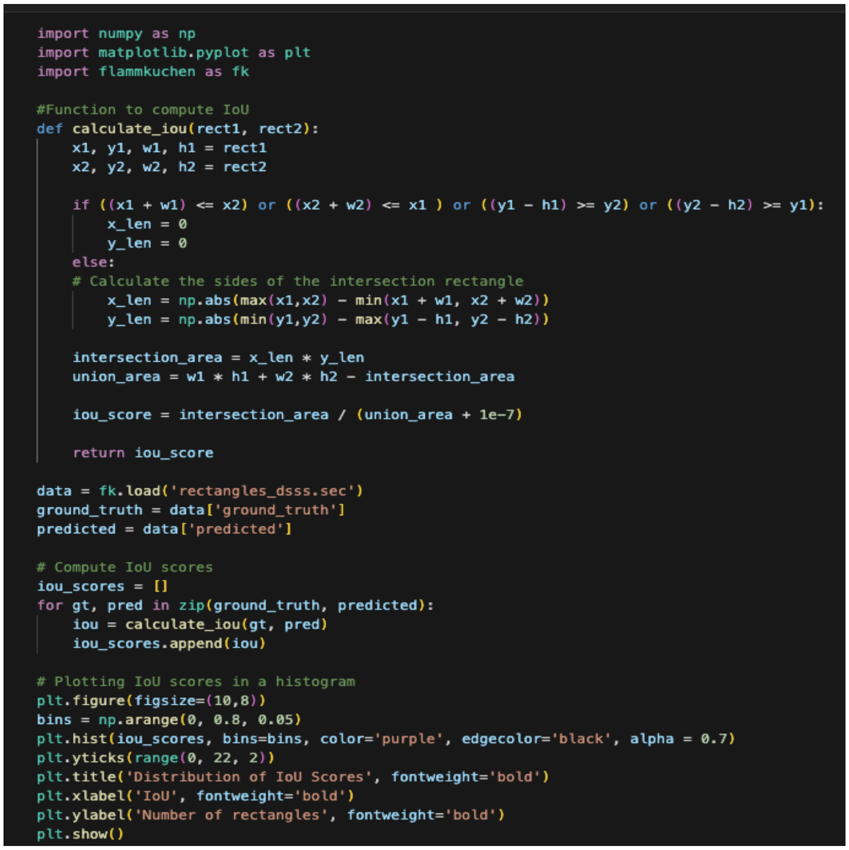
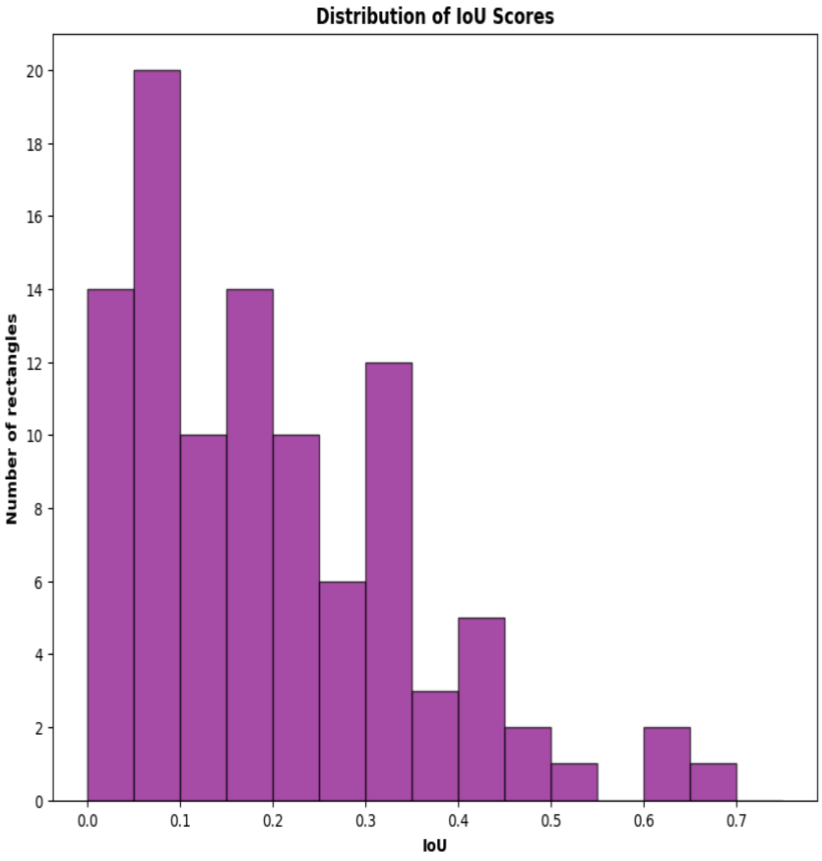
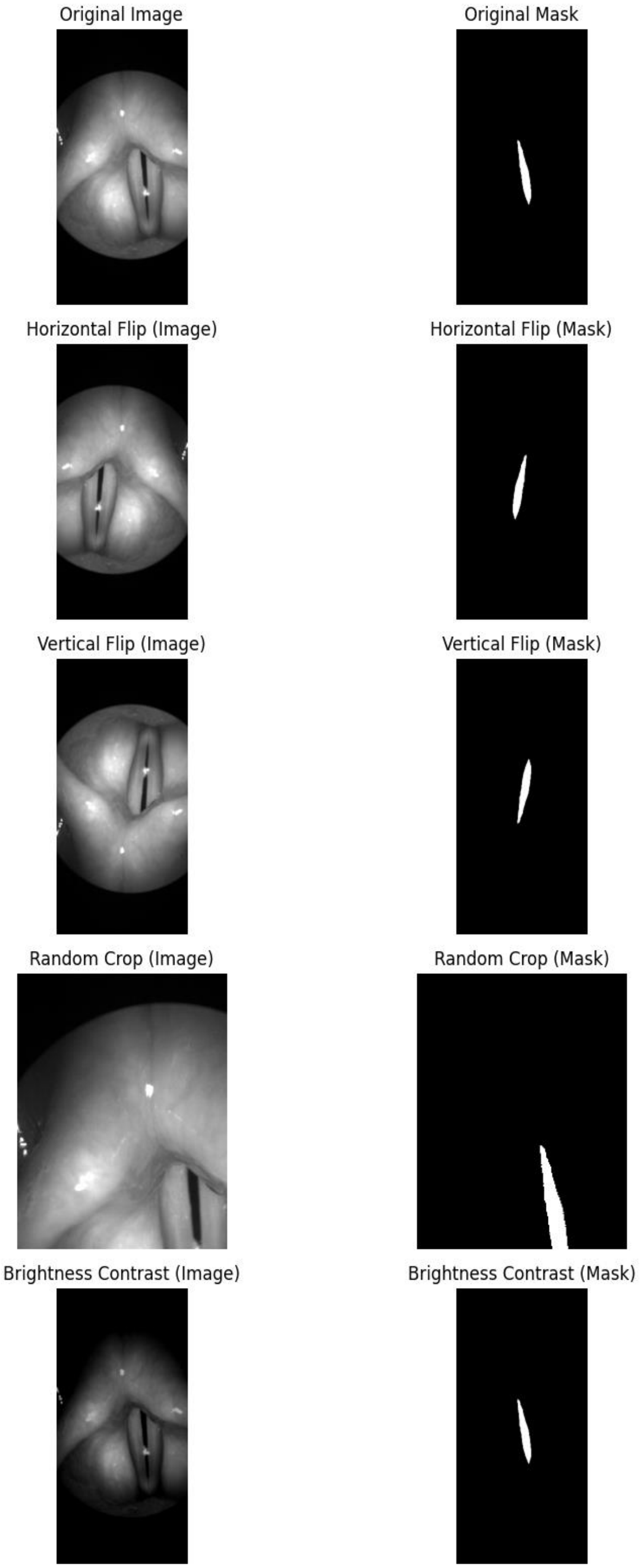
Task 1:



```
import numpy as np
import matplotlib.pyplot as plt
import flammkuchen as fk

#Function to compute IoU
def calculate_iou(rect1, rect2):
    x1, y1, w1, h1 = rect1
    x2, y2, w2, h2 = rect2

    if ((x1 + w1) <= x2) or ((x2 + w2) <= x1 ) or ((y1 - h1) >= y2) or ((y2 - h2) >= y1):
        x_len = 0
        y_len = 0
    else:
    # Calculate the sides of the intersection rectangle
        x_len = np.abs(max(x1,x2) - min(x1 + w1, x2 + w2))
        y_len = np.abs(min(y1,y2) - max(y1 - h1, y2 - h2))

    intersection_area = x_len * y_len
    union_area = w1 * h1 + w2 * h2 - intersection_area

    iou_score = intersection_area / (union_area + 1e-7)

    return iou_score

data = fk.load('rectangles_dsss.sec')
ground_truth = data['ground_truth']
predicted = data['predicted']

# Compute IoU scores
iou_scores = []
for gt, pred in zip(ground_truth, predicted):
    iou = calculate_iou(gt, pred)
    iou_scores.append(iou)

# Plotting IoU scores in a histogram
plt.figure(figsize=(10,8))
bins = np.arange(0, 0.8, 0.05)
plt.hist(iou_scores, bins=bins, color='purple', edgecolor='black', alpha = 0.7)
plt.yticks(range(0, 22, 2))
plt.title('Distribution of IoU Scores', fontweight='bold')
plt.xlabel('IoU', fontweight='bold')
plt.ylabel('Number of rectangles', fontweight='bold')
plt.show()
```



Task 2:



Rohit Potdukhe, M. N.: 22985091, IdM: yx49uxym