposed based on banknote images captured by smartphone cameras in visible-light conditions.

1.2 Problem Statement

The importance of detecting counterfeit banknotes is that the counterfeiters are elusive. Although the police did their best to confiscate the counterfeit money, it is still difficult to find the source of the counterfeit money. Because once the counterfeit banknotes are successfully circulated in the market, the police will have no way to prove who made the counterfeit banknotes. Therefore, the risk of criminals making counterfeit banknotes being caught by the police is very low. Besides that, we can classify criminals as low-level or high-level according to the quality of counterfeit banknotes. Low-level criminals use colour scanners and printers to make low-quality counterfeit banknotes. But compared to high-level criminals, they have a large budget and the most advanced technology to make counterfeit banknotes that are difficult to identify [2]. So, we need to be well informed and vigilant about counterfeit detection, use the latest technology to detect counterfeit banknotes. Moreover, counterfeit banknotes in circulation are harmful to the economy because they devalue the currency. There is also an increase in prices (inflation) due to more counterfeit banknotes circulating in the economy. Clearly, counterfeit detection is of vital importance to country’s economy [2].

Figure 1 show the statistics of counterfeiting rate of fake banknotes worldwide from 2014 to 2018. Counterfeiting rates across countries are affected by several factors including the broader crime rate, the security of a currency's banknotes, how cash is used, and the cost of equipment used to counterfeit banknotes. The United States has the highest rate of counterfeit banknotes because the dollar is a more widely used currency. For statistics of counterfeiting banknote in Malaysia, while there were isolated cases of counterfeit notes in the country, the rate was still quite low. For example, 2016, the ratio was 1.4 parts per million (PPM) and in 2019, it was only 1.2 parts per million (PPM) and up until the middle of 2020, the rate is 1.0 parts per million (PPM) [1]. However, the counterfeit notes Malaysia is decrease, this project is still important to detect the counterfeit because it is an effective tool to let the counterfeit note rate become lower in Malaysia.

Next problem statement is Visually Impaired People hard to recognise the banknote because some visually impaired people recognise banknotes through the length and width of banknote [20]. So, we need to develop to help them to recognize the true denomination of banknote and avoid them receive the fake banknote.

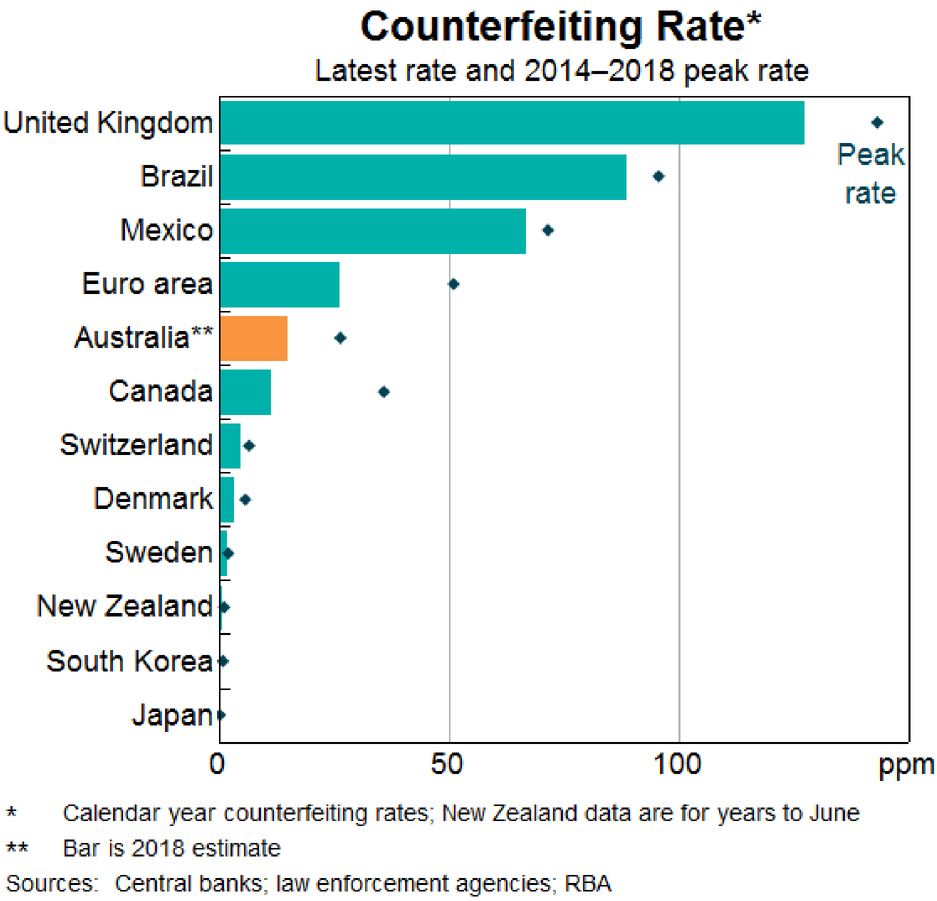


Figure 1: Statistics of counterfeiting banknote worldwide [2]

1.3 Objective

The first objective is to detect and recognised the banknote to classify their correct denomination such as RM1, RM5, RM10, RM50, and RM100. The second objective is to make sure the algorithm can perform well in real-time. The third objective is to detect and classify counterfeit money.

- Develop algorithm to identify and classify the denomination of banknotes.
- Develop algorithm to recognise and classify counterfeit banknotes
- Analyze the developed algorithm performance in real-time.

1.4 Scope

The first scope is this project would focus the Malaysia Ringgit only. The second scope is this project would focus on banknotes only and not focus on coin. The third scope is samples are taken in controlled environment such as white background, bright lighting, etc. Next scope is the model could detect front of the banknote only. Moreover, focus algorithm based on SSD-MobileNet, YOLOv5s and YOLOv5s6 for performance comparison. The last scope is developed UI by using Qt designer.

- The project will focus on Malaysia Ringgit.
- The project will focus on banknotes only.
- Samples are taken in controlled environment such as white background, bright lighting, etc.
- Detect front of the banknote only.
- Focus algorithm based on SSD-MobileNet, YOLOv5s and YOLOv5s6 for performance comparison.
- Developed UI by using Qt designer.

CHAPTER 2

LITERATURE REVIEW

Before start to do this project, I need to do the literature review to research the information with related for this project such as article, thesis, etc. A literature review is a piece of academic writing demonstrating knowledge and understanding of the academic literature on a specific topic placed in context. So, do the literature review is for me to learning more about the information of this project and that could help me to design this project. Figure 2 show the overview literature review for I had done in this project.

Based on Figure 2, much research related to recognition of the genuine and fake banknote are reviewed to find a best method to be used in this project. The method find from the research paper includes sensor-based approach, image-based approach, machine learning based approach, and real-time based approach. For sensor-based approach, various sensors are used to detect or screen the banknote such as ultraviolet (UV) sensor, Infrared sensor and so on. For image-based approach, various deep learning model are used to train the model to detect the banknote image either it is genuine or counterfeit such as convolutional neural network (CNN), Single shot multi box detector (SDD), etc. For machine learning based approach, Artificial Neural Network (ANN) and Genetic Algorithm (GA) are reviewed. For real-time based approach, various method to detect the object in real-time reviewed include Convolutional Neural Network (CNN), Recursive Neural Network (RNN) and Region-based Convolutional Neural Network (R-CNN), and MobileNet-SSD (Shot Multi Box Detector).

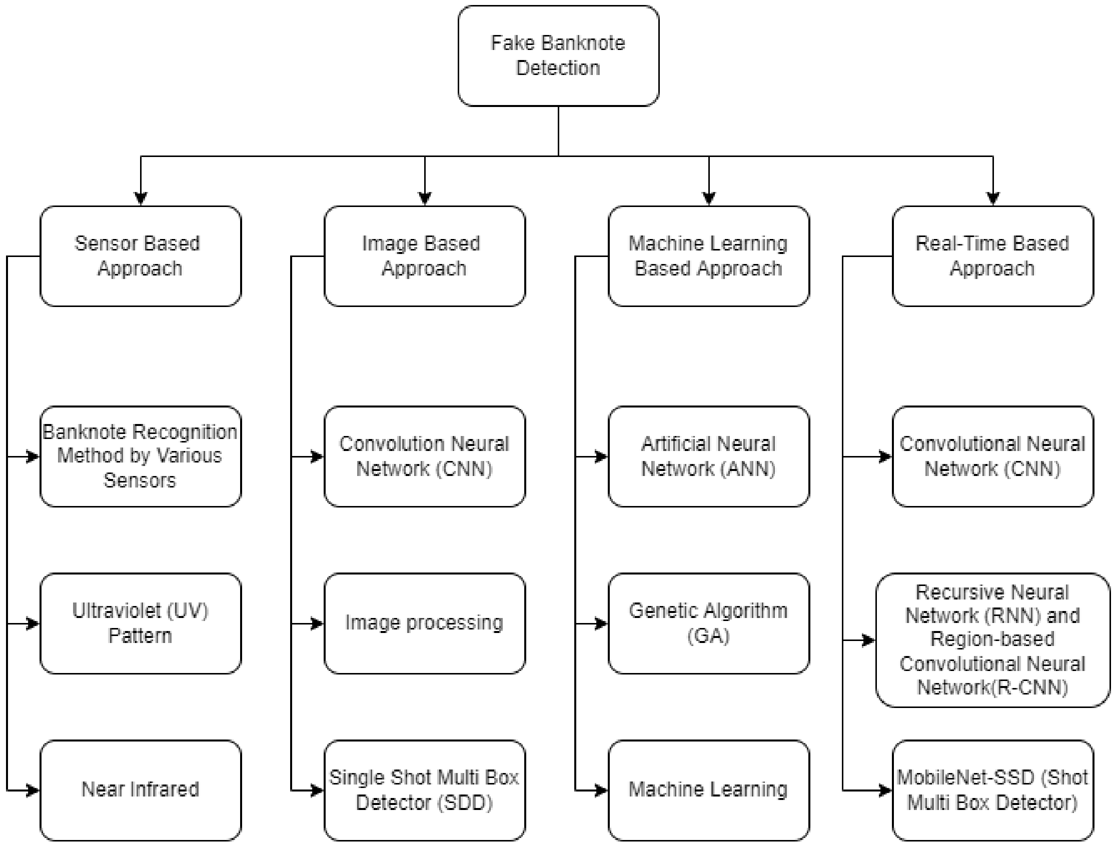


Figure 2: Overview of Literature Review

2.1 Sensor-based approach

2.1.1 A survey on Banknote Recognition Methods by Various Sensors

Research by Ji Woo Lee, Hyung Gil Hong, Ki Wan Kim and Kang Ryoung Park. Used various sensors (ultraviolet, near infrared, etc) to scan the serial number of input banknote to discern the image and data necessary for recognizing denomination and anti-counterfeiting features. Figure 3 show the process flow to recognise banknote by using automated device [4].

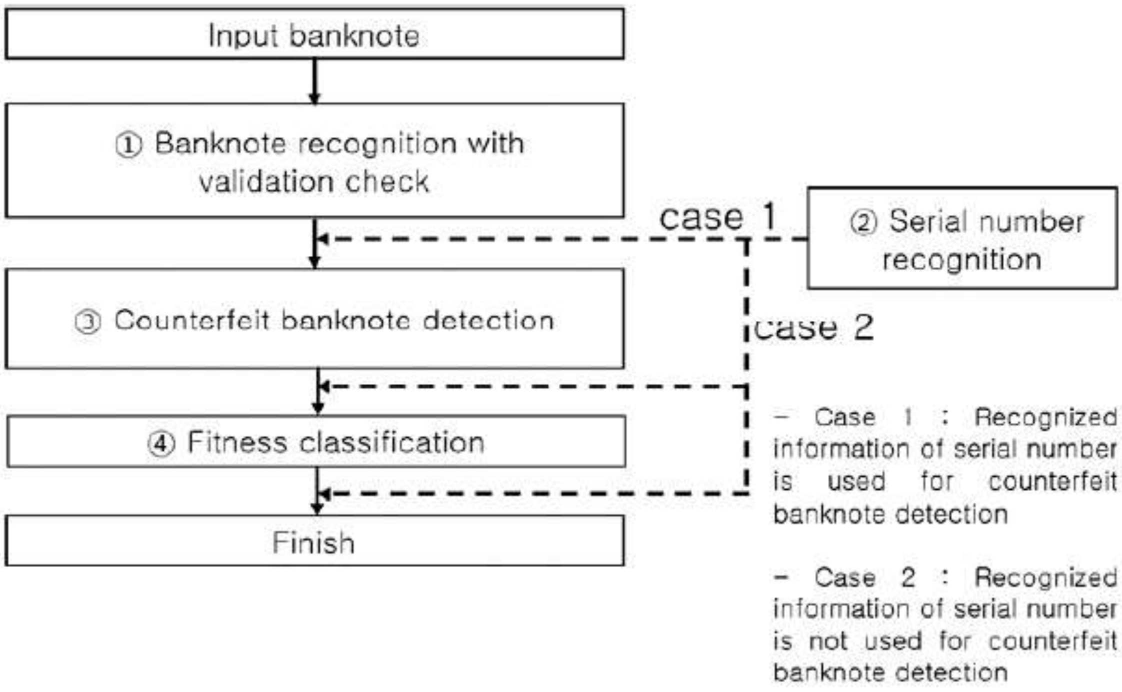


Figure 3: Banknote recognition process flow in an automated device [4]

There are two cases which are recognized information of serial number or not. If in case 1, we could classify counterfeit banknote. For case 2, the coordinate mapping between recognized banknote image and sensor data for counterfeit detection is done. For judging the genuineness of banknote, the banknote recognition step for the validation checks such as the input denomination, input side, direction, deflection, inclination to combining the input data with infrared sensors (IR), Ultraviolet Sensor (UV) and MG sensor data related to individual anti-counterfeiting is applied. After that, the sensor signals are matched in predetermined of region of interest (ROIs) to extract the anti-counterfeiting features.

The last step is fitness classification, to classify the soiling level of banknote by using visible light and near infrared (NIR) image information. The results reliability of this project is at the accuracy of 95%. For this method, many steps and many sensors.

2.1.2 Image Segmentation of UV Pattern for Automatic Paper-Money Inspection

A research Keon-Ho Lee and Tae-Hyoung Park. Used ultraviolet (UV) patterns embedded in the banknote and segmented to determine the banknote is genuine or counterfeit and Gaussian mixture model (GMM) to segment the pattern from the background. Figure 4 show the process flow of segment the banknote ultraviolet (UV)-pattern [5].

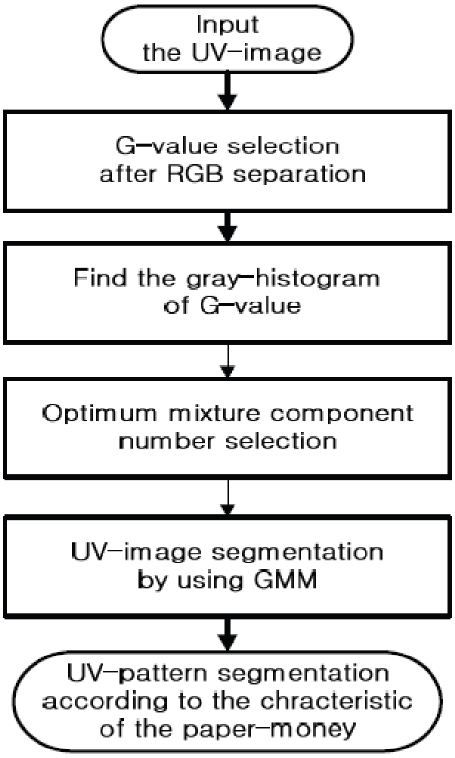


Figure 4: The method of ultraviolet (UV)-pattern segmentation [5]

The banknote is input by using ultraviolet (UV)-LED illumination as shown in Figure 5. Segmentation of the ultraviolet (UV)-image is done by using Gaussian mixture model (GMM) as shown in Figure 6 the ultraviolet (UV)-image was segmented into five parts. Judgment was performed according to the characteristic of the banknote and Figure 7 shows the segmented image of the ultraviolet (UV) pattern according to the characteristic of the banknote. For experiment, an experimental device for obtained the banknote image is required as shown in Figure 8 and the ultraviolet (UV) pattern was extracted by using Matlab.

Figure 5: Ultraviolet (UV)-image of banknote [5]

Figure 6: Segmentation of ultraviolet (UV)-image using Gaussian mixture model (GMM) [5]

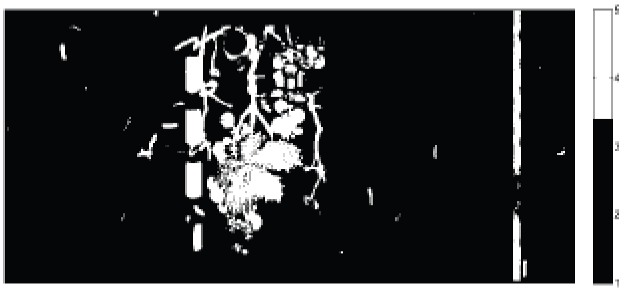


Figure 7: Ultraviolet (UV) pattern segmentation image [5]

Figure 8: An experimental device [5]

The results reliability of this research is at the accuracy of 94% for Malaysia banknote. However, this method is expensive because the experimental device requires a lot of equipments such as SCA1000-30FC CCD Camera, UV-LED illumination, and LED POWER (MDLR12-25W) illuminator.

2.1.3 Forgery Detection and Value Identification of Euro Banknotes

Research by Areangelo Bruna, Giovanni Maria Farinella, Gluseppe Claudio Guarnera and Sebastiano Battiato. Use near infrared camera to detect the banknote image. User just put the banknote on the glass, the near infrared camera would scan the banknote and show that banknote is genuine or counterfeit. Figure 9 shows the block diagram of the hardware prototype for overall system [6].

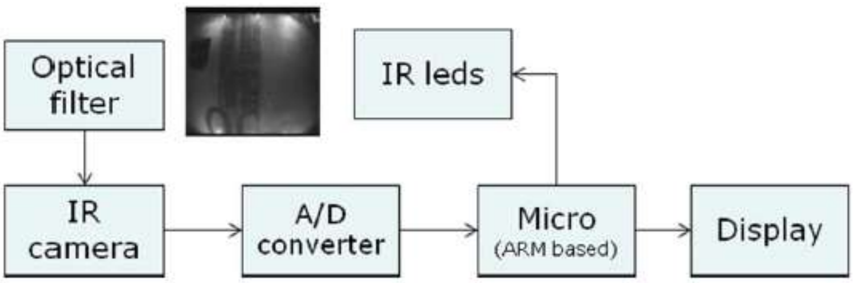


Figure 9: The schema of the hardware prototype [6]

The optical filter is to avoid an external light source from influencing the banknote image acquisition system and that filter will placed on top of the infrared camera that acquires the banknote image. Analog to digital converter is used to obtain a digital standardized format and then the digital image will be processed by microprocessor. Next, the infrared LEDs illuminate the scene and composed by six LEDs placed around the proscenium. Finally, the result is displayed eight-segment display. Figure 10 shows the final prototype of the system.

Figure 10: Final prototype of system [6]

The results reliability of this project is at the accuracy are 95.7%. This method is expensive because most of used the hardware part is quite expensive.

2.2 Image-based approach

2.2.1 Machine Learning Approach

2.2.1.1 Fake currency detection using image processing

Research by Tushar Agasti, Gajanan Burand, Pratik Wade and P Chitra. Used simple digital cameras or scanner to capture the banknote image under ultraviolet light and convert to Gray scale image. After that, image segmentation is done to extract feature and recognise the genuine or counterfeit banknote. Figure 11 show the flow diagram of the process for this project [8].

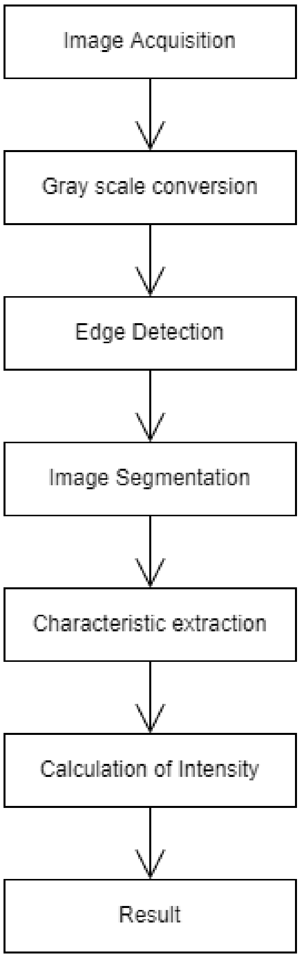


Figure 11: Flow diagram of process [8]

The banknote image is converted into gray scale image since it carries intensity information, and this image is further processed, and edges of gray scale image are detected. After that, the image segmentation is done for dividing the image into multiple parts by cropping and using edge-based segmentation to extract the features of banknote. After extracting the features, the intensity of each extracted feature is calculated for recognising the banknote is either genuine or counterfeit. If the calculated intensity is greater than the threshold of 70%, then it is classified as genuine banknote. Otherwise, it is considered as counterfeit. The results reliability of this project is very low which is at 70% only.

2.2.1.2 Banknotes Recognition in Real Time Using ANN

Research by Yueqiu Ren. Used Artificial neural network (ANN) to training the model and using digital image processing to recognise banknote in real-time. Digital image processing could extract the visual features such as colour, shape, and texture to recognise either the banknote is genuine or counterfeit [10].

In training, principle component analysis (PCA) and linear discriment analysis (LDA) are separately implemented on the LBPHistogramMatrix to reduce the dimensionality of the feature vectors. After that, the Multi-dimensional classification (MDC) and back propagation neural network (BPNN) is used to classify the banknote image. The results reliability of this project is at the accuracy of 99.4% Deep Learning Approach

2.2.1.3 Bank Note Authentication: A Genetic Algorithm Supported Neural based Approach

A Research by Spandan Sen Sarma. Used Genetic algorithm (GA) to detect authenticity of banknotes by classifying them into two separate classes [11]. The dataset is divided into two parts with 70% of the data is for training and 30% of the data is for testing. In the training phase the training dataset is supplied to different algorithms respectively to build the required classification model. In the testing phase the classification models obtained from the training phase is employed to test the accuracy of the model. Finally, the result of accuracy, precision, recall, and F-measure could be gotten by using confusion matrix as shown in Figure 12. The results reliability of this project is at the accuracy of 96.38% and the precision is 83.33%.

Figure 12: The example of Confusion Matrix [11]

2.2.2 Deep Learning Approach

2.2.2.1 Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People Using Visible-Light Images Captured by Smartphone Cameras

Research by Tuyen Danh Pham, Chanhum Park, Dat Tien Nguyen, Ganbayar Batchuluun, and Kang Ryoung Park. Is using smartphone cameras to capture the banknote image for classifying banknotes is fake or genuine based on convolution neural networks such as AlexNet, ResNet-18, GoogleNet and VGGNet. Figure 13 show the overall procedure of the proposed method to recognise genuine or counterfeit banknote [7].

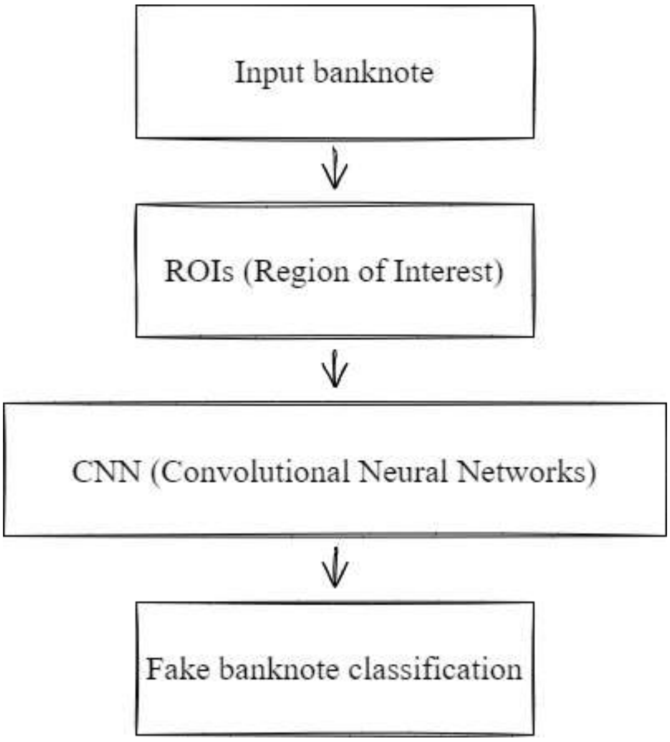


Figure 13: Overall procedure of the proposed method [7]

The banknote image with front or back side is captured using smartphone. Next, region of interest (ROIs) is used to crop center areas of banknote image to reduce the unwanted effect of the background. Also, region of interest (ROIs) will resize the banknote image to 224x224 pixel and composite image of the region of interest (ROIs) shown in Figure 14 which will be input to the Convolutional Neural Networks (CNN) model.

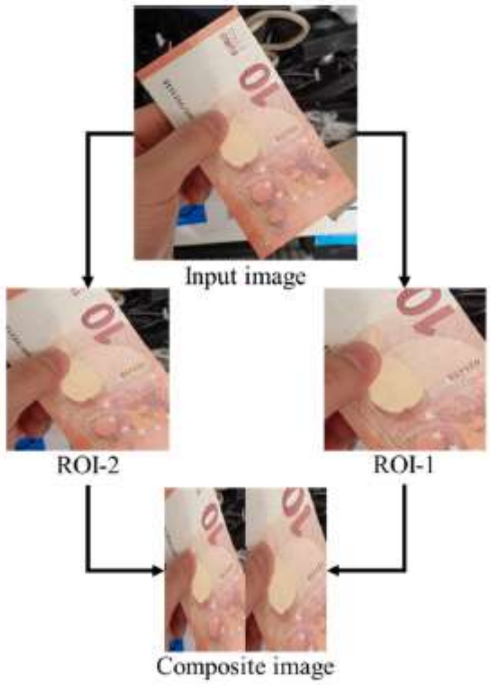


Figure 14: Example of region of interest (ROIs) selection and composite image formation [7]

The results reliability of this project is at the accuracy of 94%. This method provides high accuracy result with just simple image from smartphone camera which is similar to this project.

2.2.2.2 Currency Recognition using deep learning

Research by Qian Zhang and Wei Qi Yan. Used convolution neural networks (CNN) as a feature extractor under the framework of Single Shot Multi Bos Detector (SSD) model to recognise the banknote. Figure 15 show the process flow to create dataset and recognise the banknote [9]. The results reliability of this project is up to 96.6%.

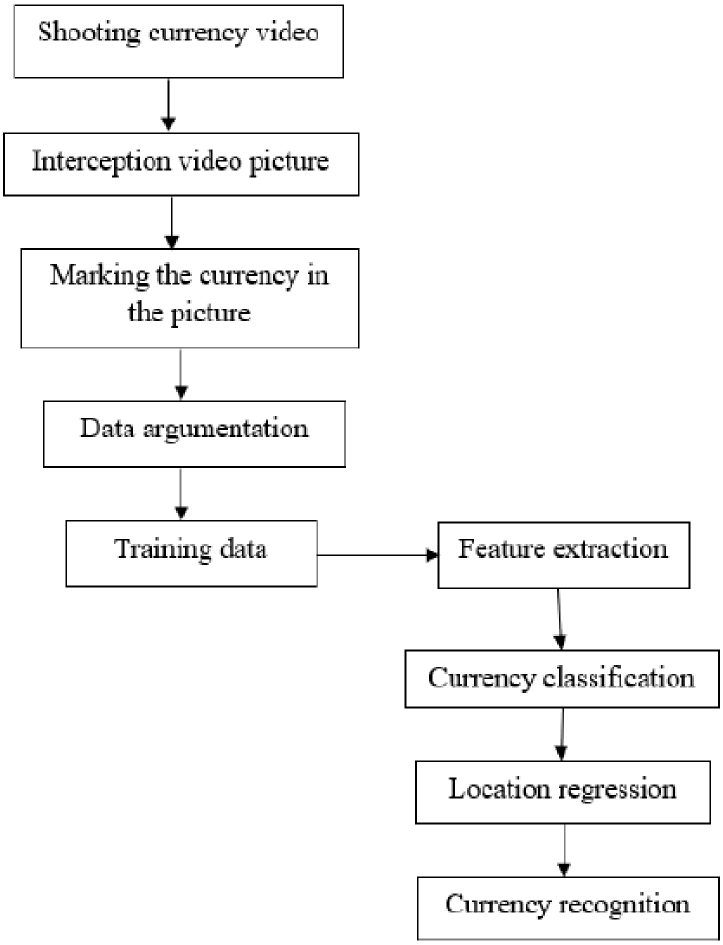


Figure 15: The steps of currency recognition [9]

2.2.2.3 Machine Learning-Based Fast Banknote Serial Number Recognition Using Knowledge Distillation and Bayesian Optimization

Research by Eunjeong Choi, Somi Chae, and Jeongtae Kim. Used machine learning to recognise the serial number of banknotes to detect either the banknote is genuine or counterfeit. It would automatically detect the region of interest (ROI) for the serial number from image banknote and used knowledge distillation to increases speed of training. Figure 16 shows the sequential region of interest (ROI) and classification system [12].

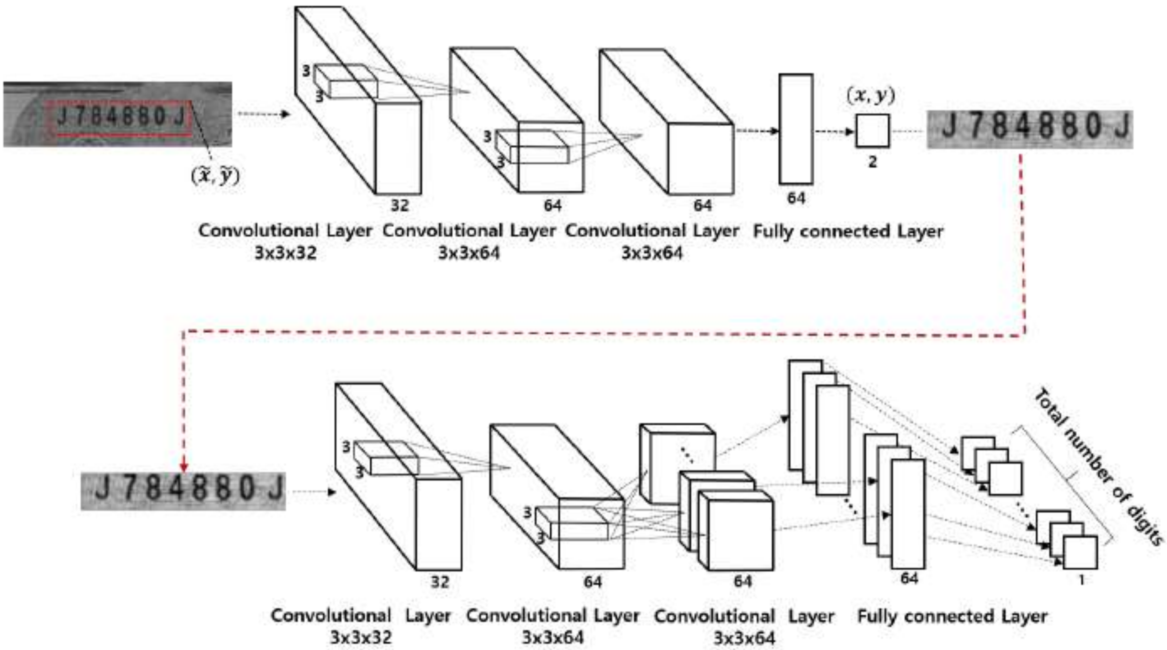


Figure 16: The sequential region of interest (ROI) detection and classification system [12]

The first convolutional neural network (CNN) detects the region of interest (ROI) for the serial number and using detected region of interest (ROI) which in the second convolutional neural network (CNN) to classifies all characters in the region of interest (ROI) simultaneously as shown in Figure 16.

The results reliability of this project is at the accuracy higher than 98%. However, this method is hard to apply in this project because it requires data serial number of the banknote.

2.2.2.4 Bangladeshi Banknote Recognition in Real-time using Convolutional Neural Network for Visually Impaired People

Research by Rahnuma Tasnim, Sadia Tasnuva Pritha, Annesha Das, and Ashim Dey. Talk about Training the Convolutional Neural Network model using 70,000 banknote images to identify the banknote in real-time. If the predictions are higher than

80% then it would show that the banknote is recognized. Otherwise, it would show unknown. Figure 17 shows the flow process of the proposed method [13].

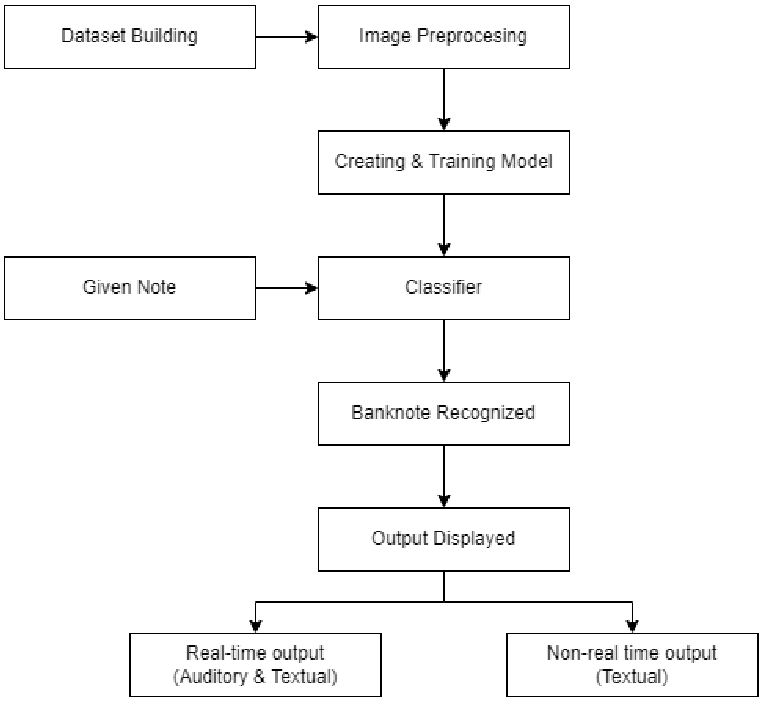


Figure 17: Overview of the Proposed Method [13]

The dataset is divided into two subsets with training is 80% of banknotes image and for testing is 20% of banknotes image. After the model has been trained, its accuracy could be tested in two ways which are real-time and non-real time. For real-time testing, it is using OpenCV to process the image by taking a frame through the webcam. Then displays the result continuously on the screen with the name of banknote. For non-real time testing, the Capture () function is using the webcam to capture the banknote image for resizes as well as reshapes it to match the dataset images. Whereas the load_predict() function is used for classifying the banknote image taken by an external camera such as mobile camera or digital cameras. If the predictions are higher than 80% then it would show the banknote is recognized. Otherwise, it would show unknown. The results reliability of this project is at the average accuracy of 92% for real-time detection and classification.

2.3 Detail Explanation of MobileNet-SSD

2.3.1 MobileNet

MobileNet-SSD is the combination of MobileNet and Single Shot Multi Bos Detector (SSD) framework for a fast and efficient deep learning-based method of object detection. In this model, the function of MobileNet is used to train the input image that have been completely removed from the associated ranking layer. The function of Single Shot Multi Bos Detector (SSD) framework is use for detecting the object in real-time. When SSD detecting the object, it will produce multiple bounding boxes to classify and label the object in real-time.

MobileNets primarily focus on optimizing for latency but also yield small networks. MobileNet also is a model that can be applied to various recognition tasks such as object detection, face attributes, fine grain classification, and landmark recognition as shown in Figure 18. The MobileNet structure is built on depthwise separable convolutions, and the first layer is a full convolution. The MobileNet architecture is defined in Table 1. All layers are followed by a batchnorm and ReLU (Rectified Linear Unit) nonlinearity except for the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification. Figure 19 shows the contrasts layer with regular convolutions, batchnorm and ReLU nonlinearity to be factorized layer with depthwise convolution, 1x1 pointwise convolution as well as batchnorm and ReLU after each convolutional layer. Countung depthwise and pointwise convolutions as separate layers, MobileNet has 28 layers [16].

Figure 18: MobileNet models can be applied to various recognition tasks for efficient on device intelligence [16]

Type / Stride	**Filter Shape**	**Input Size**
Conv / s2	3x3x3x32	224x224x3
Conv dw / s1	3x3x32 dw	112x112x32
Conv / s1	1x1x32x64	112x112x32
Conv dw / s2	3x3x64 dw	112x112x64
Conv / s1	1x1x64x128	56x56x64
Conv dw / s1	3x3x128 dw	56x56x128
Conv / s1	1x1x128x128	56x56x128
Conv dw / s2	3x3x128 dw	56x56x128
Conv / s1	1x1x128x256	28x28x128
Conv dw / s1	3x3x256 dw	28x28x256
Conv / s1	1x1x256x256	28x28x256
Conv dw / s2	3x3x256 dw	28x28x256
Conv / s1	1x1x256x512	14x14x256
5x Conv dw/s1	3x3x512 dw	14x14x512
Conv dw / s2	3x3x512 dw	14x14x512
Conv / s1	1x1x512x1024	7x7x512
Conv dw / s2	3x3x1024 dw	7x7x1024
Conv / s1	1x1x1024x1024	7x7x1024
Avg Pool / s1	Pool 7x7	7x7x1024
FC / s1	1024x1000	1x1x1024
Softmax / s1	classifier	1x1x1000

Table 1: MobileNet Body Architecture [16]

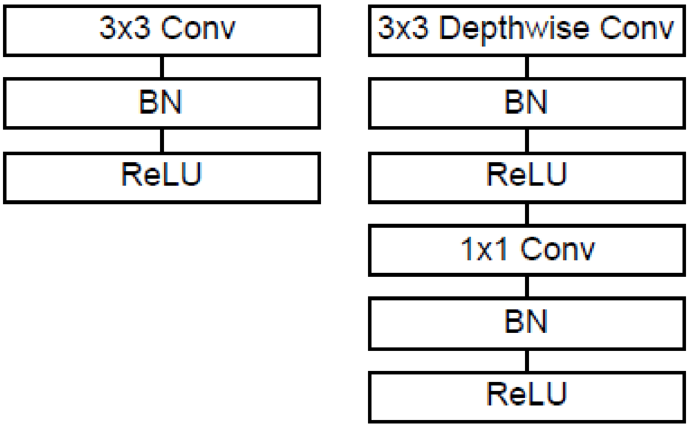


Figure 19: Left: Standard convolutional layer with batchnorm and ReLU. Right: Depthwise Separable convolutions with Depthwise and pointwise layers followed by batchnorm and ReLU [16]

2.3.2 Single Shot Multi Box Detector (SSD)

Single Shot Multi Bos Detector (SSD) is designed for object detection in real-time. SSD speeds up the process by eliminating the need for the region proposal network. To recover the drop in accuracy, SSD applies a few improvements including multi-scale features and default boxes. These improvements would let SSD match the Faster R-CNN's accuracy using lower resolution images. So, SSD is having a high processing speed to detect the object in real-time. The SSD object detection would compose of 2 parts which is extract feature maps and apply convolution filters to detect objects. Moreover, it uses the VGG16 to extract feature maps and detects the object using the Conv4_3 layer as shown in Figure 20. For convolutional predictors for object detection, SSD computes both the location and class scores using small convolution filters. After extracting the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions.

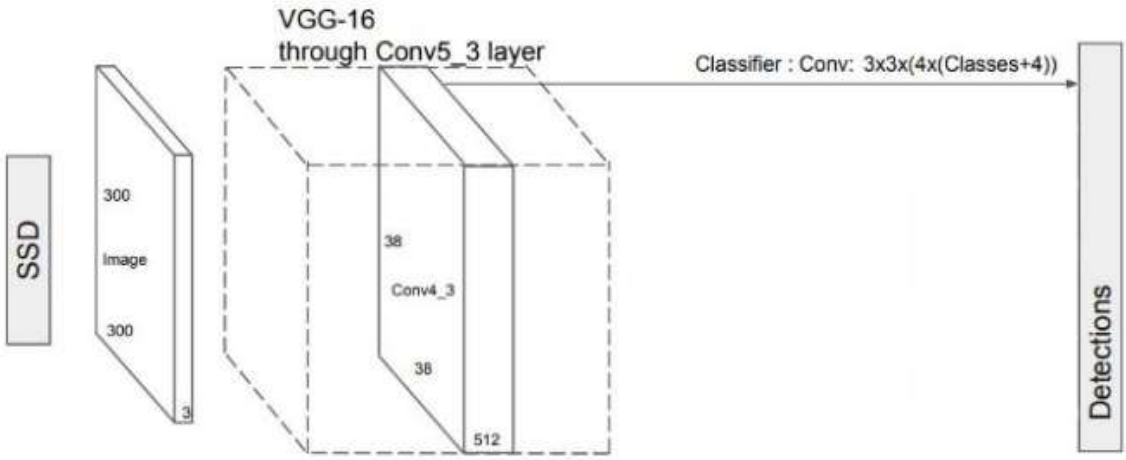


Figure 20: Single Shot MultiBox Detector (SSD) with single layer [17]

SSD detects object by using single layer. As convolutional neural network (CNN) reduces the spatial dimension gradually, the resolution of the feature maps also decreases. SSD uses lower resolution layers to detect larger scale objects. For example, the 4× 4 feature maps are used for larger scale object as shown in Figure 21. For using multiple layers to detect objects independently, would add 6 more auxiliary convolution layers were added after the VGG16 as shown in Figure 22. Five of them will be added for object detection and SSD will make 8732 predictions using 6 layers.

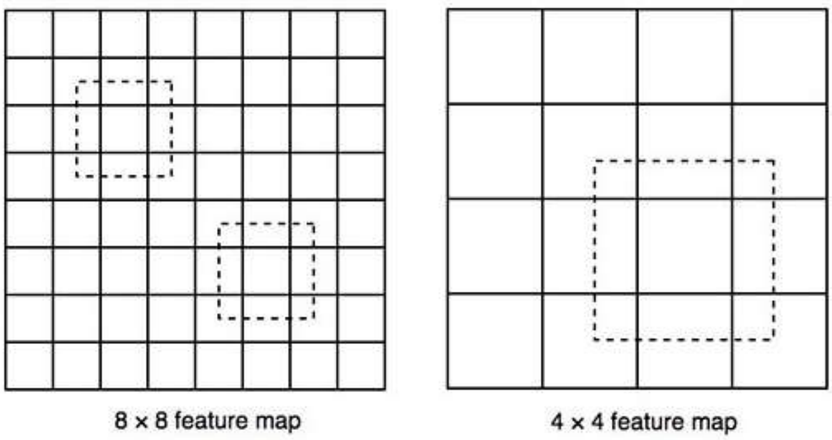


Figure 21: Lower resolution feature maps detects larger-scale objects [17]

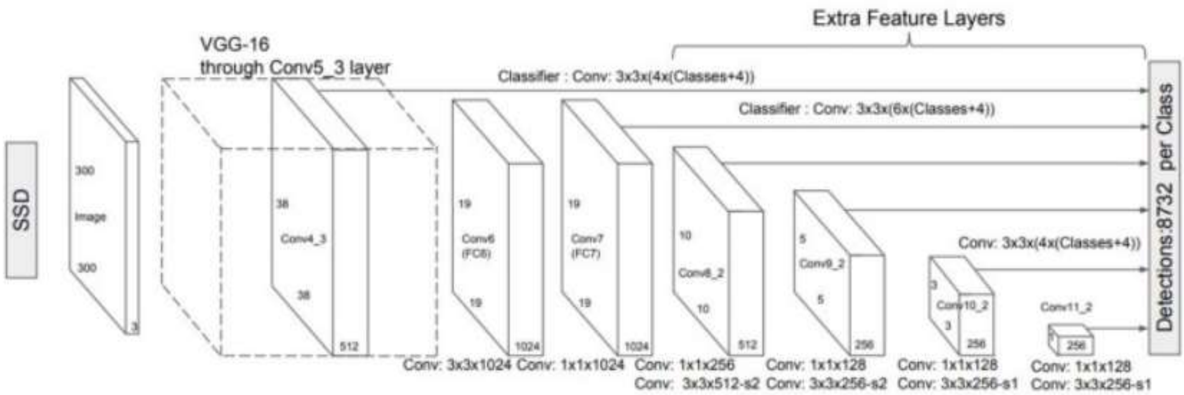


Figure 22: Single Shot MultiBox Detector (SSD) with multi-layer [17]

2.3.3 MobileNet-SSD

There are three steps to building an object detection framework. First, a deep learning model or algorithm is used to generate a large set of bounding boxes spanning the full image that is an object localization component as shown in Figure 23. Next, visual features are extracted for each of the bounding boxes. They are evaluated and it is determined whether and which objects are present in the boxes based on visual features as shown in Figure 24. In the final post-processing step, overlapping boxes are combined into a single bounding box as shown in Figure 35.

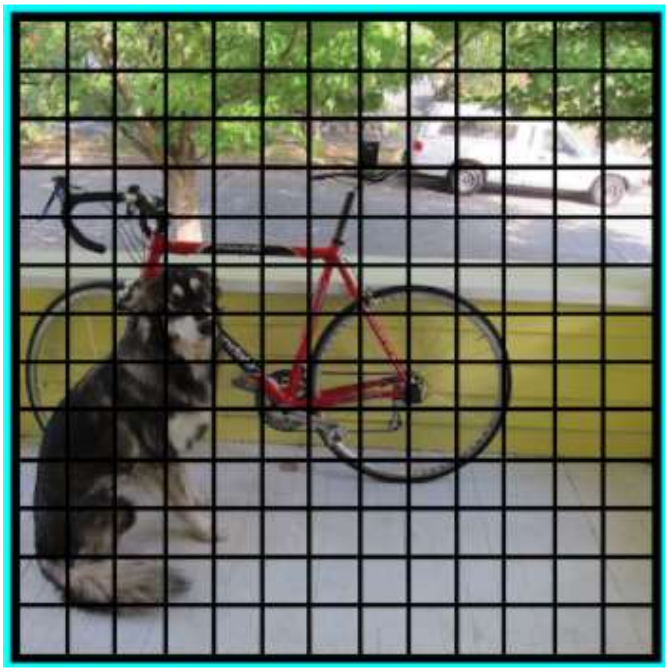


Figure 23: Large set of bounding boxes spanning [18]

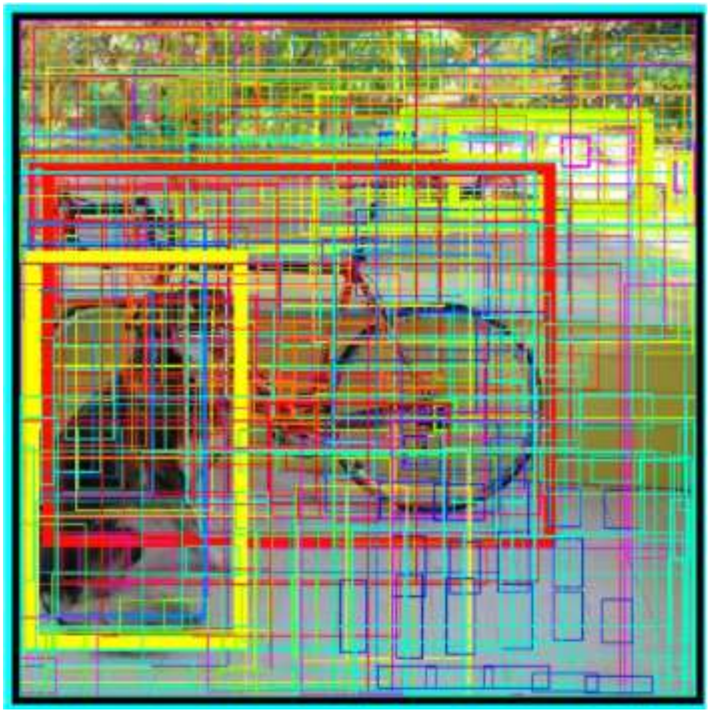


Figure 24: Extracted the features in bounding boxes [18]

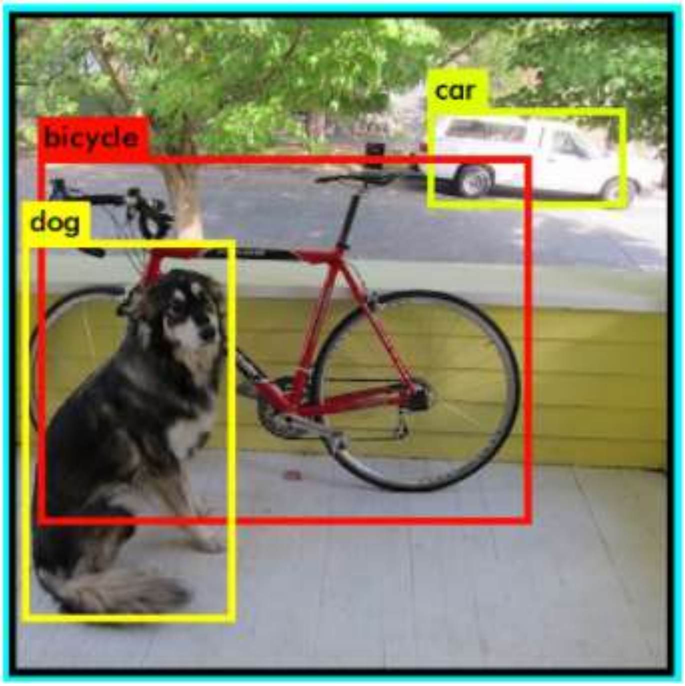


Figure 25: Bounding box [18]

MobileNet-SSD would use the Application Programming Interface (API) to detect the object in real-time. The function of API is providing developers a set of common operations so that developer don't have to write code from scratch. MobileNet-SSD would use TensorFlow object Detection API for object detection. The TensorFlow object detection API is the framework for creating a deep learning network that solves object detection problems. In TensorFlow object detection API, it already has pretrained models in their framework which they refer to as Model Zoo such as CenterNet HourGlass, EfficientDet, SSD MobileNet, SSD ResNet, and Faster R-CNN ResNET. So, we could use the MobileNet-SSD pretrained models from Model Zoo can be used to train the dataset and then detect the object in real-time. Figure 26 show the architecture of MobileNet-SSD and it Single Shot MultiBox Detector (SSD) based detection with Mobilenet as backbone.

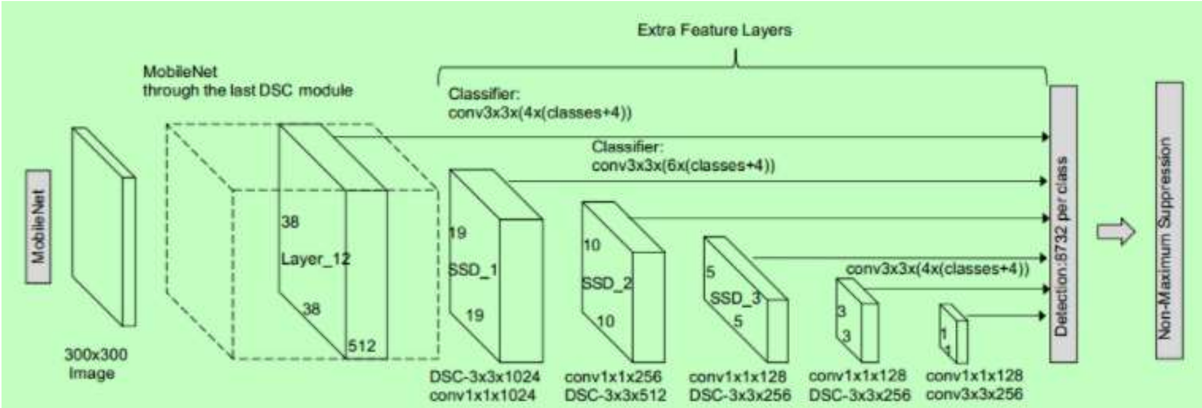


Figure 26: SSD-based detection with MobileNet as backbone [18]

2.4 Detail Explanation of YOLOv5

YOLOv5 is a family of object detection architectures and models pretrained on the coco dataset. Represents Ultralytics open-source research into future vision AL methods. The YOLOv5 repository is a natural extension of the YOLOv3 Pytorch repository and create by Glenn Jocher. Moreover, YOLOv5 has provided 10 type models and the comparison would show in Figure 27.

Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**
YOLOv5s	640	37.4	56.8	98	6.4	0.9
YOLOv5m	640	45.4	64.1	224	8.2	1.7
YOLOv5l	640	49.0	67.3	430	10.1	2.7
YOLOv5x	640	50.7	68.9	766	12.1	4.8
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5
YOLOv5x6 + TTA	1280 1536	55.0 **55.8**	72.7 **72.7**	3136 -	26.2 -	19.4 -

Figure 27: Comparison of YOLOv5 model [21]

In the Figure 27, if the value of mean Average Precision (mAP) for model is higher than another model. Then the speed of model would be lower than other models.

2.4.1 YOLOv5 Network Structure

For network structure of YOLOv5, the backbone would be used is New CSP – Darknet53 module. CSP – Darknet53 module is a convolutional neural network and backbone for object detection that uses Darknet53. It employs a CSPNet strategy to partition the feature map of the base layer into two parts and then merges them through a cross-stage hierarchy. The use of a split and merge strategy allows for more gradient flow through the network. Moreover, the neck network structure of YOLOv5 is used New CSP-PAN and mish activation is used to reduce computation by 40%. For the head of network structure, would use same with the YOLOv3 Head.

The type layer used in the network structure of YOLOv5 are convolution layer, C3 layer, and SPPF layer. A convolution layer is the main building block of a Convolution Neural Network (CNN). It contains a set of filters (or kernels), parameters of which are to be learned throughout the training many deep learning modules would be use the convolution layer such as SPPF, C3 and so on. Besides that, C3 layer is an improved version of CSP Bottleneck module, and its module can enhance learning capability of Convolution Neural Network (CNN). For C3 layer, it is simpler, faster, and lighter with similar performance and better fuse characteristics. In additions, SPPF layer is an improved of Spatial Pyramid Pooling (SPP) and the performance of SPPF is higher than SPP for do the feature extraction. Lastly, the block diagram of network structure for YOLOv5 would show in APPENDIX A and the coding of network structure would show in APPENDIX B.

The backbone New CSP – Darknet53 module and SPPF layer would show in the first line in block diagram for do the feature extraction with down sampling. Moreover, the neck New CSP-PAN module would show in the middle line in block diagram for do the feature extraction with up sampling. Furthermore, the YOLOv3 head would show in last line in the block diagram to export the training result.

2.4.2 How Convolution Layer Work

The purpose of convolution layer is do the feature extraction for the input image and the parameter could be setting in the convolution layer are kernel, stride, padding, and channel. Kernel is used to convolve the image and stride is the number of pixels moved every time. A stride of length 1 produces an image of almost the same size, and a stride of length 2 produces half the size. Padding the image will help in achieving the same size of the input. Channel is the filter map like RGB channels has 3 channels with the color as red, blue, and green. Then the kernel would convolve the input image to extract the feature of input image. After extract the feature of input image, the size of image would be reducing the size as the example show in Figure 28.

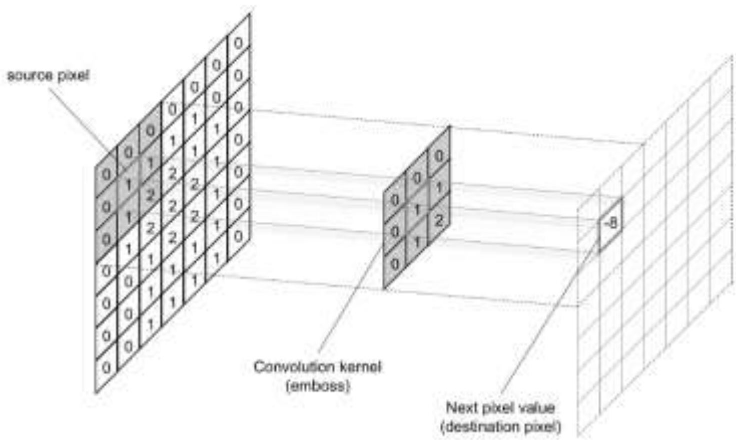


Figure 28: The process for Kernel convolves the input image [23]

Now, let me show the example how the convolution work to do the feature extraction from input image. First, we need setting the parameter of convolution layer such as kernel, stride, padding, and channel as show in Figure 29.

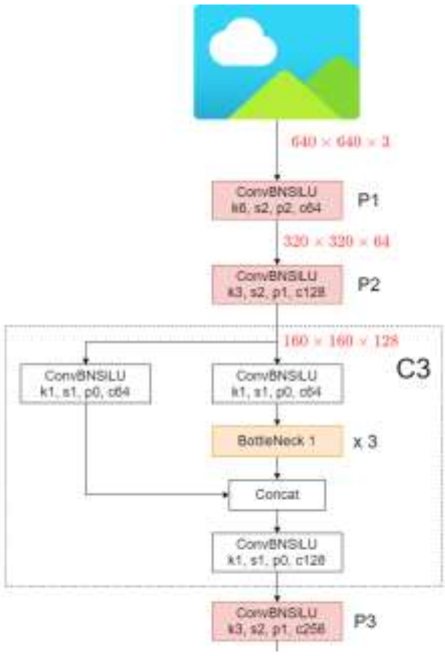


Figure 29: Example convolution layer

At first, the size of input image is 640x640 pixels and 3 channels. The input image would through the process with convolution layer and the parameter of convolution layer is set k6 (kernel is 6), s2 (stride is 2), p2 (padding is 2), and c64 (channels is 64). And then, we could use a formula to calculate output after the input image had been extract the feature by convolution layer. The formula and calculation would show as below:

$$\frac{n+2p-k}{s} x \frac{n+2p-k}{s} xc$$

$$\frac{640+2(2)-6}{2} x \frac{640+2(2)-6}{2} x64$$

$$= 320x320x64$$

After the process convolution layer done, the input image would be reduced size from 640x640 pixels to 320x320 pixels and the channels would be added from 3 to 64 channels.

2.4.3 Data Augmentation

YOLOv5 had provided a hyperparameter evolution method to optimization model for training. The initial hyperparameter had been write in the coding and the coding would show in APPENDIX C. In this coding, has include the method of data augmentation such as mosaic, flip, mix up, rotation, and augment HSV (Hue, Saturation, Value). Data augmentation for Mosaic is to combine four images into an image to expansion the dataset of diversity as example shows in Figure 30. Next, flip is added horizontal or vertical flips to help the model be insensitive to subject orientation as example show in Figure 31. Moreover, mix up is the image of transparency would adjust and combine with another image as example show in Figure 32. Furthermore, data augmentation for rotation is add variability to rotations to help the model be more resilient to camera roll as example show in Figure 33. Lastly, augment HSV is random adjust the Hue, Saturation, and Value for the image as example show in Figure 34.

Figure 30: Data augmentation for mosaic

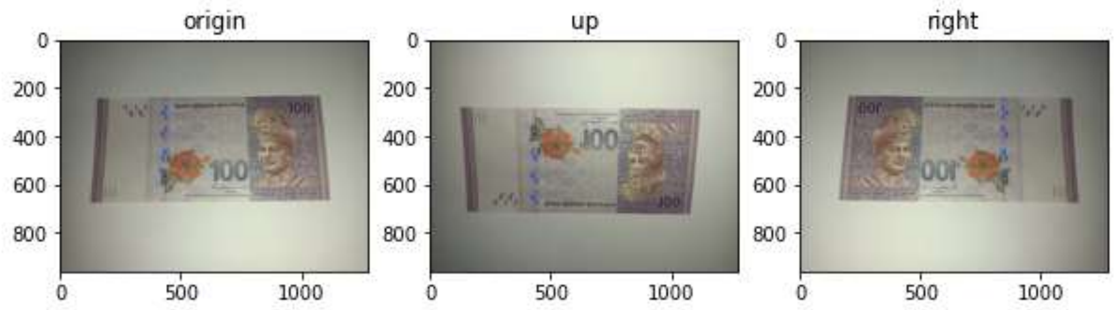


Figure 31: Data augmentation for flip

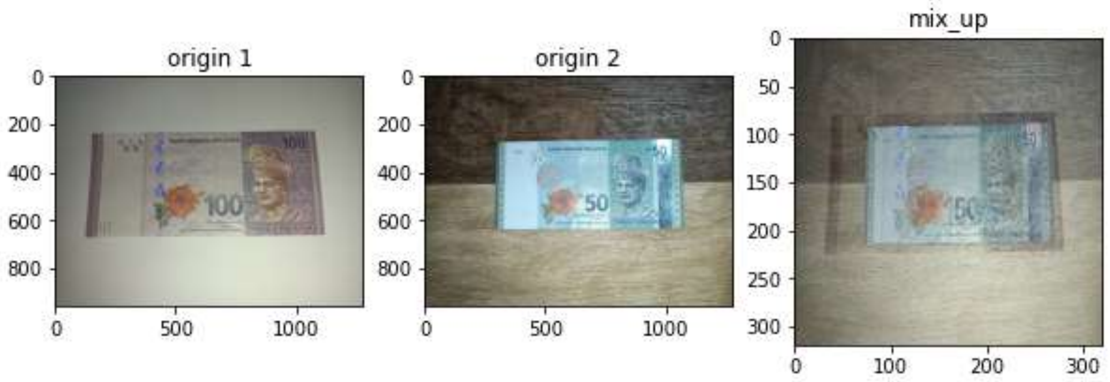


Figure 32: Data augmentation for mix up

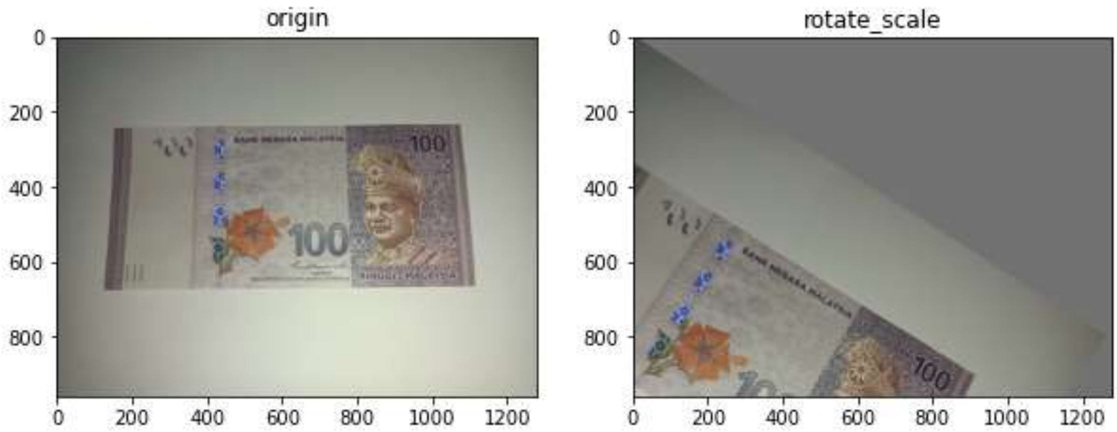


Figure 33: Data augmentation for rotation

Figure 34: Data augmentation for augment HSV (Hue, Saturation, Value)

2.5 Review Conclusion

In conclusion, I select the method is YOLOv5 and YOLOv5s6 model will be used in this project. Because this method able to detect the object in real-time which is the main objective for this project. Also, this model has a high speed to detect the object in real-time with speed in 8.2ms [] and available develop to APP application.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The process flow diagram for this project is shown as show in Figure 35. Firstly, collect the banknote images to create the dataset. After that, label each banknote image from dataset. Next, set the parameters of the model in coding and starting train model at Google Colaboratory. Moreover, create an UI to detect the banknote more convenient. Also, edit the coding for the UI able to show the results in text box and has audio feedback. Next, using the UI to detect the banknote in image or video or real-time and save the result in the folder.

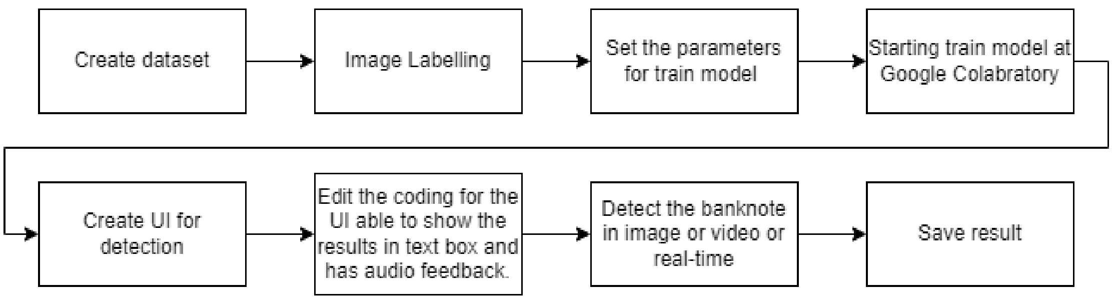


Figure 35: Process flow diagram of project

3.2 Create Dataset

First, banknote image is collected by using smartphone to create dataset with 6 types of Malaysia banknote such as RM1, RM5, RM10, RM20, RM50, RM100. Total banknote images had been collected is 952 images. The 80% of total banknote image would been used for train (765 banknote image) and 20% of total banknote image would been used for test (187 banknote image). Figures 36 and 37 show a part of banknote image put in the train and test folder. Moreover, Table 2 would show the quantity of banknote image for each denomination.

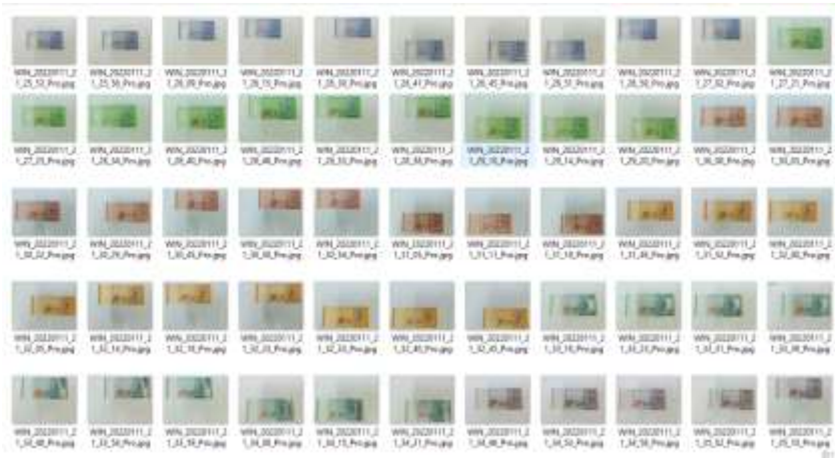


Figure 36: Banknote image in train folder

Figure 37: Banknote image in test folder

Denomination	Quantity
RM1	155
RM5	158
RM10	159
RM20	159
RM50	159
RM100	162
Total	952

Table 2: Quantity for each denomination

3.3 Image Labelling

The purpose of label the banknote image is to generate the bounding box and show the object name when detecting the banknote in real-time. The tools for label each banknote images are using Labelimg tools and labels each banknote image in YOLO darknet TXT annotation format. There is total 9 classes labels to label the banknote image. First 6 classes labels would been used to classify the denomination of banknote such as RM1, RM5, RM10, RM20, RM50, and RM100. Other 3 classes labels are used for labelling the security feature of banknote for recognize the fake banknote such as SF_RM1 for Ringgit Malaysia 1, SF_RM5 for Ringgit Malaysia 5, and SF_AGONG for Ringgit Malaysia 10 to 100. Figure 38 would show example how label Ringgit Malaysia 1 and Figure 39 would show the example on how label the security feature for Ringgit Malaysia 1.

Figure 38: Label classes RM1

Figure 39: Label classes SF_RM1

3.4 Training the Model

After completed the previous steps, the model is trained using YOLOv5s6 model downloaded from Github YOLOv5. Before start to train the model, we need to set the parameters for train model. Firstly, set 9 label classes with RM1, RM5, RM10, RM50, RM100, SF_AGONG, SF_RM5, and SF_RM1 for classify the denomination and the security features. Next is set the number of epochs is 100 because 100 epochs have reached a high accuracy and more epochs would occurs overfitting problem. Next is set the number of batch size is 16 and set the number of batch size is based on the performance of computer GPU RAM. If the computer of GPU RAM is high, then could increase the number of batch size to speed up train the model. However, if the computer of GPU RAM is low and still set the number of batch size is high then the computer would been crashing due the GPU RAM is not enough to support train the model. After setting all parameters needed, could starting train model at Google Colaboratory. Google Colaboratory is a product from Google Research and could allow anybody to write and execute arbitrary python code through the browser. So, we could train the model in the Google Colaboratory because it provided free 12 GPU RAM for train the model. Finally, I successful training the model in the Google Colaboratory as the result show in Figure 40.

```
     Epoch   gpu_mem       box       obj       cls    labels  img_size
     99/99     4.24G  0.005696   0.00472   0.00117        34       640: 100% 48/48 [01:29<00:00,  1.87s/it]
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 6/6 [00:04<00:00,  1.23it/s]
                 all        187        187      0.849      0.971      0.961      0.959

100 epochs completed in 2.743 hours.
Optimizer stripped from drive/.shortcut-targets-by-id/1W1ZrKMXDuNlqV3YdwQl1eERgXqZ7EK1f/YOLOv5/yolov5-6.1/runs/train/exp8/weights/last.pt, 25.1MB
Optimizer stripped from drive/.shortcut-targets-by-id/1W1ZrKMXDuNlqV3YdwQl1eERgXqZ7EK1f/YOLOv5/yolov5-6.1/runs/train/exp8/weights/best.pt, 25.1MB

Validating drive/.shortcut-targets-by-id/1W1ZrKMXDuNlqV3YdwQl1eERgXqZ7EK1f/YOLOv5/yolov5-6.1/runs/train/exp8/weights/best.pt...
Fusing layers...
Model Summary: 280 layers, 12339016 parameters, 0 gradients, 16.2 GFLOPs
               Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 6/6 [00:06<00:00,  1.12s/it]
                 all        187        187      0.814          1      0.962       0.96
                 RM1        187         24      0.875          1      0.989      0.989
                 RM5        187         22      0.869          1      0.986      0.986
                RM10        187         24      0.879          1       0.99      0.984
                RM20        187         24       0.88          1      0.987      0.987
                RM50        187         24      0.852          1      0.987      0.987
               RM100        187         27      0.889          1      0.989      0.989
            SF_AGONG        187         28      0.697          1      0.916      0.901
              SF_RM5        187          7      0.684          1      0.895      0.895
              SF_RM1        187          7        0.7          1      0.918      0.918
Results saved to drive/.shortcut-targets-by-id/1W1ZrKMXDuNlqV3YdwQl1eERgXqZ7EK1f/YOLOv5/yolov5-6.1/runs/train/exp8
```

Figure 40: Completed train the model at Google Colaboratory

3.5 Create UI for Detection

Create the UI by using Qt Designer. Qt Designer is a Qt tool that provided design and build cross-platform Desktop Applications (GUIs) with Pyqt5. In Qt Designer, could design a UI by adding button, window, text box, and so on. After finish design the UI, Qt designer would generate the coding for UI. Then its coding could combine with the detection coding to let the model detect the banknote by using UI. The Figure 41 would show this project of detection UI and this UI has a few functions.

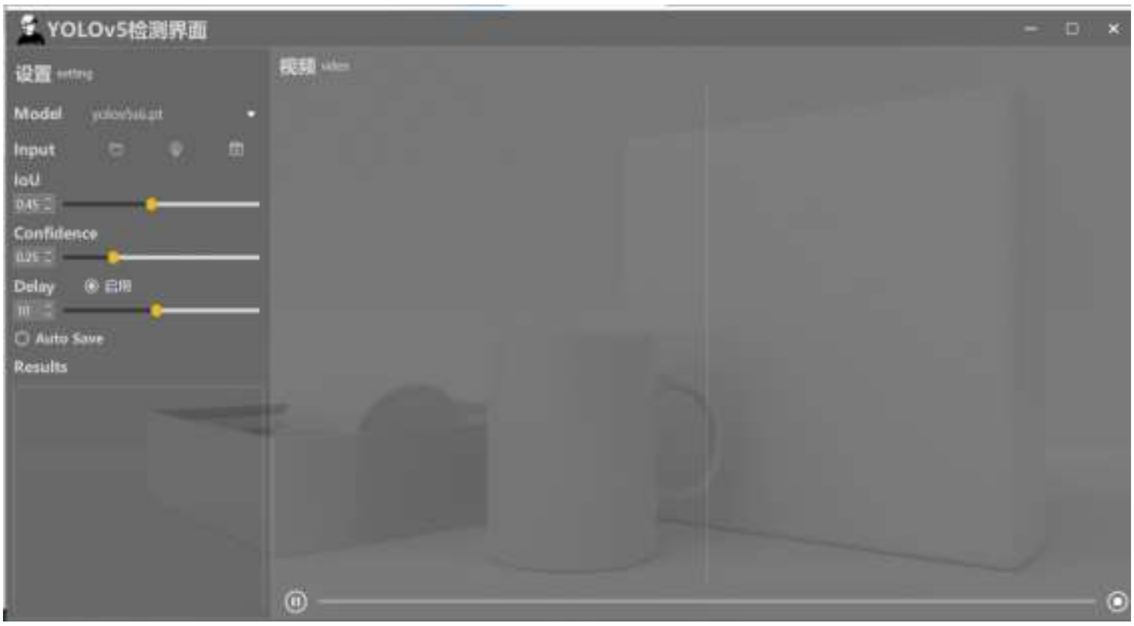
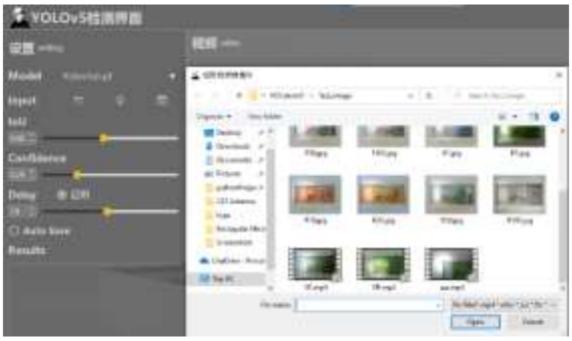


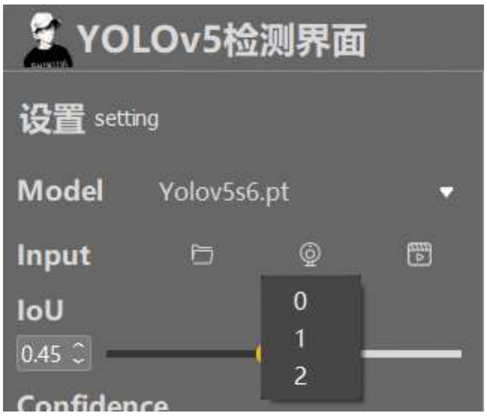
Figure 41: UI detection

The included function is selection model to do the detection such asYOLOv5s and YOLOv5s6 as show in Figure 42. Next function is input image or video for detection as show in Figure 43. Moreover, open the webcam for detect the object in real-time as show in Figure 44. Furthermore, adjust the value of IoU threshold, confidence threshold, and delay as show in Figure 45. Adjust the value of IoU threshold is for limit the intersect over Union Threshold during detection. For example, if adjust the value of IoU threshold high then more easily to detect the multiple objects in an image. For confidence threshold, only show predictions whose predicted probability exceeds confidence threshold. For example, if set the value of confidence threshold is 0.5 then the detection would show the result when the predicted was higher than 0.5 only. Adjust delay is for let the model could delay generating result.

Next function is auto saving the results in folder such as detection results for image, video, and real-time as show in Figure 46. The function of Results box is showing the text result when model detected the object as show in Figure 47. For the window, it's for display the detection screen with the original image would show in left side and right side would show the image had been detection as show in Figure 48. Also, could move the line on the window to left so the window would show the detected image only as show in Figure 49. Lastly, press the start button for starting the detection and press the stop button is for stopping the detection. And the detection result would been saved in folder after press the stop button.



Figures 42 & 43: Selection model & Input image or video for detection



Figures 44 & 45: Open the webcam for detection in real-time & Adjust the value of IoU threshold, confidence threshold, and delay

Figures 46 & 47: Auto save and results box & Window, start and stop button

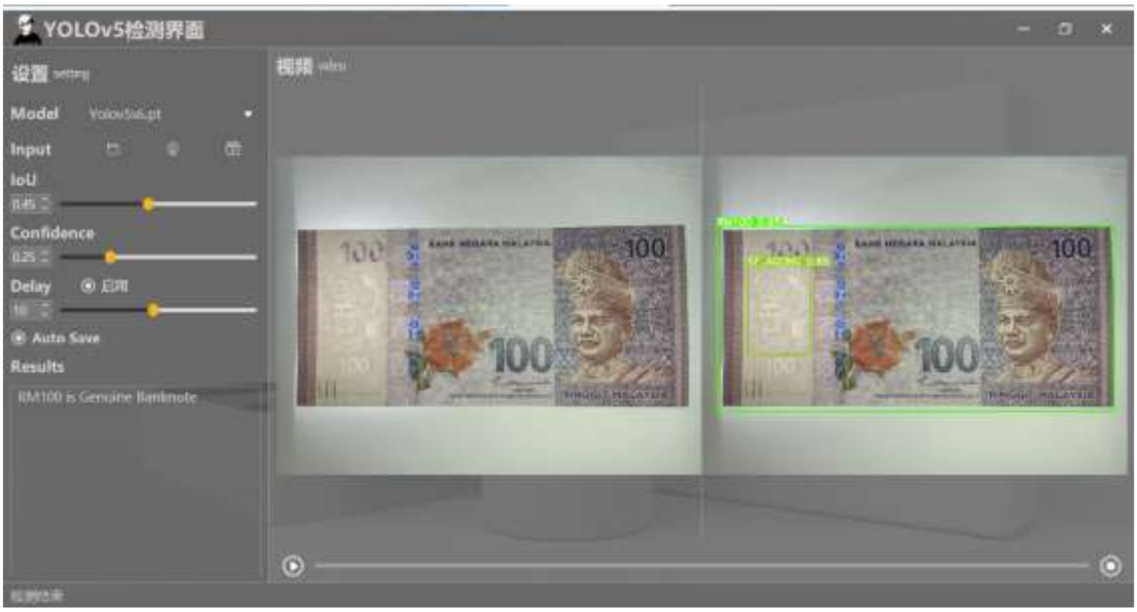


Figure 48: Window of left side show original image and right side show detected image

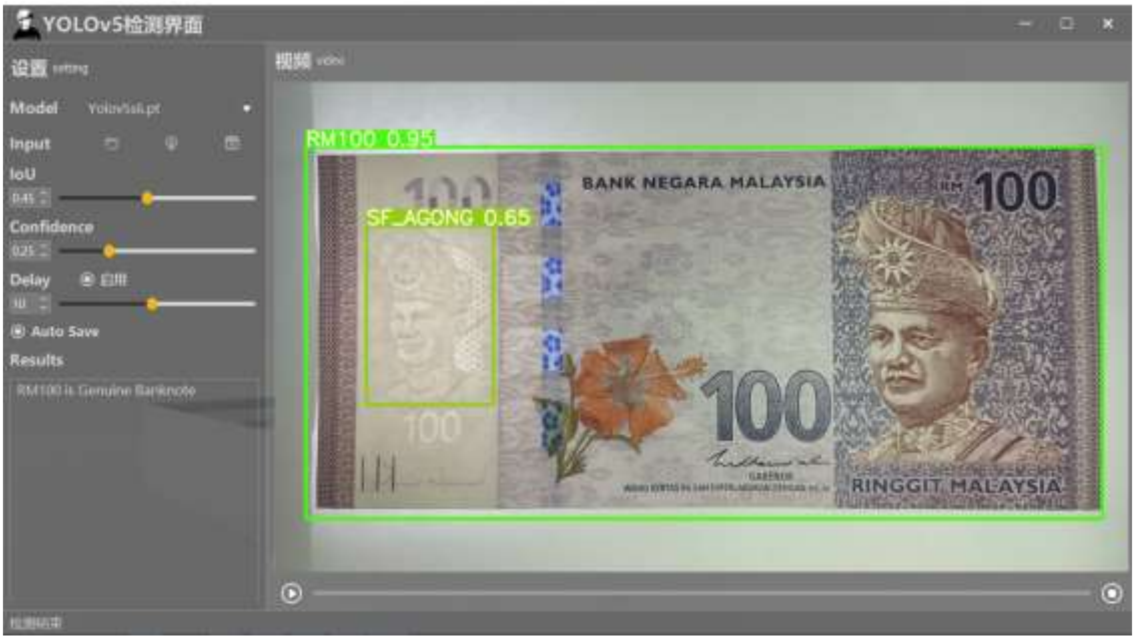


Figure 49: Move the line on the window to left to show the detected image only

3.6 Edit the coding for the UI able to show the results in text box and has audio feedback

After completed create the UI, I tried edit the detection coding to adding the extra function for the model. The first function had added was the text box would show the detection result when detect the banknote and the coding edited would show in APPENDIX D. The second function had added was the model would have the audio feedback when detect banknote and the coding edited would show in APPENDIX E.

After edited the coding, the detection result would show on the text box. The text box would show the denomination of banknote and the banknote is genuine or fake. Have one condition would show the text 'RM? is genuine banknote' on the text box and the condition is detect the security feature and that confidence score is higher than 0.5 (50%). And has two conditions would show the text 'RM? is fake banknote' such as detect the security feature and that confidence score is lower than 0.5 (50%) and detect the denomination of banknote only. For audio feedback, all condition is same with the text box. But would not voice the banknote denomination and only voice ‘REAL’ or ‘FAKE’.

So, there has 3 conditions trigger function the text box and audio feedback. Below would show that 3 conditions of block diagram.

If conf >= 0.5 and (SF_AGONG or SF_RM5 or SF_RM1)

RM? Is Genuine Banknote && Voice = REAL

If conf < 0.5 and (SF_AGONG or SF_RM5 or SF_RM1)

RM? Is Fake Banknote && Voice = FAKE

If only detect RM1 or RM5 or RM10 or RM20 or RM50 or RM100

RM? Is Fake Banknote && Voice = FAKE

3.7 Detect the Banknote in Real-Time

After that, could starting to detect the banknote in real-time. Firstly, open the Jupyter notebook to run the coding of detection with UI as show in Figure 50. Jupyter Notebook is an open-source software that is a place to run the coding or program such as Python, Pytorch, Tensorflow 2, OpenCV, and so on. After running the detection coding in the Jupyter Notebook, the detection UI would been open and standby to detect the banknote.

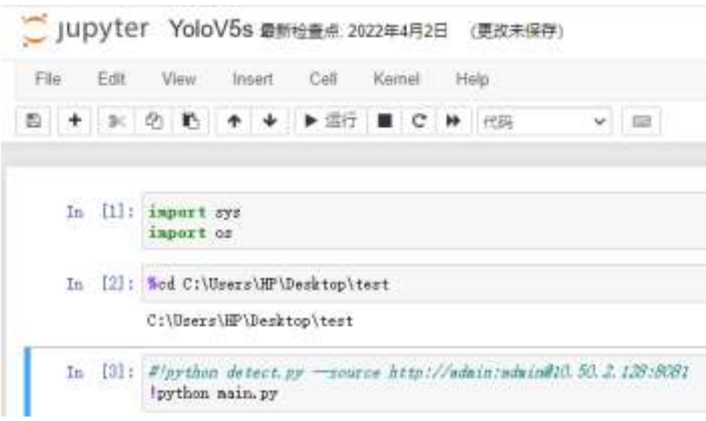
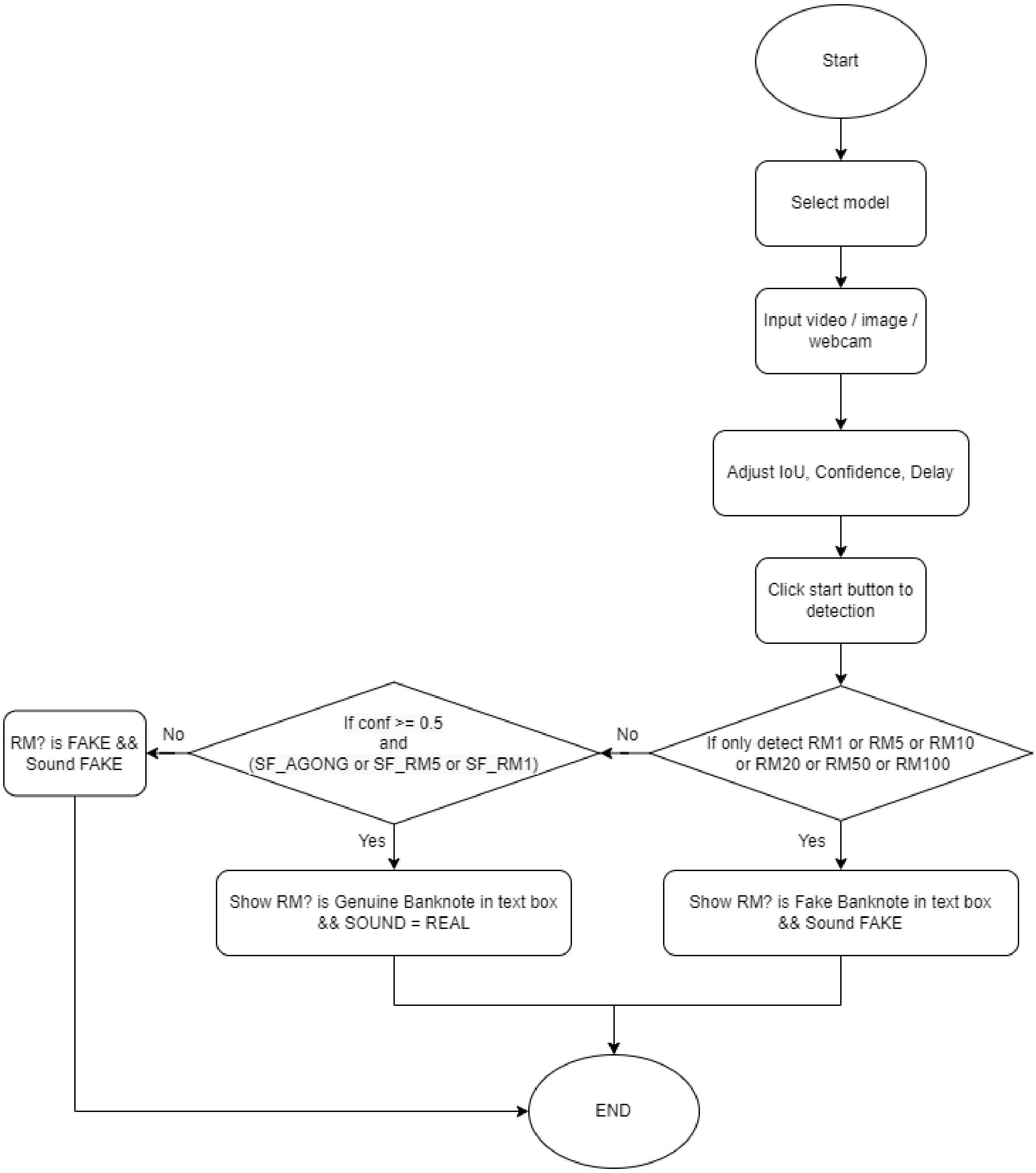


Figure 50: Jupyter Notebook

But before start detect the banknote, need to open APP Iriun webcam application to sync my smartphone camera to laptop for be a webcam to detect the banknote. After that, could start using the UI to detect the banknote and below would show the flowchart to use the UI detect the banknote.



At the first, need to select the model had trained such YOLOv5s, YOLOv5s6, and so on. Next, select image or video for do the detection. Also, could select open the webcam for do the detection in real-time. After selected type of input, could adjust the value of Intersection over Union (IoU) threshold, confidence threshold, and delay. Then could click start button to detect the banknote. After finish detection, could click stop button to stop the detection and if have on the Auto Save function on the UI then the result would be saved in the folder as show in Figure 51.

Figure 51: Save result in folder

3.8 Gantt Chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. Each activity is represented by a bar, the position and length of the bar reflects the start date, duration, and end date of the activity. Figure 52 show the Gantt chart of Semester 1 and Figure 53 shows the Gantt chart of Semester 2. The objective 1 which is develop algorithm to identify and classify the denomination of banknotes is had completed in Semester 1. The objective 2 which is develop algorithm to recognise and classify the counterfeit banknote is had completed in Semester 2. In additions, the objective 3 which is develop algorithm to detect the banknote in real-time also had completed in Semester 2. Finally, the developed algorithm for detecting the banknote in real-time will be analysed in detail and improvement is done accordingly based on the findings during the analysis.

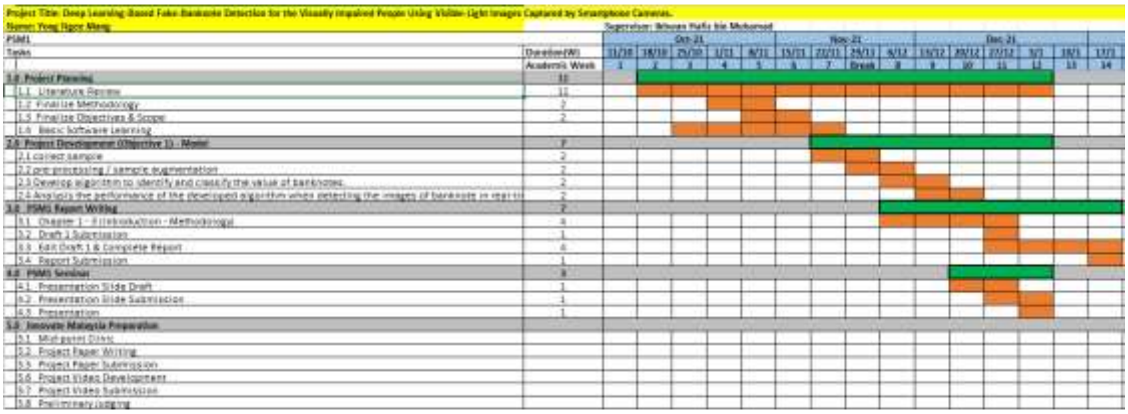


Figure 52: Gantt chart of PSM 1

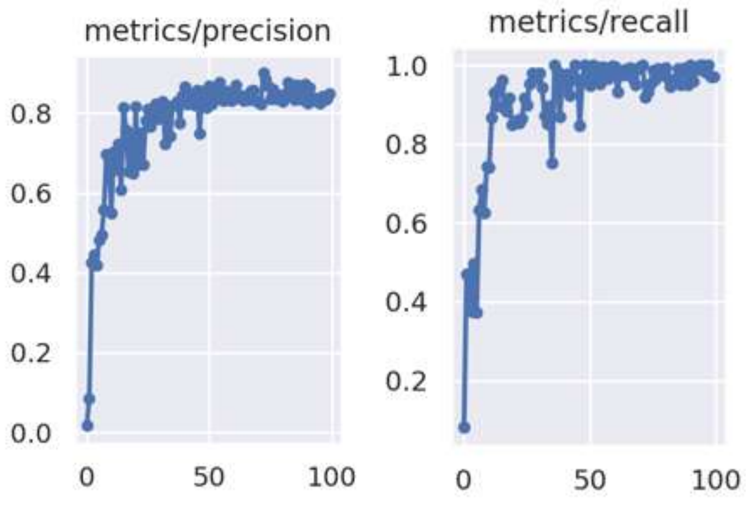
Figure 53: Gantt chart of PSM 2

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Result and Discussion for Precision and Recall

The result of precision and recall would been generated after completing train the YOLOv5s6 model and the Figures 54 & 55 would show the curve graph for precision and recall. In this project, precision and recall is important to take a look at the performance of the model to detect the banknote in real-time. Precision is the number of proportions of the detected banknote were correct and measure how accurate is the predictions in terms of percentage. Recall on the other hand, is the number of proportions of the actual objects did capture.



Figures 54 & 55: Curve graph for Precision & Recall

Intersection over union (IoU) measures the overlap between 2 boundaries. It uses that to measure how much our predicted boundary overlaps with the ground truth as shown in Figure 56. The result of average precision and recall for all area are 0.849 (84.9%) and 0.971 (97.1%). The result for precision need been improve because lower than 85% such as adding more or different banknote image in the dataset to train model. However, the result for recall is best because the result was higher than 95%. The following formula shows the calculation for precision and recall as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where,

TP is True positive (Predicted as positive as was correct)

FP is False positive (Predicted as positive but was incorrect)

FN is False negative (Failed to predict an object that was there)

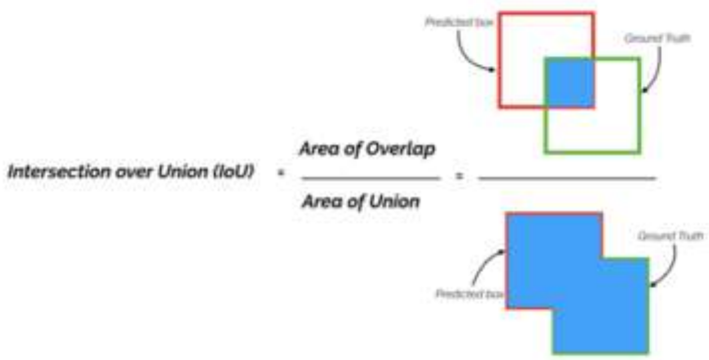


Figure 56: Intersection over Union (IoU) [19]

In addition, we calculate precision and recall using intersection over Union (IoU) Value for a given intersection over Union (IoU) threshold. For example, if intersection over Union (IoU) threshold is 0.5 and the intersection over Union (IoU) value for prediction is 0.7. Then classify the prediction as True Positive (TF). On the other hand, if IoU is 0.3, we classify it as False Positive (FP) as show in Figure 57. That also means that for a prediction, could get different binary true or false positive by changing the intersection over Union (IoU) threshold as shown in Figure 58.

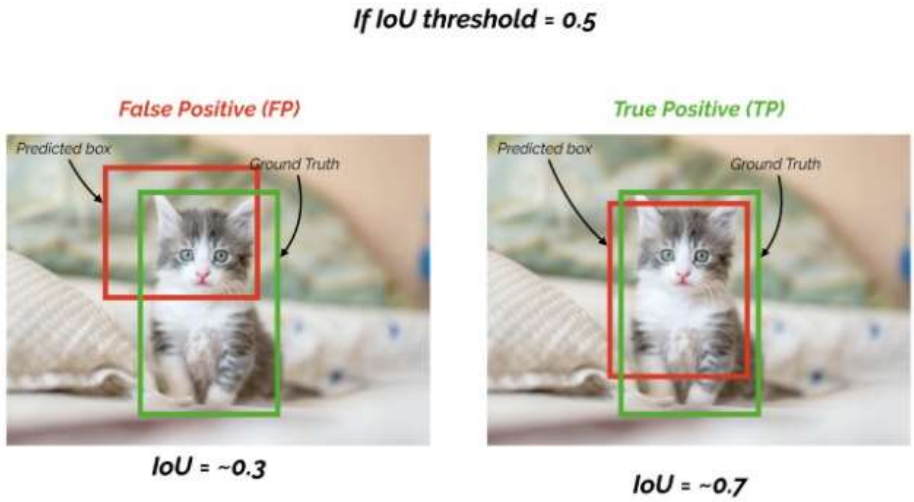


Figure 57: False positive (FP) and True Positive (TP) [19]

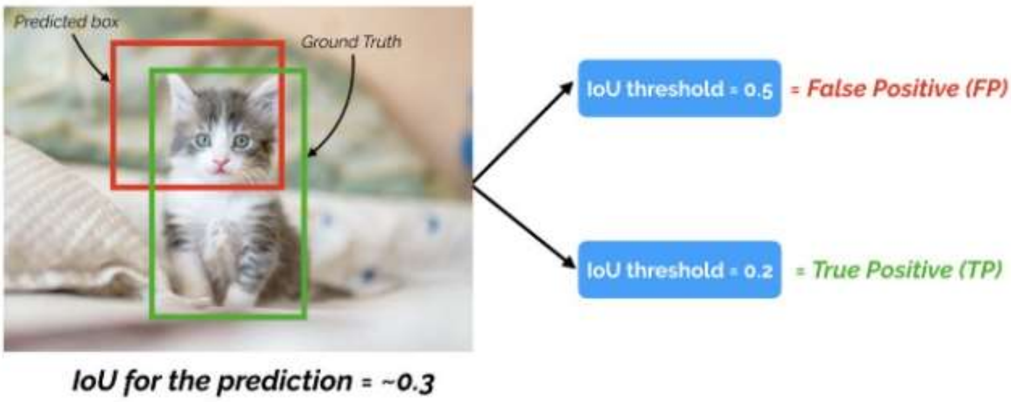


Figure 58: Changing intersection over Union (IoU) threshold to get the different binary for False positive (FP) and True Positive (TP) [19]

4.2 Result and Discussion for Losses Graph

The curve graph for losses also would been generated after completely training the YOLOv5s6 model such as Bounding box loss, Objectness loss, Classification loss as show in Figure 59. The Bounding box loss is the error between the predicted and ground truth bounding box. Objectness loss is an intersection over Union (IoU) prediction and term teaches the network to predict a better box. Classification is a loss function; classification problem involves predicting a discrete class output. It involves dividing the dataset into different and unique classes based on different parameters so that a new and unseen record can be put into one of the classes. The loss needs to be low for an efficient system because to reduce the error in prediction.

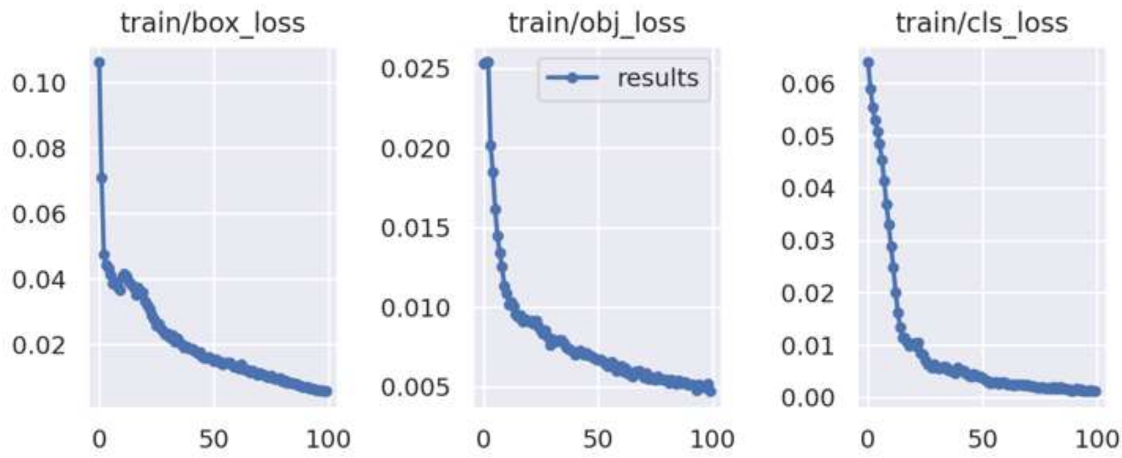


Figure 59: Losses graph for Bounding loss, Objectness loss, and classification loss

Total losses are the sum from Bounding box loss, Objectness loss, Classification loss. The Bounding box loss is 0.005696, Objectness loss is 0.00472, and Classification loss is 0.00117. So, the total losses are 0.011586 and percentages is 1.16%. Based on the total losses graph, the higher the epochs, the losses would be lower. So, the epochs to training the model need to be 100 to decrease the losses. Reducing the loss would increase the accuracy to recognize banknote in real-time but must consider the time taken for training the model. It also would increase the time taken to train the model when the epochs are increased. Because as the number of epochs increases, a greater number of times the weight is changed in the neural network. Besides that, the adjustment of batch size and adding more banknote image for training model also could increase the accuracy of recognize banknote in real-time. If the greater the loss, huger is the errors made on the data and accuracy can be seen as the number of errors made on the data. So, we need to reduce loss to increase the accuracy of model.

4.3 Data Analysis for Real-Time Detection

After detected the banknote image and save result in the folder, could start to do the data analysis for the results of real-time detection because could clearly find out the problem and find a way to solve this problem. Moreover, could select the best model after do the analysis and comparison for each model has been used. Firstly, would do the data analysis for the SSD-MobileNet model of real-time detection result and the detail result would show in Table 3.

Model: SSD-MobileNet				
Type banknote	Label classes detected	Accuracy (%)	Recognize result	Detection speed (ms)
Ringgit Malaysia 1	RM1	100	Genuine Banknote	22
Ringgit Malaysia 5	RM5	100	Genuine Banknote	22
Ringgit Malaysia 10	RM10	100	Genuine Banknote	22
Ringgit Malaysia 20	RM20	100	Genuine Banknote	22
Ringgit Malaysia 50	RM50	100	Genuine Banknote	22
Ringgit Malaysia 100	RM100	100	Genuine Banknote	22
Calculator	RM5	100	Genuine Banknote	22

Table 3: The Result for SSD-MobileNet model detect each denomination of banknote in real-time

Data Analysis for the SSD-MobileNet and images detected show in APPENDIX F, this model had a problem which is the model would show the result when detecting calculator and the result would show that calculator was RM5 and have prediction of 100%. I think the reason cause this problem is the model detect the object based on the colour only. So, this model is not suitable use for this project because this model cannot recognize the banknote is fake or genuine. So, I change the model from SSD-MobileNet to YOLOv5s to detect the banknote in real-time and the detection result would been show in Table 4.

Model: YOLOv5s				
Type banknote	Label classes detected	Accuracy (%)	Recognize result	Detection speed (ms)
Ringgit Malaysia 1	RM1 SF_RM1	86 Lower than 50	Fake Banknote	0.1
Ringgit Malaysia 5	RM5 SF_RM5	86 Lower than 50	Fake Banknote	0.1
Ringgit Malaysia 10	RM10 SF_AGONG	85 54	Genuine Banknote	0.1
Ringgit Malaysia 20	RM20 SF_AGONG	89 64	Genuine Banknote	0.1
Ringgit Malaysia 50	RM50 SF_AGONG	91 Lower than 50	Fake Banknote	0.1
Ringgit Malaysia 100	RM100 SF_AGONG	88 69	Genuine Banknote	0.1
Printer_RM1	RM1	86	Fake Banknote	0.1

Printer_RM5	RM5	90	Fake Banknote	0.1
Printer_RM10	RM10	89	Fake Banknote	0.1
Printer_RM20	RM20	90	Fake Banknote	0.1
Printer_RM50	RM50	92	Fake Banknote	0.1
Printer_RM100	RM100	89	Fake Banknote	0.1

Table 4: The result for YOLOv5s model detect each denomination of banknote and print banknote in real-time

Data Analysis for the YOLOv5s and images detected show in APPENDIX G, this model had the accuracy problem. The problem is the accuracy of detection is lower than 50% when detecting the RM1, RM5, and RM50. This means this model cannot recognize the banknotes for RM1, RM5, and RM50. I think the reason cause this problem is the precision of the model is lower. So, I change the model from YOLOv5s to YOLOv5s6 because the mean Average Precision (mAP) of YOLOv5s6 (44.8 mAP [21]) is higher than YOLOv5s (37.4 Map [21]) to do the data analysis. And the detection result would been show in Table 5.

Model: Yolov5s6				
Type banknote	Label classes detected	Accuracy (%)	Recognize result	Detection speed (ms)
Ringgit Malaysia 1	RM1 SF_RM1	81 79	Genuine Banknote	0.1

Ringgit Malaysia 5	RM5 SF_RM5	88 83	Genuine Banknote	0.1
Ringgit Malaysia 10	RM10 SF_AGONG	89 75	Genuine Banknote	0.1
Ringgit Malaysia 20	RM20 SF_AGONG	85 77	Genuine Banknote	0.1
Ringgit Malaysia 50	RM50 SF_AGONG	92 78	Genuine Banknote	0.1
Ringgit Malaysia 100	RM100 SF_AGONG	90 81	Genuine Banknote	0.1
Printer_RM1	RM1	89	Fake Banknote	0.1
Printer_RM5	RM5	90	Fake Banknote	0.1
Printer_RM10	RM10	94	Fake Banknote	0.1
Printer_RM20	RM20	91	Fake Banknote	0.1
Printer_RM50	RM50	93	Fake Banknote	0.1
Printer_RM100	RM100	89	Fake Banknote	0.1
Calculator	Not detected	-	-	-

Table 5: The result for YOLOv5s6 model detect each denomination of banknote, print banknote, and calculator in real-time

Data Analysis for the YOLOv5s6 and images detected show in APPENDIX H, this best because could correctly show the denomination for each type of Malaysia banknote and recognize the Malaysia banknote is genuine or fake. Also, there would not show any result when detecting the calculator. So, the YOLOv5s6 model had been choose for this project to detect the banknote. In addition, I also used the YOLOv5s6 model detect the Singapore Dollar with 4 type denomination to test the performance of this model and the detection result would been show in Table 6.

Model: YOLOv5s6				
Type banknote	Label classes detected	Accuracy (%)	Recognize result	Detection speed (ms)
Singapore Dollar 2	Not detected	-	Fake Banknote	0.1
Singapore Dollar 5	RM5	54	Fake Banknote	0.1
Singapore Dollar 10	Not detected	-	Fake Banknote	0.1
Singapore Dollar 50	SF_AGONG	Lower than 50	Fake Banknote	0.1

Table 6: The result for YOLOv5s6 model detect 4 type denomination for Singapore Dollar

Data Analysis for the YOLOv5s6 model detect Singapore Dollar and images detected show in APPENDIX I, the result is not bad because all Singapore Dollar had been detected with been Fake Banknote. But there would show the RM5 result of 54% when detecting Singapore Dollar 5. Then I think the reason cause this problem is the colour of Singapore Dollar is green and same with the RM5. But it’s not a serious problem because the accuracy is low and hard to detected for Singapore Dollar 5. Another problem is would show the label classes of SF_AGONG when detecting Singapore Dollar. But it’s also not a serious problem because the accuracy is lower than 50% and hard to detected. Thus, the recognize result for Singapore Dollar 50 is Fake Banknote.

CHAPTER 5

CONCLUSION

5.1 Conclusion

In this project, all objectives had been achieved which is develop algorithm to identify and classify the value of banknote. And achieved the objective 2 which is develop algorithm to recognise and classify counterfeit banknote. Also, the last objective had been achieved which is analyze the developed algorithm performance in real-time. Moreover, I also successful create a UI for detect the banknote more convenience. Next is successful edit the detection coding to allow the model able to show the results in the text box on the UI and has the audio feedback when detecting the banknote in video or real-time.

All above is by using the YOLOv5s6 to be achieved. Furthermore, the precision and recall of the trained YOLOv5s6 model is 84.9% and 94.1% by setting the parameter are epochs is 100 and batch size is 16. And the learning rate is 0.0033. Finally, I think I successfully developed this project.

5.2 Recommendation

For the future work for this project, the first work is development this project to APP application because Yolov5 is support and available to become an APP for banknote detection. Moreover, accuracy improvement by adding more banknote image in dataset. Next is adding the image with different lighting and background in the dataset to training a model that could detect the banknote in different lighting and background. Furthermore, develop a function for coin detection. Lastly, adding other country banknotes in the dataset to train a model that could recognize other country banknote.

REFERENCES

1. Martin Carvalho, H. S. (6 Aug 2020). *One in a million: Finance Ministry cites low rate of counterfeit currency in Malaysia.* Malaysia: thestar. Retrieved from https://www.thestar.com.my/news/nation/2020/08/06/one-in-a-million-finance-ministry-cites-low-rate-of-counterfeit-currency-in-malaysia

2. Ball, M. (21 March 2020). *Recent Trends in Banknote Counterfeiting.* Australia: Reserve Bank of Australia. Retrieved from https://www.rba.gov.au/publications/bulletin/2019/mar/recent-trends-in-banknote-counterfeiting.html

3. "models/tf2_detection_zoo.md at master · tensorflow/models - GitHub." https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md. Accessed 18 Jan. 2022.

4. Lee, J.W.; Hong, H.G.; Kim, K.W.; Park, K.R. A Survey on Banknote Recognition Methods by Various Sensors. *Sensors* **2017**, *17*, 313. https://doi.org/10.3390/s17020313

5. K. Lee and T. Park, "Image segmentation of UV pattern for automatic paper-money inspection," 2010 11th International Conference on Control Automation Robotics & Vision, 2010.

6. Bruna, Arcangelo & Farinella, Giovanni & Guarnera, Giuseppe & Battiato, Sebastiano. (2013). Forgery Detection and Value Identification of Euro Banknotes. Sensors (Basel, Switzerland).

7. T. D. Pham, C. Park, D. T. Nguyen, G. Batchuluun and K. R. Park, "Deep Learning-Based Fake-Banknote Detection for the Visually Impaired People

Using Visible-Light Images Captured by Smartphone Cameras," in IEEE Access.

8. Tushar Agasti, Gajanan Burand, Pratik Wade and P Chitra, " Fake currency detection using image processing," in IopScience Access.

9. Q. Zhang and W. Q. Yan, "Currency Detection and Recognition Based on Deep Learning," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018.

10. Ren, Yueqiu. "Banknote Recognition in Real Time Using ANN." (2017).

11. Sarma, Spandan. (2017). Bank Note Authentication: A Genetic Algorithm Supported Neural based Approach. International Journal of Advanced Research in Computer Science. 7.

12. Choi, Eunjeong & Chae, Somi & Kim, Jeongtae. (2019). Machine Learning-Based Fast Banknote Serial Number Recognition Using Knowledge Distillation and Bayesian Optimization. Sensors.

13. R. Tasnim, S. T. Pritha, A. Das and A. Dey, "Bangladeshi Banknote Recognition in Real-time using Convolutional Neural Network for Visually Impaired People," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2021.

14. R. Priyatharshini, Aswath Ram. A.S, R. Shyam Sundar, G. Nethaji Nirmal, " Real-Time Object Recognition using Region based Convolution Neural Network and Recursive Neural Network," in IJRTE Access.

15. Younis, Ayesha & Shixin, Li & Jn, Shelembi & Hai, Zhang. (2020). Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD CCS Concepts.

16. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, " MobileNets:

Efficient Convolutional Neural Networks for Mobile Vision Applications," in arvix Access.

17. "SSD object detection: Single Shot MultiBox Detector for real-time ...." 13 Mar. 2018, https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06.

18. "Object Detection using the TensorFlow API - Analytics Vidhya." 7 Apr. 2020, https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/.

19. "mAP (mean Average Precision) might confuse you! - Towards Data ...." 9 Jun. 2020, https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2.

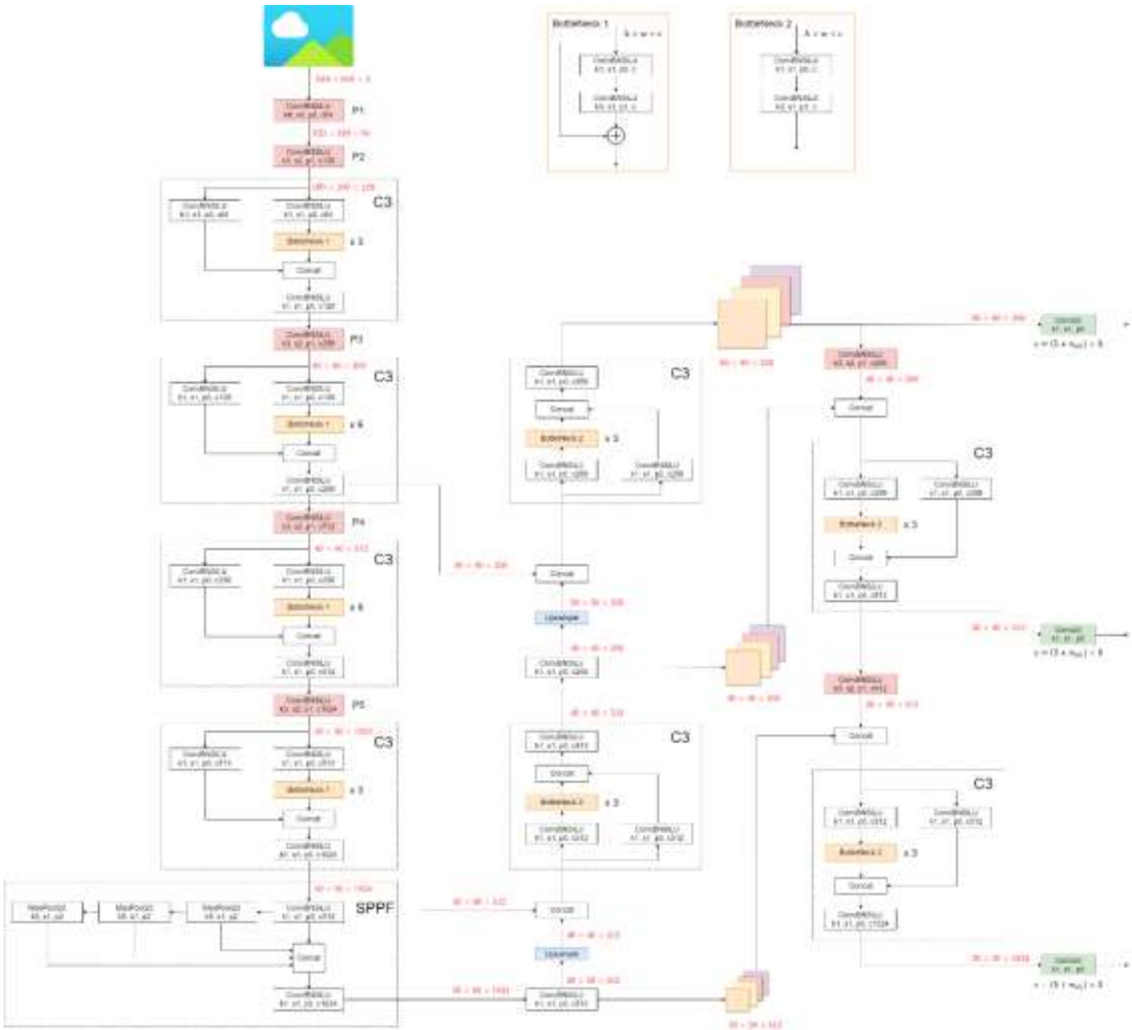
20. "New currency notes pose problems for visually-impaired." 24 Dec. 2018, https://www.newindianexpress.com/cities/bengaluru/2018/dec/24/new-currency-notes-pose-problems-for-visually-impaired-1915955.html.

21. "YOLOv5 in PyTorch > ONNX > CoreML > TFLite - GitHub." https://github.com/ultralytics/yolov5.

22. "Content - GitHub." https://github.com/ultralytics/yolov5/issues/6998.

23. "Using Deep Learning Models / Convolutional Neural Networks." https://docs.ecognition.com/v10.0.2/eCognition_documentation/User%20Guide%20Developer/8%20Classification%20-%20Deep%20Learning.htm.

APPENDIX A
YOLOV5 NETWORK STRUCTURE



APPENDIX B
CODING OF NETWORK STRUCTURE

```
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# Parameters
nc: 9  # number of classes
depth_multiple: 1.0  # model depth multiple
width_multiple: 1.0  # layer channel multiple
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Conv, [64, 6, 2, 2]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, C3, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 6, C3, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, C3, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 3, C3, [1024]],
   [-1, 1, SPPF, [1024, 5]],  # 9
  ]

# YOLOv5 v6.0 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, C3, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, C3, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, C3, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, C3, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

APPENDIX C
CODING INITIALIZE HYPERPARAMETER

```
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# Hyperparameters for VOC training
# python train.py --batch 128 --weights yolov5m6.pt --data VOC.yaml --epochs 50 --img 512 --hyp hyp.scratch-med.yaml --evolve
# See Hyperparameter Evolution tutorial for details https://github.com/ultralytics/yolov5#tutorials
# YOLOv5 Hyperparameter Evolution Results
# Best generation: 319
# Last generation: 434
# metrics/precision, metrics/recall, metrics/mAP_0.5, metrics/mAP_0.5:0.95, val/box_loss, val/obj_loss, val/cls_loss
# 0.86236, 0.86184, 0.91274, 0.72647, 0.0077056, 0.0042449, 0.0013846
lr0: 0.0033
lrf: 0.15184
momentum: 0.74747
weight_decay: 0.00025
warmup_epochs: 3.4278
warmup_momentum: 0.59032
warmup_bias_lr: 0.18742
box: 0.02
cls: 0.21563
cls_pw: 0.5
obj: 0.50843
obj_pw: 0.6729
iou_t: 0.2
anchor_t: 3.4172
fl_gamma: 0.0
hsv_h: 0.01032
hsv_s: 0.5562
hsv_v: 0.28255
degrees: 0.0
translate: 0.04575
scale: 0.73711
shear: 0.0
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 0.87158
mixup: 0.04294
copy_paste: 0.0
anchors: 3.3556
```

APPENDIX D
EDITED CODING FOR TEXT BOX

```
c = int(cls)  # integer class
f = 'FAKE'
if conf >= 0.5:
   label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {conf:.2f}')
   # im0 = plot_one_box_PIL(xyxy, im0, label=label, color=colors(c, True), line_thickness=line_thickness)
   plot_one_box(xyxy, im0, label=label, color=colors(c, True),
                line_thickness=line_thickness)
   if c == 6 or c == 7 or c == 8:
      statistic_dic[names[c]] = 1

if conf < 0.5:
   if c == 6 or c == 7 or c == 8:
      statistic_dic[names[c]] = 2
      label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {f}')
      plot_one_box(xyxy, im0, label=label, color=colors(c, True),
                   line_thickness=line_thickness)

if c == 0:
   statistic_dic[names[c]] = 3

if c == 1:
   statistic_dic[names[c]] = 4

if c == 2:
   statistic_dic[names[c]] = 5

if c == 3:
   statistic_dic[names[c]] = 6

if c == 4:
   statistic_dic[names[c]] = 7

if c == 5:
   statistic_dic[names[c]] = 8
```

```
a = str(i[1])
global b

if a=='1':
   self.resultWidget.clear()
   results = [' ' + b + ' ' + 'is' + ' ' + 'Genuine Banknote']
   self.resultWidget.addItems(results)

elif a=='2':
   self.resultWidget.clear()
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='3':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='4':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='5':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='6':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='7':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)

elif a=='8':
   b = str(i[0])
   results = [' ' + b + ' ' + 'is' + ' ' + 'Fake Banknote']
   self.resultWidget.addItems(results)
```

APPENDIX E
EDITED CODING FOR AUDIO FEEDBACK

```
now = datetime.datetime.now()
s = now.second
ms = now.microsecond


def test():
    if conf >= 0.5 and s%5 == 0 and ms >= 850000:
       if c == 6 or c == 7 or c == 8:
          voice = '1'
          self.send_voice.emit(voice)
       elif c == 0 or c == 1 or c == 2 or c == 3 or c == 4 or c == 5:
          voice = '2'
          self.send_voice.emit(voice)
    elif conf < 0.5 and s%5 == 0 and ms >= 850000:
       voice = '2'
       self.send_voice.emit(voice)
thread = threading.Thread(target=test)
thread.start()


def show_voice(self, voice):
    try:
        if voice == '1':
           mixer.init()
           mixer.music.load('Voice Feedback/GenuineBanknote.mp3')
           mixer.music.play()
        elif voice == '2':
           mixer.init()
           mixer.music.load('Voice Feedback/FakeBanknote.mp3')
           mixer.music.play()

    except Exception as e:
        print(repr(e))
```

APPENDIX F
DETECTION RESULT FOR SSD-MOBILENET


object detection
RM5: 100%

APPENDIX G
DETECTION RESULT FOR YOLOV5S


RM1 0.86


RM5 0.90


RM10 0.89


RM10 0.90


RM50 0.92


RM100 0.89

APPENDIX H
DETECTION RESULT FOR YOLOV5S6


RM10 0.94

RM20 0.91

RM50 0.93

RM100 0.89


APPENDIX I
DETECTION RESULT FOR SINGAPORE DOLLAR