R&D PROJECT

Hemoglobin Level Estimation from Photographic images

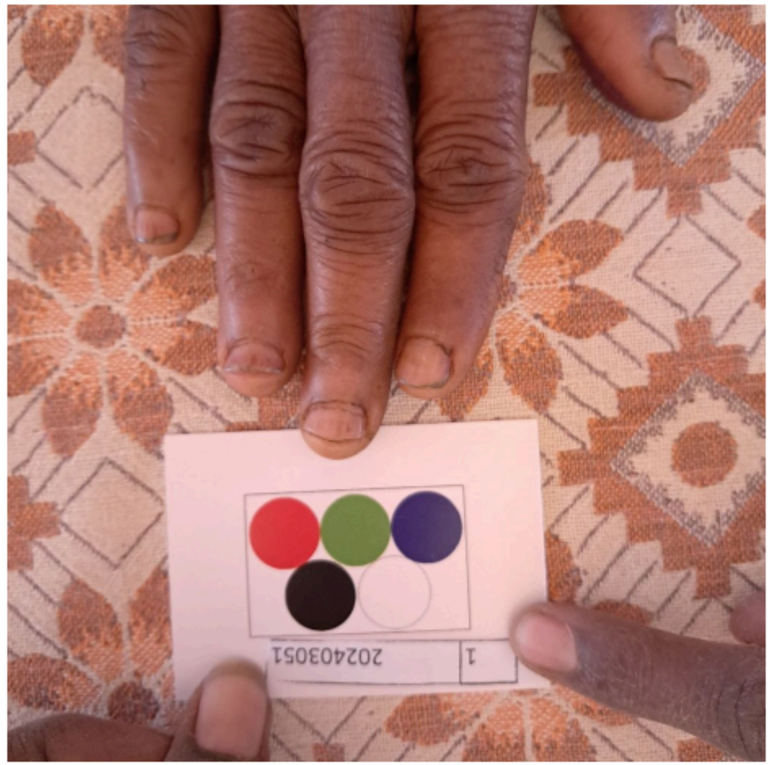
NIRANJAN VERMA
210020085

PROF. NIRMAL PUNJABI

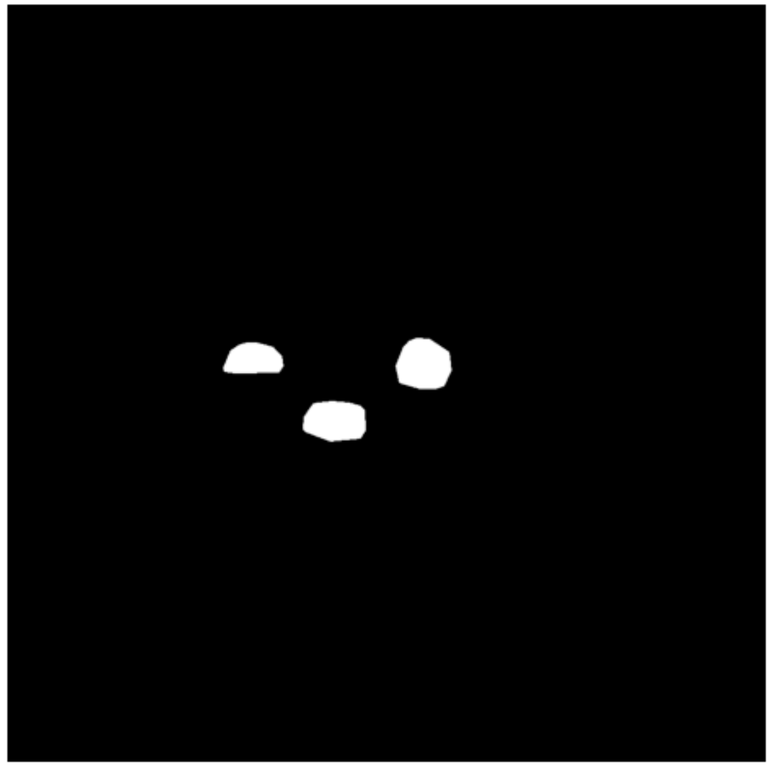
DH 307

Segmented Fingernail using the given mask

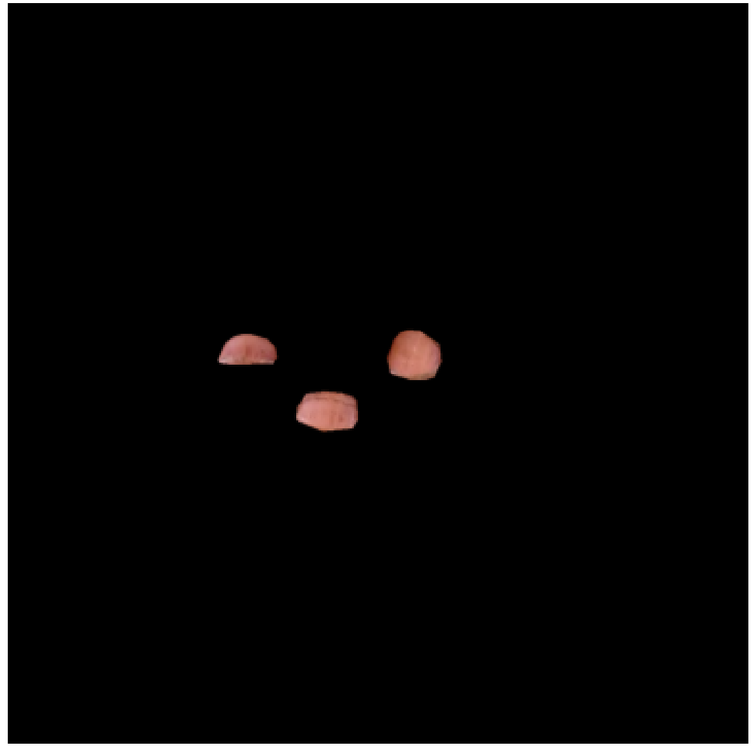
Original Image



Mask



Masked Image



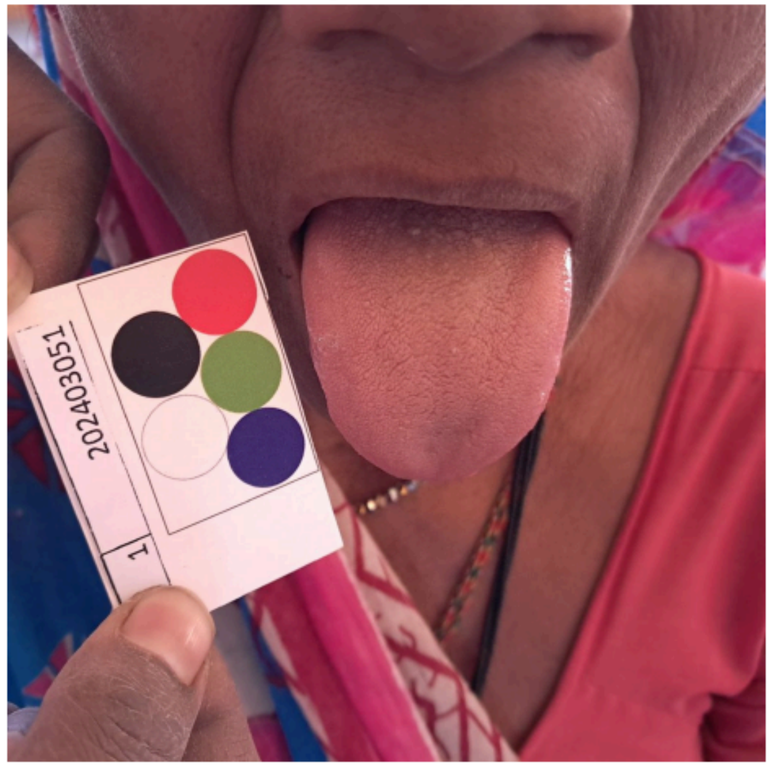
Segmented Fingernail using the given mask

Logic Used

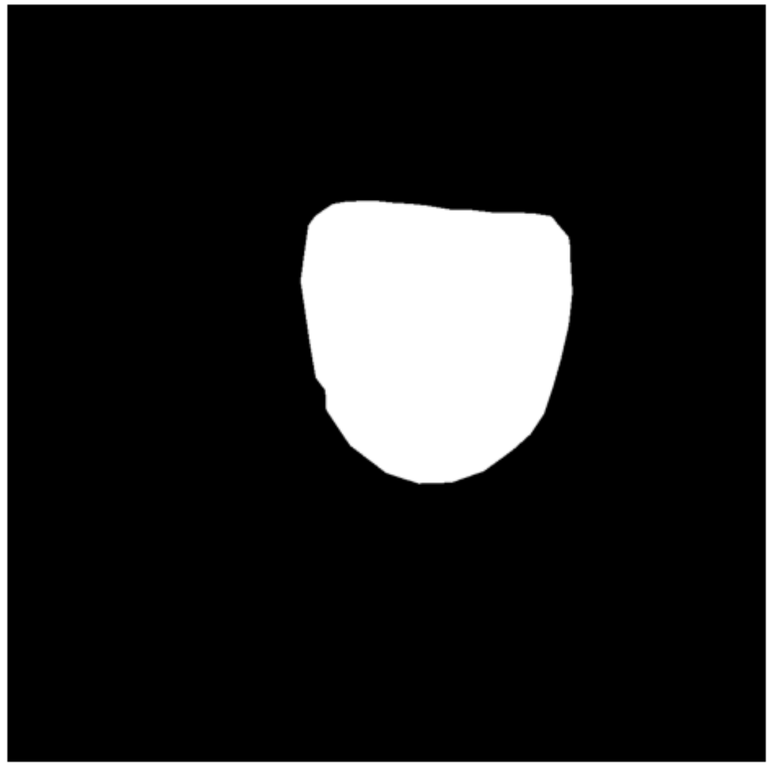
- **Image Loading**: Load the original image and mask; convert the original to RGB for visualization and the mask to grayscale for processing.
- **Binary Mask Creation**: Threshold the mask (values < 125 as True) to identify regions for modification.
- **Channel Splitting**: Split the original image into Blue, Green, and Red channels for individual manipulation.
- **Mask Application**: Set pixel values to 0 (black) in the identified mask regions for all three channels.
- **Image Reconstruction and Visualization**: Merge the modified channels, convert to RGB, and display the original, mask, and masked image side by side.

Segmented tongue using the given mask

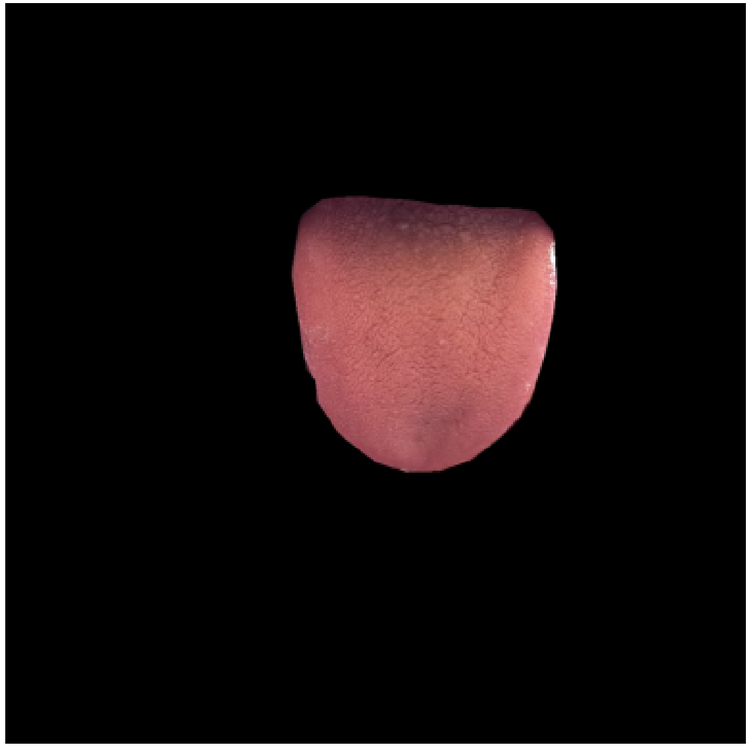
Original Image



Mask



Masked Image



Segmented tongue using the given mask

Logic Used

Same logic has been used here which is used for fingernail extraction from the image using the segmented mask provided.

Image has been loaded and binary mask is created by thresholding the mask (values < 75 as True).

Original image has been split into 3 RGB channel and binary mask is applied on each individual channel.

The spilt image has been merged to generate the output.

Color Palette Detection

- Contour Detection
- Detection using a defined template
- Object Detection using YOLO

Contour Detection

Logic Used

- **Convert to Grayscale and apply Gaussian blur :** Converted the image to grayscale for edge detection and applied gaussian blur to reduce noise
- **Edge Detection:** Used Canny edge detector to detect edges in blurred image
- **Contour Detection:** Identifies External Contour from the detected edges
- **Identify the Largest Contour:** Loops through all the contour to find the contour with maximum area which is expected to be the color palette region
- **Extract Palette Region:** Crop the palette from the extracted contour

Contour Detection

Original Image

- Loaded the original image using cv2

Next Step

- Convert the original image into gray scale

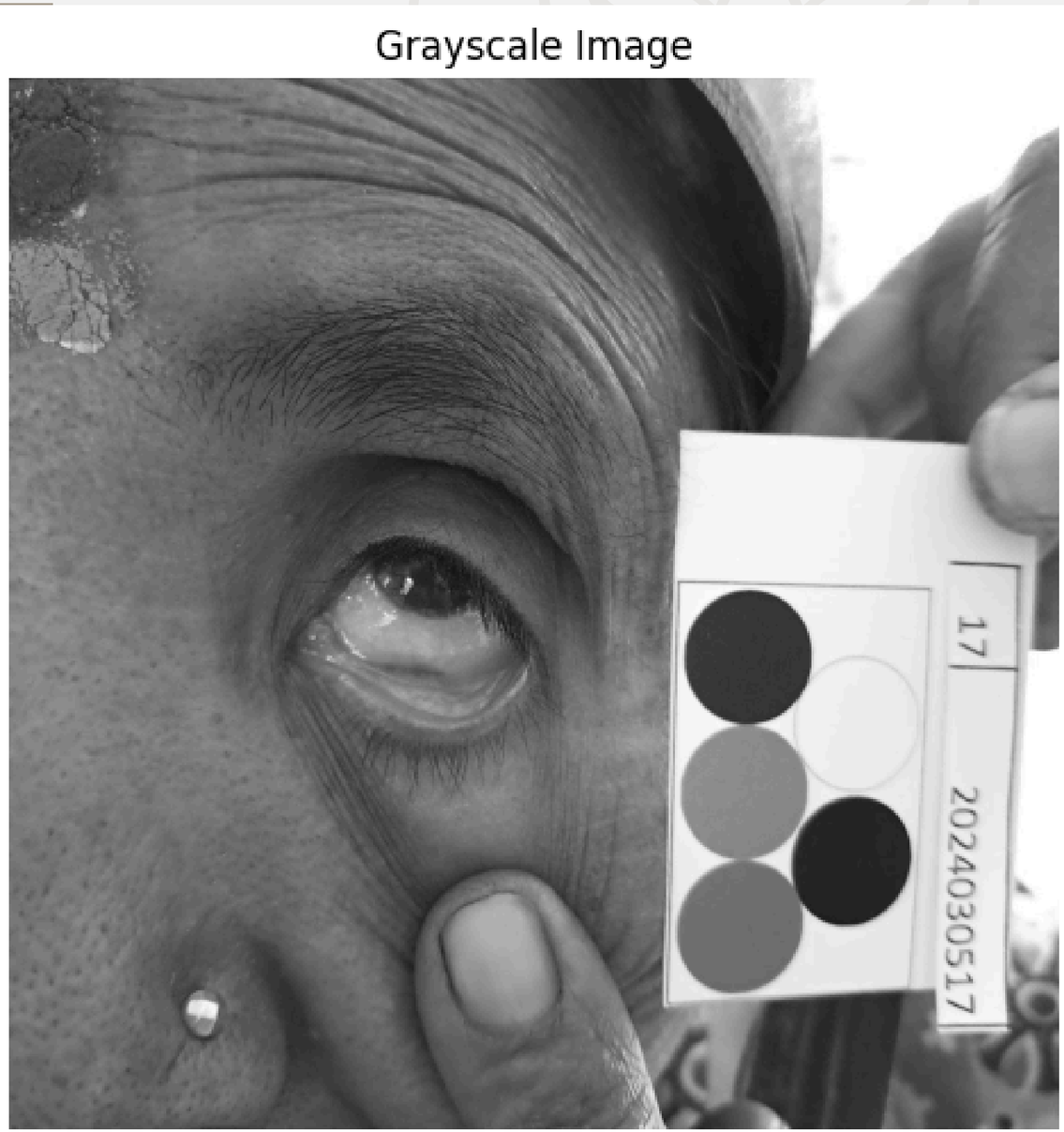
Contour Detection

GrayScale Image

- Converted the image into gray scale for edge detection using cv2.cvtColor

Next Step

- Apply gaussian blur



Contour Detection

Gaussian Blur

- Applied Gaussian blur of 5x5 to reduce noise

Next Step

- Detect the edges

Blurred Image

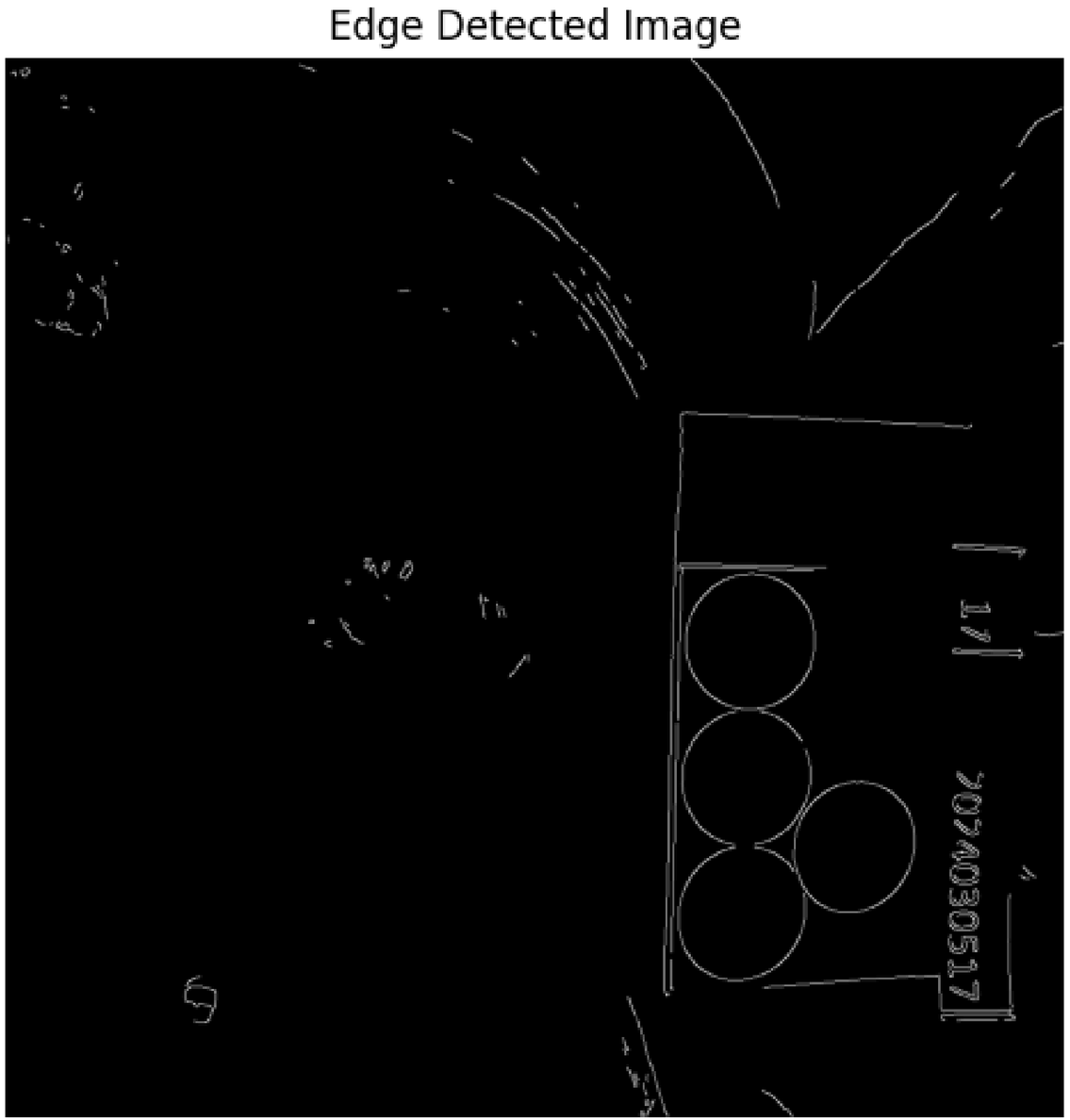
Contour Detection

Edge Detection

- Detected the edges using Canny Edge detection
- cv2.Canny(blurred_image, 200, 150)
- pixel gradient intensity above 200 is considered an edge & below 150 is ignored

Next Step

- Find Contours



Contour Detection

Contour Detection

- Function returns a list of all detected contours, where each contour is a list of points forming a boundary.

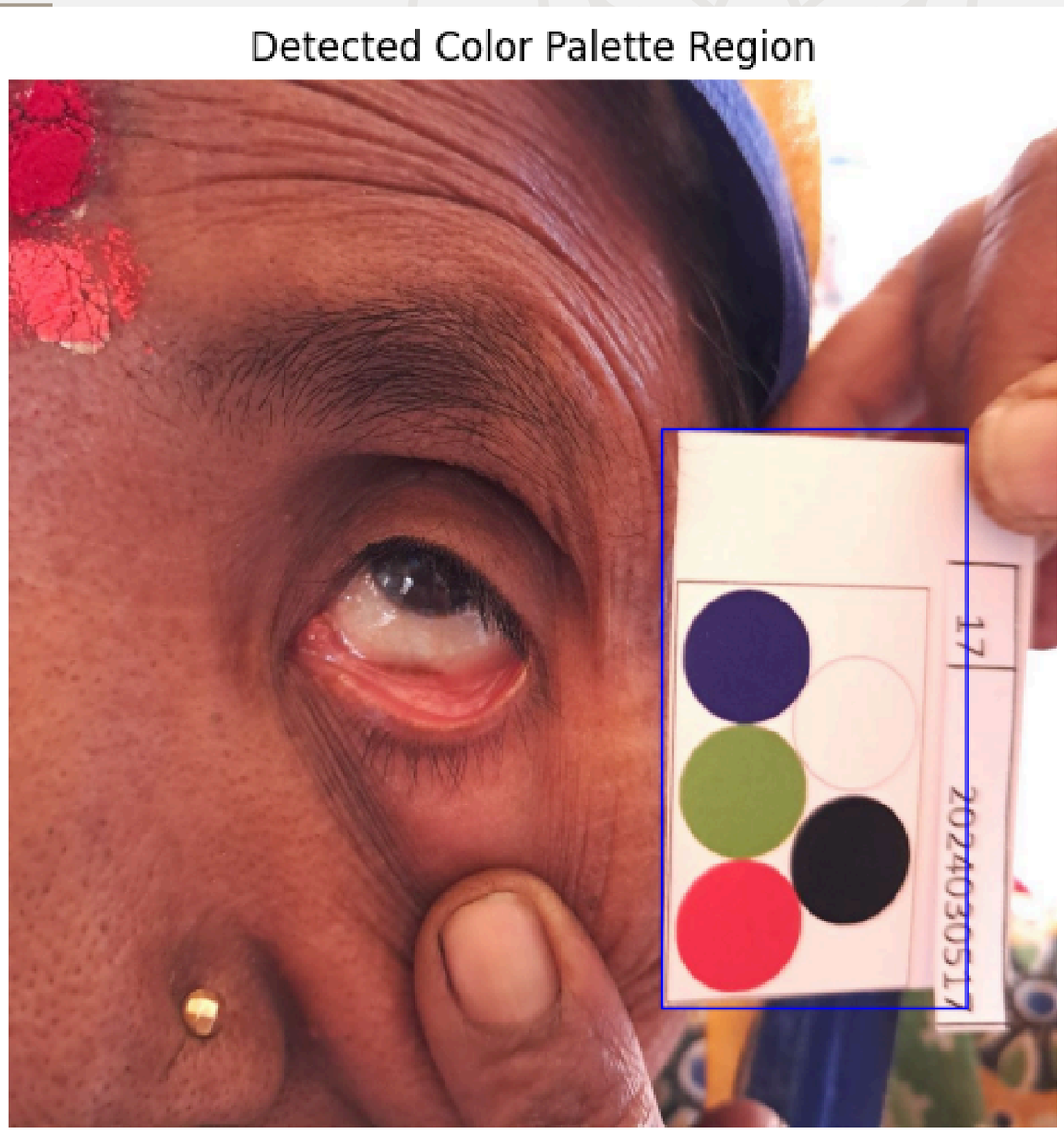
Next Step

- Find the largest bounding box

Contour Detection

Largest Contour Detection

- Used cv2.boundingRect
  - (x, y): The top-left corner of the bounding box.
  - (w, h): The width and height of the bounding box.
- Area of each bounding box is calculated as w*h and box with max area is selected



Contour Detection

Cropping the largest contour

- Cropped the largest contour using the x,y coordinates and height and width obtained

Contour Detection

1

Original Image

2

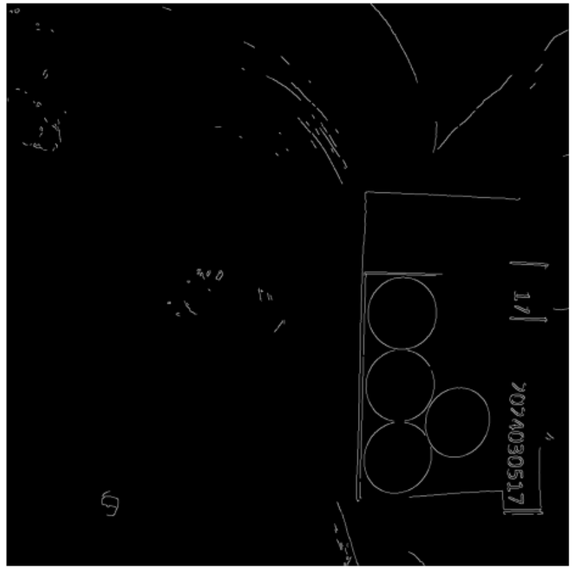
Grayscale Image

3

Blurred Image

4

Edge Detected Image



5

Contours Detected

6

Detected Color Palette Region

Contour Detection GOOD Results:

Original Image

Color Palette

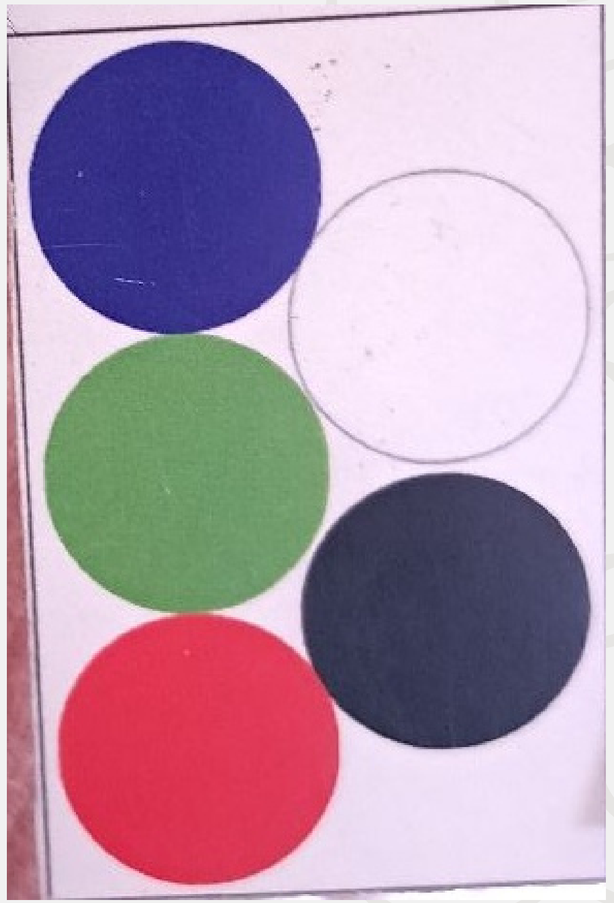
Contour Detection GOOD Results:

Original Image

Color Palette

Contour Detection GOOD Results:

Original Image

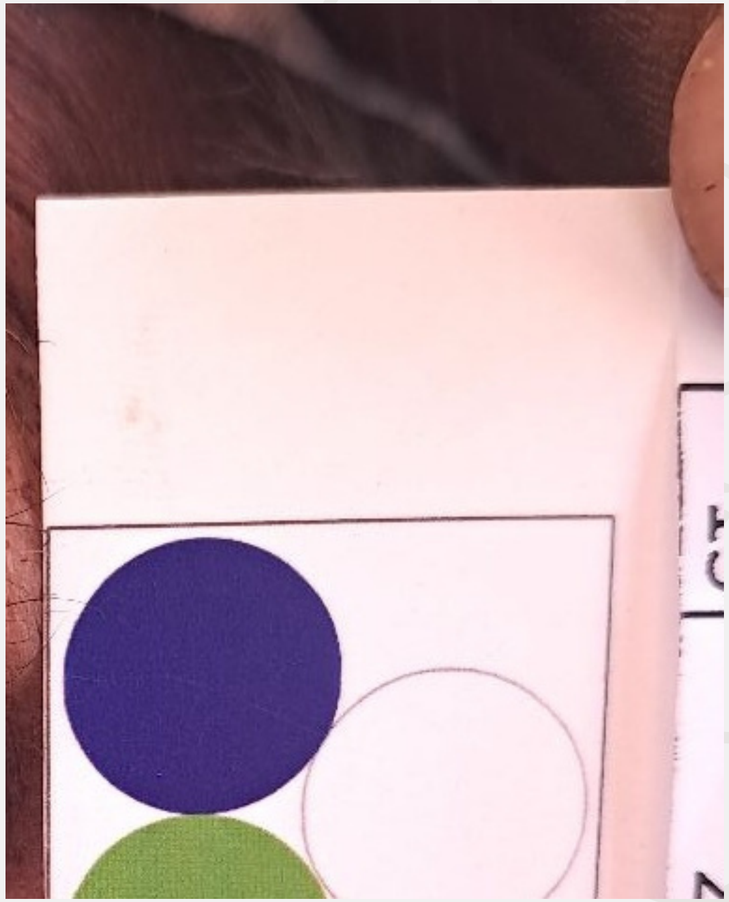


Color Palette

Contour Detection Problems:

1. Not able to capture the whole palette

Original Image

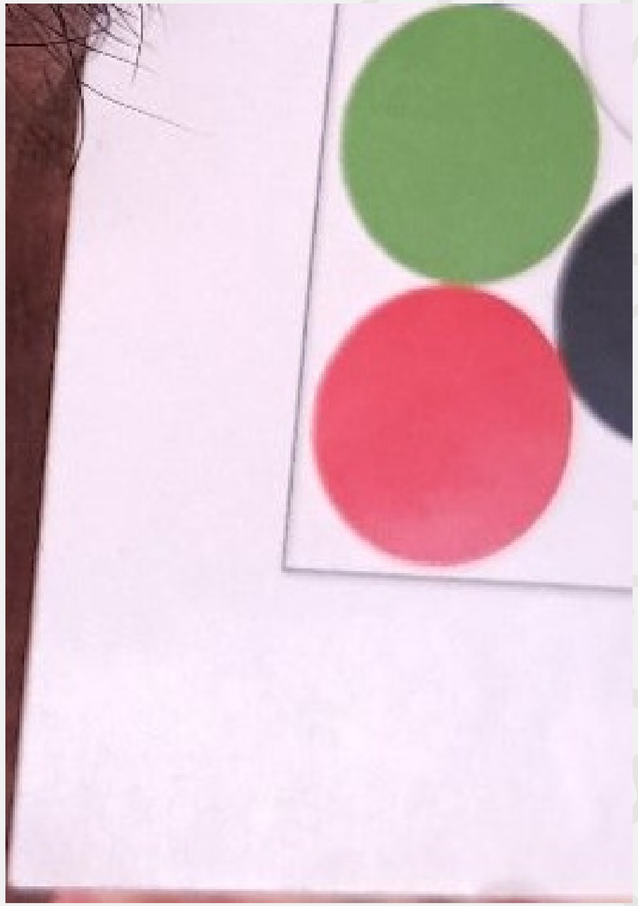


Color Palette

Contour Detection Problems:

1. Not able to capture the whole palette

Original Image

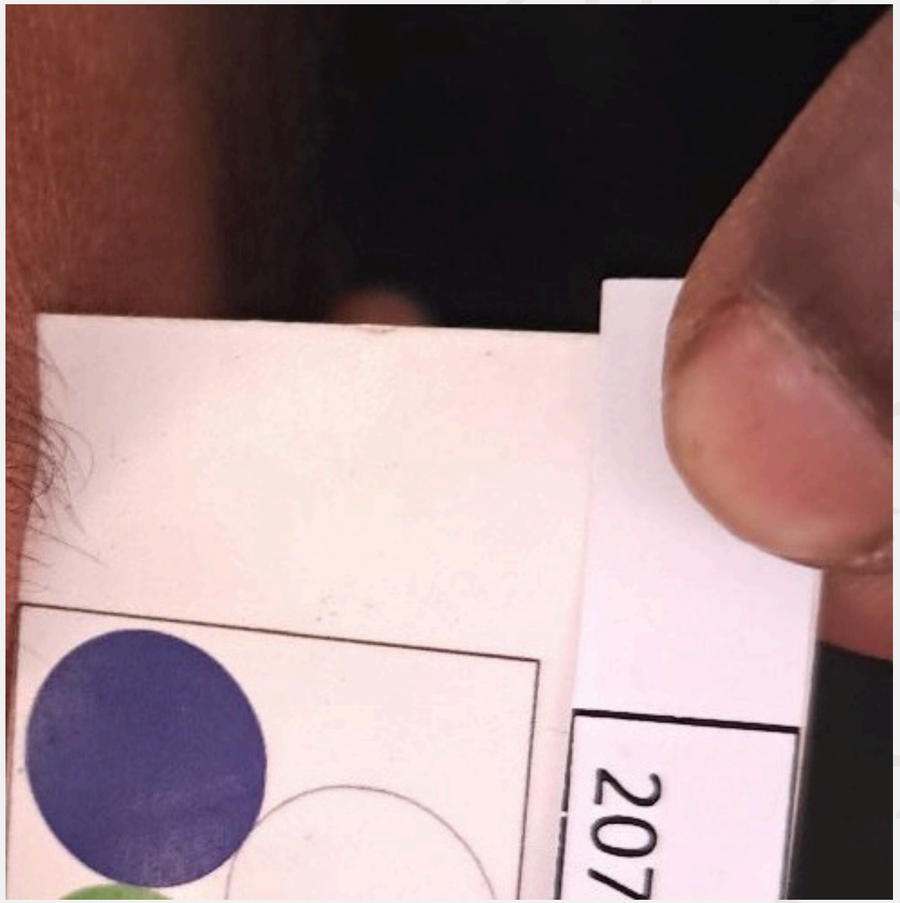


Color Palette

Contour Detection Problems:

1. Not able to capture the whole palette

Original Image

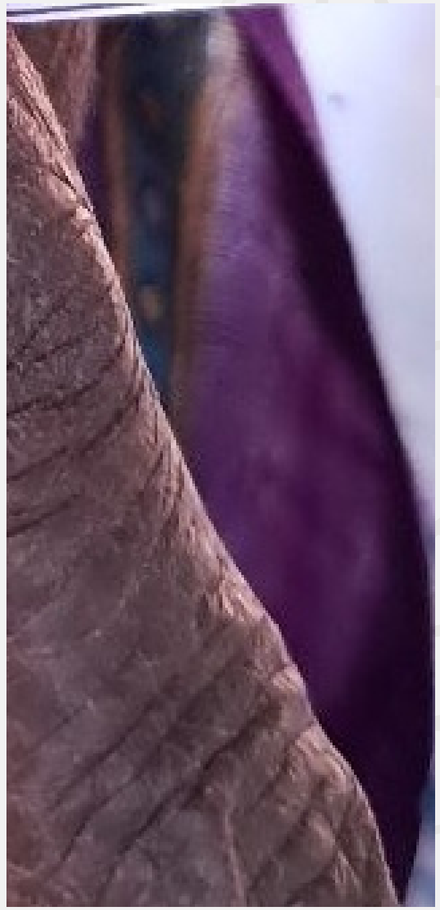


Color Palette

Contour Detection Problems:

2. Not able to capture any considerable portion of pallete

Original Image



Extracted Palette

Contour Detection Problems:

2. Not able to capture any considerable portion of pallete

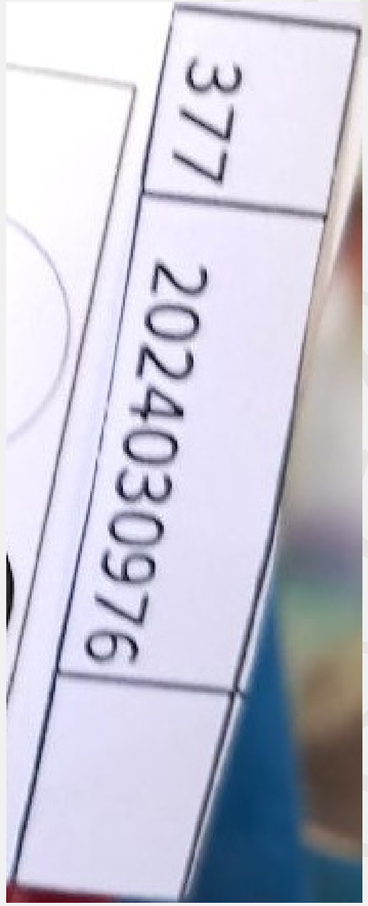
Original Image

Extracted Palette

Contour Detection Problems:

2. Not able to capture any considerable portion of pallete

Original Image



Extracted Palette

Contour Detection Problems:

2. Not able to capture any considerable portion of pallete

Original Image

Extracted Palette

Contour Detection Conclusion:

- Contour detection was used only for left eye images.
- At least **25%** of the obtained color palette images from this contour detection are **faulty** (estimate).
- Ignoring the fault, I tried the same method for palm images where the algorithm performed worse. **50%** of the images were faulty.

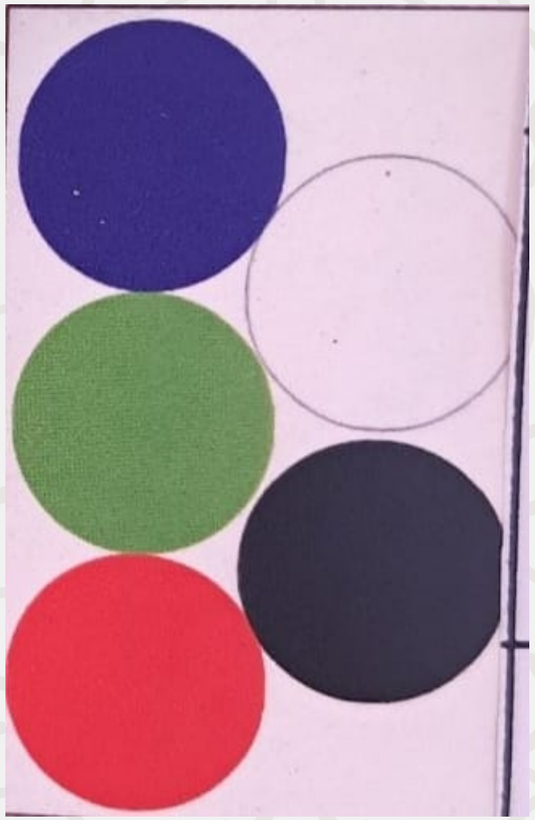
WHY?

- Edge detection (cv2.Canny()) can pick up unwanted small contours
- If the color palette blends into the background or lacks strong edges, it may not be detected properly

Template Matching:

Tried using template matching to detect the palette but the performance was far worse.

Mainly because the orientation of color palette is different in all the images.



Template Image

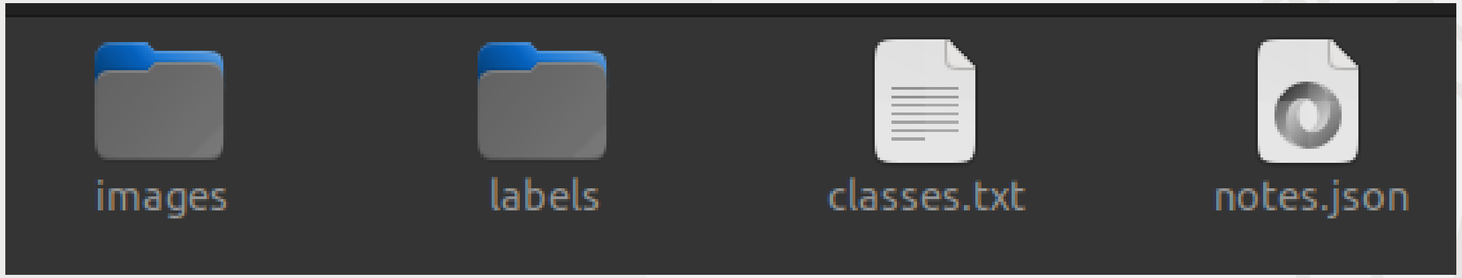
Object Detection using YOLO

- **Created a custom dataset**
  - Took 60 images from eye, palm, fingernail and tongue each
  - Total dataset created is of 240 images

- **Labelling the custom dataset for training**
  - Manually labelled all 240 images using label studio

Object Detection using YOLO

- **Labelling the custom dataset for training**
  - Manually labelled all 240 images using label studio
  - After labelling the images, label studio give the file on which YOLO model can be trained



- images folder contains the training images i.e. 240 images
- labels folder contains 240 .txt file containing coordinates of segmentation boxes
- classes.txt contains the all the classes of segmentation, in our case it is only one i.e. color_palette

Object Detection using YOLO

Model Selection

- There are various version of YOLO available.

- For beginners and ease of use → **YOLOv5** is a good choice.
- For state-of-the-art performance → **YOLOv8** is the best.
- For edge devices → **YOLOv6** is optimized.
- For research and deep learning customization → **YOLOv7** provides advanced features.

Object Detection using YOLO

YOLOv5

- It is easier to use and train, so i chose yolov5(v5x) to train
- Offers small (s), medium (m), large (l), and extra-large (x) models

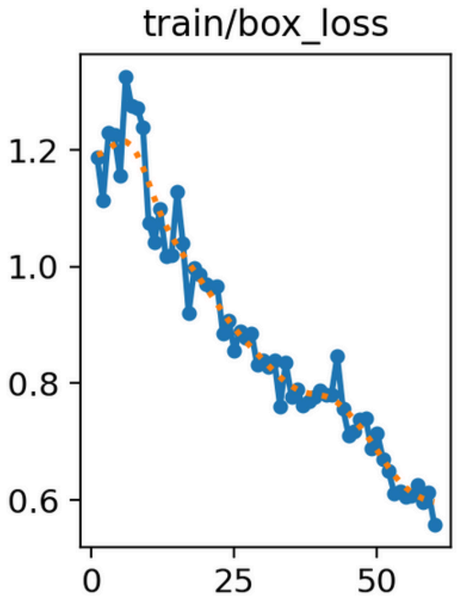
Model	Size (Params)	Speed (FPS on GPU)	mAP@0.5 (Accuracy)
YOLOv5s (small)	7.2M	Fastest (140+ FPS)	Lowest
YOLOv5m (medium)	21.2M	Fast (100+ FPS)	Medium Accuracy
YOLOv5l (large)	46.5M	Slower (80+ FPS)	Higher Accuracy
YOLOv5x (extra-large) ✅	86.7M	Slowest (60+ FPS)	Best Accuracy

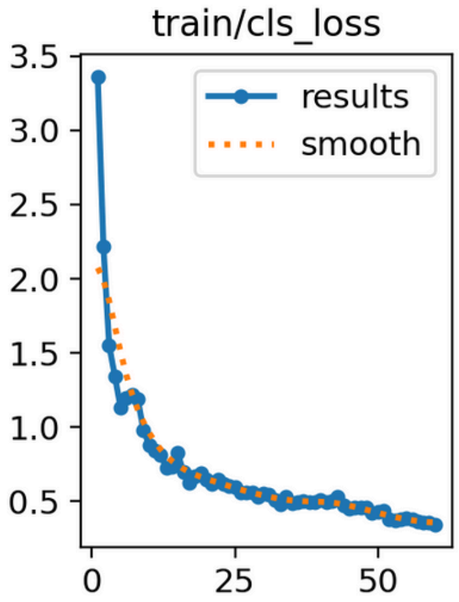
Object Detection using YOLO

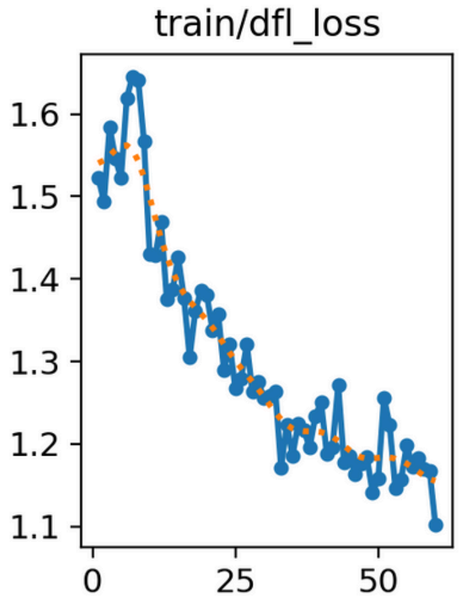
YOLOv5x

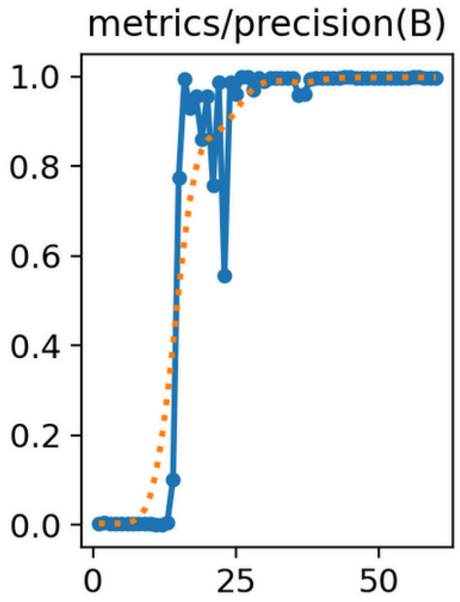
- Created Training (90%) and Validation Set (10%)
  - Training had 216 images
  - Validation had 24 images
- Trained the pretrained YOLO model for 60 epochs which took around 60-70 minutes to train

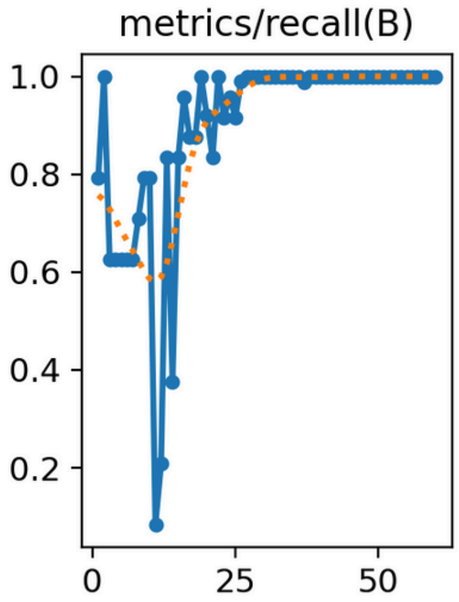
Object Detection using YOLO

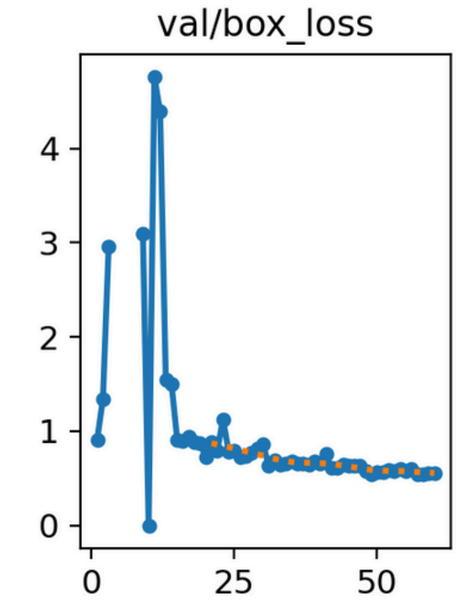


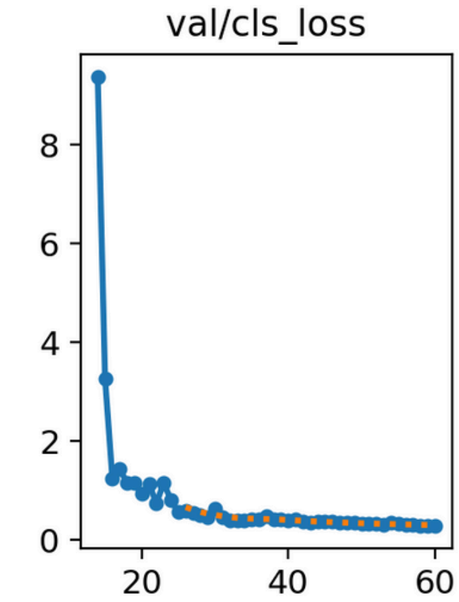


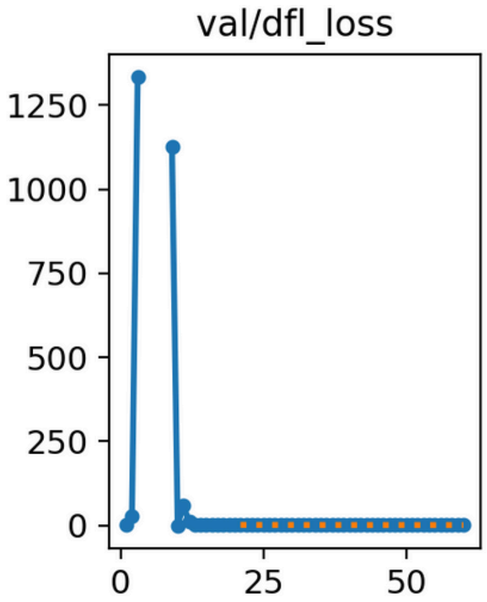


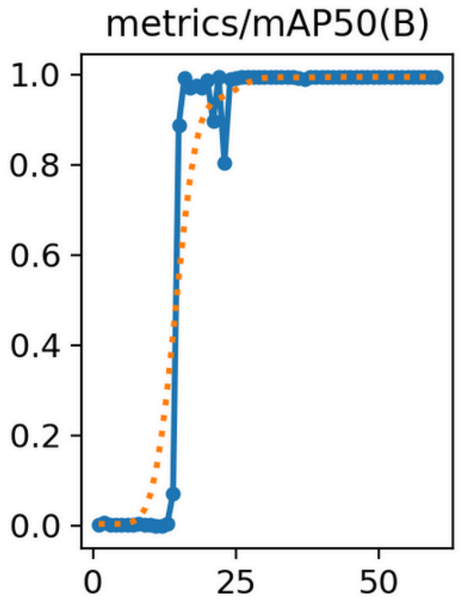


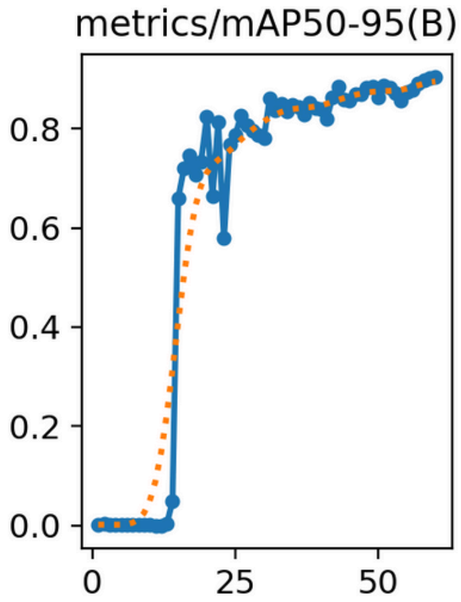






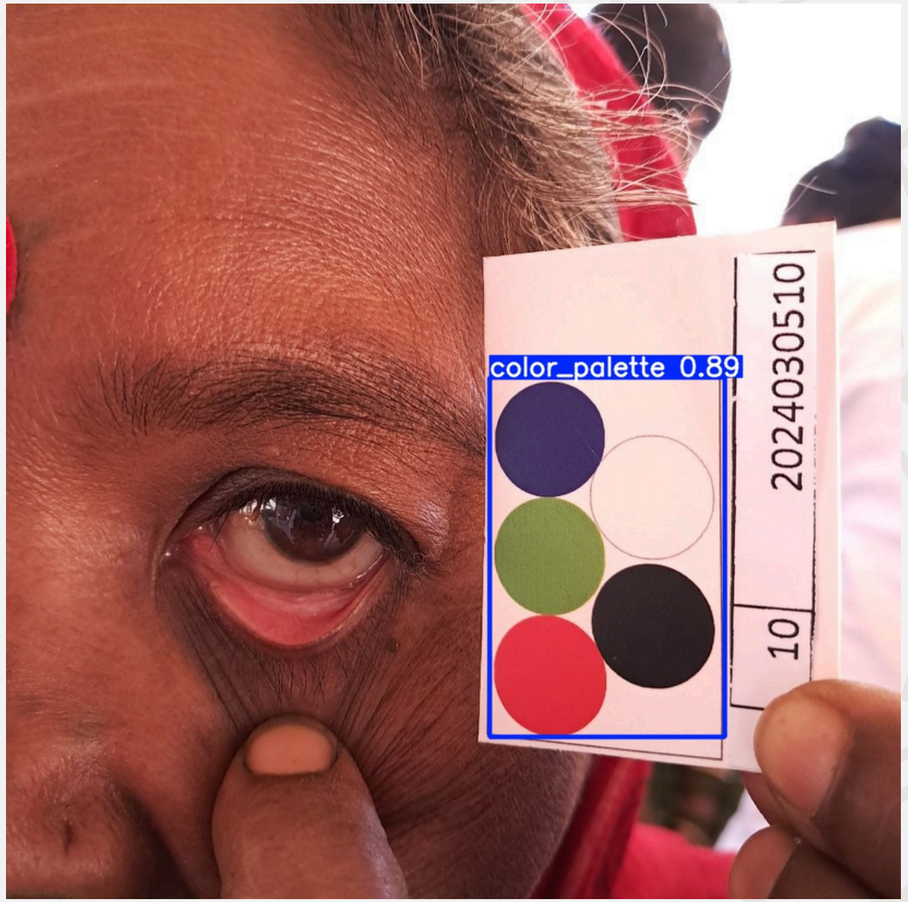






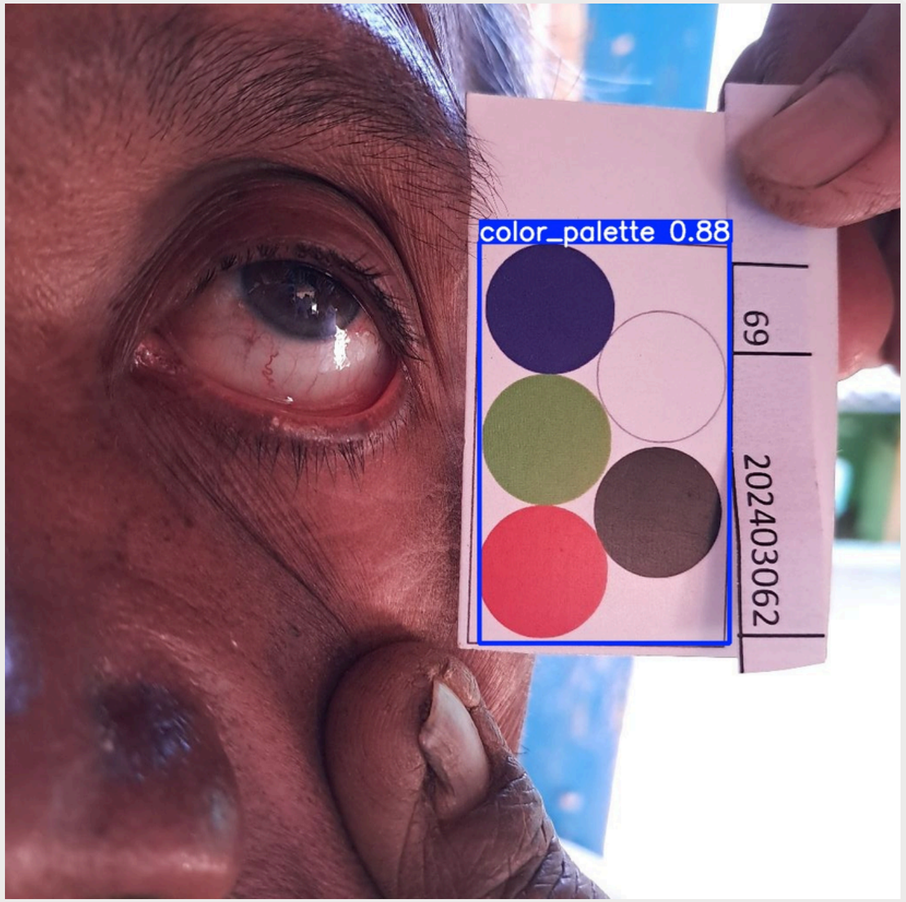
Object Detection using YOLO

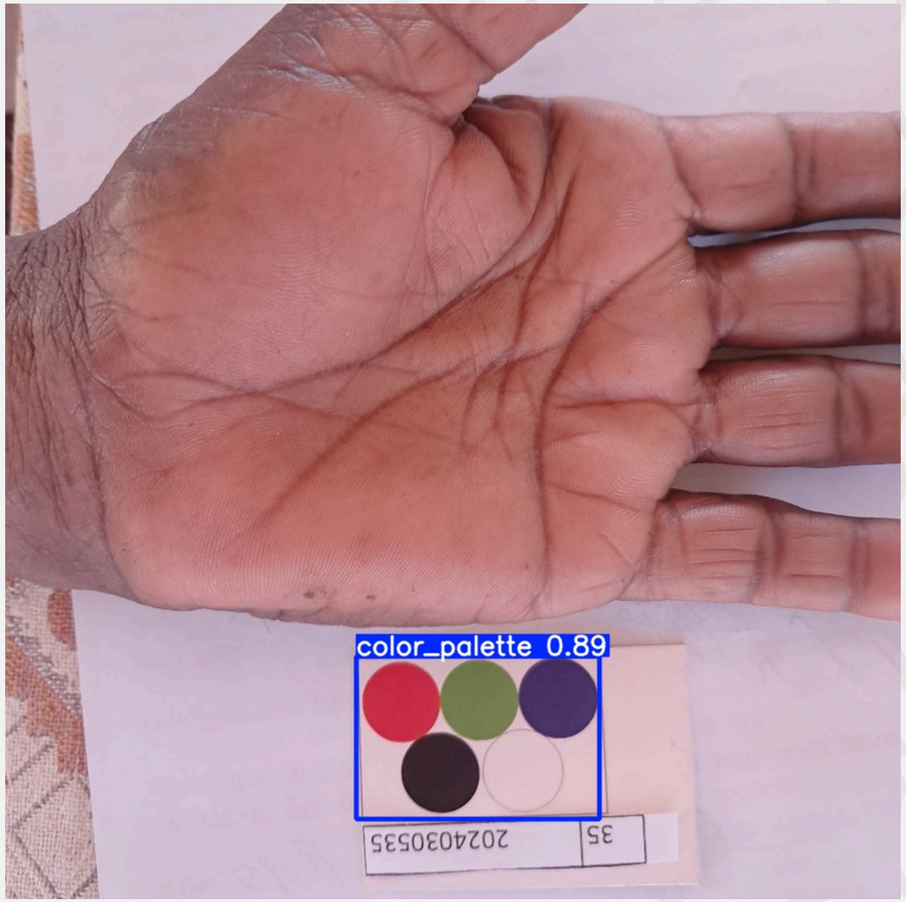
YOLOv5x Results on validation set



Object Detection using YOLO

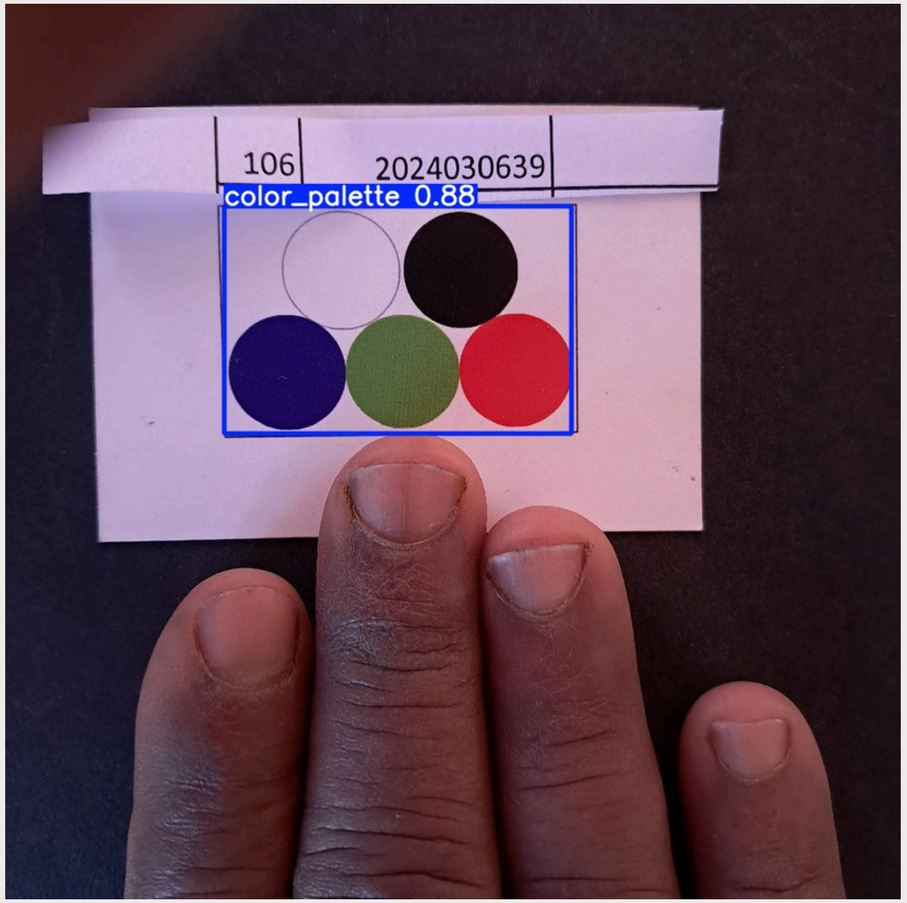
YOLOv5x Results on validation set

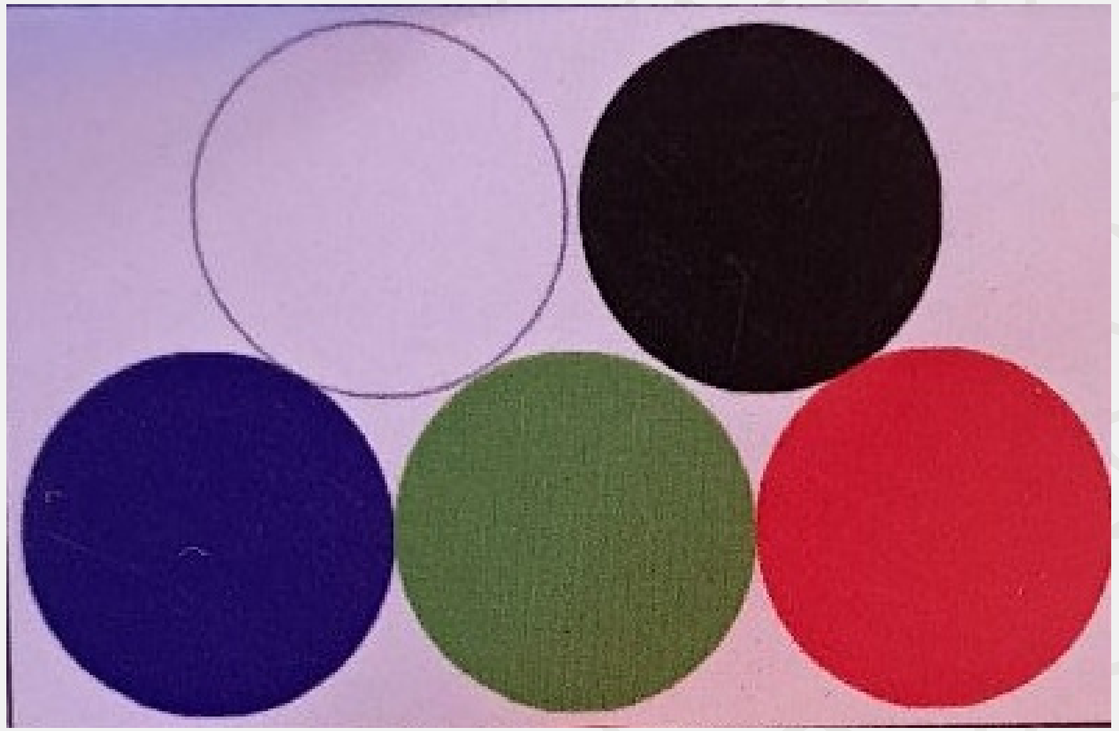




Object Detection using YOLO

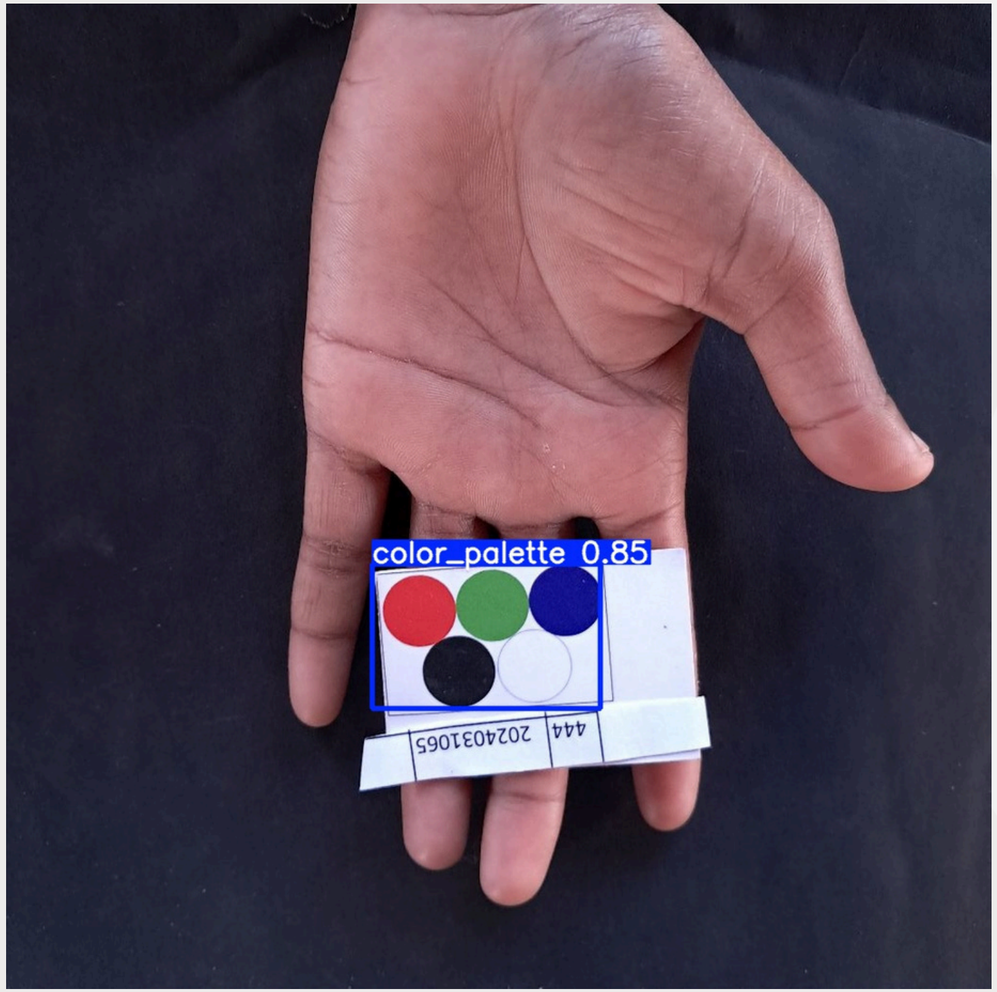
Cropped the color palette Part

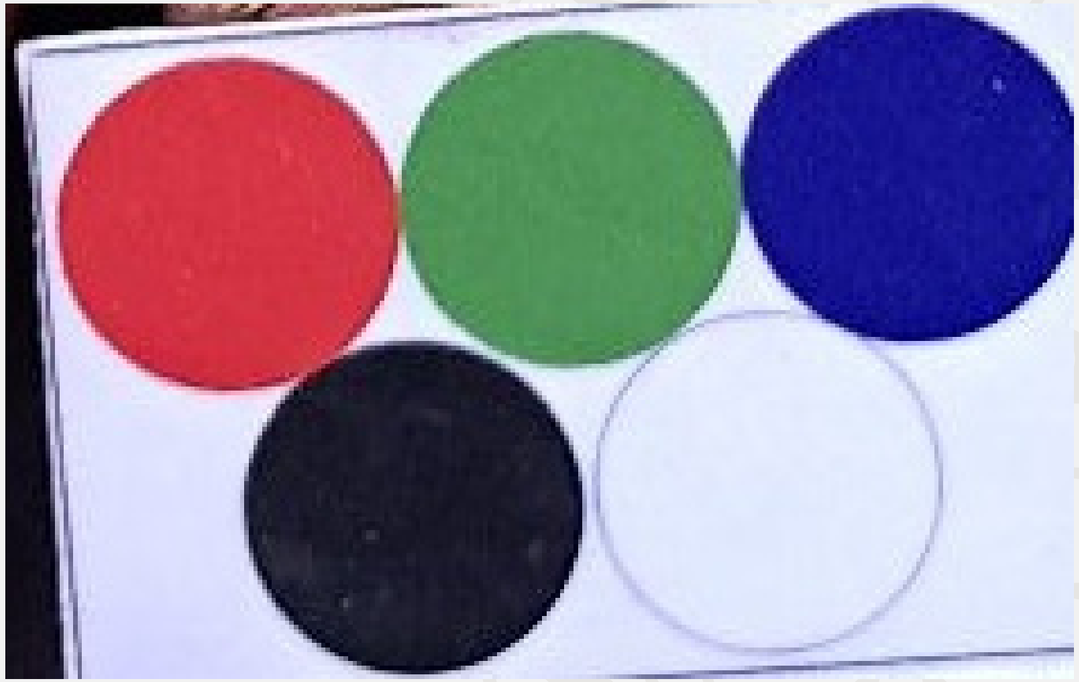




Object Detection using YOLO

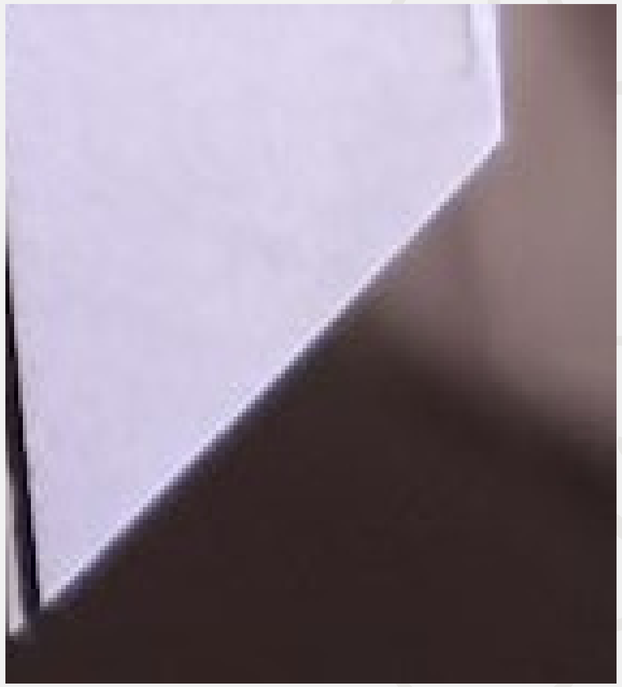
Cropped the color palette Part





Object Detection using YOLO

Some Problems:



Cropped segmented image

Object Detection using YOLO

Some Problems:

Cropped segmented image

Object Detection using YOLO

How to solve these Problems:

- The mistake is from my side as I didn’t thought that there can be more than one segmentation box

- This mistake is very less though, 2 images in eye out of 400 and 3 images in tongue out of 400 so I will manually see through them

Solved!!

Future Research

RECOMMENDATION FOR NEXT RESEARCH

- Make model without using the color palette

Thank you