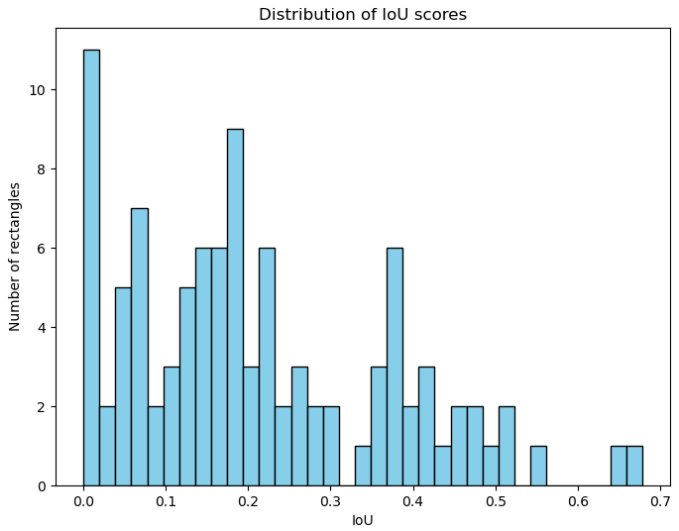
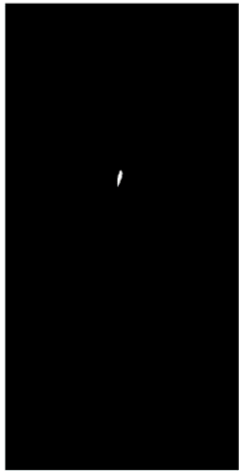
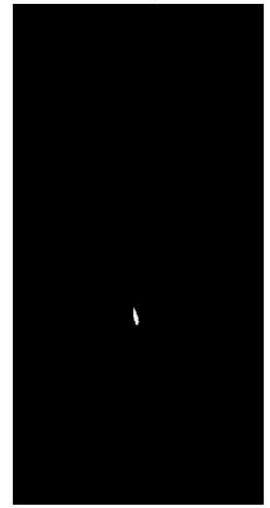
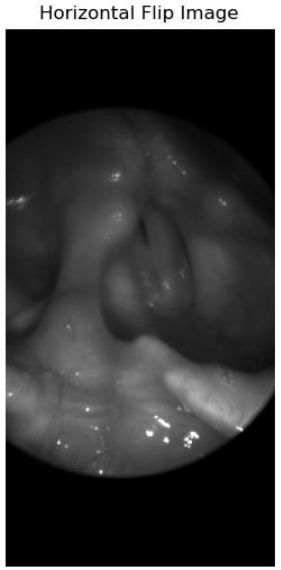
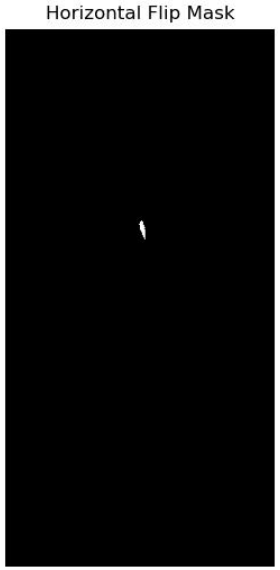
```python
def calculate_iou(rect1, rect2):
    # Convert rectangle coordinates and dimensions to NumPy arrays
    rect1 = np.array(rect1)
    rect2 = np.array(rect2)

    # Calculate coordinates of bounding boxes
    x1_gt, y1_gt, w_gt, h_gt = rect1
    x2_gt, y2_gt = x1_gt + w_gt, y1_gt + h_gt  # Bottom-right corner of ground truth

    x1_p, y1_p, w_p, h_p = rect2
    x2_p, y2_p = x1_p + w_p, y1_p + h_p  # Bottom-right corner of predicted box

    # Calculate coordinates of the intersection rectangle
    x_left = np.maximum(x1_gt, x1_p)
    y_top = np.maximum(y1_gt, y1_p)
    x_right = np.minimum(x2_gt, x2_p)
    y_bottom = np.minimum(y2_gt, y2_p)

    # Handle cases where there's no intersection
    intersection_width = np.maximum(0, x_right - x_left)
    intersection_height = np.maximum(0, y_bottom - y_top)
    intersection_area = intersection_width * intersection_height

    # Calculate areas of both rectangles
    area_gt = w_gt * h_gt
    area_pred = w_p * h_p

    # Calculate IoU
    # Union area = Ground Truth area + Predicted area - Intersection area
    union_area = area_gt + area_pred - intersection_area
    iou = intersection_area / union_area

    return iou
```

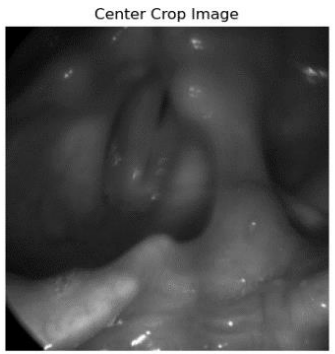


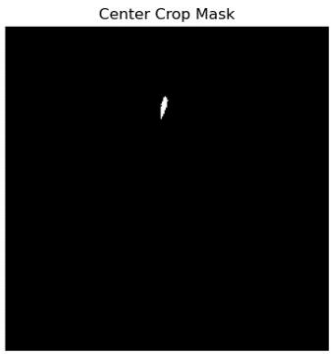


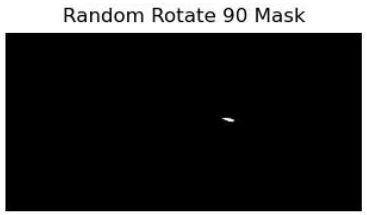












I-En Lee, 23365648, di65haji