```
#The darknet library cannot be found in my IDE and I cannot figure it
out, so I used the 'cv2.dnn' module in opencv

import cv2
import numpy as np


def detect_objects(img_path, conf_threshold=0.5, nms_threshold=0.4):
    # Load YOLOv3 network
    config_file = "/Users/pananqi/darknet/cfg/yolov3.cfg"
    weights_file = "/Users/pananqi/darknet/yolov3.weights"
    net = cv2.dnn.readNetFromDarknet(config_file, weights_file)

    # Get layer names
    layer_names = net.getLayerNames()
    output_layers = [layer_names[i - 1] for i in
net.getUnconnectedOutLayers()]

    # Load image and perform forward pass
    img = cv2.imread(img_path)
    img_height, img_width, _ = img.shape
    blob = cv2.dnn.blobFromImage(img, scalefactor=1/255, size=(416,
416), swapRB=True, crop=False)
    net.setInput(blob)
    detections = net.forward(output_layers)

    # Process detections
    boxes = []
    confidences = []
    class_ids = []
    for detection in detections:
        for object_detection in detection:
            scores = object_detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > conf_threshold:
                center_x = int(object_detection[0] * img_width)
                center_y = int(object_detection[1] * img_height)
                width = int(object_detection[2] * img_width)
                height = int(object_detection[3] * img_height)
                left = int(center_x - width / 2)
                top = int(center_y - height / 2)
                boxes.append([left, top, width, height])
                confidences.append(float(confidence))
                class_ids.append(class_id)

    # Apply non-maximum suppression
    indices = cv2.dnn.NMSBoxes(boxes, confidences, conf_threshold,
nms_threshold)
```

```
    # Create list of detected objects
    objects = []
    for i in indices:
        i = i
        x, y, w, h = boxes[i]
        object_class = str(class_ids[i])
        objects.append((object_class, (x, y, w, h)))

    return objects

def draw_boxes(img_path, objects):
    # Load image
    img = cv2.imread(img_path)

    # Draw bounding boxes around detected objects

    for obj in objects:
        x, y, w, h = obj[1]
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255),
thickness=3)
        cv2.putText(img, obj[0], (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (0, 255, 0), 2)

    # Show image
    cv2.imshow('image', img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Example usage
img_path = 'cattest.jpg'
objects = detect_objects(img_path)
draw_boxes(img_path, objects)
```

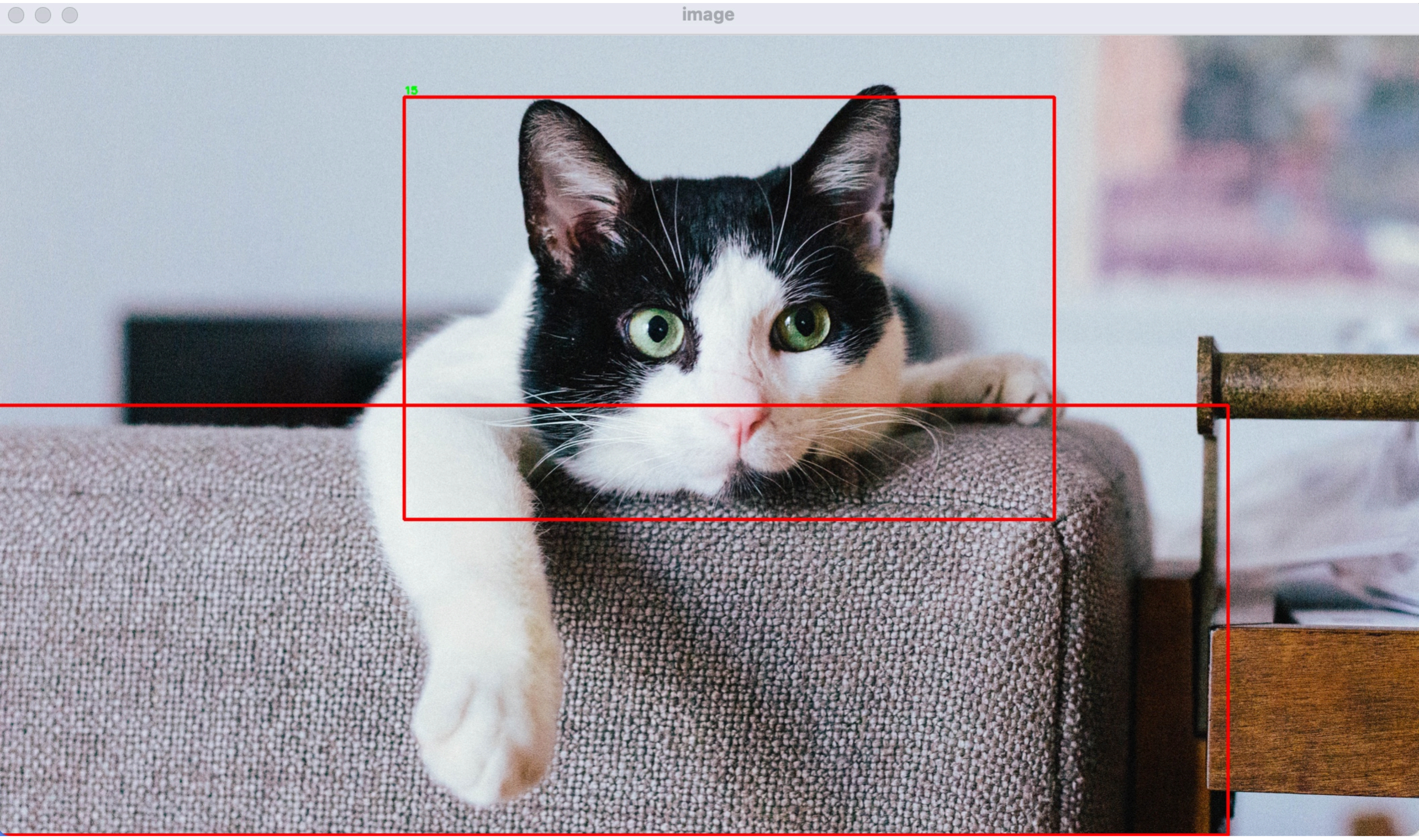
image
15