PRICE

CALCULATOR

OBJECTIVES

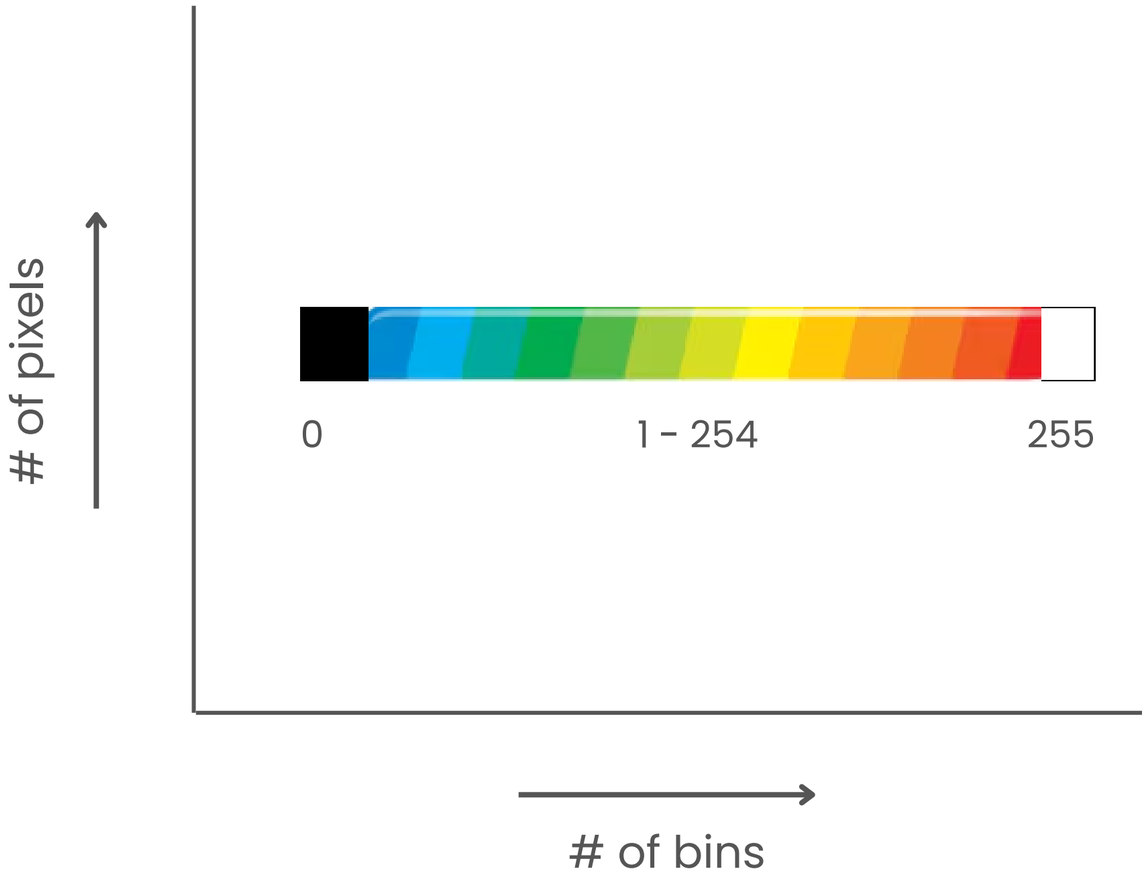
- Solve the dilemma of every student or customer when they want to print files – with the use of a histogram

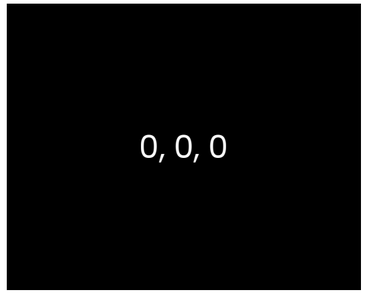
PRICE RANGE

- ₱5.00 - Black Text Color
- ₱5.01 → ₱19.99 - Colored
- ₱20.00 - Whole Page Colored

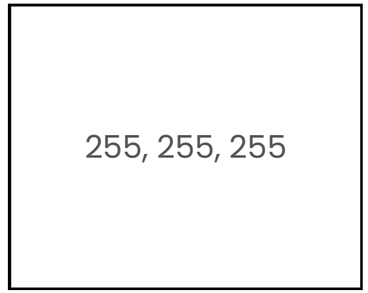
Note: ₱5.00 - ₱20.00 Price Range

HISTOGRAM

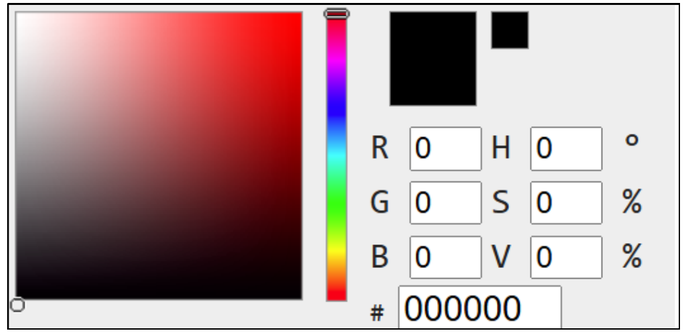




RGB Format

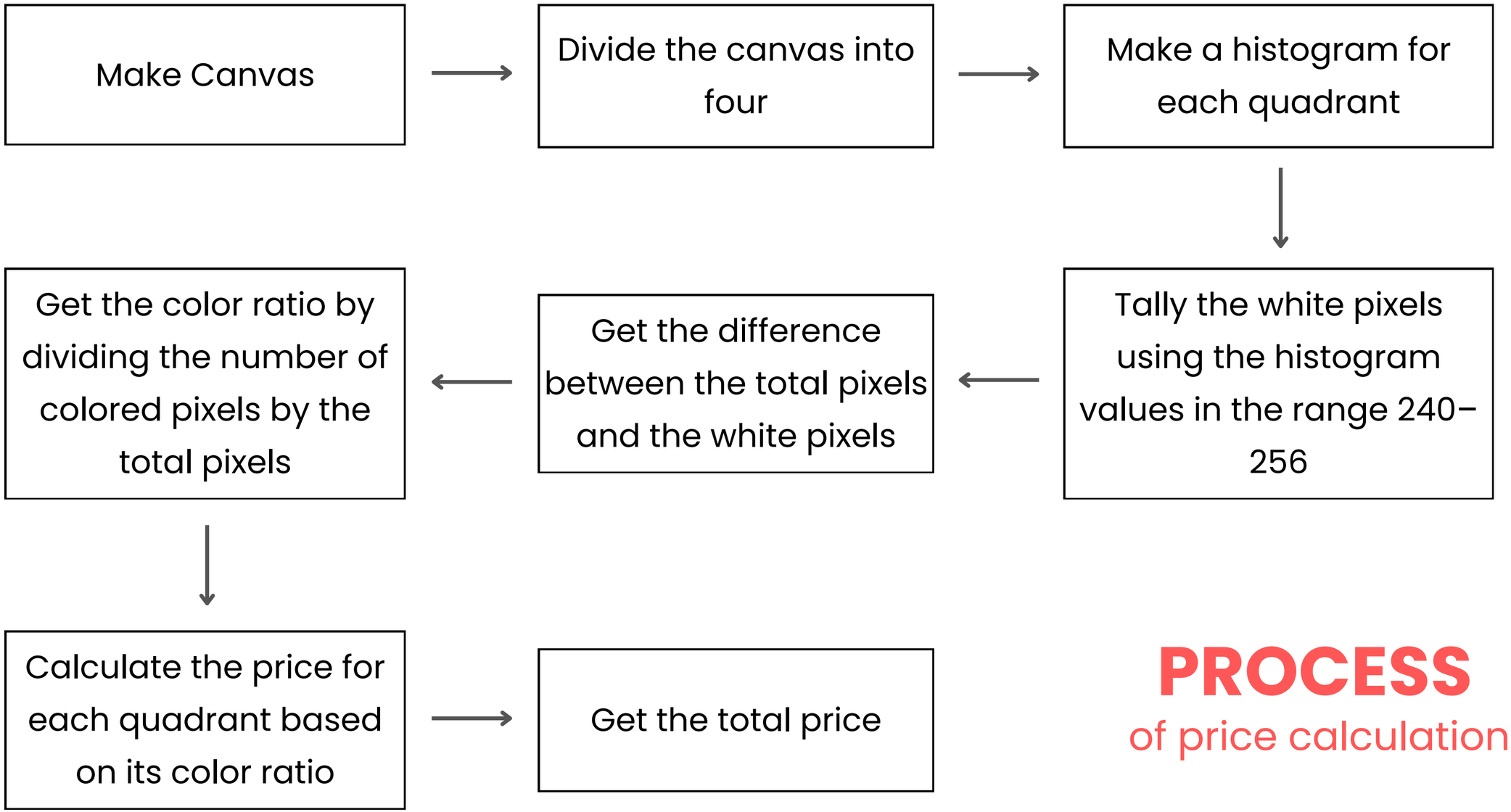


RGB Format



Most of the students in MSU-IIT (or even in other universities) experienced going to computer cafes to print their documents. Typically, a page with black text costs 2 pesos. However, if a student wants to print a colored page, the teller in the cafe will put up a price based on how large or small the colored portions are on the page. If the colored portion on the page is around a quarter, the price will be 5 pesos. If it is around half the page, the price will be around 10 pesos. If the whole page contains colored figures, the price will go up to 20 pesos. As a student-customer, there will be instances that we find the heuristic determination of the price "unfair" due to a lack of standards in the computation.

To address this dilemma, you will create a solution to calculate the price of the colored page according to the distribution of colors (or color histograms). Let us assume that the image resolution will be 600 by 900 pixels. Generate sample images for your test bench. Create a strategy on how to price the images. For uniformity, the minimum price will be 5 pesos and the maximum price will be 20 pesos. Also, your calculated price must be in two decimal places only, such as 11.23, 10.02, 5.00, etc.

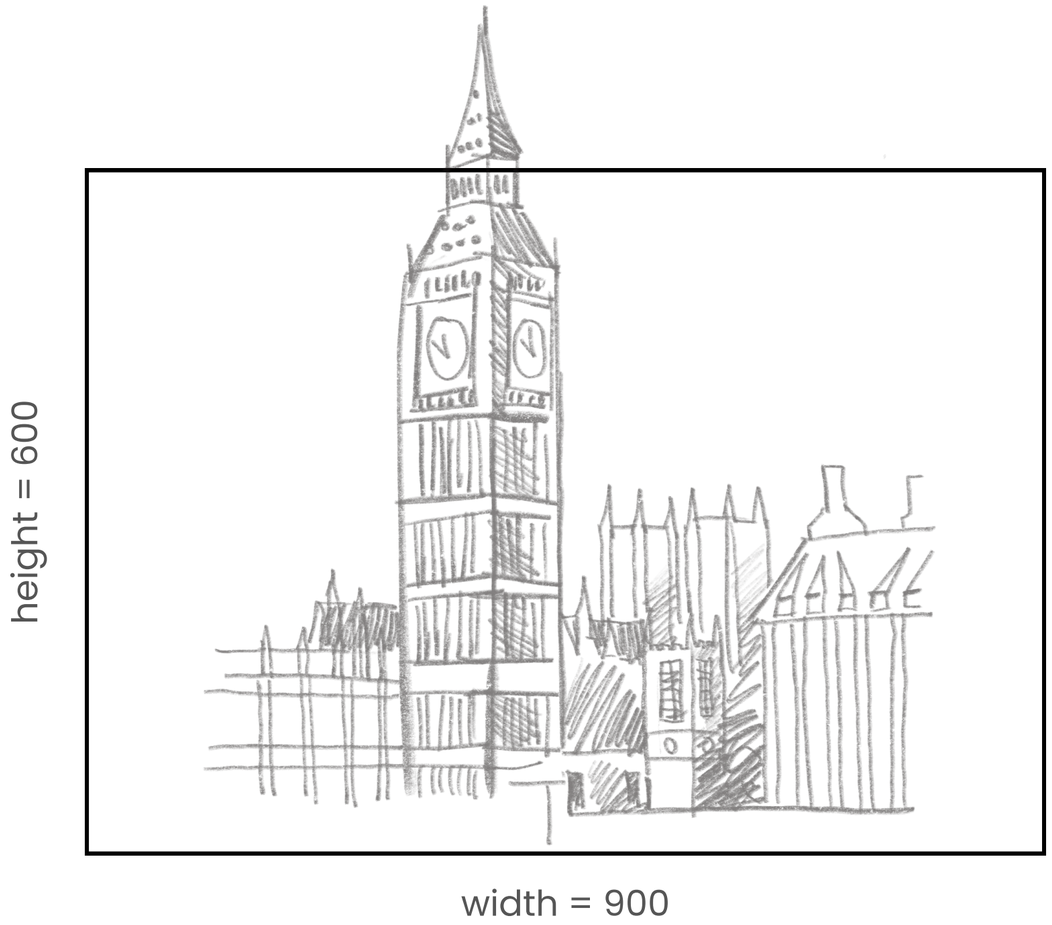


PROCESS

of price calculation

Make canvas

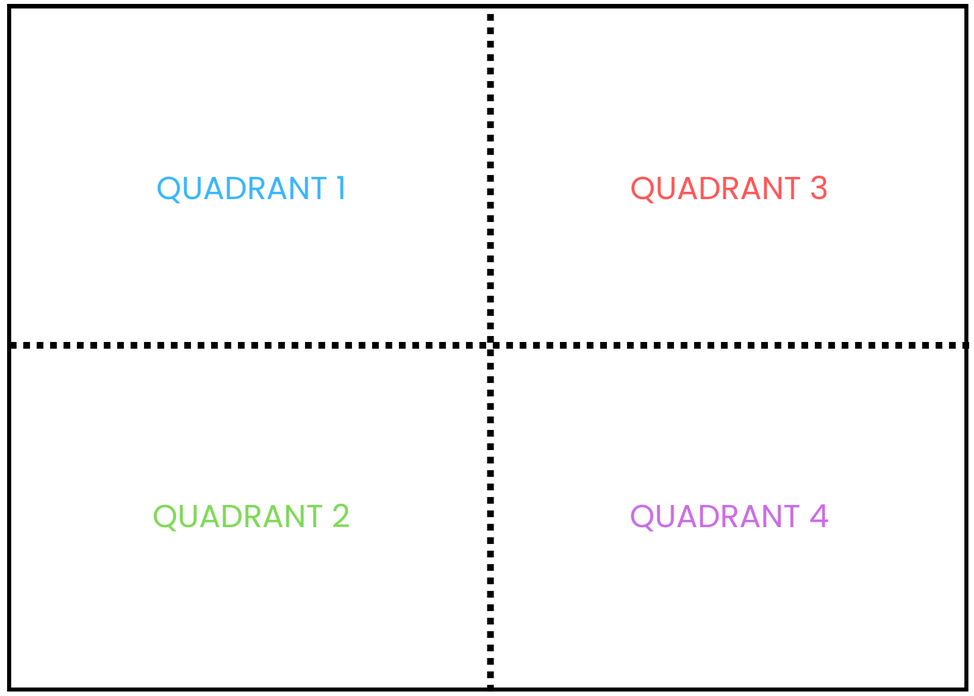
```
image = cv2.imread(args["image"])
image = cv2.resize(image, (900, 600))
cv2.imshow("Original", image)
```



Divide into 4

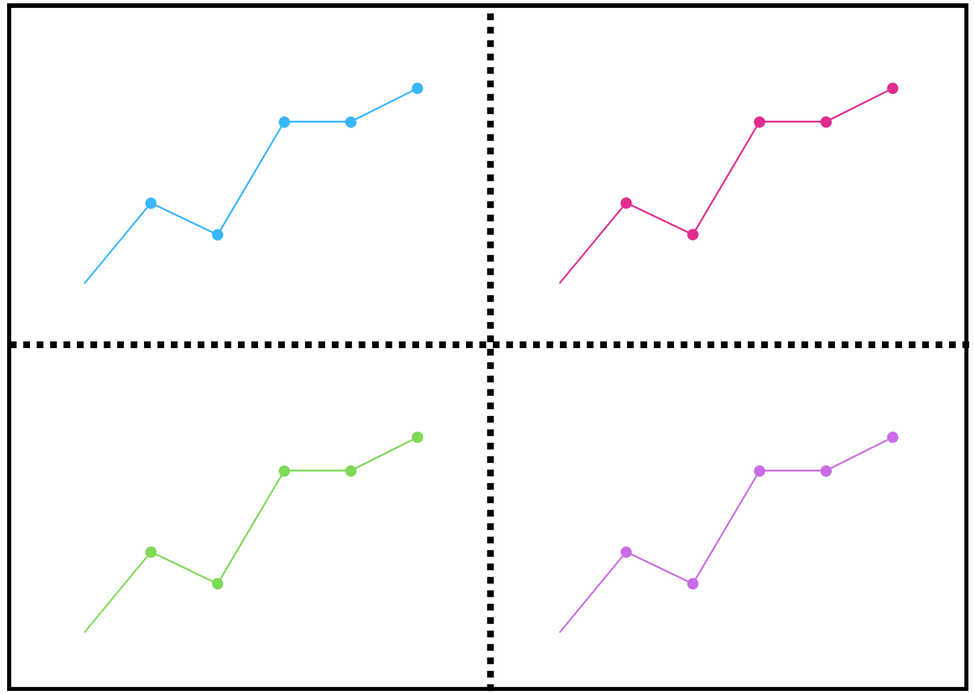
```
quadrants= {
    "Quadrant 1 (Top-Left)": image[0:300, 0:450],
    "Quadrant 2 (Bottom-Left)": image[300:600, 0:450],
    "Quadrant 3 (Top-Right)": image[0:300, 450:900],
    "Quadrant 4 (Bottom-Right)": image[300:600, 450:900]
}
```

image[0:300, 0:450] = (225, 0, 0)
image[300:600, 0:450] = (0, 255, 0)
image[0:300, 450:900] = (0, 0, 225)
image[300:600, 450:900] = (225, 0, 225)



Make a histogram

```
chans = cv2.split(corner)
white_pixels = 0

plt.figure()
plt.title("Flattened ColorHistogram")
plt.xlabel("Bins")
plt.ylabel("# of Pixels")

for (chan, color) in zip (chans, ("b", "g", "r")):
    hist = cv2.calcHist([chan], [0], None, [256], [0, 256])
    white_pixels += hist[240:256].sum()
    plt.plot(hist, color=color)
    plt.xlim([0, 256])
```



Price calculation

```
white_pixels /= 3.0
total_pixels = corner.shape[0] * corner.shape[1]
colored_pixels = total_pixels - white_pixels
color_ratio = colored_pixels / total_pixels

price = 1.25 + color_ratio * (5.0 - 1.25)
price = round(price, 2)

print(f"{title} = Price: {price:.2f}")
return price
```

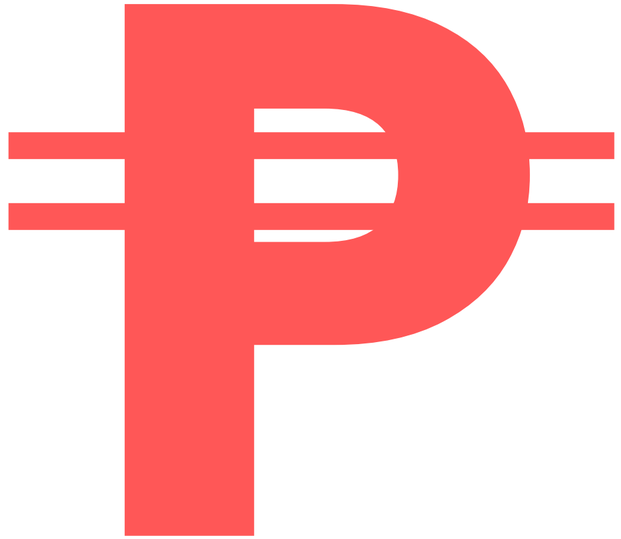
Price Range

- ₱5.00 - ₱20.00

$$\frac{20}{4} = 5 \longrightarrow \frac{5}{4} = 1.25$$

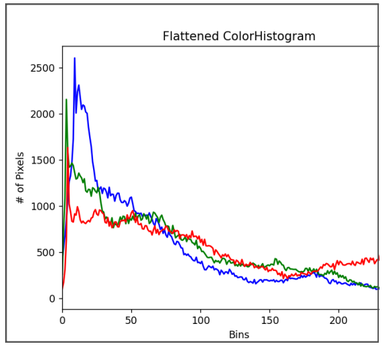
Total Price

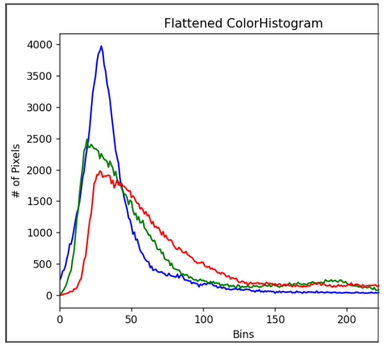
```
for name, corner in quadrants.items():
  price = price_calculator(corner, name)
  total_price += price

print(f"\nTOTAL PRICE for the page = {total_price:.2f} pesos")
```

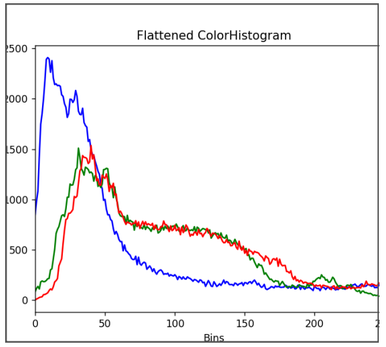


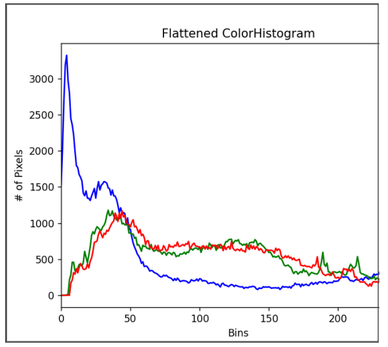
TESTBENCH


MAX
OUT


Flattened ColorHistogram
# of Pixels
Bins
0
500
1000
1500
2000
2500
0
50
100
150
200


Flattened ColorHistogram
# of Pixels
Bins
0
500
1000
1500
2000
2500
3000
3500
4000
0
50
100
150
200

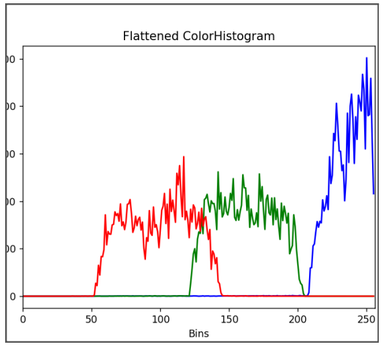

Flattened ColorHistogram
Bins
0
500
0
50
100
150
200

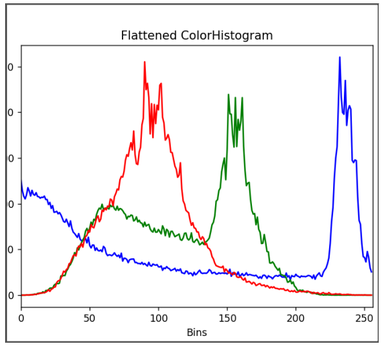

Flattened ColorHistogram
# of Pixels
Bins
0
500
1000
1500
2000
2500
3000
0
50
100
150
200

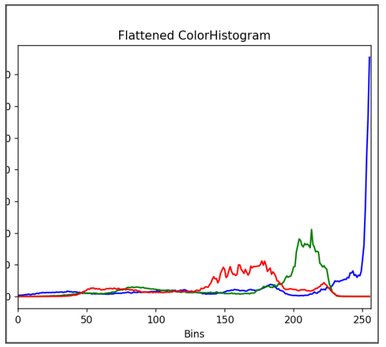
TESTBENCH 1

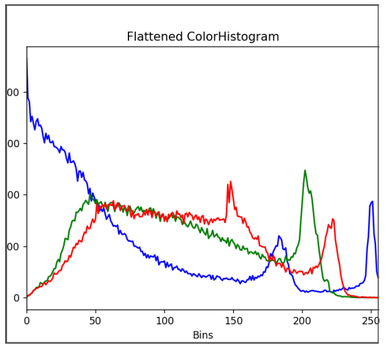
Quadrant 1 (Top-Left) = Price: 4.94
Quadrant 2 (Bottom-Left) = Price: 4.95
Quadrant 3 (Top-Right) = Price: 4.96
Quadrant 4 (Bottom-Right) = Price: 4.8

TOTAL PRICE for the page = 19.72 pesos


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins

TESTBENCH 2

Quadrant 1 (Top-Left) = Price: 4.44
Quadrant 2 (Bottom-Left) = Price: 4.44
Quadrant 3 (Top-Right) = Price: 4.87
Quadrant 4 (Bottom-Right) = Price: 4.94

TOTAL PRICE for the page = 18.69 pesos

Six Sigma Document Control Plan Template

Company Name	Project ID	Document Number	Initiation Date
Process Name	**Process Owner**	**Revision Number**	**Revision Date**

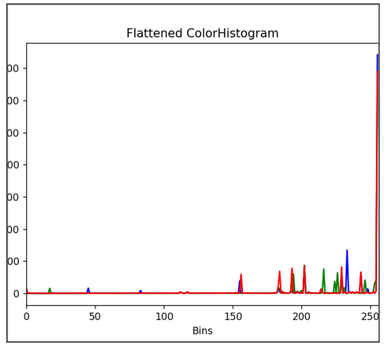
Prepared by	Date Prepared
Approved by	**Date Approved**

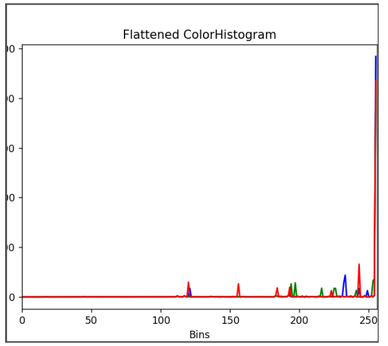
Project Scope

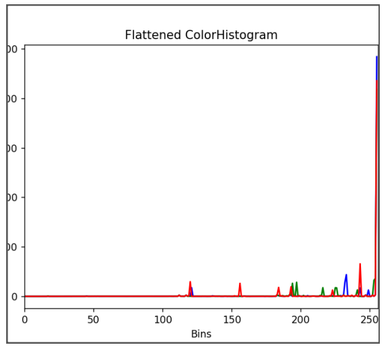
Control Plan

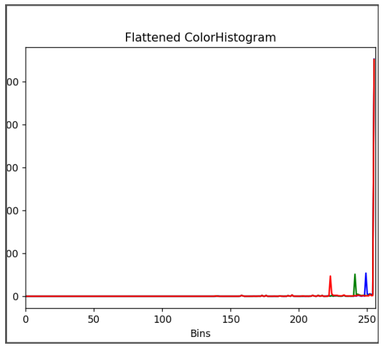
Start Date	End Date	Page No.	– of –	Total Pages

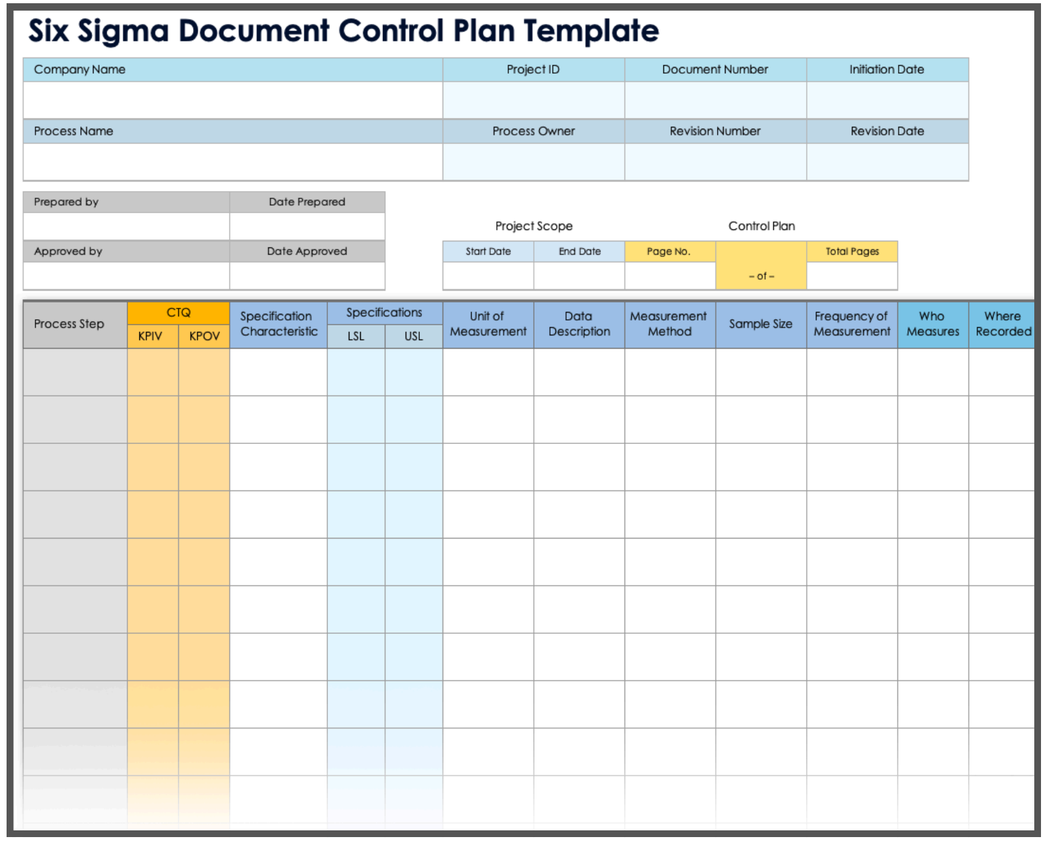
Process Step	CTQ		Specification Characteristic	Specifications		Unit of Measurement	Data Description	Measurement Method	Sample Size	Frequency of Measurement	Who Measures	Where Recorded
	KPIV	KPOV		LSL	USL							











Six Sigma Document Control Plan Template

Company Name	Project ID	Document Number	Initiation Date
Process Name	**Process Owner**	**Revision Number**	**Revision Date**

Prepared by	Date Prepared
Approved by	**Date Approved**

Project Scope

Control Plan

Start Date	End Date	Page No.	– of –	Total Pages

Process Step	CTQ		Specification Characteristic	Specifications		Unit of Measurement	Data Description	Measurement Method	Sample Size	Frequency of Measurement	Who Measures	Where Recorded
	KPIV	KPOV		LSL	USL							

TESTBENCH 3

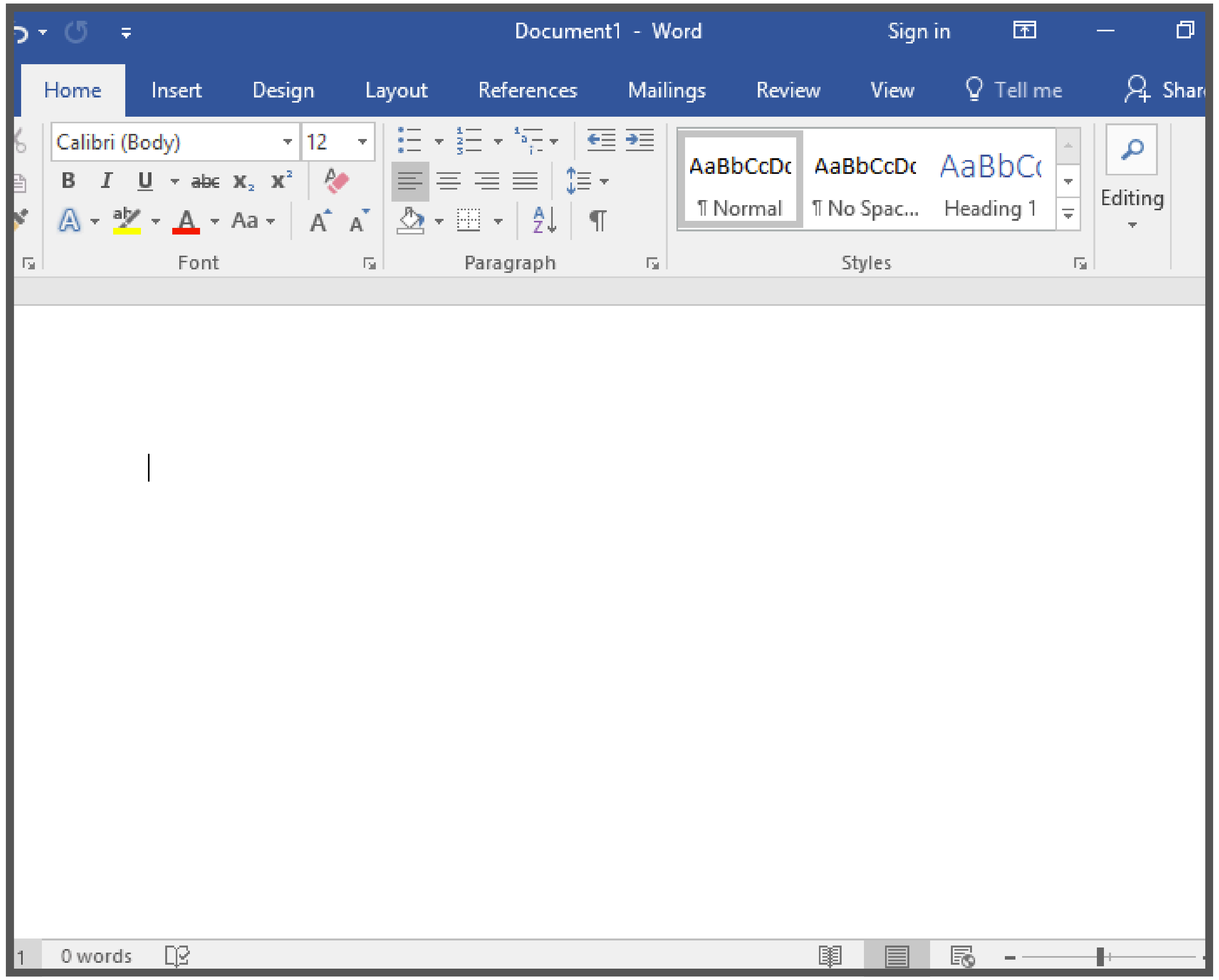
Quadrant 1 (Top-Left) = Price: 2.77

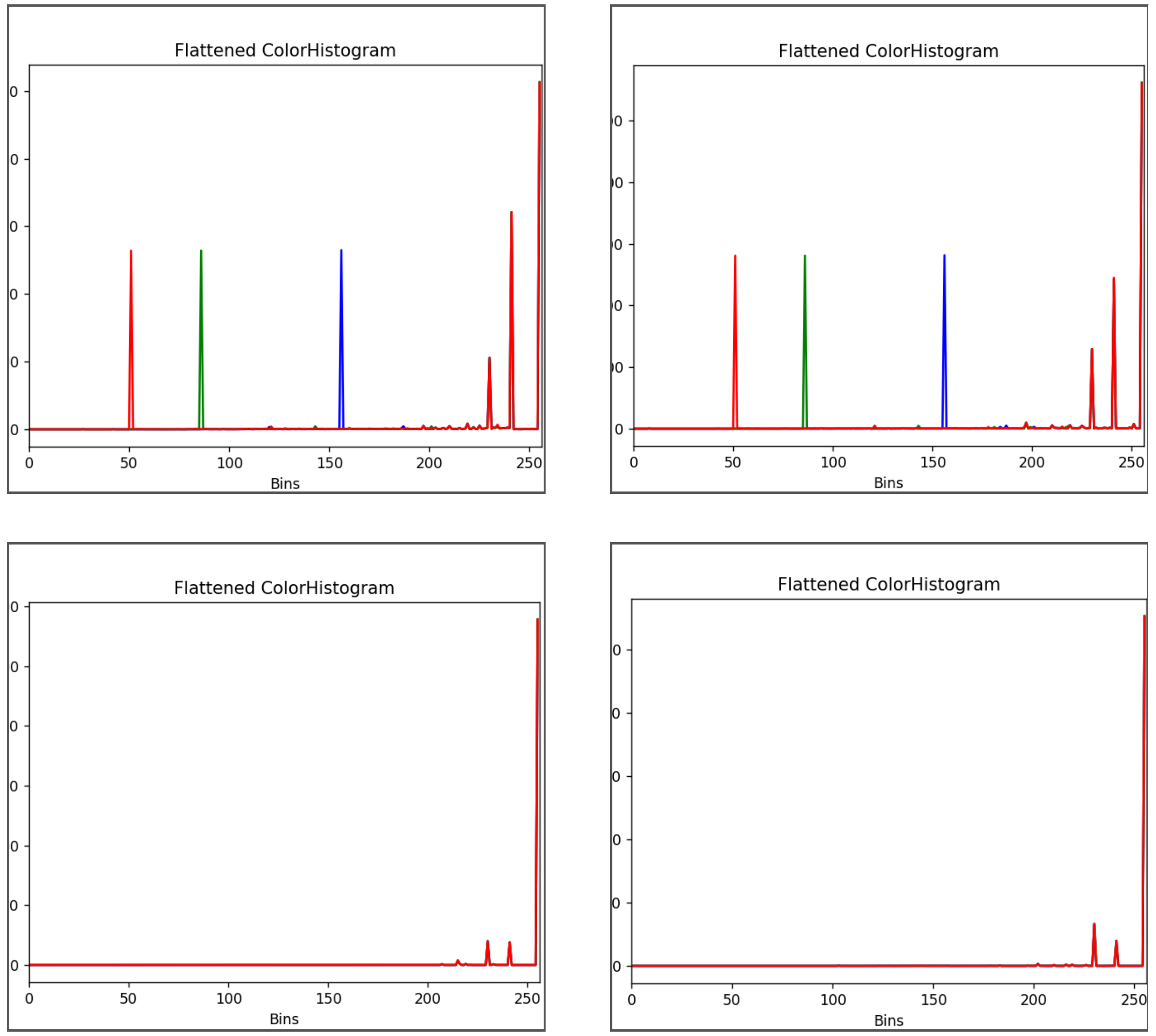
Quadrant 2 (Bottom-Left) = Price: 2.14

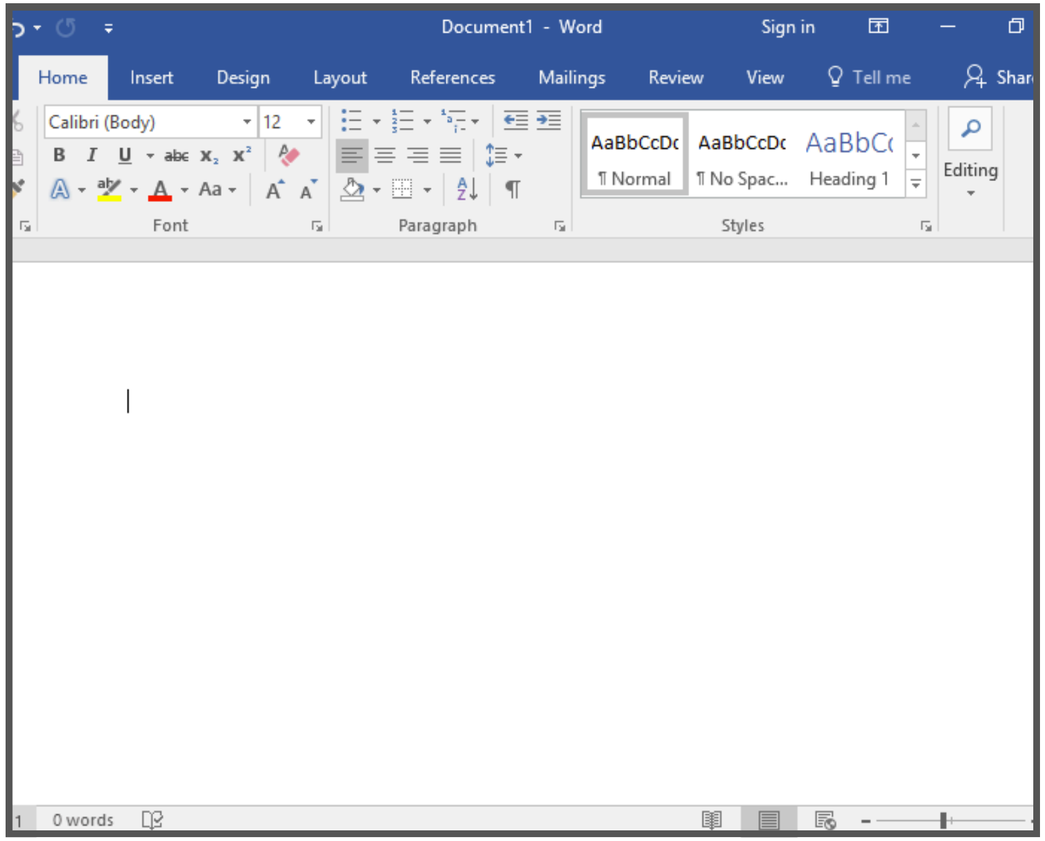
Quadrant 3 (Top-Right) = Price: 2.48

Quadrant 4 (Bottom-Right) = Price: 1.58

TOTAL PRICE for the page = 8.97 pesos


Document1 - Word
Sign in
Home
Insert
Design
Layout
References
Mailings
Review
View
Tell me
Calibri (Body)
12
AaBbCcDc
¶ Normal
AaBbCcDc
¶ No Spac...
AaBbCc
Heading 1
Editing
Font
Paragraph
Styles
0 words

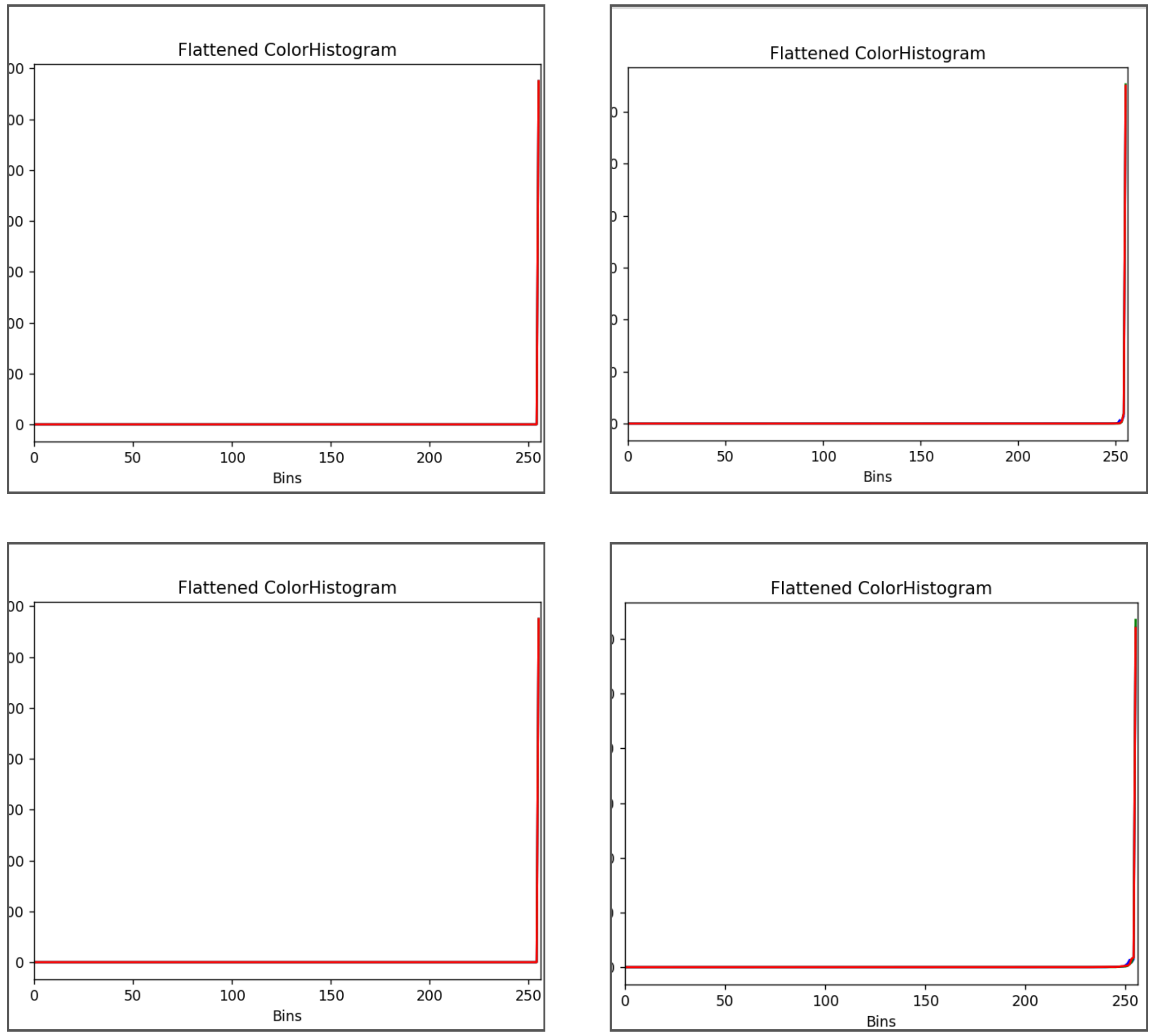

Flattened ColorHistogram
Flattened ColorHistogram
Flattened ColorHistogram
Flattened ColorHistogram
0
50
100
150
200
250
Bins



TESTBENCH 4

Quadrant 1 (Top-Left) = Price: 2.67
Quadrant 2 (Bottom-Left) = Price: 1.58
Quadrant 3 (Top-Right) = Price: 2.71
Quadrant 4 (Bottom-Right) = Price: 1.71

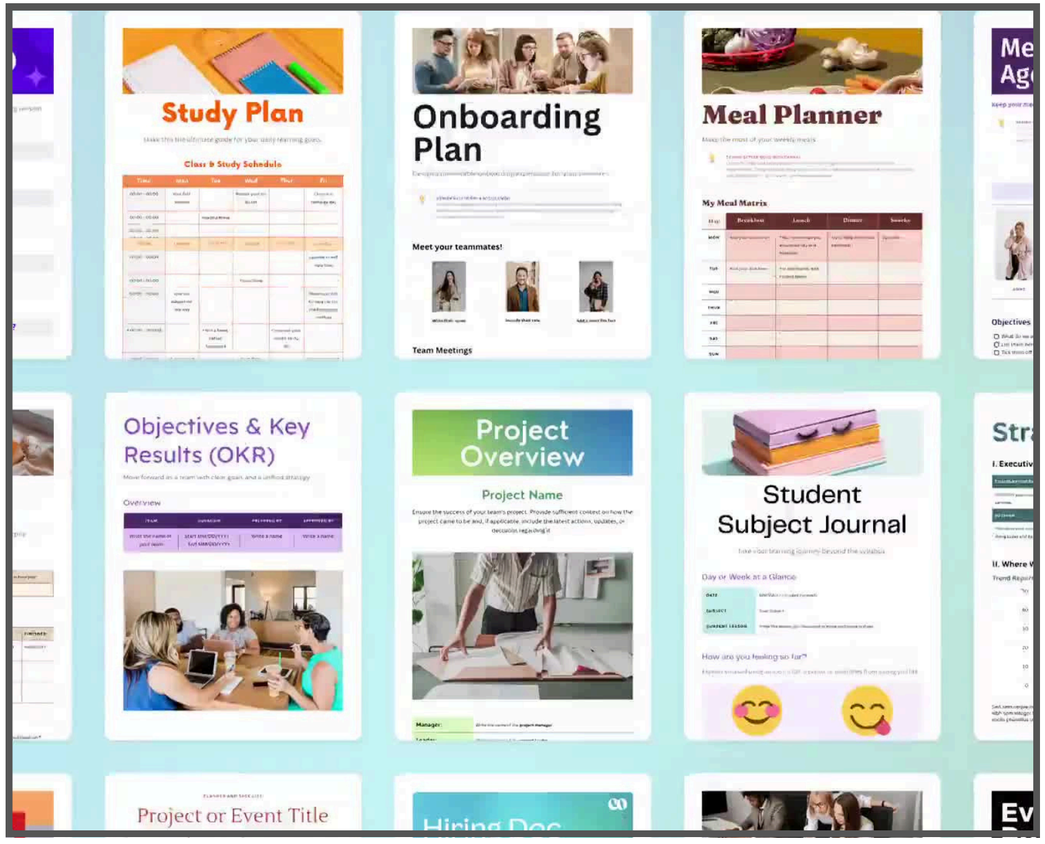
TOTAL PRICE for the page = 8.67 pesos

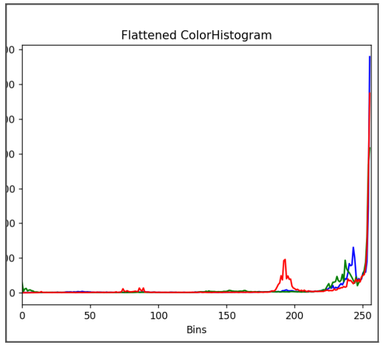

Flattened ColorHistogram
0
0 50 100 150 200 250
Bins
Flattened ColorHistogram
0 50 100 150 200 250
Bins
Flattened ColorHistogram
0
0 50 100 150 200 250
Bins
Flattened ColorHistogram
0 50 100 150 200 250
Bins

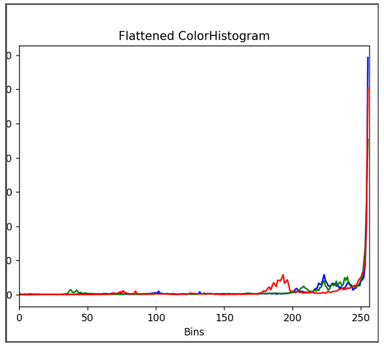
TESTBENCH 5

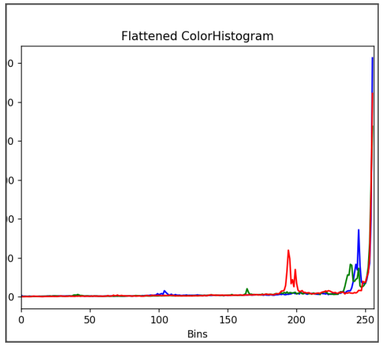
Quadrant 1 (Top-Left) = Price: 1.25
Quadrant 2 (Bottom-Left) = Price: 1.25
Quadrant 3 (Top-Right) = Price: 1.25
Quadrant 4 (Bottom-Right) = Price: 1.26

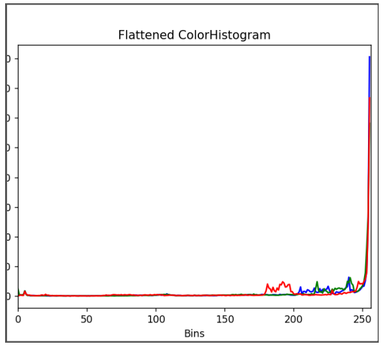
TOTAL PRICE for the page = 5.01 pesos

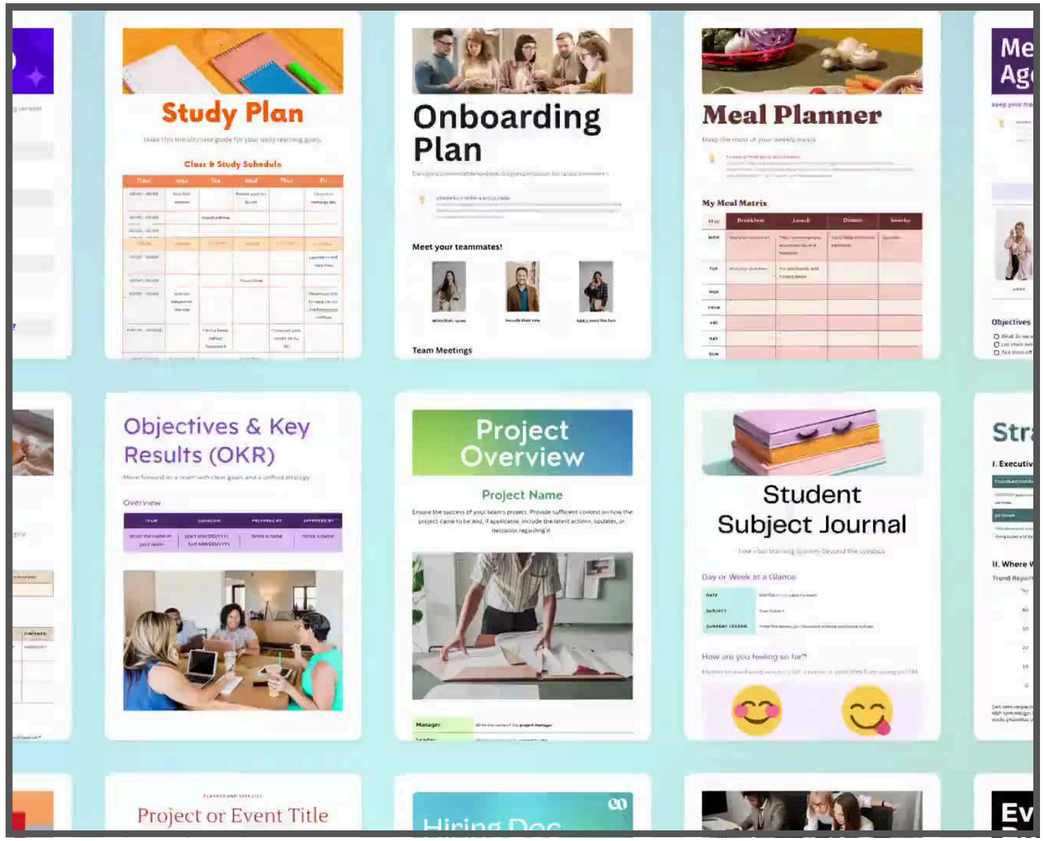

Study Plan
Class & Study Schedule
Onboarding Plan
Meet your teammates!
Team Meetings
Meal Planner
My Meal Matrix
Objectives & Key Results (OKR)
Overview
Project Overview
Project Name
Student Subject Journal
Day or Week at a Glance
How are you feeling so far?
Project or Event Title


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins


Flattened ColorHistogram
0
50
100
150
200
250
Bins



TESTBENCH 6

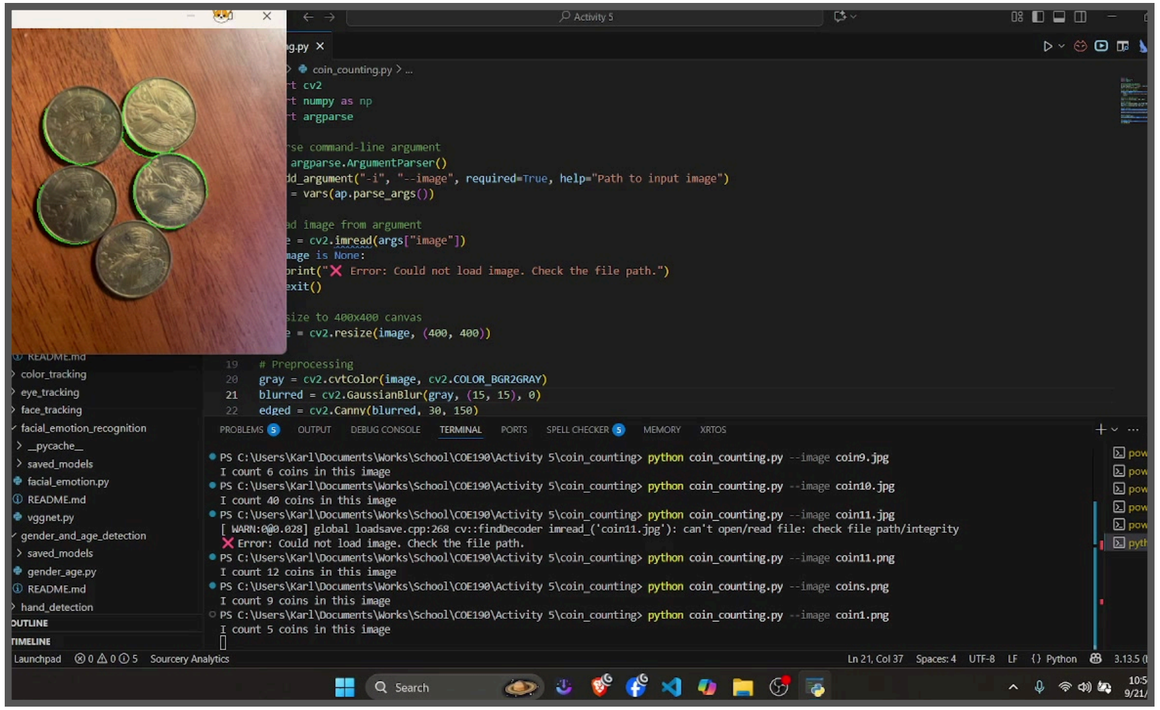
Quadrant 1 (Top-Left) = Price: 2.71
Quadrant 2 (Bottom-Left) = Price: 2.99
Quadrant 3 (Top-Right) = Price: 3.09
Quadrant 4 (Bottom-Right) = Price: 2.99

TOTAL PRICE for the page = 11.78 pesos

COIN COUNTING

COIN COUNTING

TESTBENCH




```
image = cv2.resize(image, (400, 400))

# Preprocessing
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (15, 15), 0)
edged = cv2.Canny(blurred, 30, 150)
```

5 coins

21 coins

10 Coins

10 coins

7 coins

6 Coins

10 coins

40 Coins

12 coins

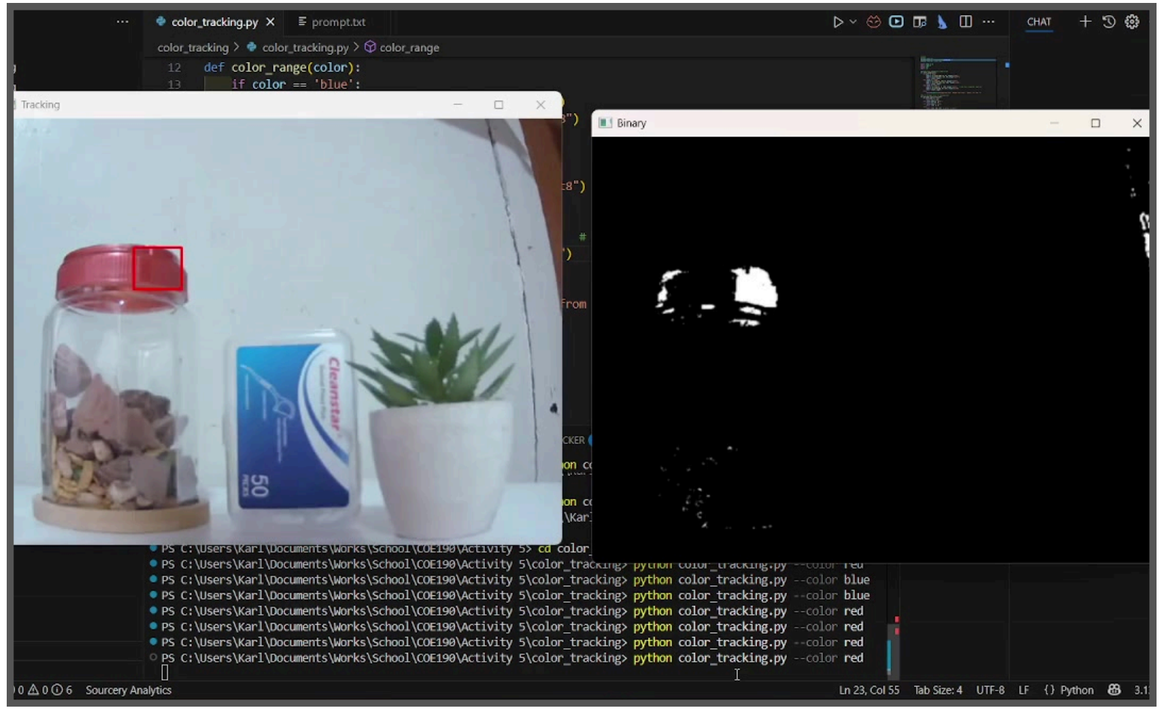
9 coins

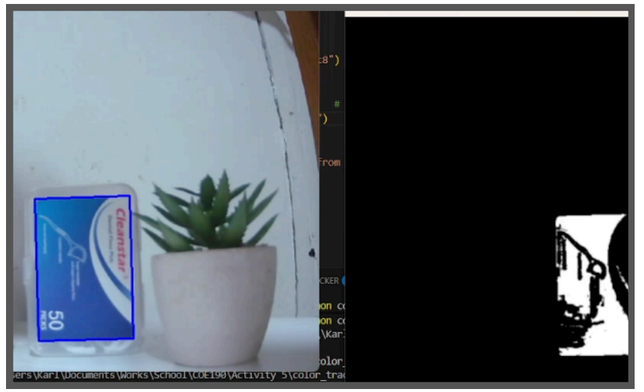
COLOR TRACKING

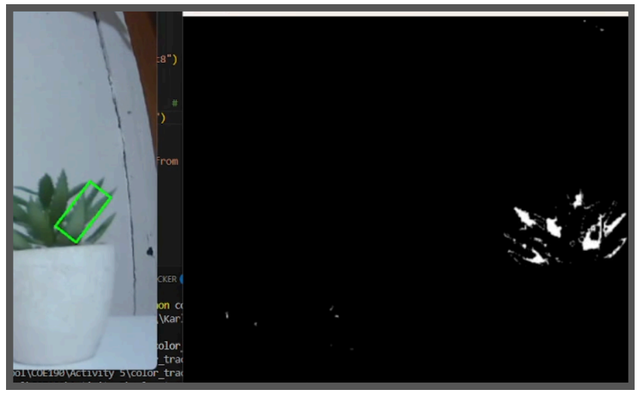

```
def color_range(color):
  if color == 'blue':
    lower = np.array([100, 67, 0], dtype="uint8")
    upper = np.array([255, 128, 50], dtype="uint8")
    return (lower, upper)
  elif color == 'green':
    lower = np.array([0, 100, 0], dtype="uint8")
    upper = np.array([100, 255, 100], dtype="uint8")
    return (lower, upper)
  elif color == 'red':
    lower = np.array([0, 0, 100], dtype="uint8") # low blue,
low green, high red
    upper = np.array([80, 80, 255], dtype="uint8")
    return (lower, upper)
  else:
    raise ValueError("Unsupported color. Choose from
'blue', 'green', or 'red'.")
```

COLOR TRACKING

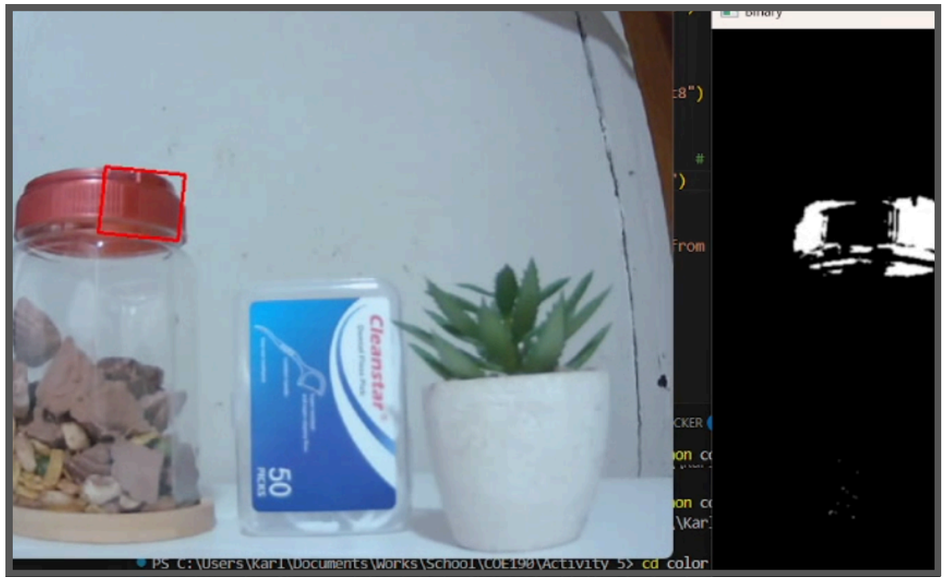
TESTBENCH

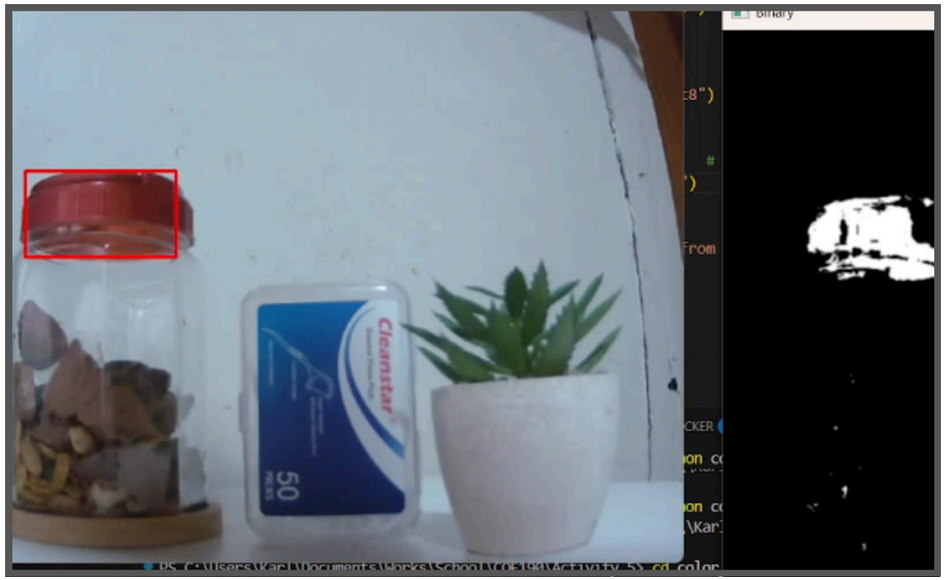






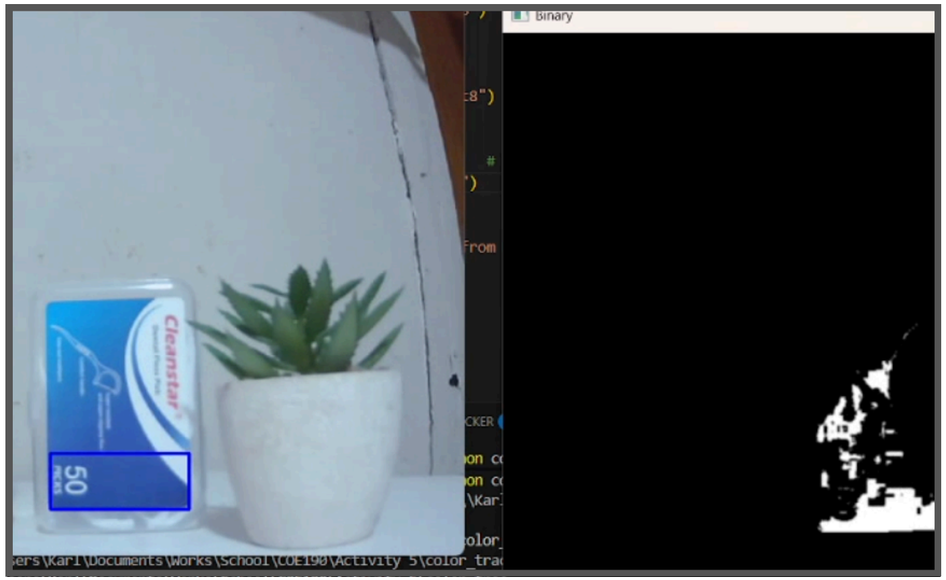
LIGHTING DIFFERENCE *(red)*

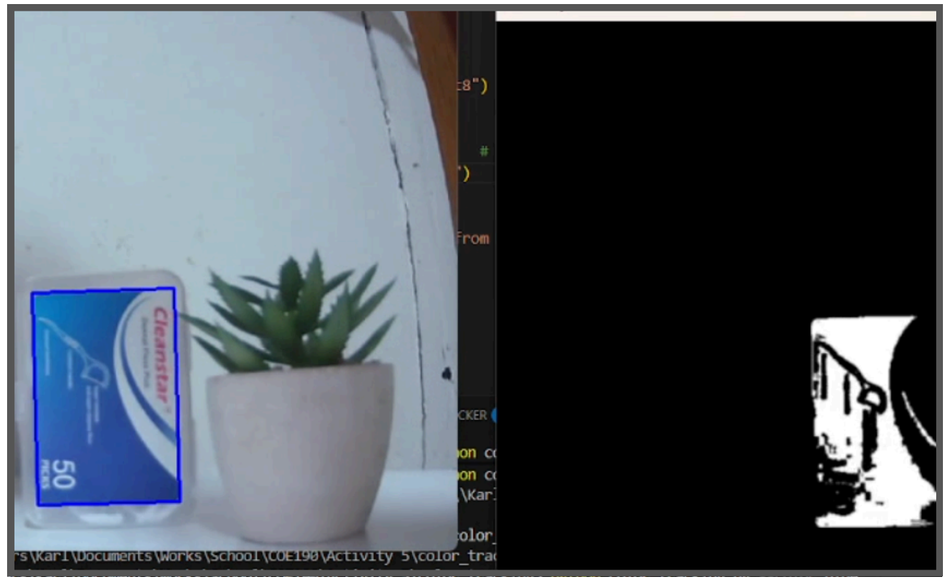




BRIGHT ⟶ **LOW LIGHT**

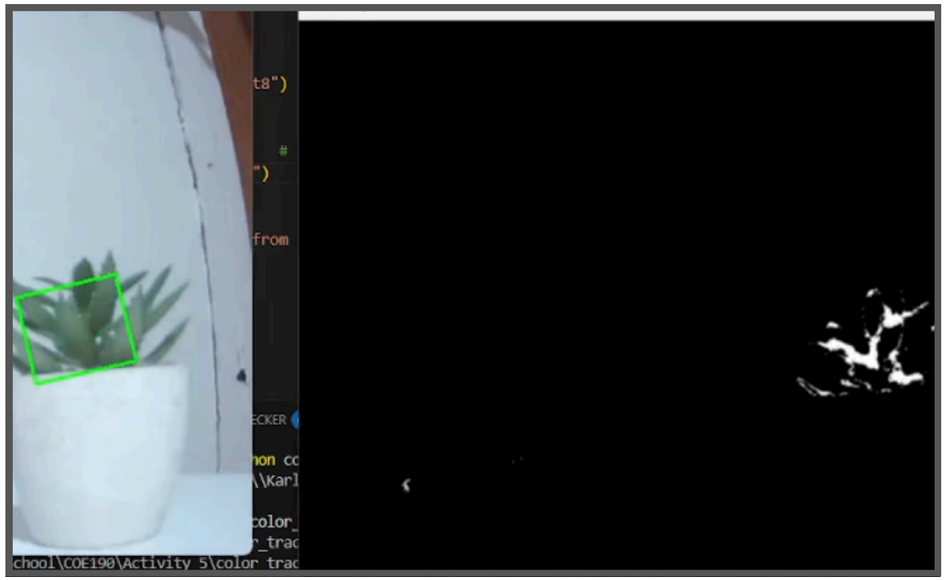
LIGHTING DIFFERENCE *(blue)*

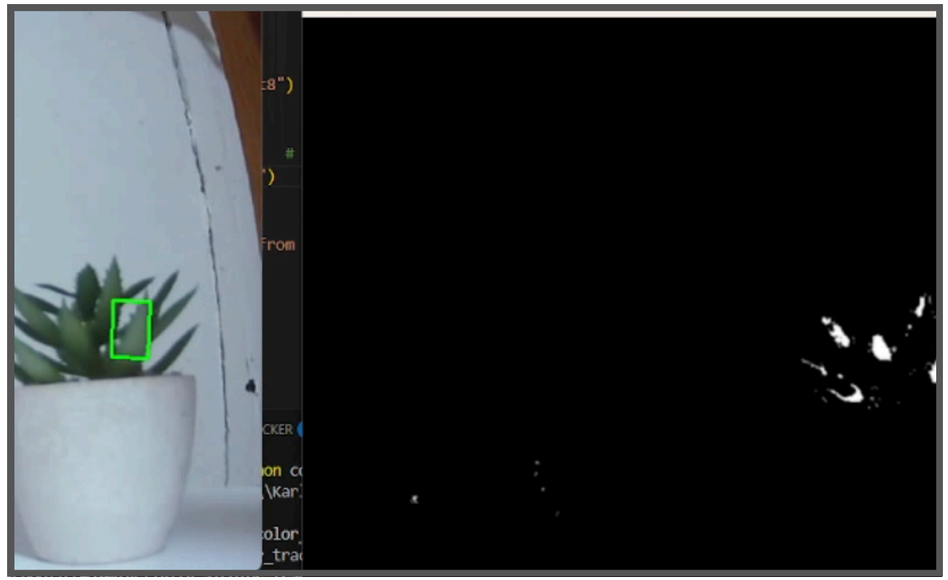




BRIGHT ⟶ **LOW LIGHT**

LIGHTING DIFFERENCE *(green)*



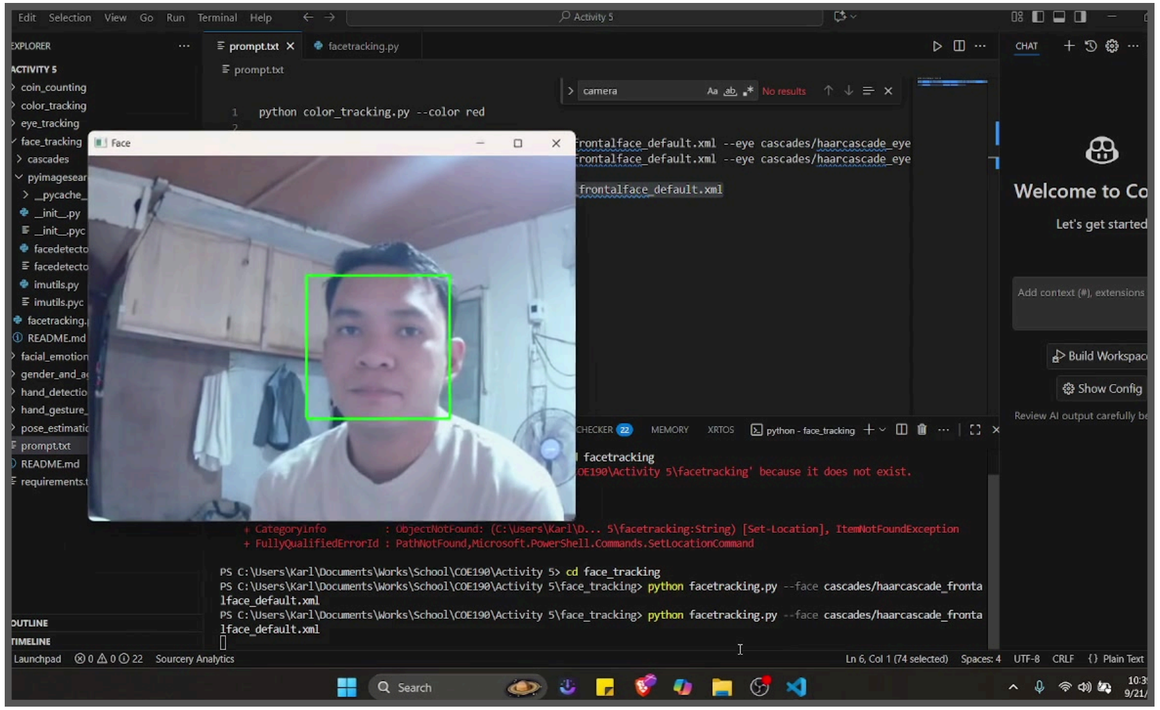


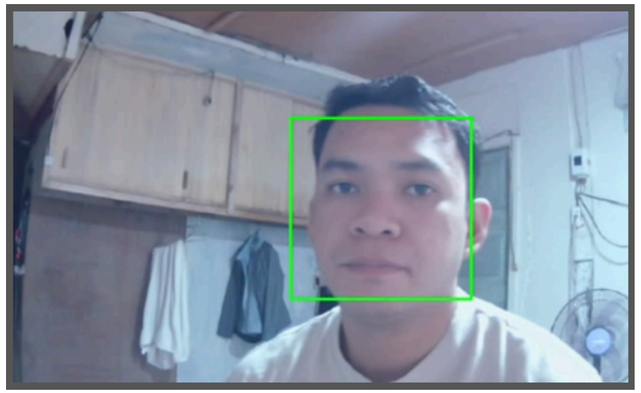
BRIGHT ⟶ **LOW LIGHT**

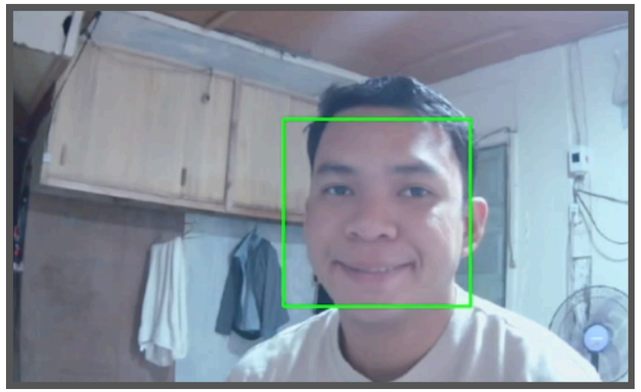
FACE TRACKING

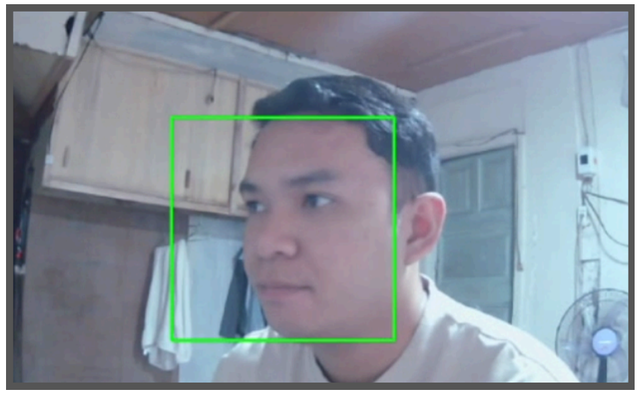
FACE TRACKING

TESTBENCH





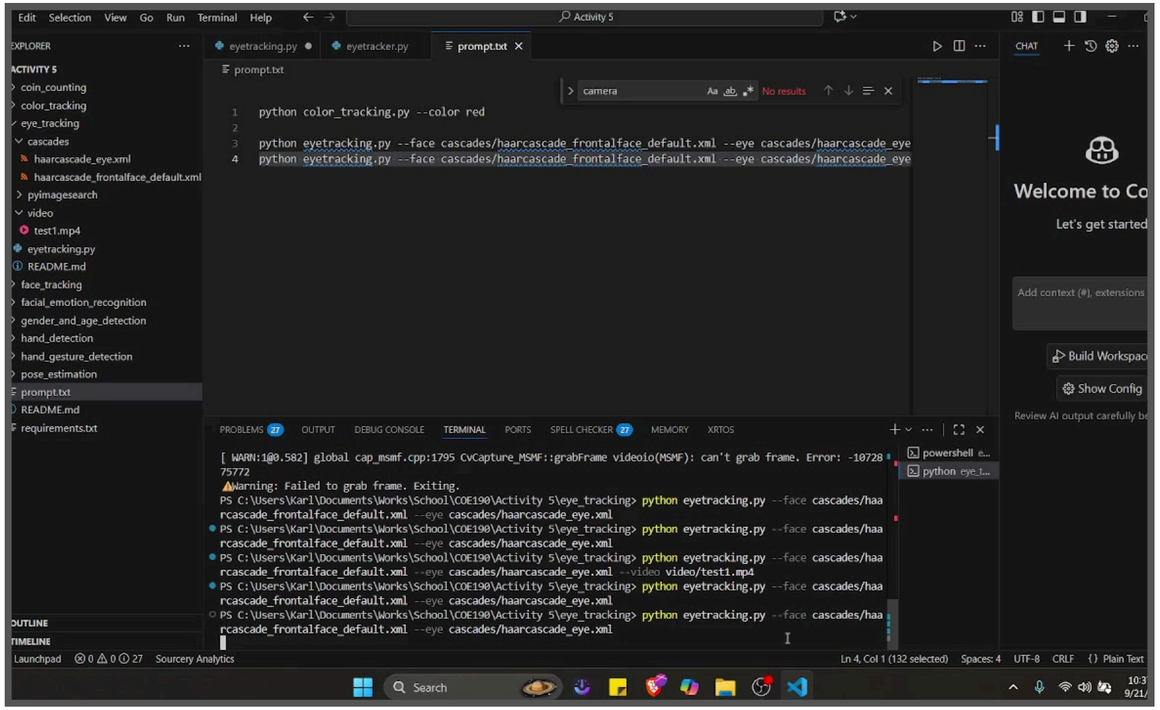


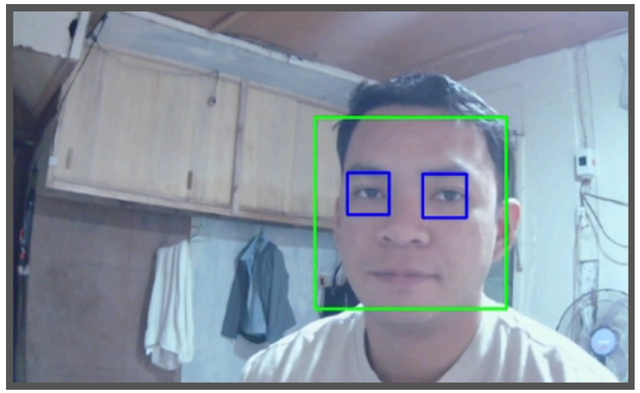


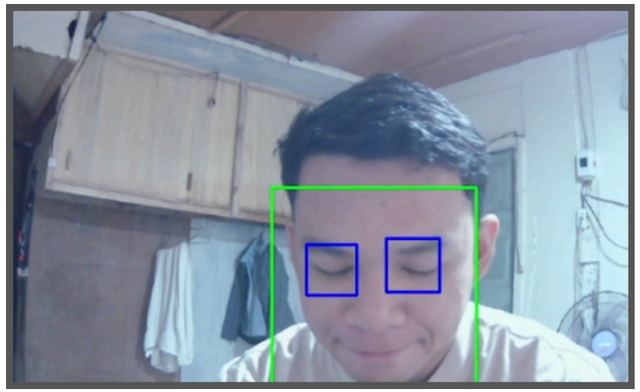
EYE TRACKING

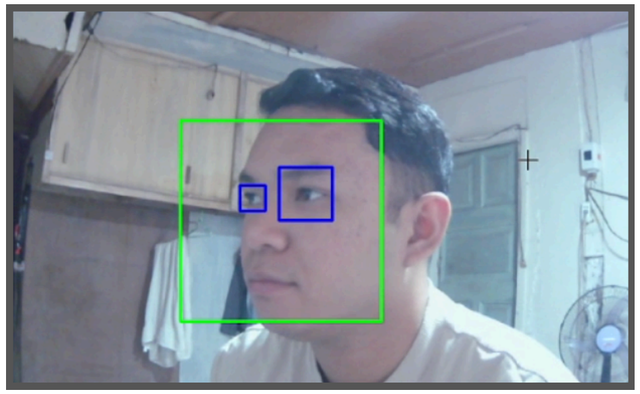
EYE TRACKING

TESTBENCH





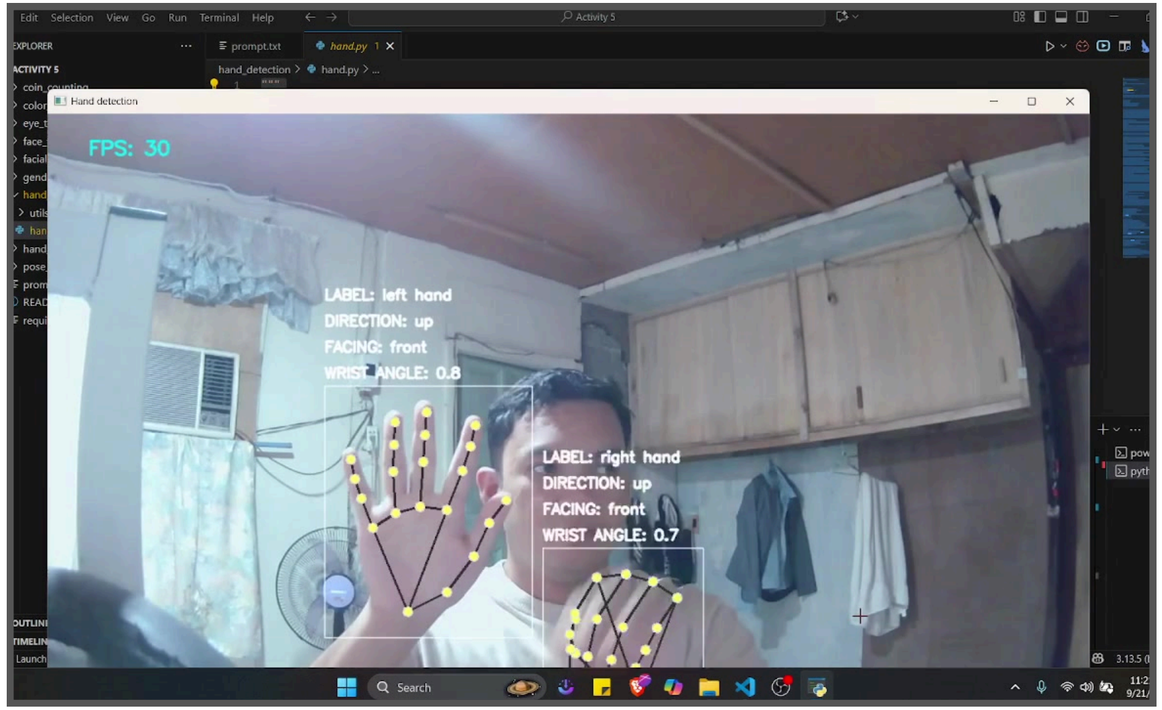


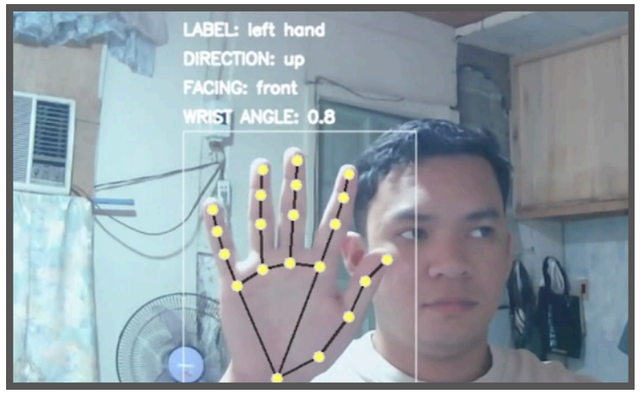


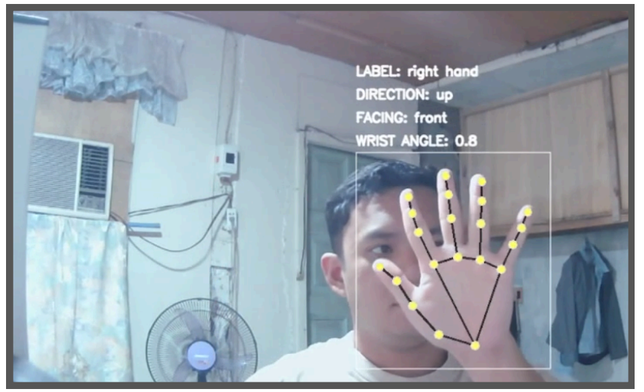
HAND DETECTION

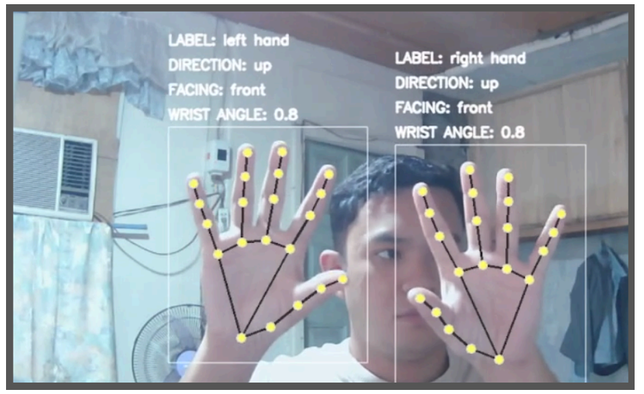
HAND DETECTION

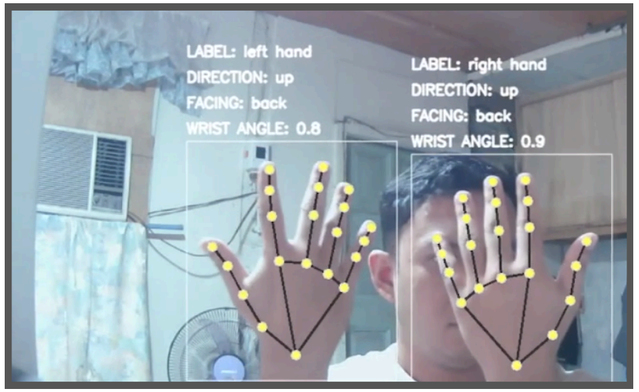
TESTBENCH

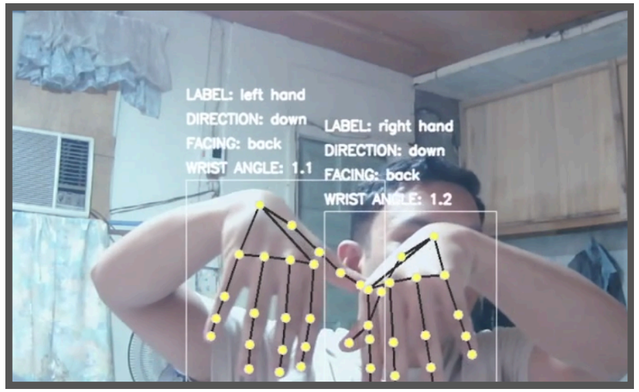


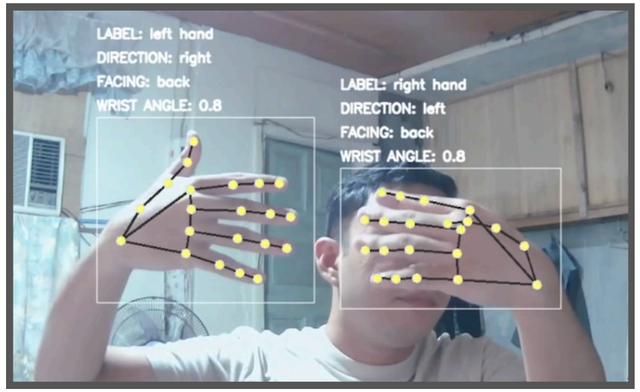


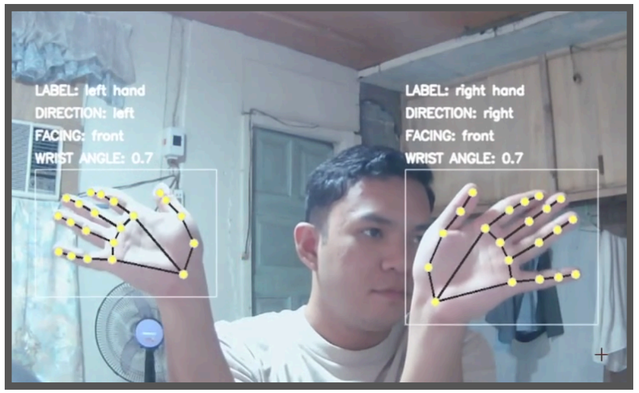


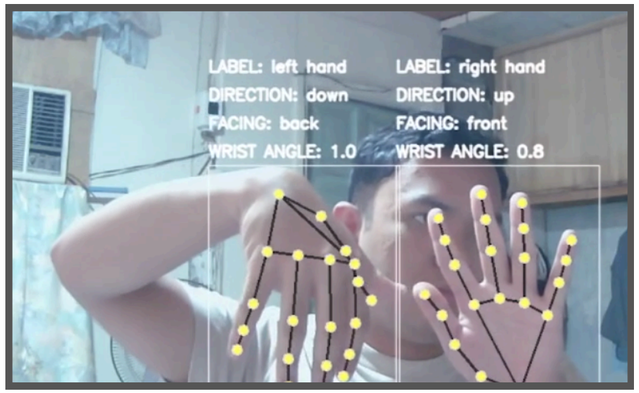


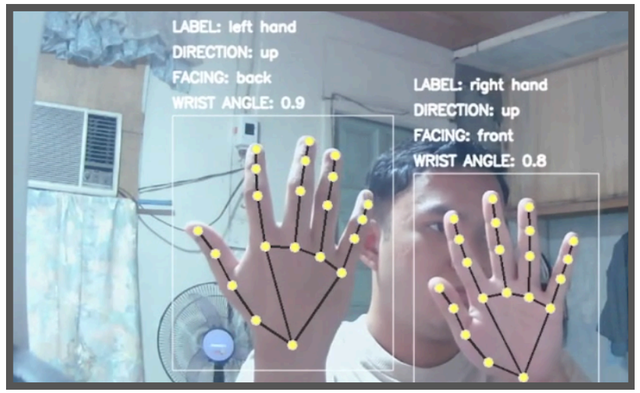

LABEL: left hand
DIRECTION: up
FACING: back
WRIST ANGLE: 0.8
LABEL: right hand
DIRECTION: up
FACING: back
WRIST ANGLE: 0.9


LABEL: left hand
DIRECTION: down
FACING: back
WRIST ANGLE: 1.1
LABEL: right hand
DIRECTION: down
FACING: back
WRIST ANGLE: 1.2


LABEL: left hand
DIRECTION: right
FACING: back
WRIST ANGLE: 0.8
LABEL: right hand
DIRECTION: left
FACING: back
WRIST ANGLE: 0.8


LABEL: left hand
DIRECTION: left
FACING: front
WRIST ANGLE: 0.7
LABEL: right hand
DIRECTION: right
FACING: front
WRIST ANGLE: 0.7

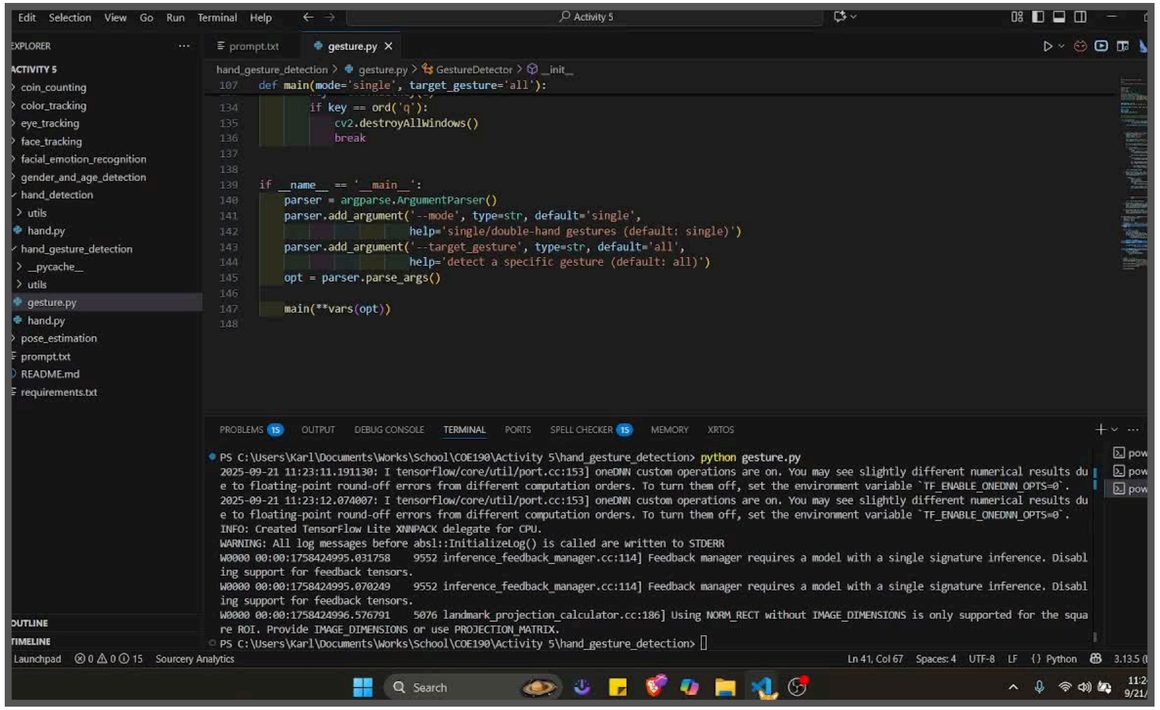

LABEL: left hand
DIRECTION: down
FACING: back
WRIST ANGLE: 1.0
LABEL: right hand
DIRECTION: up
FACING: front
WRIST ANGLE: 0.8

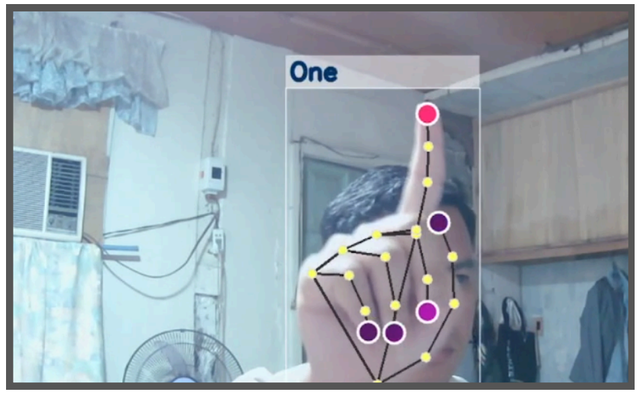

LABEL: left hand
DIRECTION: up
FACING: back
WRIST ANGLE: 0.9
LABEL: right hand
DIRECTION: up
FACING: front
WRIST ANGLE: 0.8

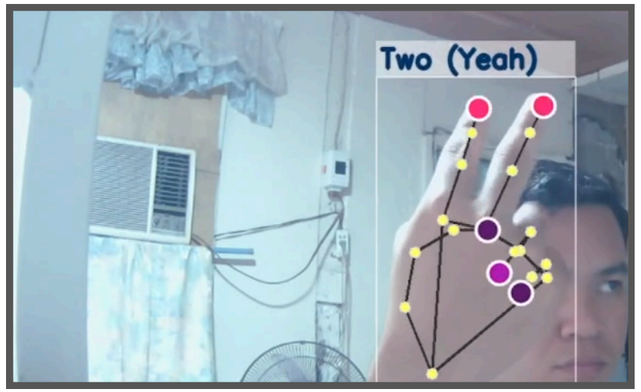
HAND GESTURE

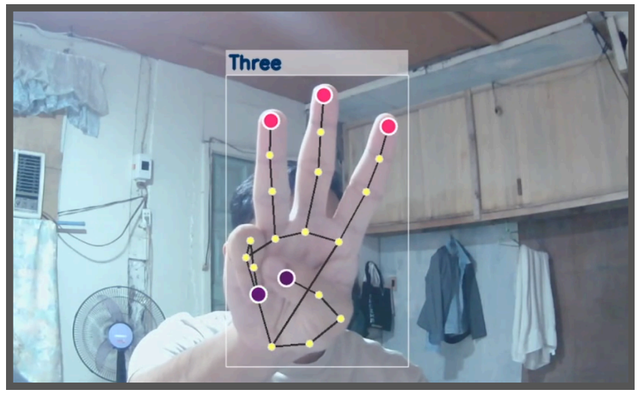
HAND GESTURE

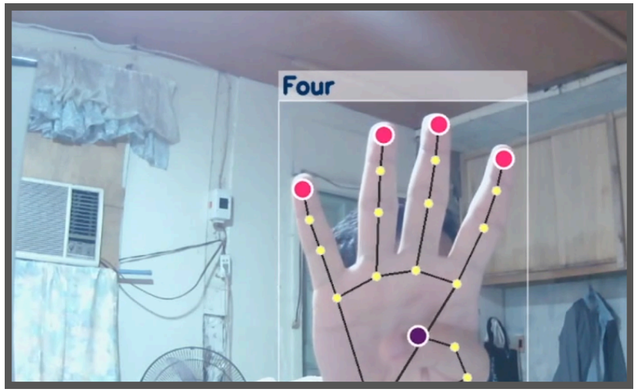
TESTBENCH

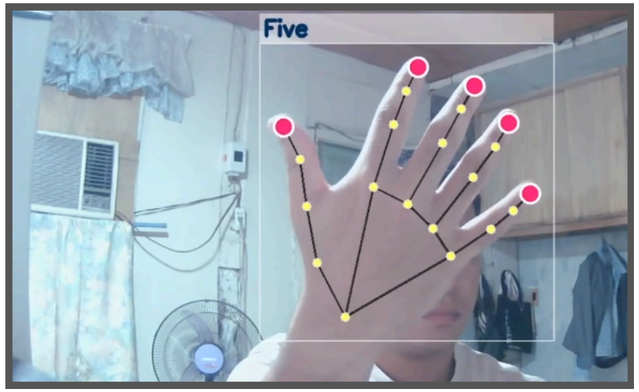


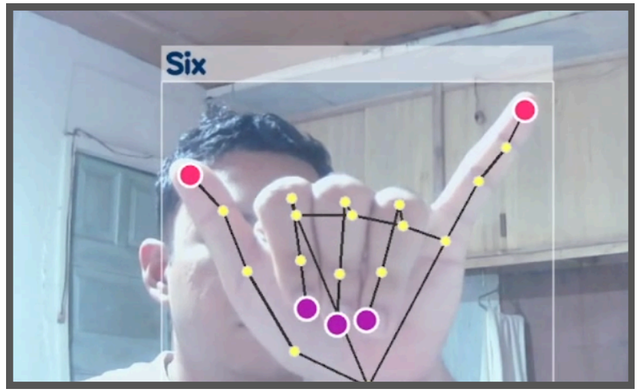


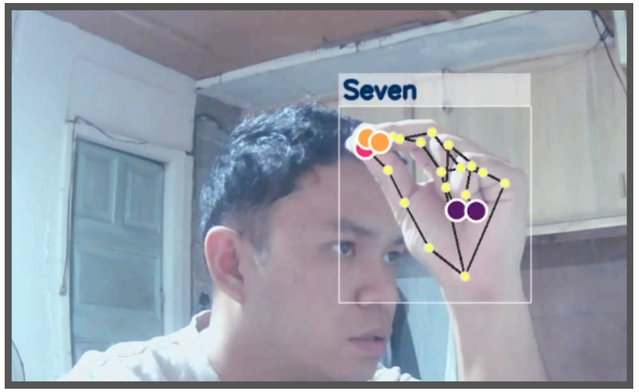


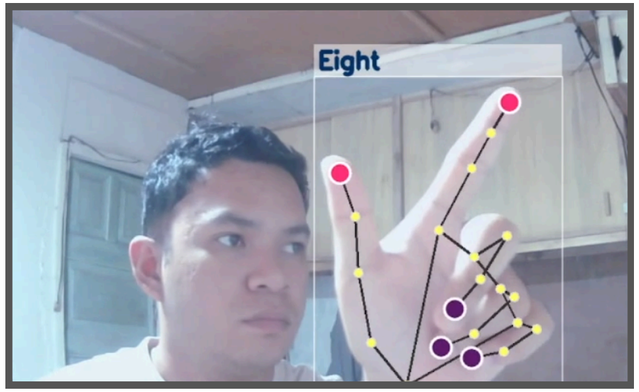


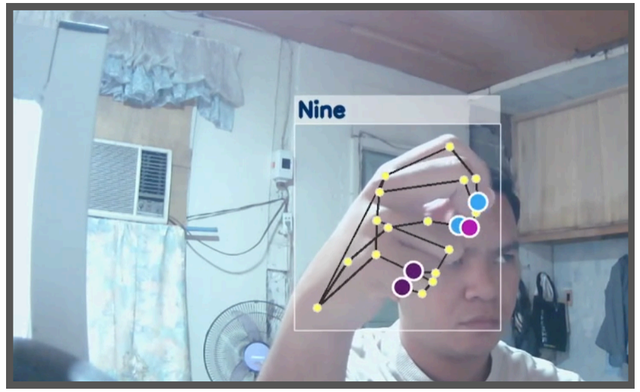

Four

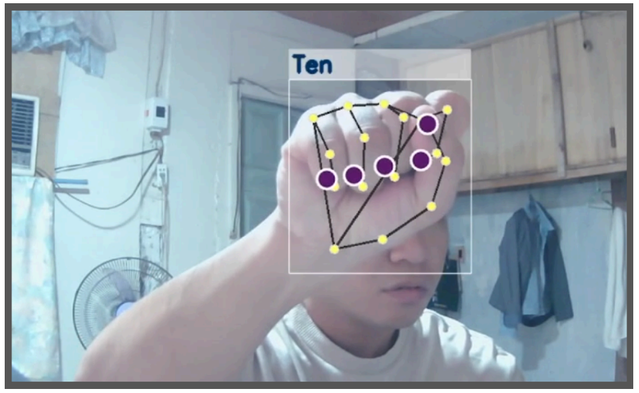

Five

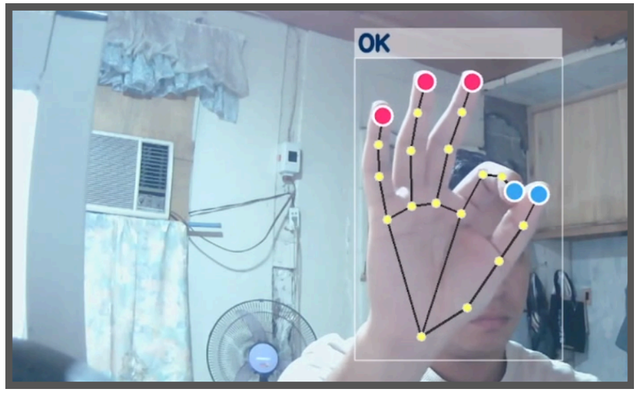

Six

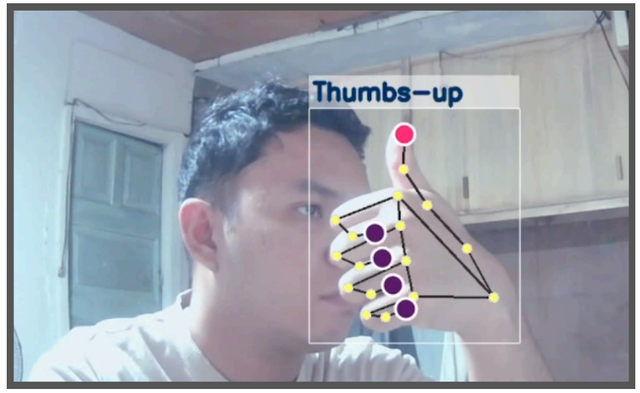

Seven

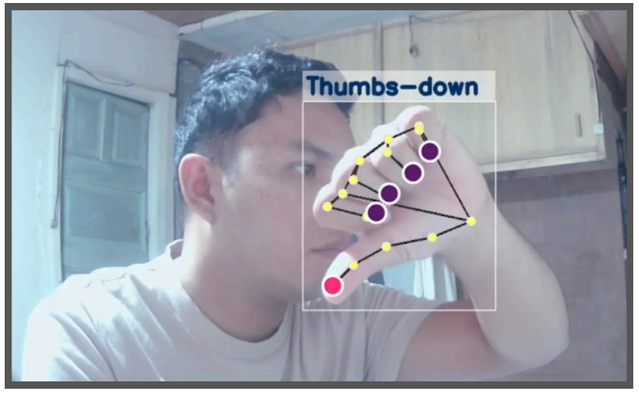

Eight

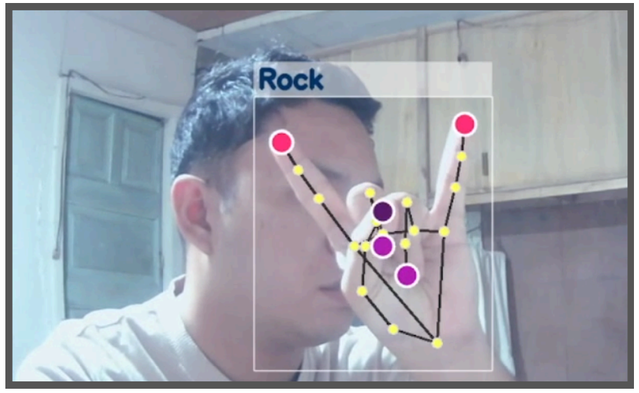

Nine

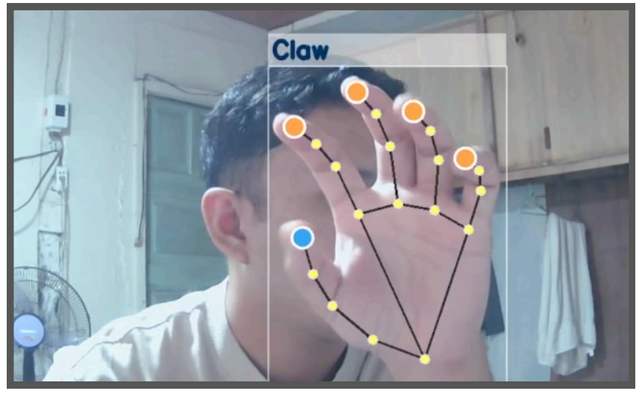

Ten

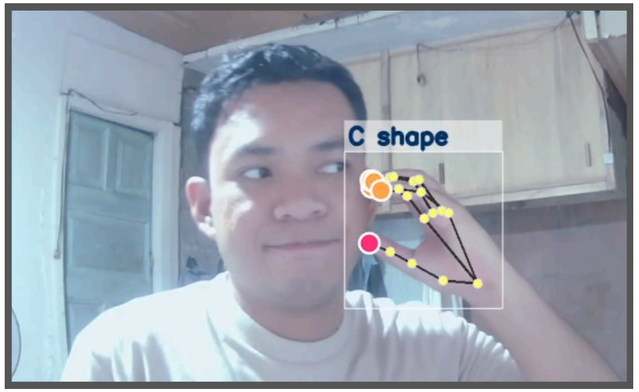

OK

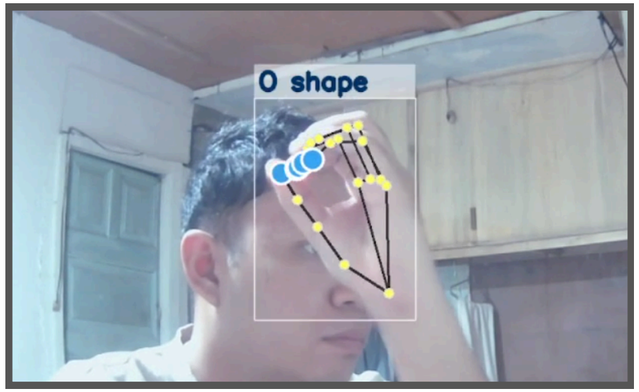

Thumbs-up

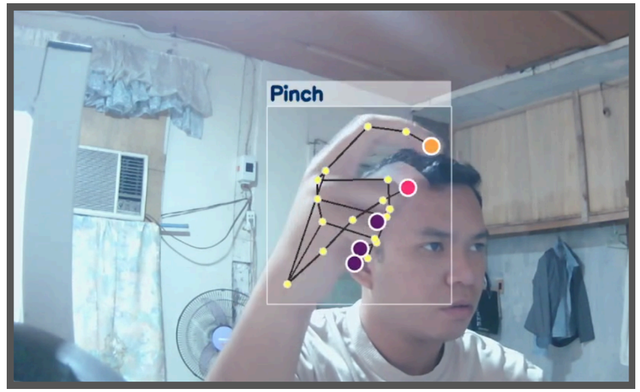

Thumbs-down

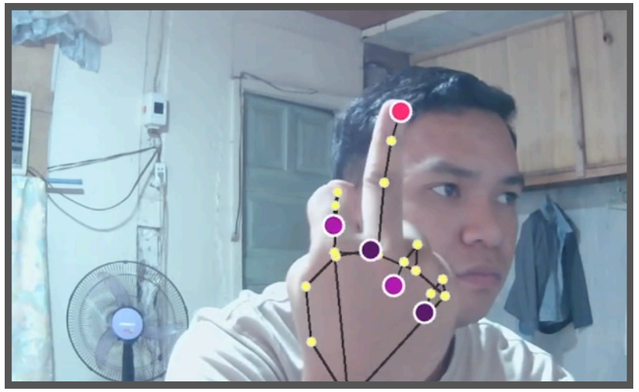

Rock


Claw


C shape


O shape

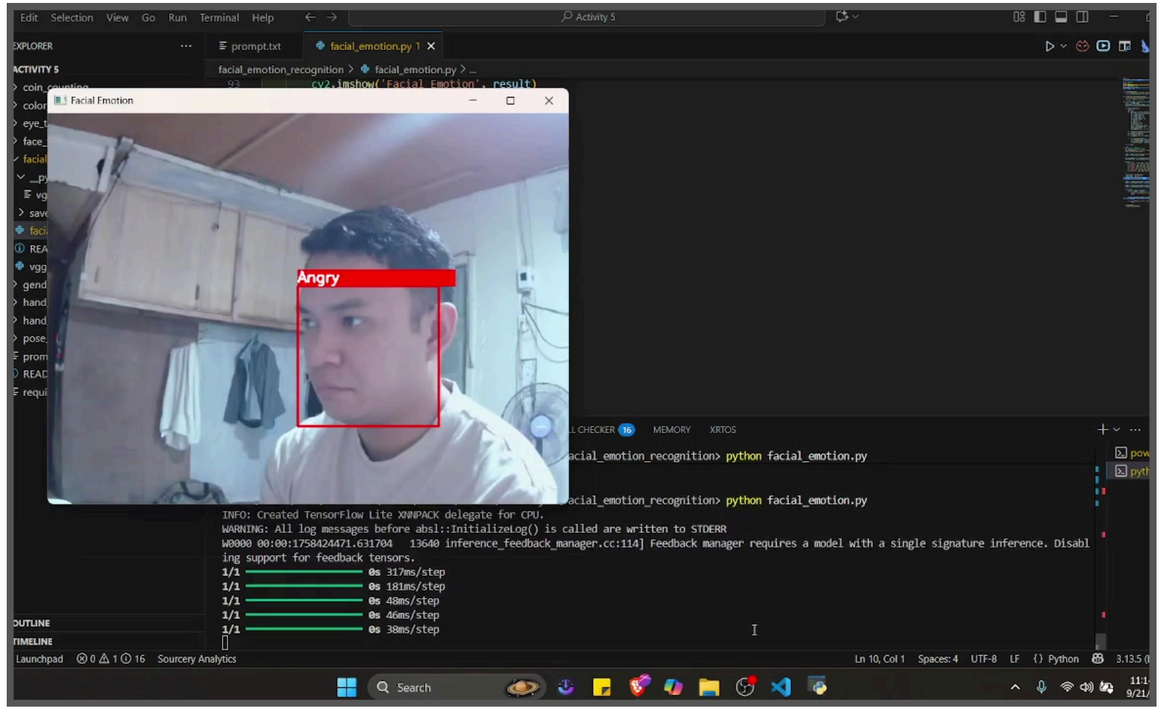

Pinch

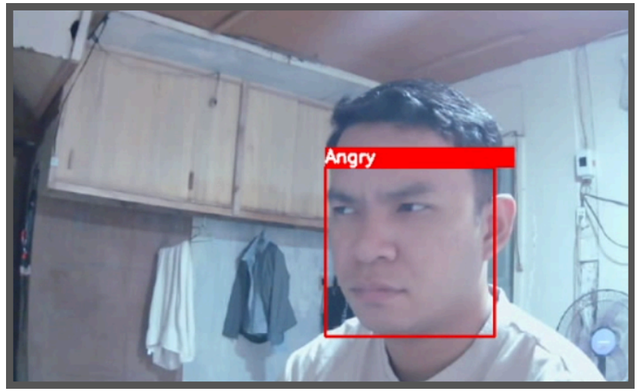


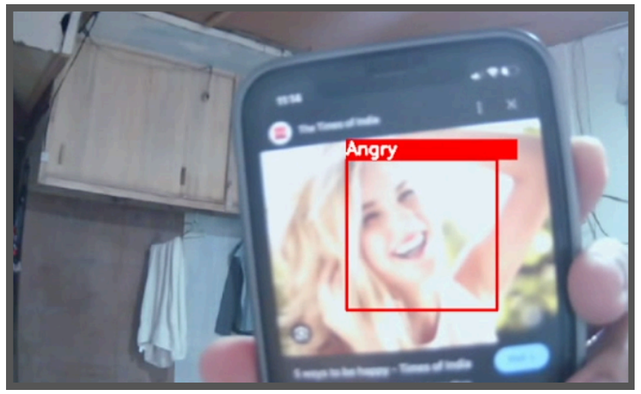
FACIAL EMOTION

FACIAL EMOTION

TESTBENCH



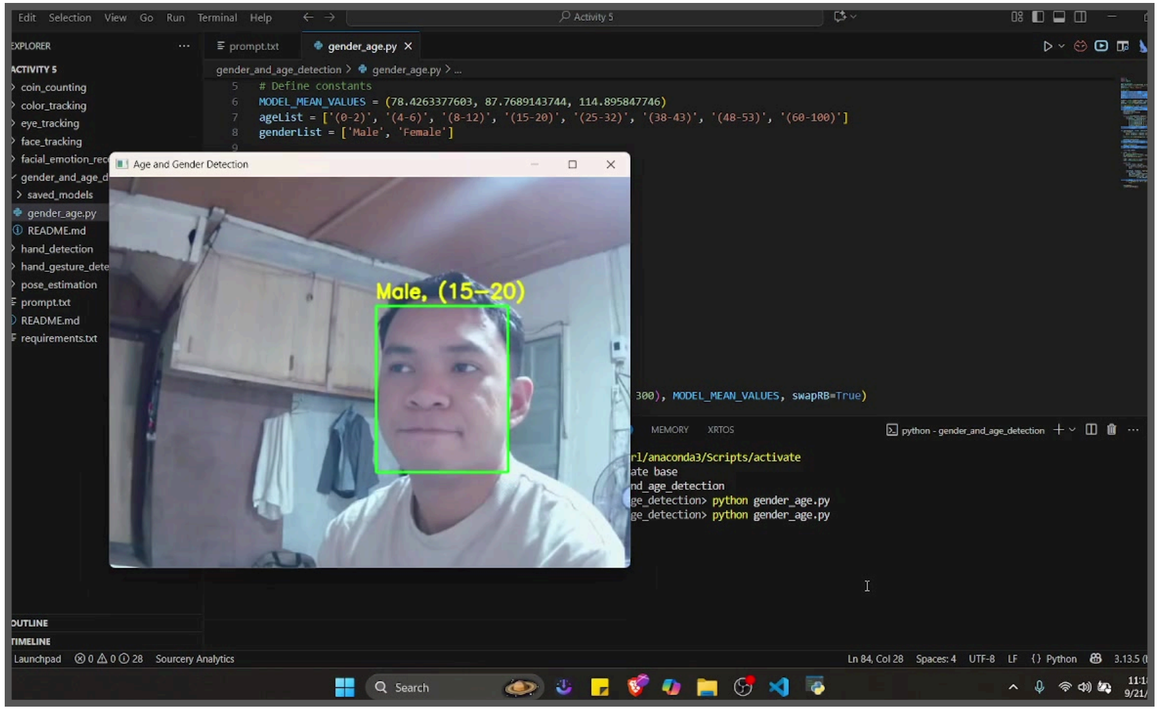


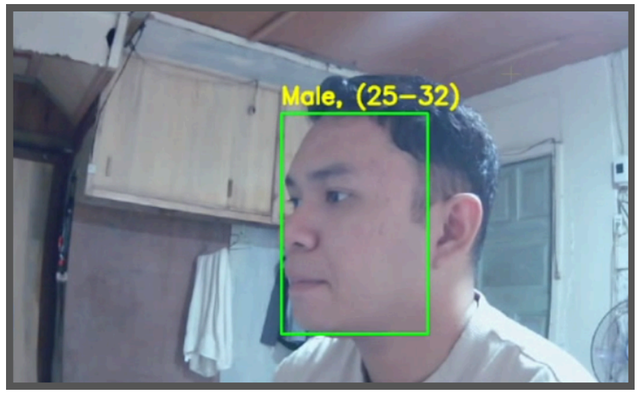


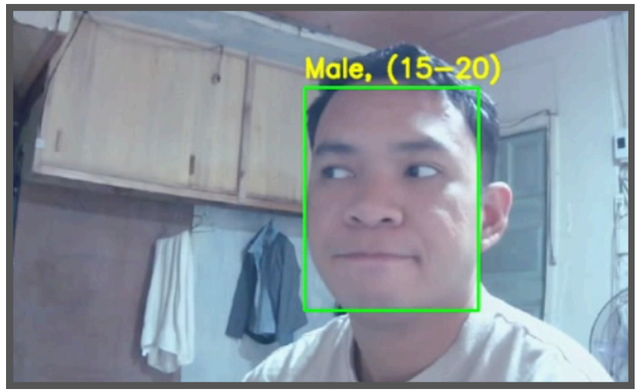
GENDER AND AGE

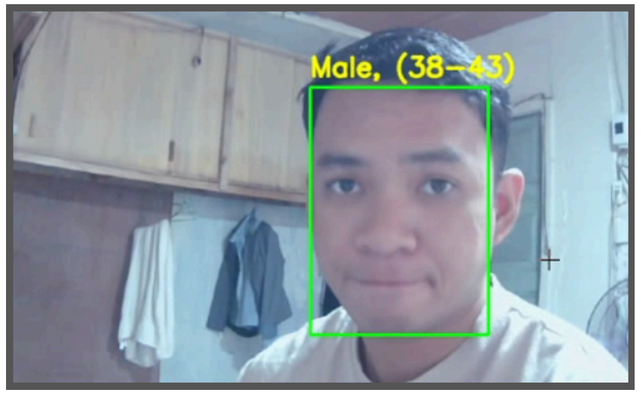
GENDER AND AGE

TESTBENCH









POSE ESTIMATION

POSE ESTIMATION

TESTBENCH