Philippine Currency Counterfeit Detector using Image Processing

By:

Jude Michael Apoloni

Sean Derrick Escueta

ECE/3

A Research Proposal Submitted to the School of Electrical Electronics and Computer Engineering

In Partial Fulfilment of the Requirements for the Degree

Bachelor of Science in Electronics Engineering

Mapua University

July 2021

CHAPTER 1

INTRODUCTION

Currency counterfeiting is a crime that constantly threatens a country's economy and causes financial loss to the population. Money is being transmitted every second from person-to-person with the intention of making an exchange for a product or service. Some people intentionally produce and utilize counterfeit currency when it comes to their daily expenses, because of this, counterfeit currency is circulated in the economy. Currency counterfeiting causes problems in many ways. Some major effects include the reduction of the value of legitimate money and inflation. According to the BSP, the rate of currency counterfeiting dropped down to 11 parts per million (PPM) in 2019, compared to 12.9 PPM recorded in 2018. Although the results showed a low rate of currency counterfeiting, it also makes it harder to find or detect counterfeit currency that is currently being circulated in the money transmission cycle. To combat currency counterfeiting, the BSP regularly changes the design of Philippine currency banknotes and also adds security features to serve as preventive measures to combat currency counterfeiting. Security features are separated in levels 1 to 4. Level 1 includes security features that are easily recognizable by the public without the use of specialized devices. These features involve the “feel”, “look”, and “tilt” features applied to banknotes. Level 2 includes security features that are professionally recognizable by bank personnel with the use of ultraviolet light or magnifying lens. Level 3 are features that are disclosed to the public. Lastly, level 4 includes security features that are used in forensic investigations.

In 2019, the Philippine News Agency published an article that the National Capital Region Police Office (NCRPO) called the attention of the public to be cautious of counterfeit currency, after arresting three (3) men in Pasay City for peddling fake currencies in casinos. The NCRPO Director, Major Gen. Guillermo Eleazar stated to the Philippine News Agency, "We advise the residents to carefully examine their bills, so as not to be victimized by this illegal activity. We also call on the public to call the authorities for information that will lead to the arrest of those behind these counterfeit money". In 2020, BSP asked inhabitants of Negros Occidental to report any case of counterfeit currency to their office. BSP told the public if ever they received suspicious currency, the local branch of BSP will prearrange if the currency is counterfeit and contains real value, and doubtful currency will be sent to the main office that is in Manila. The BSP Currency Analysis Redemption Division will release their final verdict to determine if the currency has real value. According to section 50 of the Republic Act (R.A.) No. 7653, as amended, the BSP "shall have the authority to investigate, make arrests, conduct searches and seizures in accordance with law, for the purpose of maintaining the integrity of the currency.". Furthermore, the act of imitation and reproduction without prior authority of Philippine currency banknotes is punishable by law under Republic Act 11211 or the New Central Bank Act of 2019. As part of the mandate stated, the BSP encourages the protection of the integrity of the Philippine currency and acts upon counterfeiting and defacement of Philippine currency banknotes.

Recent studies and existing counterfeit detection machines suggest the use of UV light. This process is often unreliable due to it still being tricked by bleached banknotes. One of the processes of currency counterfeiting is by bleaching, which involves placing banknotes in a bleaching solution, and using the output to print it in a higher denomination. Another existing problem is the maintenance, as the process of existing money counterfeit detection machines involves inserting money into the system for it to be scanned, requiring the machine to be cleaned regularly.

The main objective of this research is to develop a Philippine currency detector that utilizes image processing. The specific objectives are as follows; (1) To develop a device that detects security features used in Philippine currency banknotes such as security fibers, watermark, asymmetric serial number, see-through mark, and security thread; (2) to utilize different feature extraction algorithms related to image processing such as Optical Character Recognition (OCR), pattern recognition, Canny Edge Detector, and Hough Line Transform; (3) to determine if there is a significant difference between the implementation of security features in legitimate and counterfeit banknotes.

The proposed design of the Philippine currency counterfeit detector using image processing will redound to the benefit of society considering that counterfeit currency runs rampant as of today. Furthermore, this study is beneficial with currency transactions. The concept of image processing when applied to detecting counterfeit currency can be beneficial to currency analysts since it is easier to detect counterfeit currency.

The scope of this study is to determine counterfeit Philippine currency from the 20 peso bills up to 1000-peso bills by detecting security features such as security fibers, watermark, asymmetric serial number, see-through mark, and security thread by utilizing feature extraction algorithms such as Optical Character Recognition (OCR), pattern recognition, Canny Edge Detector, Hough Line Transform. The study does not carry on to any other currencies.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter of the study comprises the intent of the literature review, reviews on image processing and machine learning capable of detecting counterfeit currency that were accomplished in various countries and universities and several previous various conference papers. In this chapter, the focuses are findings in Image Processing and Machine Learning.

2.1 Applications of Image Processing

In the past decades, progress in catch gadgets and the increment of accessible computerized picture information have animated the production of philosophies for information preparation that produce different types of important models, for example, descriptors, classifiers, approximations, and perceptions. Image enhancement incorporates the cycles of evolving pictures, regardless of whether they are customary photochemical photos, advanced photos, or outlines. Ordinary simple image enhancement is known as photograph correcting, utilizing apparatuses like a digitally embellish to change photos, or altering plan with any mechanism of Traditional craftsmanship. Realistic programming programs, which can be comprehensively assembled into raster illustrations editors, and three-Dimensional modelers and vector designs editors are the essential devices with which a client may impact, upgrade, and change pictures. A few image altering programs are additionally used to deliver or make computer workmanship without any preparation [7].

Currency bills were used every day through the mode of traditional transaction; therefore, it is very common for fraudsters to create counterfeit currency. Individual bills have unique identities that cannot be noticed by the naked eye. Utilizing the algorithm of the Hough Line Transform will help to identify the specific placement of the security thread present in the bill by giving it more detailed results [4]. The Hough Line Transform will recognize the line present in the bill, it will take the binary edge map as inputs and locate the edges that are placed in a straight line. The thought behind the Hough transform is that every edge point in the map is transformed to all possible lines and to pinpoint their locations. The characteristics of the bills will be discussed later in this chapter. In general, the number of dimensions of the accumulator corresponds to the number of unknown parameters in the Hough transform problem [10].

In the Philippines, both cash sums have very similar estimations. This suggests that visually impaired people can't separate between some categories. Visually impaired people physically arrange a currency unit in a specific way. The lack of identification devices motivated the need for a device that can be used as a counterfeit detector.

Image processing comprises the control of pictures utilizing advanced computers. Its utilization has been expanding dramatically recently. Its applications range from medication to diversion, passing by geographical handling and far off detecting. Sight and sound frameworks, one of the mainstays of the advanced data society, depend intensely on image processing.

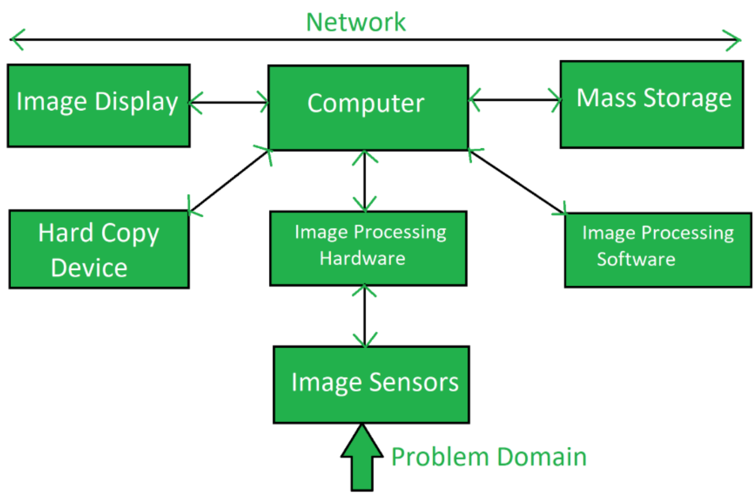


Figure 2.1: Image Processing System

In the field of image processing, an image is defined as a two dimensional function f(x,y), whereas the amplitude of "f" is considered as intensity of the image at a certain point, and x and y are plane coordinates [20]. Each coordinate position is called a pixel. The smallest unit of an image is the pixel, which is also known as a picture element. Pixels make up digital images; each pixel represents the color at a particular place in the image. A pixel is a little dot of a specific color. A digital image, often known as a Bitmap, is a rectangular array of pixels.

Components of Image Processing System

Image Processing System is the blend of the various components engaged with the digital image processing. Advanced picture preparation is the handling of a picture through a computerized computer. Advanced picture handling utilizes diverse computer calculations to perform picture preparing on the computerized images. The most prerequisites for image processing of pictures are that the pictures be accessible in digitized structure i.e., array of finite length binary words. A run of the mill approach of putting away a picture carefully on a computer is by inspecting the picture at a rectangular matrix. The tone or power at every one of these focuses are changed over into a numeric worth and put away in the computer. Aside from the shading/power at those focuses, everything else is disposed of when the picture is put away in the computer [6].

1. **Image Sensors**

   The size of an Image Sensor is quite possibly the main specs in deciding by and large execution. Even the least complex machine vision and imaging gadgets have these sensors; they would be futile. An image sensor is a gadget that permits the camera to change over photons into electrical signs that can be deciphered by the gadget. The image sensors identify the magnitude, co-ordinates, amplitude, intensity, and the RGB of the images, the outcomes will be passed through the image processing hardware.

1. **Image Processing Hardware**

   The image processing hardware is the devoted equipment that is utilized to handle the guidelines acquired from the picture sensors. The hardware utilizes advanced performances that can render digital prints to images. It passes the outcome to broadly useful computers.

1. **Computer**

A programmable electronic gadget intended to acknowledge information, perform recommended numerical and coherent tasks at high velocity, and show the consequences of these activities. Centralized servers, work area and PCs, and cell phones are a portion of the various kinds of PCs. Computer utilized in the image processing system is the universally useful PC that is utilized by us in our everyday life.

1. **Image Processing Software**

The image processing software is the product that incorporates every one of the instruments and calculations that are utilized in the picture preparation framework. The software that can display and preparing the input images as the output becomes enhanced.

1. **Mass Storage**

The mass storage stores the pixels that are present in the image during the process. The ability is a must in picture handling applications. Advanced capacity for picture preparing applications falls into three rule classes: (1) momentary capacity (2) on-line stockpiling for somewhat quick review, (3) chronicled capacity, portrayed by rare access.

1. **Image Display**

It is where the captured image will be displayed after the process. Picture shows being used today are primarily shading TV screens, Monitors are driven by the yields of picture and design display cards that are necessary pieces of the PC framework. Rarely are their prerequisites for image display applications that cannot be met by show cards accessible monetarily as a feature of the PC framework.

1. **Hardcopy Devices**

Utilized for recording pictures incorporate laser printers, film cameras, heat delicate gadgets, inkjet units and computerized units, for example, optical and CDROM circles. Although large-scale image processing systems still are being utilized for monstrous imaging applications, for example, preparing of satellite pictures, the pattern proceeds towards scaling down and mixing of broadly useful little PCs with image processing equipment.

1. **Network**

The network is the web that connects all the components that were mentioned. Because of the enormous measure of information intrinsic in image processing applications, the key thought in image transmission is transfer speed. In committed organizations, this ordinarily is not an issue, however interchanges with distant locales through the web are not continuously as productive.

2.2 Image Acquisition

Image acquisition is the first step of image processing. Imaging systems are a significant part in the process of image acquisition by the sensing of an image. The next process is scaling, this allows production of high-quality output images, allowing a precise image analysis [19].

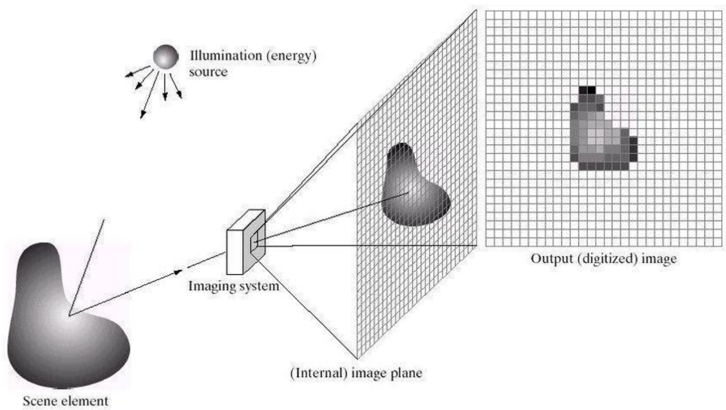


Figure 2.2: Image Acquisition

Figure 2.3: Raspberry Pi Camera Module V2

Table 2.3: Technical Specification of Raspberry Pi Camera Module V2

Camera parameter	Specification
Lens focal length	3.04 mm
Aperture f-number	2.0
Full-frame field of view	59.17 deg ✖ 58.3 deg

Table 2.3 shows the technical specifications of Raspberry Pi Camera Module V2[23].

2.3 Image Filtering

Image Filtering is usually associated with the concepts of image processing. Image filtering is the process of adjusting the colors of pixels in an image to change its appearance. Filters can be used to increase contrast and provide a range of special effects to photographs. The main goal of filters is to minimize the noise surrounding characters or patterns in the image to make image processing faster and more precise. The most common filters used in image processing are Box filter, Gaussian filter, and bilateral filters.

2.4 Philippine Peso Banknote

The Philippine currency issued by the Central Bank of the Philippines has changed a lot throughout the years, improving its looks, but importantly, improving its security features to combat counterfeit bills.

Figure 2.4: Banknotes of the Philippine Peso

There are multiple ways to tell if a Philippine peso bill is considered legitimate. These include the look, the feel, and the tilt, but when it comes to image processing detection, the properties that are mostly used and what is going to be used in this research is the look of the bill. A thorough on-hand inspection can show if the Philippine peso bill is legitimate or a counterfeit. The following are some of the security features:

Serial Number

Each bill is assigned with a unique set of alphanumeric characters at both the upper right and lower left corners of the paper bill. These sets of alphanumeric characters are composed of one to two prefix letters and six to seven numerical digits, the font size and thickness is increasing.

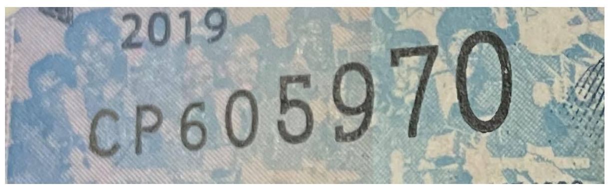


Figure 2.5: The serial number of a one-thousand-peso banknote

Security Fibers

Randomized scattering of red and blue fibers are found in the front and back of the banknote. Security Fibers are another type of safety included in Philippine currency that protects against falsification. Certified security strands in Philippine banknotes can be effectively culled out with the guide of a needle. Fake cash generally just prints the strands on paper; hence, they can't be culled out.

Watermark

When viewing the banknote against the light, a visible mirror image of the portrait and a numerical value that represents the amount of the banknote in pesos can be seen from either side of the banknote on the blank space found in the white part of the banknote.

Figure 2.6: The watermark found on a one-thousand-peso banknote

See-Through print

Located below the watermark, the word "Filipino" written in a pre-Hispanic scripture called "Baybayin", which can only be seen in full form when the banknote is viewed against the light because the half part of the word is revealed the same way as the watermark is revealed.

Figure 2.7: the word "Filipino" written in "Baybayin" on a Philippine peso banknote

Security Thread

Different values of the Philippine peso bill have varying sizes, 20 and 50 peso bills have 2mm wide thread, while 100, 200, 500, and 1000 bills have 4mm wide thread. Tilting the 100 to 1000 paper bill will result in a change of color. The security thread found on 20 and 50 peso bills are embedded inside the paper bill, while the security thread found on 100 to 1000 peso bills are embedded through the paper bill.

Figure 2.8: Security Fiber found on Philippine peso banknotes

2.5 Field Programmable Gate Arrays

Field Programmable Gate Arrays (FPGAs) are increasingly being utilized in the development of image processing algorithms. This is particularly true for real-time embedded applications, wherein latency and power are critical factors. An FPGA integrated in a programming language such as python can mostly perform the process of image processing [14]. Utilizing an FPGA to stimulate complex algorithms in image processing is difficult. The most common approach used is by utilizing an open-source programming language.

2.6 PYNQ-Z2

PYNQ is a programmable system that comprises a multi-core processor and a Field Programmable Gate Array (FPGA) applied into a single integrated circuit. The TUL PYNQ-Z2 board allows the acceleration of Python functions such as image processing, as well as the utilization and compatibility of Raspberry Pi.

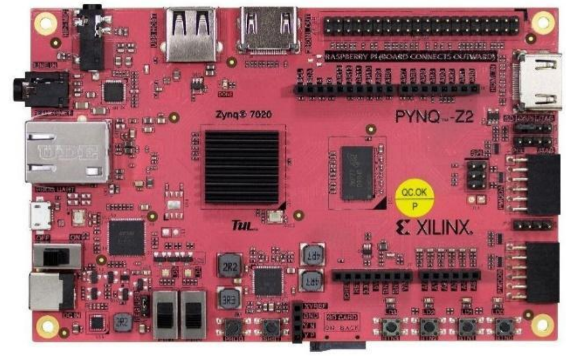


Figure 2.9: TUL PYNQ-Z2 board

CHAPTER 3

METHODOLOGY

3.1 Conceptual Framework

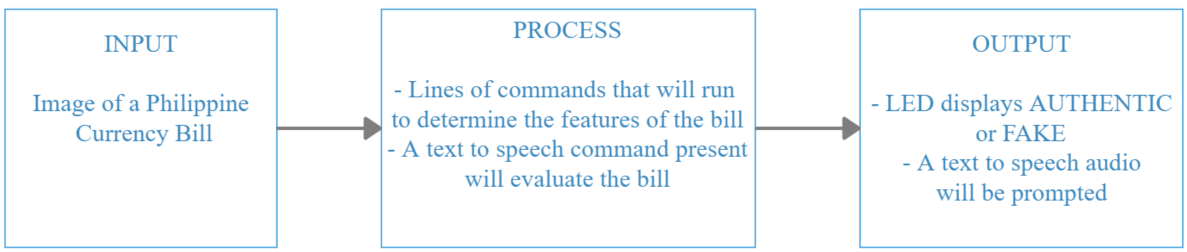


Figure 3.1: IPO of the Counterfeit Detector

The figure above shows the Input, Process, and Output (IPO) of the Counterfeit Currency Detector. The image of a Philippine currency bill will be used as an Input, then for the process, Python OpenCV will be utilized and the algorithm of the Hough Line Transform that is in the program will determine the Output that displays an LED if the Philippine currency bill, green for Authentic or red for Fake, an audio for denomination will prompt which helps those that are visually impared.

3.2 Image Processing System

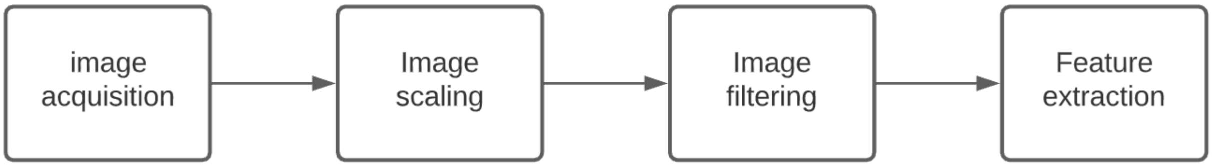


Figure 3.2: Block image processing system

3.3 System Flowchart

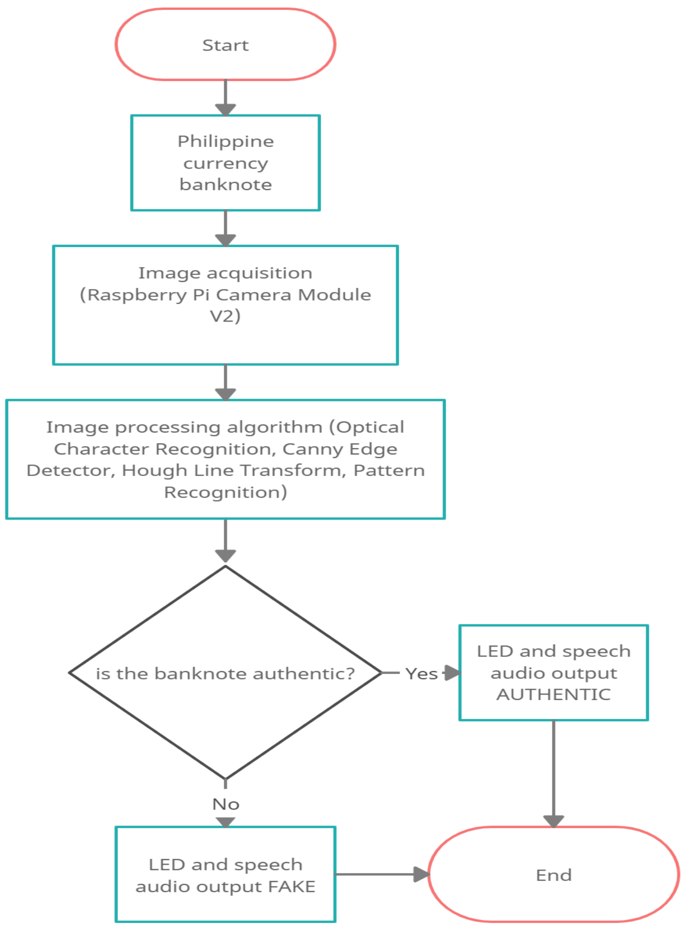


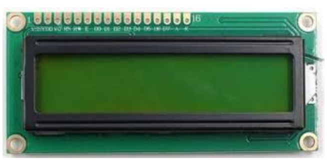
Figure 3.3: System flowchart of the Counterfeit Currency Detector

The figure above shows the system flowchart of the Philippine Counterfeit Currency Detector. The flow of the chart will begin from START, based on **Figure 3.1**, it was mentioned that the Input of the study will be an image of a Philippine currency bill, this is where the information will be obtained. The next step will be the Python program that will mostly rely on OpenCV since it will perform the algorithm that is needed to obtain the data. The Hough Line Transform will be the algorithm that will be utilized. It will map the lines present in the bills, due to the unique characteristics of a genuine bill, the program will determine if it lacks any of the characteristics specifically the Security Thread present in all genuine currency bills. If the program does not recognize the security bill, it can be concluded that the bill is counterfeit. If the bill displays lines, it is known that Security Thread is present and therefore can conclude that the currency bill is authentic.

3.4 Materials and Equipment

Table 3.4: Materials/Equipment and Description

	Equipment	Description
	Raspberry Pi Camera Module V2	The Raspberry Pi Camera Module V2 is a 8 megapixel camera system capable of capturing 3280 x 2464 pixel images and supports up to a maximum of 1080p30 video.
	TUL PYNQ-Z2 board	The TUL PYNQ-Z2 board comprises a 650MHz dual-core Cortex-A9 processor and a built in FPGA, and is capable of utilizing Python OpenCV and Raspberry Pi
	HDMI cable	The High Definition Multimedia Interface (HDMI) cable allows port to port transfer of high-resolution digital video and sound. This will be used for connecting the TUL PYNQ-Z2 board to the output monitor.
	Ethernet cable	The ethernet cable is commonly used for wired network connections. This will be used by the TUL PYNQ-Z2 board so that updating the board is possible.

	USB A to Micro-USB Cable	A micro-USB is a smaller version of the Universal Serial Bus (USB) interface, allowing such connections but for compact devices. This will be used to power the TUL PYNQ-Z2 board.
	Speaker	This will be connected to the audio output of the PYNQ-Z2 board, and it is where the audio prompt is going to be outputted.
	LCD screen	This will be used for the output of the words “FAKE” and “AUTHENTIC”.

3.5 Proposed Prototype

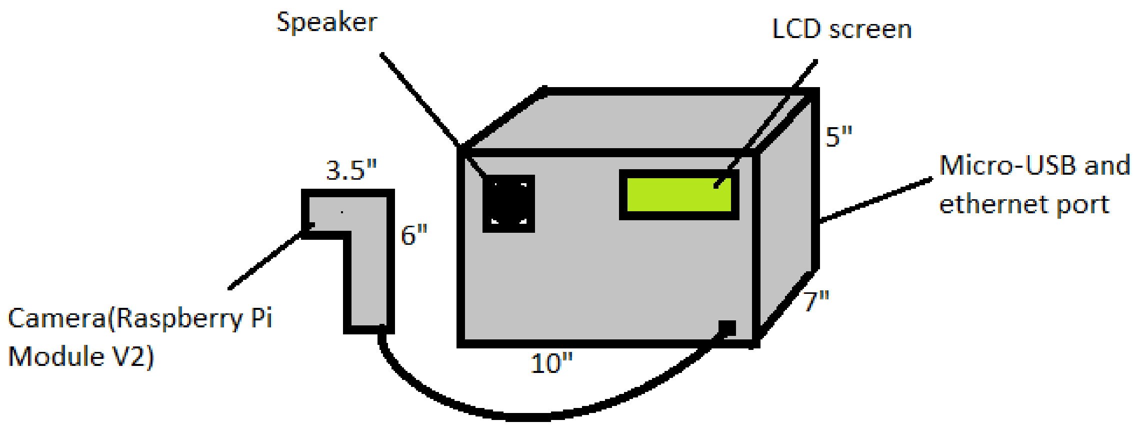


Figure 3.5.1: Proposed Prototype

The proposed prototype on the outside involves two parts. The first part would be the image scanner, which would serve as the image acquisition device. This part would include the Raspberry Pi Camera Module V2. The second part would include the PYNQ-Z2 board, the output LCD screen, and the speaker.

3.6 Calibration of the Counterfeit Currency Detector

The calibration of the image processing will be conducted on different trials for the counterfeit currency and the genuine currency and the data that will be gathered will be compared with each other. The codes that will be used for the study will be compiled in this section of the paper. Codes for the OpenCV Python will focus on the Hough Line Transform algorithm that will determine if the currency is counterfeit. Lack of data and lines for the output will be classified as a counterfeit currency.

```python
"""
@file hough_lines.py
@brief This program demonstrates line finding with the Hough transform
"""
import sys
import math
import cv2 as cv
import numpy as np


def main(argv):

    default_file = 'sudoku.png'
    filename = argv[0] if len(argv) > 0 else default_file

    # Loads an image
    src = cv.imread(cv.samples.findFile(filename), cv.IMREAD_GRAYSCALE)

    # Check if image is loaded fine
    if src is None:
        print ('Error opening image!')
        print ('Usage: hough_lines.py [image_name -- default ' + default_file + '] \n')
        return -1



    dst = cv.Canny(src, 50, 200, None, 3)


    # Copy edges to the images that will display the results in BGR
    cdst = cv.cvtColor(dst, cv.COLOR_GRAY2BGR)
    cdstP = np.copy(cdst)
```

Figure 3.6A: Sample code for the OpenCV Python

```python
    lines = cv.HoughLines(dst, 1, np.pi / 180, 150, None, 0, 0)

    if lines is not None:
        for i in range(0, len(lines)):
            rho = lines[i][0][0]
            theta = lines[i][0][1]
            a = math.cos(theta)
            b = math.sin(theta)
            x0 = a * rho
            y0 = b * rho
            pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
            pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))

            cv.line(cdst, pt1, pt2, (0,0,255), 3, cv.LINE_AA)



    linesP = cv.HoughLinesP(dst, 1, np.pi / 180, 50, None, 50, 10)

    if linesP is not None:
        for i in range(0, len(linesP)):
            l = linesP[i][0]
            cv.line(cdstP, (l[0], l[1]), (l[2], l[3]), (0,0,255), 3, cv.LINE_AA)

    cv.imshow("Source", src)
    cv.imshow("Detected Lines (in red) - Standard Hough Line Transform", cdst)
    cv.imshow("Detected Lines (in red) - Probabilistic Line Transform", cdstP)

    cv.waitKey()
    return 0


if __name__ == "__main__":
    main(sys.argv[1:])
```

Figure 3.6B: Sample code for the OpenCV Python

```python
default_file = 'sudoku.png'
filename = argv[0] if len(argv) > 0 else default_file

# Loads an image
src = cv.imread(cv.samples.findFile(filename), cv.IMREAD_GRAYSCALE)

# Check if image is loaded fine
if src is None:
    print ('Error opening image!')
    print ('Usage: hough_lines.py [image_name -- default ' + default_file + '] \n')
    return -1
```

Figure 3.6.1: Codes use to load an image

```python
# Edge detection
dst = cv.Canny(src, 50, 200, None, 3)
```

Figure 3.5.2: Codes use to detect the edge of the image using Canny detector

```python
#  Standard Hough Line Transform
lines = cv.HoughLines(dst, 1, np.pi / 180, 150, None, 0, 0)
```

```python
# Draw the lines
if lines is not None:
    for i in range(0, len(lines)):
        rho = lines[i][0][0]
        theta = lines[i][0][1]
        a = math.cos(theta)
        b = math.sin(theta)
        x0 = a * rho
        y0 = b * rho
        pt1 = (int(x0 + 1000*(-b)), int(y0 + 1000*(a)))
        pt2 = (int(x0 - 1000*(-b)), int(y0 - 1000*(a)))

        cv.line(cdst, pt1, pt2, (0,0,255), 3, cv.LINE_AA)
```

Figure 3.6.3: Codes used to display the result for the Standard Hough Line Transform

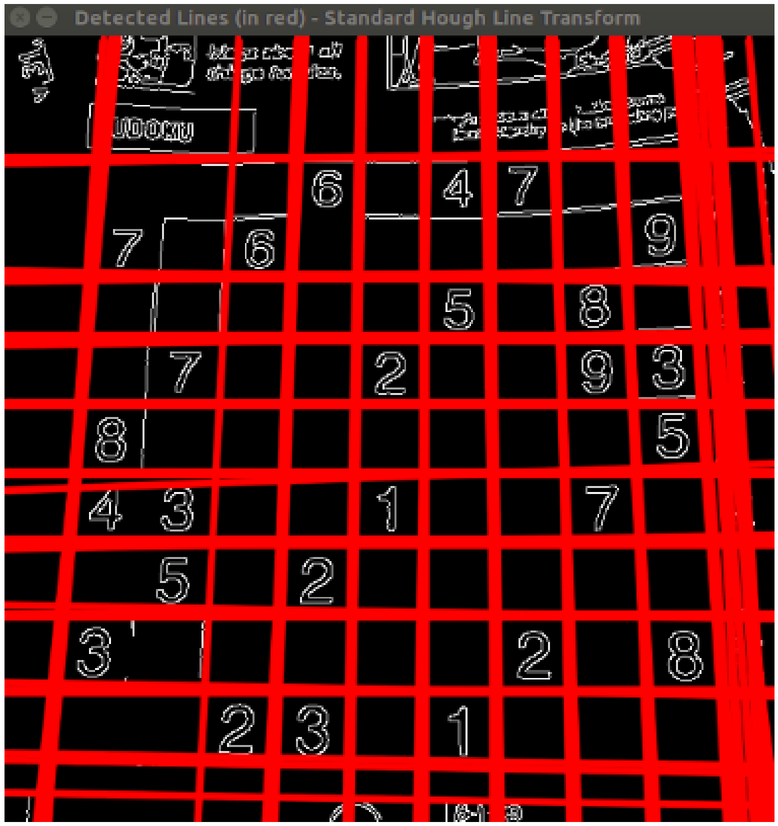


Figure 3.6.4: Sample Output 1

```
# Probabilistic Line Transform
linesP = cv.HoughLinesP(dst, 1, np.pi / 180, 50, None, 50, 10)
```

```
# Draw the lines
if linesP is not None:
    for i in range(0, len(linesP)):
        l = linesP[i][0]
        cv.line(cdstP, (l[0], l[1]), (l[2], l[3]), (0,0,255), 3, cv.LINE_AA)
```

Figure 3.6.5: Codes used to display the result for the Probabilistic Hough Line Transform

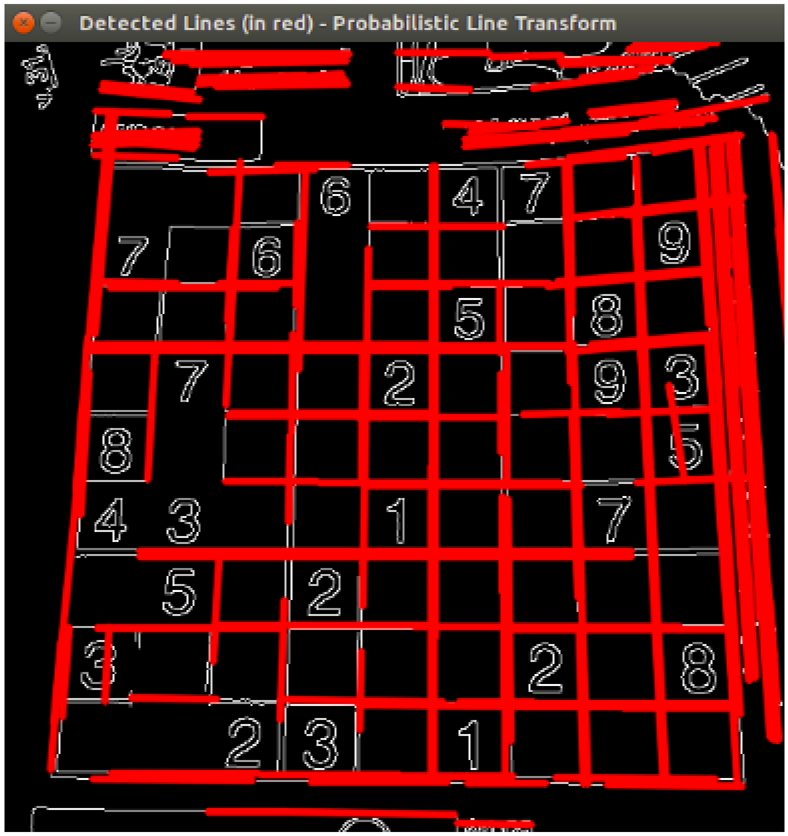


Figure 3.6.6: Sample Output 2

For the calibration of the TUL PYNQ-Z2 board, the board must first be set up, this can be done by first setting the boot jumper located on the right side of the TUL PYNQ-Z2 board to the SD position, this implies that the board would boot from a Micro-SD card, similar to how a computer can boot from a compact disc or a universal serial bus (USB). The next step is to power the board using the micro USB cable, this can be done by setting the power jumper to the usb position, which can be located on the lower part of the board. The next step is to boot up the TUL PYNQ-Z2 board using a Micro SD card loaded with the PYNQ-Z2 image, which is a downloadable file that allows the board to be programmed. The next step is to connect the ethernet cable to the board, to start the process of connecting the board to a network connection to allow the use of Jupyter notebook, this can be done by connecting it to a computer. The first step is to assign your computer a static IP address, this prevents the IP address from changing. The last step is to connect the board to the ethernet port of the computer and browse to http://192.168.2.99, this directs to a log-in screen, whereas the username and the password are both “xilinx”. After logging in, the board is now connected to Jupyter Notebook.

The following codes written in Python will focus on the setting up of an HDMI connection between the PYNQ TUL PYNQ-Z2 board and the Raspberry Pi Camera Module V2, as well as the connection between the PYNQ TUL PYNQ-Z2 board and the output monitor:

```
from pynq import Overlay
from pynq.lib.video import *

base = Overlay('base.bit')
hdmi_in = base.video.hdmi_in
hdmi_out = base.video.hdmi_out
```

Figure 3.6C: Codes used to set up an HDMI instance, in and out.

```
hdmi_in.configure()
hdmi_out.configure(hdmi_in.mode)
```

Figure 3.6.5: Codes used to configure an HDMI connection.

```
hdmi_in.start()
hdmi_out.start()
```

Figure 3.6.6: Codes used to activate an HDMI controller.

```python
from pynq.overlays.base import BaseOverlay
base = BaseOverlay("base.bit")
pAudio = base.audio
pAudio.load("/home/xilinx/jupyter_notebooks/base/audio/data/recording_0.wav")

pAudio.play()
```

Figure 3.6.7: Codes used to activate audio prompt using PYNQ-Z2

3.7 Data Gathering

For the data gathering of the study, the calculation of the proposed framework has been detailed in two sections. The initial segment subtleties the distinguishing proof also, order of notes dependent on void districts. The second part manages notes for which the researchers have to utilize layout coordinating. Note that the worth of r utilized in the calculation is the ratio of the quantity of dark pixels to white pixels inside a specific currency bill.

Determine the currency value by:

1: Convert the image to black and white

2: Extract the center region

3: Calculate r as the ratio of number of black pixels to the number of white pixels

4: else if r < 4.1% then

4.1: Output: Philippine Peso

4.2: Determine denomination based on color using K-means clustering algorithm

The algorithm present here will show the characteristics present in a Philippine currency bill. The following steps are provided:

i. READ the input RGB currency image

ii. Separate into Red, Green and Blue channels

iii. Denoise each channel of the image

iv. Recombine channels

a. SET fet to [color feature, edge feature, texture

feature]

b. SET size of the database to zero

v. READ the size of the database

vi. FOR counter is less than the size of the database

a. READ database image [counter]

b. Denoise each channel of the image

c. Recombine channels

d. SET fet1 to [color feature, edge feature, texture

feature]

e. SET D[counter] to [CALCULATE Euclidean

distance between fet & fet1]

vii. END FOR

viii. GET the index & distance for minimum D

ix. READ the value of index from the database

x. DISPLAY the value

a. IF distance < 0.003 && distance == 0

b. ADD image & value to the database

xi. END IF

Table 3.7.1 Recognition Accuracy for the characteristics of Philippine Currency Notes (Genuine)

Note Type	**No. of Testing Samples**	**Recognition Accuracy**
20 pesos	1000	
50 pesos	1000	
100 pesos	1000	
200 pesos	1000	
500 pesos	1000	
1000 pesos	1000	
Average Accuracy		

Table 3.7.1 Recognition Accuracy for the characteristics of Philippine Currency Notes (Counterfeit)

Note Type	No. of Testing Samples	Recognition Accuracy
20 pesos	1000	
50 pesos	1000	
100 pesos	1000	
200 pesos	1000	
500 pesos	1000	
1000 pesos	1000	
Average Accuracy		

Statistical analysis

Compare the detection of security fibers in legitimate currency and counterfeit currency.

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\sqrt{(}s^2(\frac{1}{n_1} + \frac{1}{n_2}))}$$

REFERENCES

[1] T. Treebupachatsakul and S. Poomrittigul, “Bacteria Classification using Image Processing and Deep Learning,” *34th Int. Tech. Conf. Circuits/Systems, Comput. Commun. ITC-CSCC 2019*, pp. 2–4, 2019, doi: 10.1109/ITC-CSCC.2019.8793320.

[2] A. Ranjan, A. K. S. Arya, and M. Ravinder, “Quantum Techniques for Image Processing,” *Proc. - IEEE 2020 2nd Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2020*, pp. 1035–1039, 2020, doi: 10.1109/ICACCCN51052.2020.9362910.

[3] Y. Cheng and B. Li, “Image segmentation technology and its application in digital image processing,” *Proc. IEEE Asia-Pacific Conf. Image Process. Electron. Comput. IPEC 2021*, pp. 1174–1177, 2021, doi: 10.1109/IPEC51340.2021.9421206.

[4] G. Vladimir, I. Evgen, and N. L. Aung, “Automatic detection and classification of weaving fabric defects based on digital image processing,” *Proc. 2019 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. ElConRus 2019*, no. c, pp. 2218–2221, 2019, doi: 10.1109/EIConRus.2019.8657318.

[5] A. Chakraborty, “Image processing and image pattern recognition a programming tutorial,” *Proc. - 2018 1st IEEE Int. Conf. Artif. Intell. Ind. AI4I 2018*, pp. 122–123, 2019, doi: 10.1109/AI4I.2018.8665702.

[6] P. J. A. D. S. S. Shaveta Malik3, K. Radhika, and P. Vishalini, “DIGITAL IMAGE PROCESSING– Sequence , Components and Pros,” *K.Radhika al, / Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 2, pp. 922–924, 2016.

[7] J. Akter, M. K. Hossen, and M. S. A. Chowdhury, “Bangladeshi Paper Currency Recognition System Using Supervised Learning,” *Int. Conf. Comput. Commun. Chem. Mater. Electron. Eng. IC4ME2 2018*, pp. 2–5, 2018, doi: 10.1109/IC4ME2.2018.8465595.

[8] R. R. Palekar, S. U. Parab, D. P. Parikh, and V. N. Kamble, “Real time license plate detection using openCV and tesseract,” *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-January, pp. 2111–2115, 2018, doi: 10.1109/ICCSP.2017.8286778.

[9] J. Iacovacci and L. Lacasa, “Visibility graphs for image processing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 974–987, 2020, doi: 10.1109/TPAMI.2019.2891742.

[10] V. Abburu, S. Gupta, S. R. Rimitha, M. Mulimani, and S. G. Koolagudi, “Currency Recognition System Using Image Processing,” no. August, pp. 10–12, 2017.

[11] R. M. Colaco and R. Fernandes, “Efficient Image Processing Technique for Authentication of Indian Paper Currency,” 2021.

[12] S. Arya and M. Sasikumar, “Fake Currency Detection,” *2019 Int. Conf. Recent Adv. Energy-Efficient Comput. Commun. ICRAECC 2019*, pp. 2019–2022, 2019, doi: 10.1109/ICRAECC43874.2019.8994968.

[13] A. S. Alon, R. M. Dellosa, N. U. Pilueta, H. D. Grimaldo, and E. T. Manansala, “EyeBill-PH: A Machine Vision of Assistive Philippine Bill Recognition Device for

Visually Impaired,” *2020 11th IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2020 - Proc.*, no. August, pp. 312–317, 2020, doi: 10.1109/ICSGRC49013.2020.9232557.

[14] D. G. Bailey, “Image processing using FPGAs,” *J. Imaging*, vol. 5, no. 5, 2019, doi: 10.3390/jimaging5050053.

[15] M. Vashisht and M. Bhatia, “Role of Mathematics in Image Processing,” *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 538–543, 2019, doi: 10.1109/COMITCon.2019.8862438.

[16] E. Provenzi, "*Color Image Processing" J. Imaging,* . 2018.

[17] L. Likforman-Sulem and E. Kavallieratou, “Document image processing,” *J. Imaging*, vol. 4, no. 7, 2018, doi: 10.3390/jimaging4070084.

[18] B. B., “A Study on the Importance of Image Processing and Its Apllications,” *Int. J. Res. Eng. Technol.*, vol. 03, no. 15, pp. 155–160, 2014, doi: 10.15623/ijret.2014.0315029.

[19] Shonima Vasudevan and Dr. M. Nagarajan, “A Basic Study of Image Processing and Its Application Areas,” *Int. J. Eng. Res.*, vol. V6, no. 07, pp. 343–348, 2017, doi: 10.17577/ijertv6is070217.

[20] F. Siddiqui *et al.*, “FPGA-based processor acceleration for image processing applications,” *J. Imaging*, vol. 5, no. 1, 2019, doi: 10.3390/jimaging5010016.

[21] K. Haeublein, W. Brueckner, S. Vaas, S. Rachuj, M. Reichenbach and D. Fey, "Utilizing PYNQ for Accelerating Image Processing Functions in ADAS Applications," *ARCS*

Workshop 2019; 32nd International Conference on Architecture of Computing Systems, 2019, pp. 1-8.

[22] P. Ogden, “PYNQ for Compute Acceleration PYNQ for Compute Acceleration,” 2020.

[23] M. Pagnutti *et al.*, “Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes,” *J. Electron. Imaging*, vol. 26, no. 1, p. 013014, 2017, doi: 10.1117/1.jei.26.1.013014.