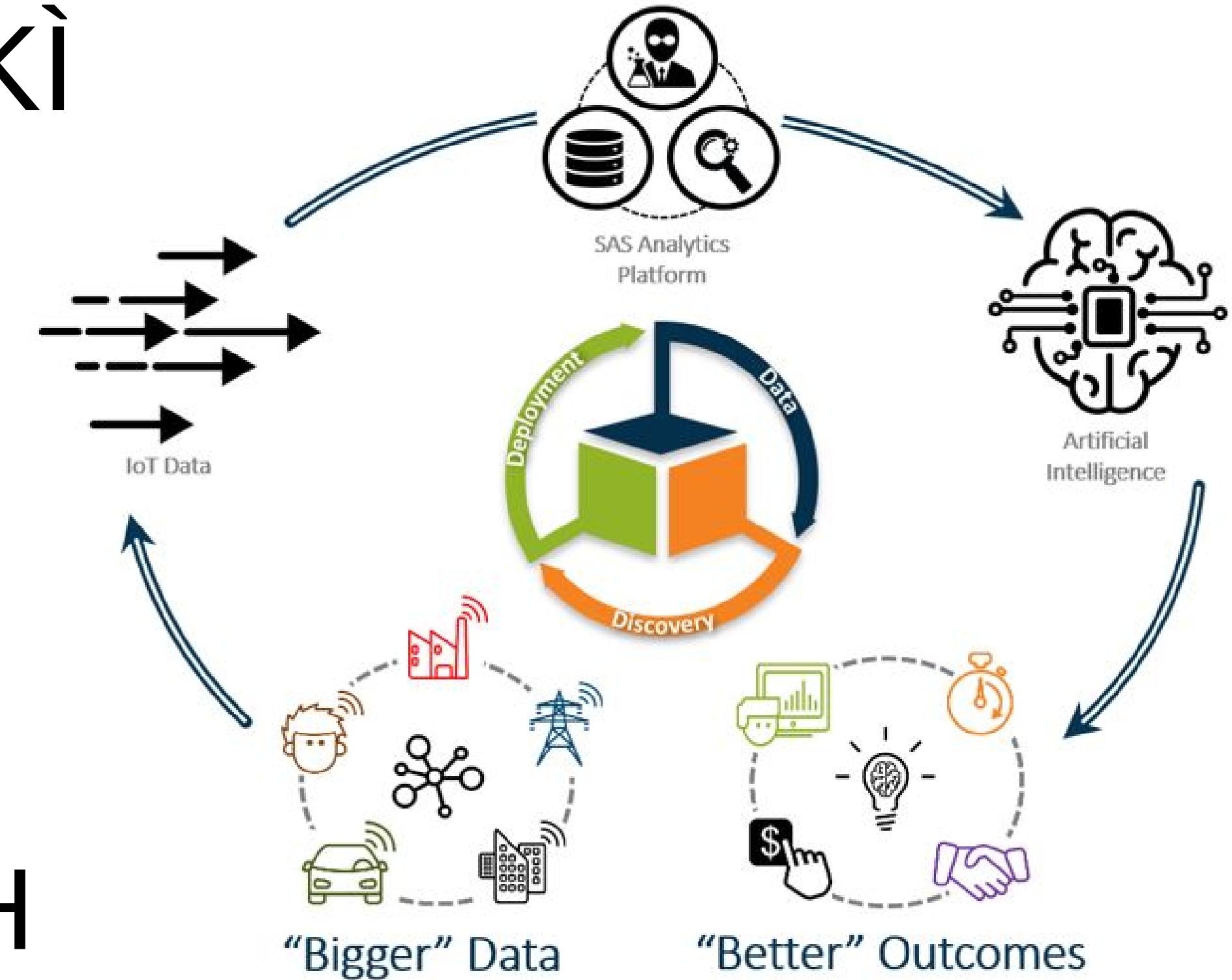


# BÁO CÁO CUỐI KÌ

AIoT – 20TGMT  
FIT.HCMUS

PROJECT:  
OBJECT  
DETECTION WITH  
YOLOV3 AND  
ESP32-CAM



20127674 – Lê Đức Đạt

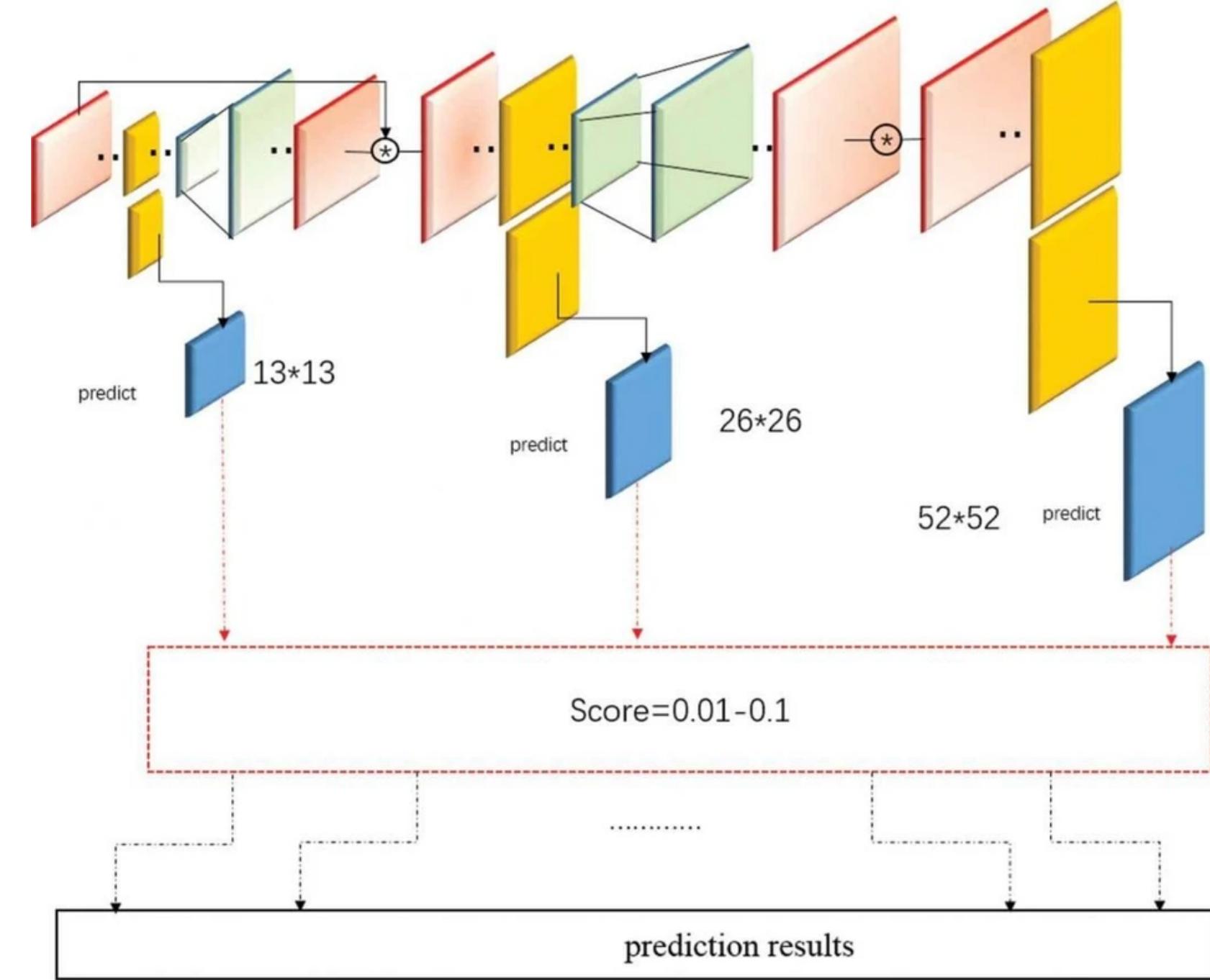
# MỤC LỤC

1. Giới thiệu
2. Tổng quan về ESP32-CAM
3. Tổng quan về YOLOv3
4. Xây dựng hệ thống – giải thích code
5. Ứng dụng và kết quả
6. Thảo luận
7. Kết luận
8. Tư liệu tham khảo

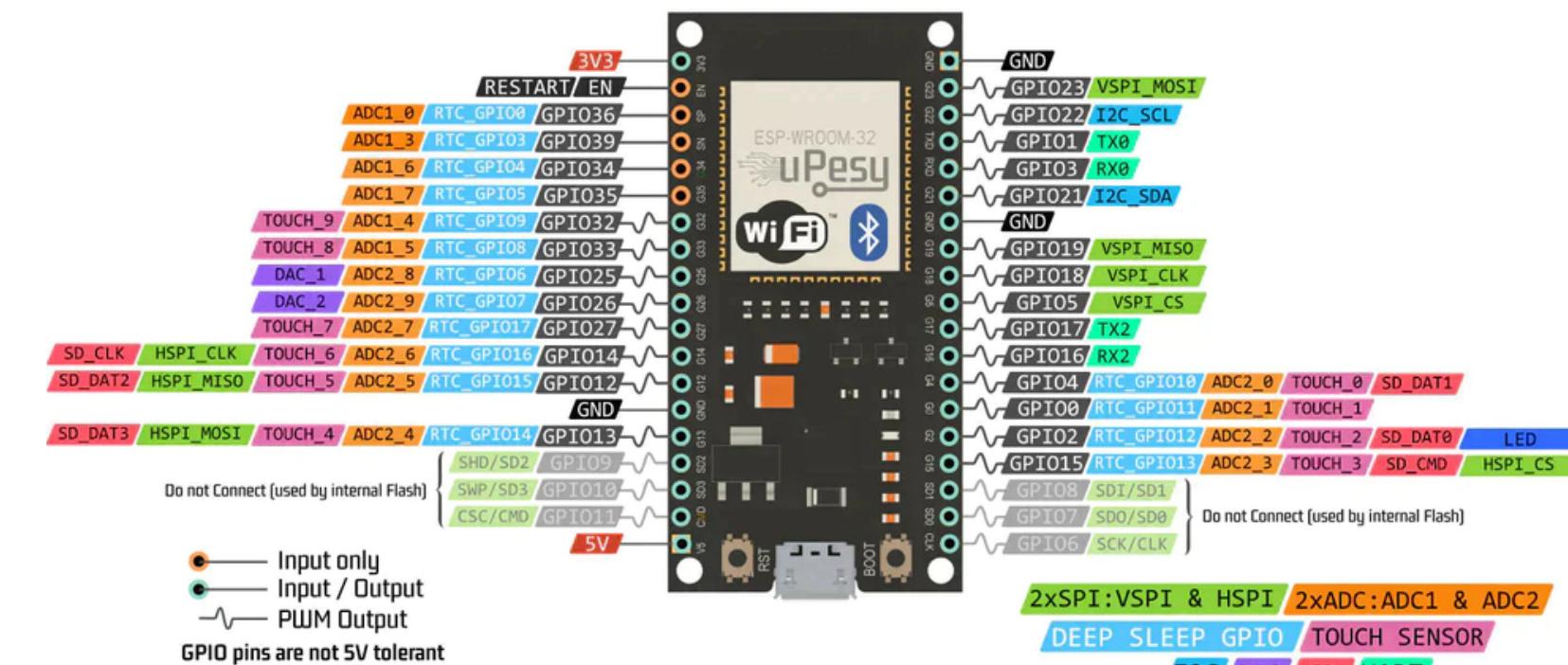
# GIỚI THIỆU



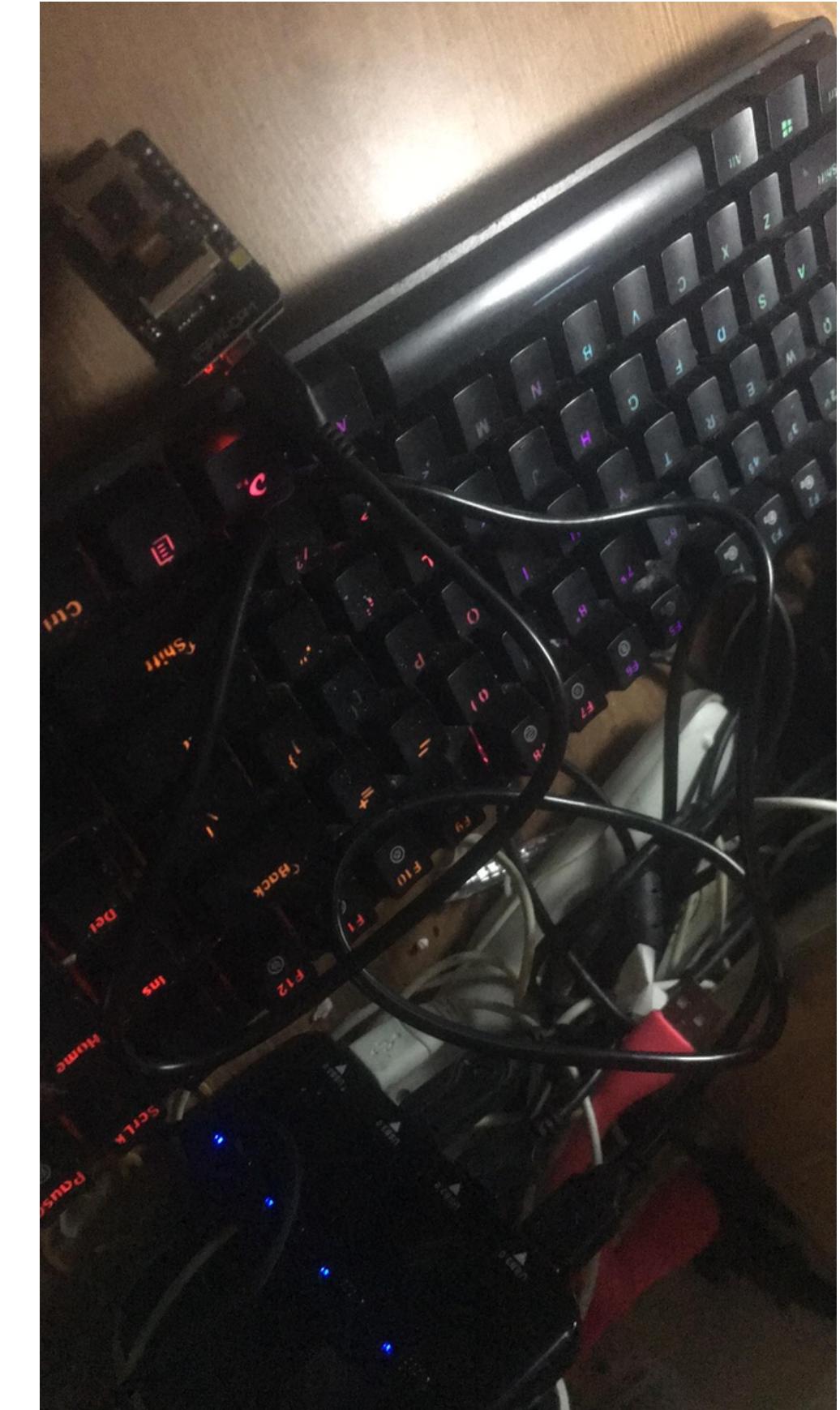
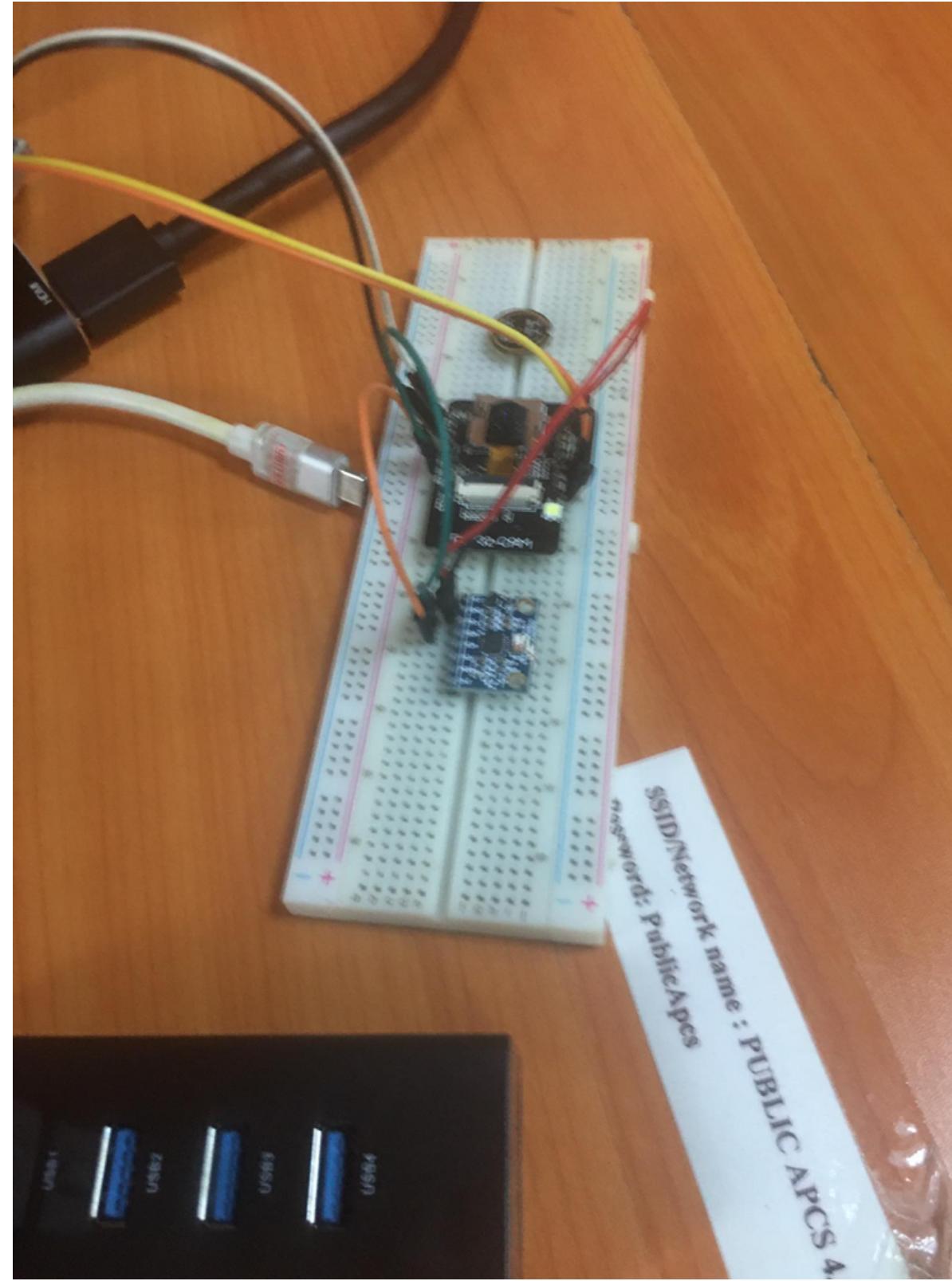
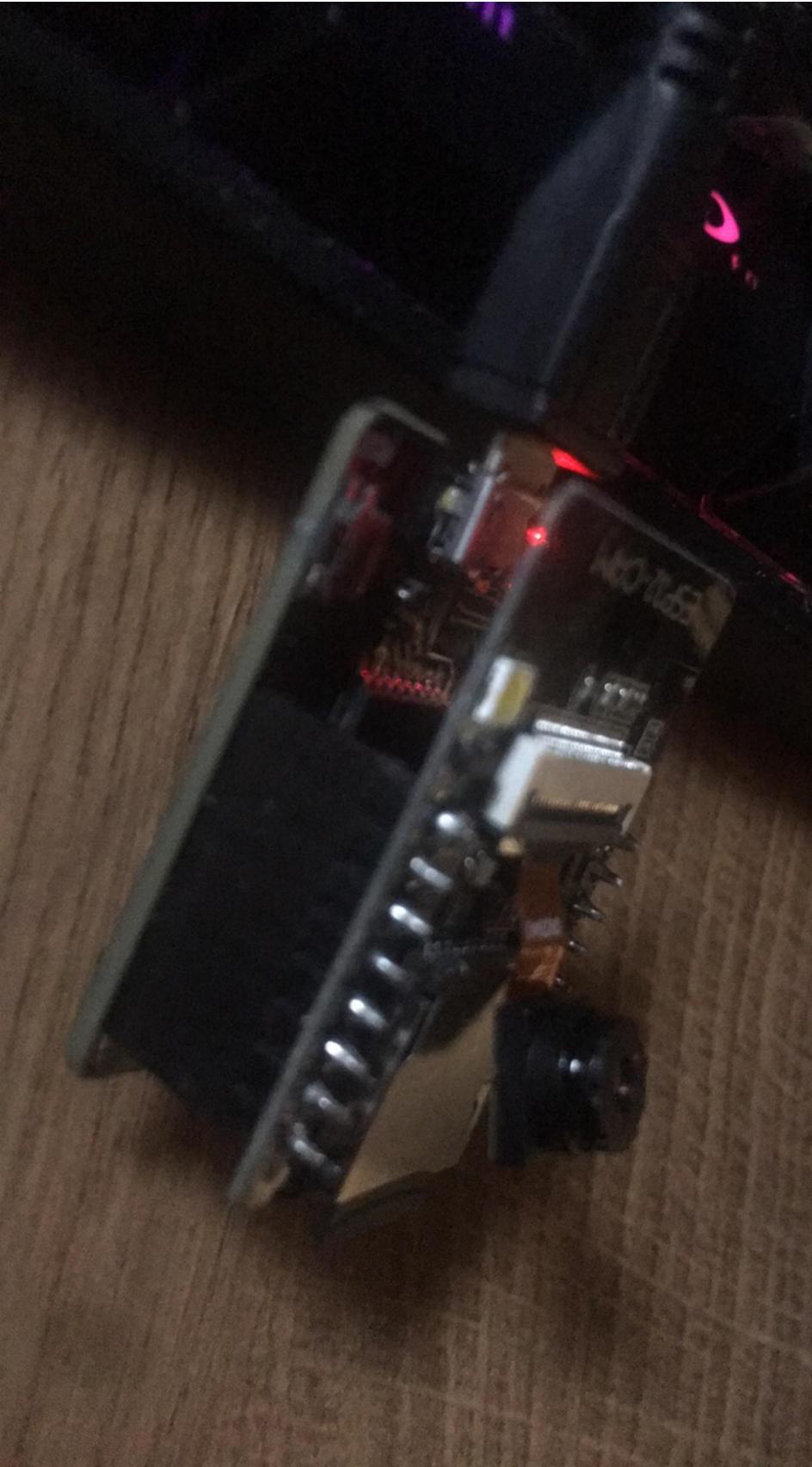
Thế giới hiện đại ngày càng số hóa, và việc xử lý hình ảnh trở nên quan trọng trong nhiều lĩnh vực như an ninh, y tế, và tự động hóa. Bài toán nhận dạng đối tượng, một vấn đề trong thị giác máy tính, đã được giải quyết bằng cách sử dụng mạng nơ-ron sâu, và mô hình YOLO là một phương pháp tiêu biểu. Phiên bản YOLOv3 cải tiến hiệu suất so với phiên bản trước, nhưng việc áp dụng nó trên thiết bị nhúng như ESP32-CAM gặp nhiều thách thức về tài nguyên. Mục tiêu của đồ án là tạo ra một hệ thống nhận dạng đối tượng thời gian thực trên ESP32-CAM sử dụng YOLOv3, mang lại một giải pháp nhúng giá rẻ và hiệu quả cho việc nhận dạng đối tượng.



ESP32 Wroom DevKit Full Pinout



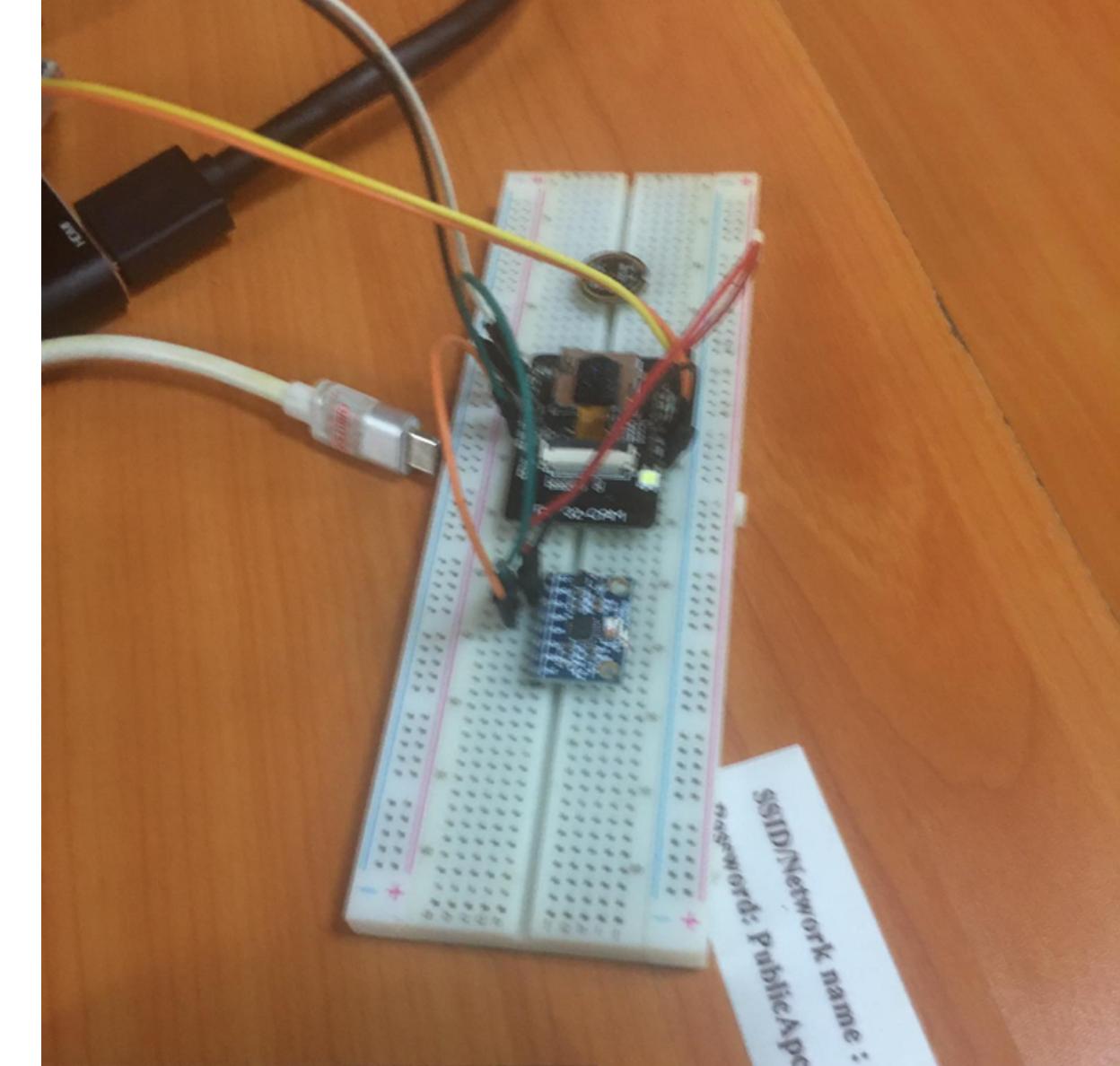
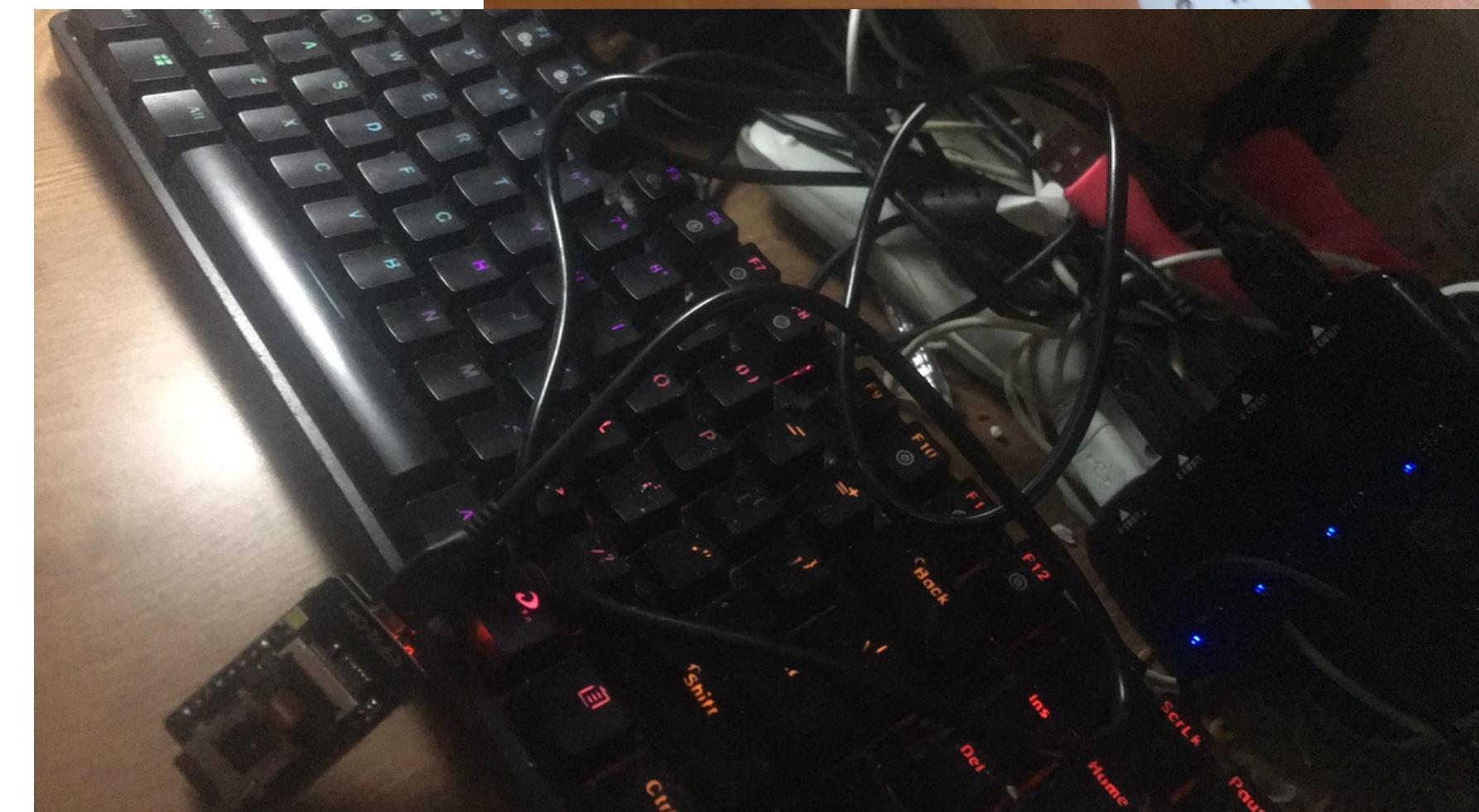
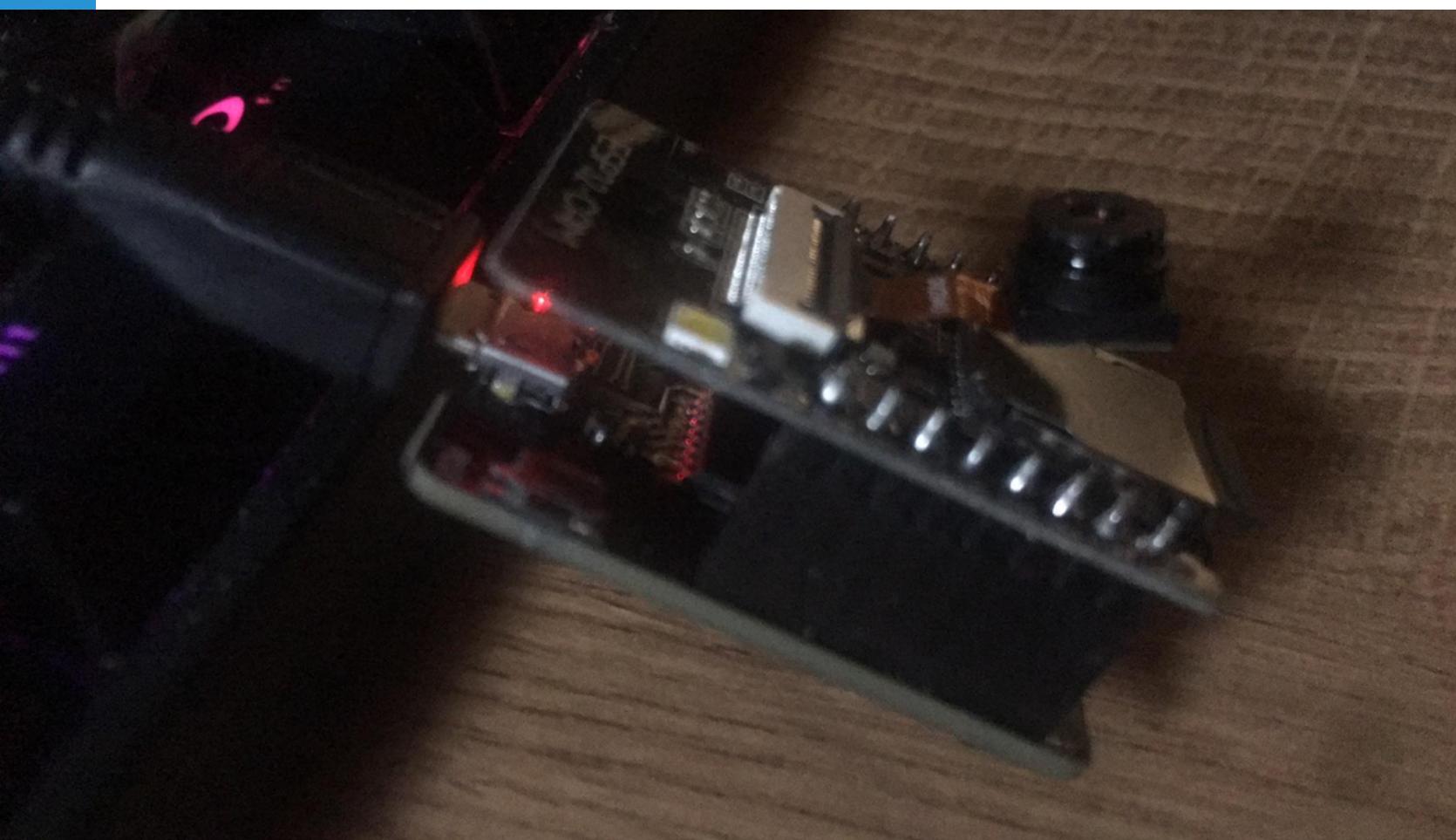
# TỔNG QUAN VỀ ESP32-CAM

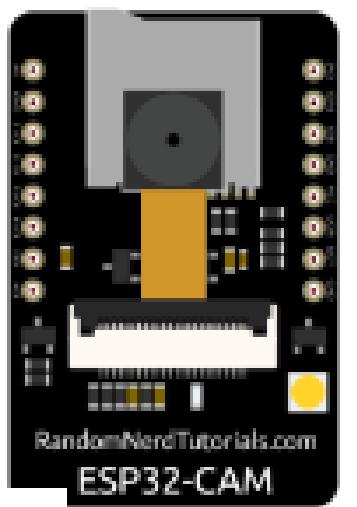
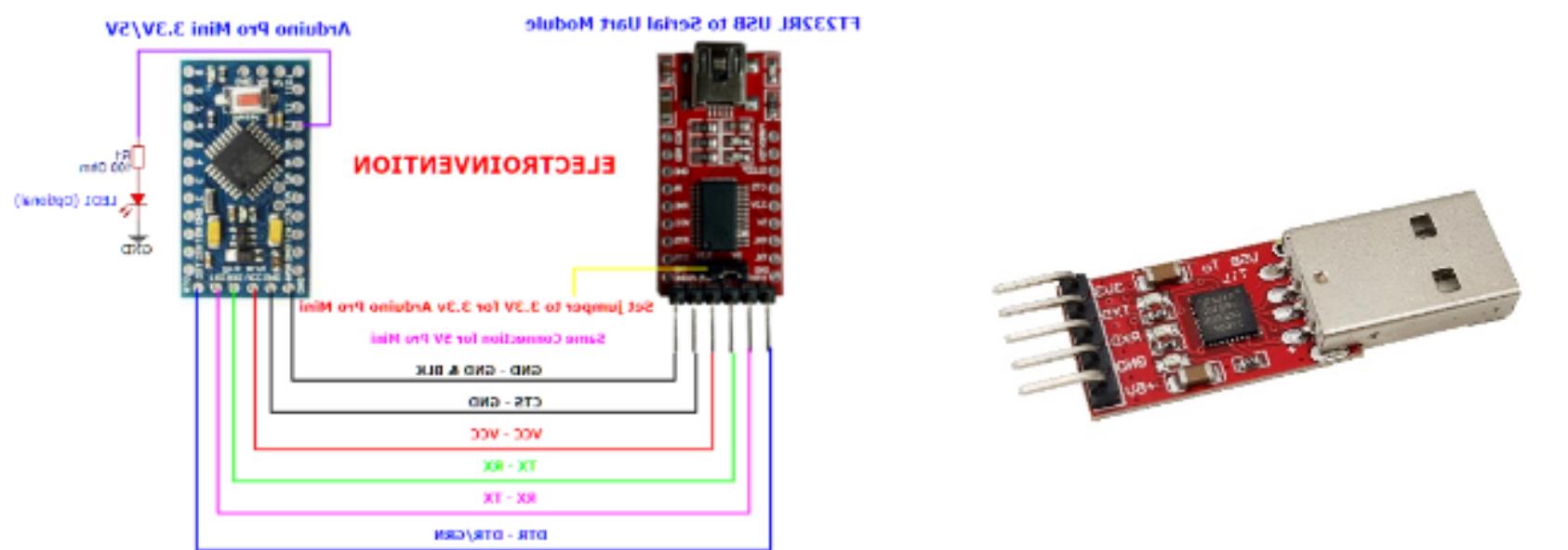
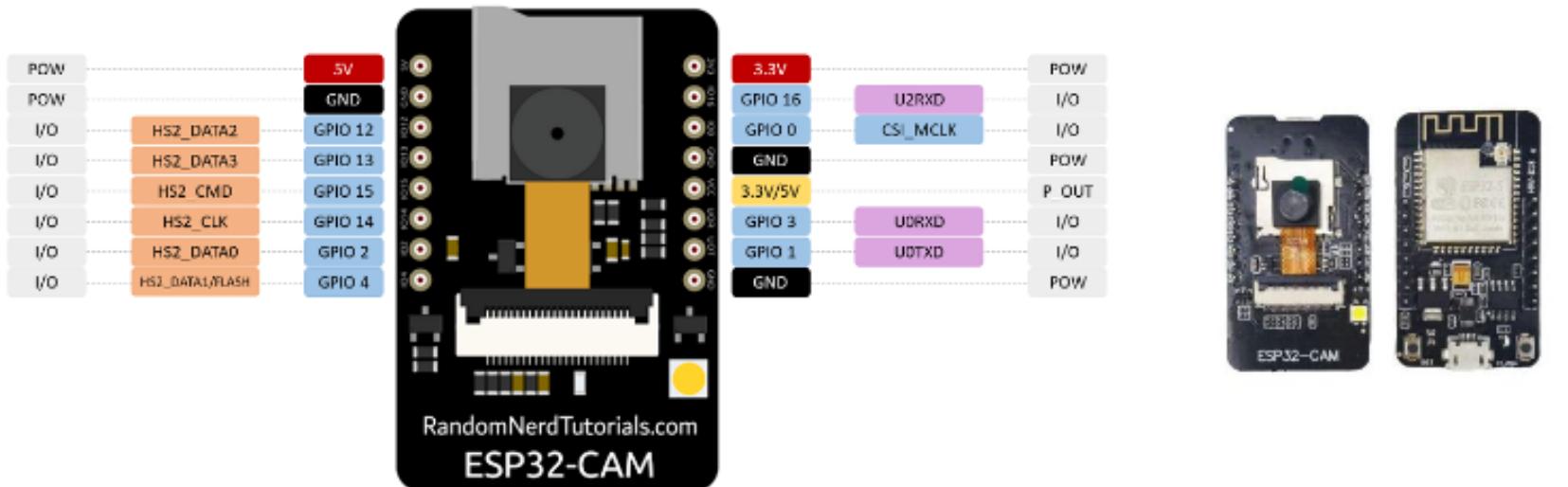


# Tính năng và ưu điểm

## Cách hoạt động

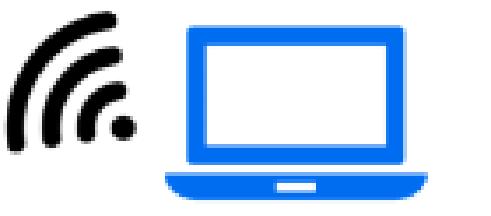
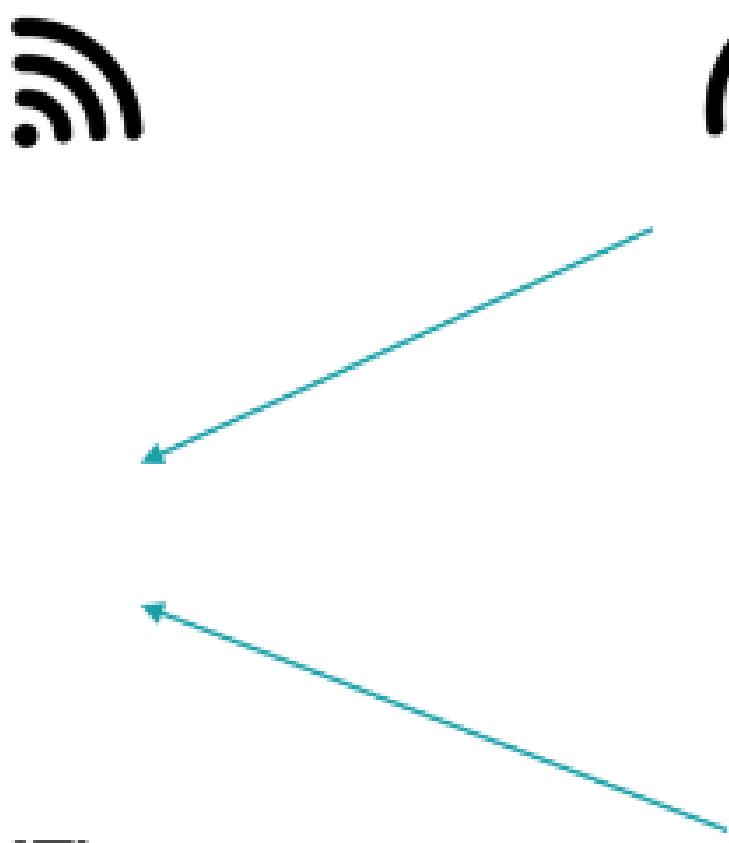
## Cách kết nối và cấu hình





ESP32-CAM  
Web Server  
**ESS POINT)**

```
const char* WIFI_SSID = "HTD";
const char* WIFI_PASS = "20012002";
```

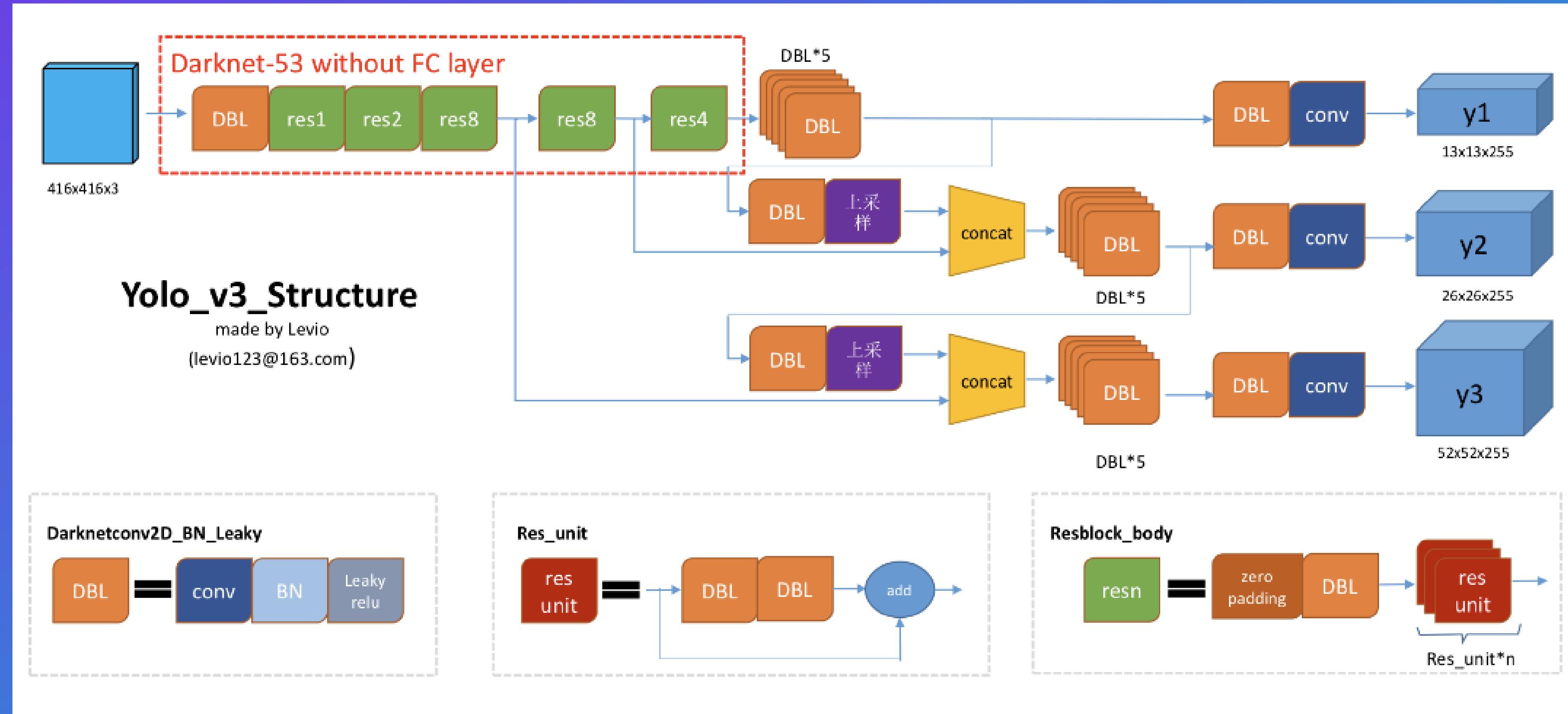


Wi-Fi Client  
**(STATION)**



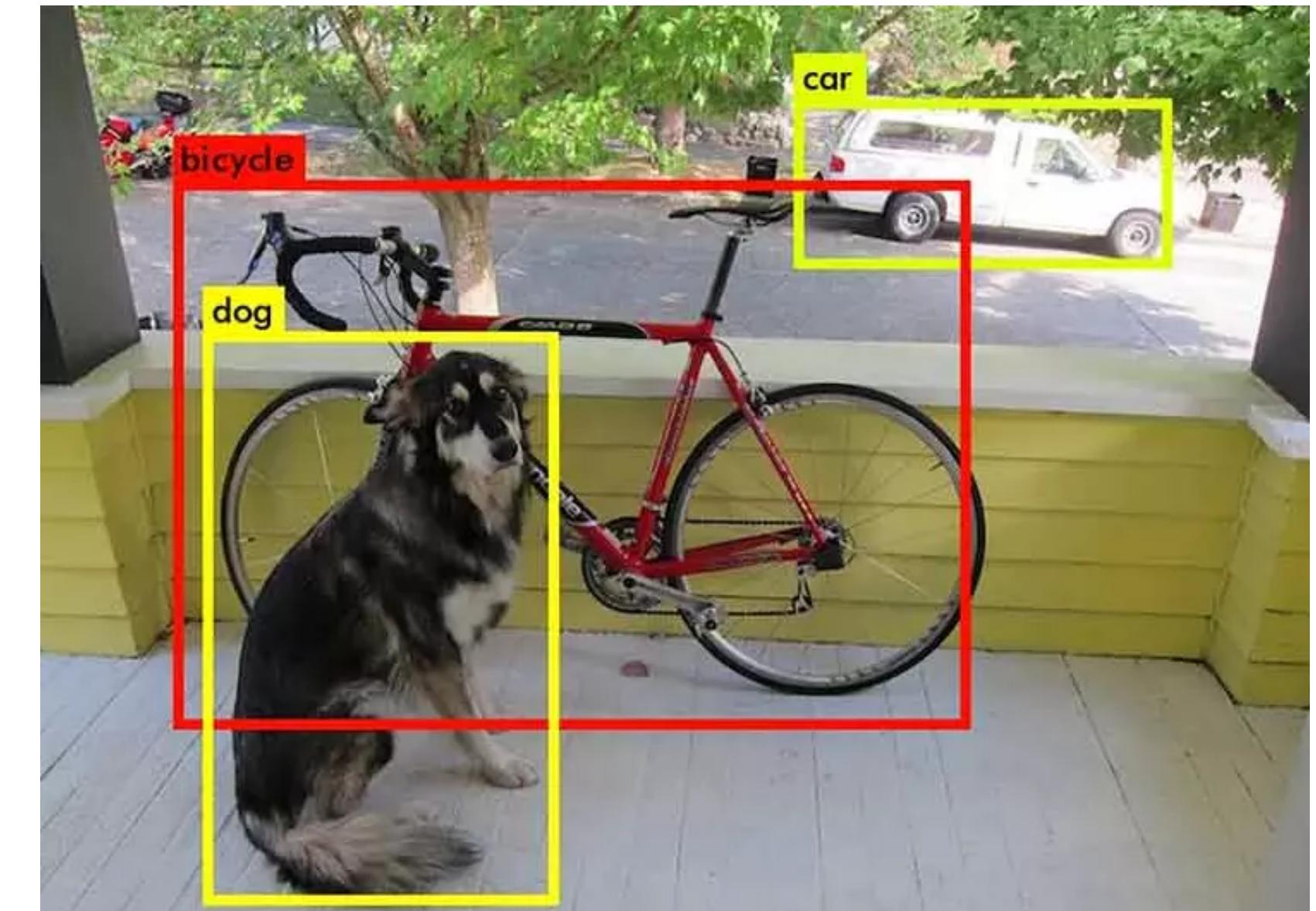
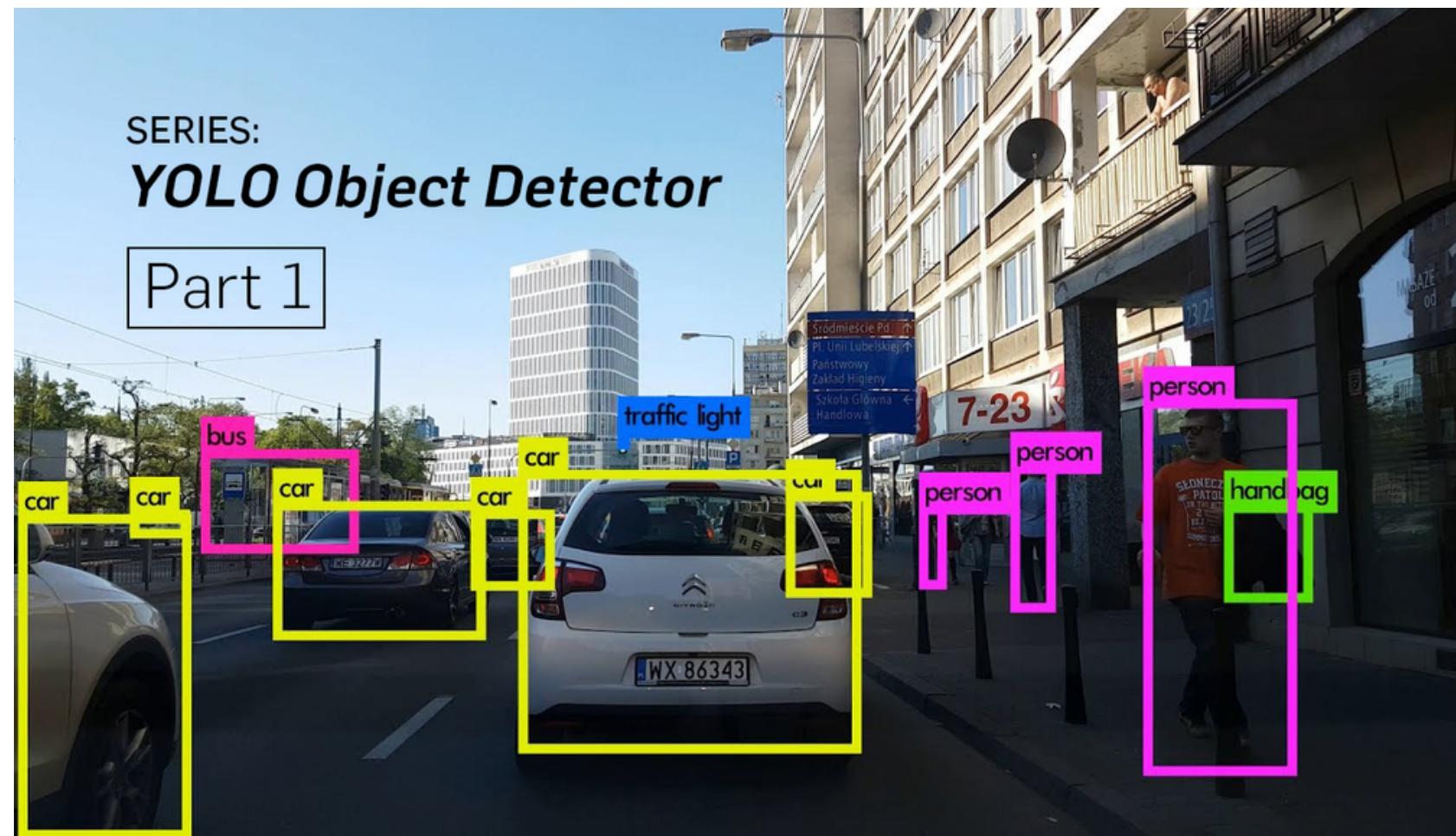
Wi-Fi Client  
**(STATION)**

# TỔNG QUAN VỀ YOLO V3

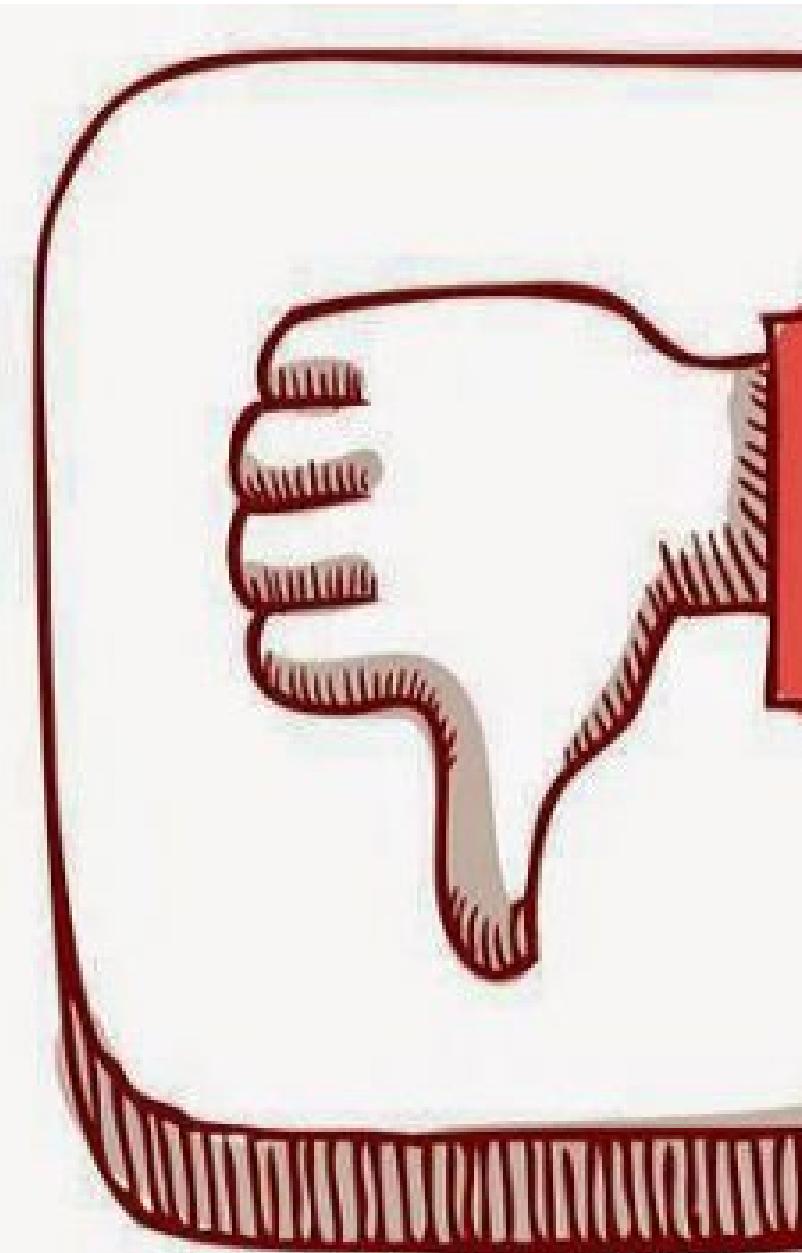


# Nguyên lý hoạt động – những cải tiến

- YOLOv3, phiên bản thứ ba của YOLO, đưa ra nhiều cải tiến đáng kể:
  - + Bộ phát hiện nhiều tỷ lệ: YOLOv3 sử dụng ba kích thước khác nhau của anchor boxes, giúp tăng khả năng phát hiện đối tượng ở nhiều kích thước.
  - + Sử dụng hàm kích hoạt Leaky ReLU: Cải thiện hiệu suất và giảm thiểu vấn đề vanishing gradient.
  - + Phát hiện đối tượng ở ba tỷ lệ khác nhau: Cho phép phát hiện đối tượng từ lớn đến nhỏ trong cùng một hình ảnh.



# ƯU - NHƯỢC ĐIỂM VÌ SAO CHỌN YOLO V3?



# XÂY DỰNG HỆ THỐNG - GIẢI THÍCH CODE



# XÂY DỰNG HỆ THỐNG - GIẢI THÍCH CODE

```
#include <WiFi.h>
#include <esp32cam.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebSrv.h>
```

# Kết nối ESP32-CAM với WIFI

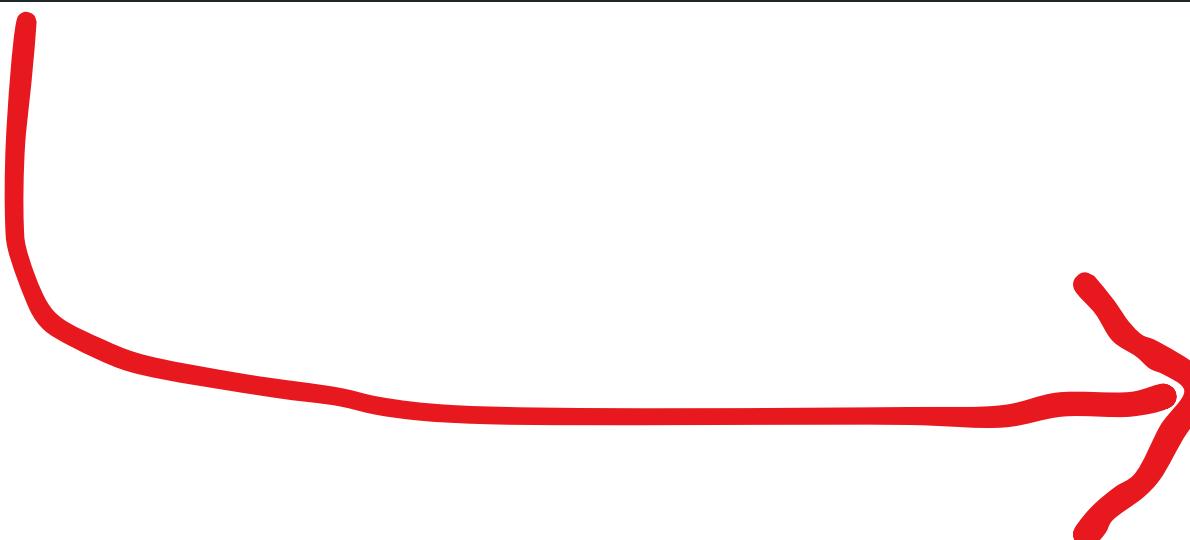
## QUÁ TRÌNH KẾT NỐI



# Kết nối ESP32-CAM với WIFI

## CÁC HÀM CHÍNH

```
void serveJpg(AsyncWebServerRequest *request) {
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("CAPTURE FAIL");
        request->send(503, "image/jpeg", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
                 static_cast<int>(frame->size()));
    AsyncWebServerResponse *response = request->beginResponse_P(200, "image/jpeg", frame->data(), frame->size());
    request->send(response);
}
```



```
void handleJpgLo(AsyncWebServerRequest *request){
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg(request);
}

void handleJpgHi(AsyncWebServerRequest *request){
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
    serveJpg(request);
}

void handleJpgMid(AsyncWebServerRequest *request){
    if (!esp32cam::Camera.changeResolution(midRes)) {
        Serial.println("SET-MID-RES FAIL");
    }
    serveJpg(request);
}
```

# Xử lý video trực tiếp từ ESP32-CAM

## 20127674\_yolov3.ipynb

```
url = 'http://192.168.1.13/cam-hi.jpg'
save_folder = 'C:/Users/dat20/OneDrive/Desktop/20127674/'

while True:
    # Get the image from the Arduino server
    img_resp = urllib.request.urlopen(url)
    imgnp = np.array(bytarray(img_resp.read()), dtype=np.uint8)
    im = cv2.imdecode(imgnp, -1)

def detect_objects(image):
    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    layerOutputs = net.forward(ln)

    boxes = []
    confidences = []
    classIDs = []
    for output in layerOutputs:
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]
            if confidence > 0.5: # bạn có thể thay đổi ngưỡng này
                box = detection[0:4] * np.array([image.shape[1], image.shape[0], image.shape[1], image.shape[0]])
                (centerX, centerY, width, height) = box.astype("int")
                x = int(centerX - (width / 2))
                y = int(centerY - (height / 2))
                boxes.append([x, y, int(width), int(height)])
                confidences.append(float(confidence))
                classIDs.append(classID)

    idxs = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.3)

    if len(idxs) > 0:
        for i in idxs.flatten():
            (x, y) = (boxes[i][0], boxes[i][1])
            (w, h) = (boxes[i][2], boxes[i][3])
            color = [int(c) for c in np.random.randint(0, 255, size=(3,))]
            cv2.rectangle(image, (x, y), (x + w, y + h), color, 2)
            text = "{}: {:.4f}".format(labels[classIDs[i]], confidences[i])
            cv2.putText(image, text, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

# Save the detected image to the folder
timestamp = time.strftime("%Y-%m-%d_%H-%M-%S")
save_path = save_folder + f"detected_{timestamp}.jpg"
cv2.imwrite(save_path, im_with_objects)

# Display the image
cv2.imshow('live Cam Testing', im_with_objects)
```

# Xử lý video trực tiếp từ ESP32-CAM

20127674\_yolov3\_fix.ipynb

```
url = 'http://192.168.1.13/cam-hi.jpg'
save_folder = 'C:/Users/dat20/OneDrive/Desktop/20127674/'

def fetch_frames(queue):
    while True:
        img_resp = urllib.request.urlopen(url)
        imgnp = np.array(bytarray(img_resp.read()), dtype=np.uint8)
        im = cv2.imdecode(imgnp, -1)
        queue.put(im)
        time.sleep(0.02) # Sleep for a short duration before fetching the next frame

frame_queue = Queue(maxsize=2)
threading.Thread(target=fetch_frames, args=(frame_queue,), daemon=True).start()
|
frame_skip = 5 # Run detection every 5 frames
frame_count = 0

while True:
    frame = frame_queue.get()
    frame_count += 1

    if frame_count % frame_skip == 0:
        im_with_objects = detect_objects(frame)

        # Check if any objects were detected (by checking changes in the image)
        if not np.array_equal(frame, im_with_objects):
            timestamp = time.strftime("%Y-%m-%d_%H-%M-%S")
            save_path = save_folder + f"detected_{timestamp}.jpg"
            cv2.imwrite(save_path, im_with_objects)

            cv2.imshow('live Cam Testing', im_with_objects)
    else:
        cv2.imshow('live Cam Testing', frame)

    key = cv2.waitKey(5)
    if key == ord('q'):
        break

cv2.destroyAllWindows()
```

- Đa luồng (Threading):

+ Trong đoạn mã mới, sử dụng đa luồng để thu thập hình ảnh từ ESP32-CAM. Hàm `fetch_frames` được thiết kế để thu thập các khung hình và đặt chúng vào một hàng đợi (`frame_queue`).

+ Một luồng riêng biệt (`threading.Thread`) được bắt đầu để chạy hàm `fetch_frames` một cách đồng thời với chính chương trình chính.

- Hàng đợi (Queue)

- Bỏ qua một số khung hình

- Kiểm tra sự khác biệt giữa hình ảnh  
Những khác biệt này làm cho model mới hiệu quả và linh hoạt hơn. Sử dụng đa luồng cho phép thu thập hình ảnh mà không làm chậm việc xử lý và hiển thị hình ảnh. Việc bỏ qua một số khung hình giảm bớt tải xử lý và tăng tốc độ chương trình.

# Phát triển và huấn luyện mô hình YOLOv3:

## Xây dựng hàm nhận dạng đối tượng

Hàm detect\_objects(image) được xây dựng để thực hiện việc nhận diện đối tượng

trên một hình ảnh đầu vào.

Hàm này tiến hành các bước sau:

Chuyển hình ảnh đầu vào thành blob.

Thiết lập blob làm đầu vào cho mô hình và tiến hành phát hiện.

Lọc và lấy ra những đối tượng được phát hiện dựa trên mức độ tin cậy (confidence).

Áp dụng Non-Maximum Suppression để loại bỏ những đối tượng bị trùng lặp.

Vẽ hình chữ nhật và nhãn cho mỗi đối tượng được phát hiện trên hình ảnh.

Xử lý video trực tiếp:

Em sử dụng URL <http://192.168.1.13/cam-hi.jpg> để lấy hình ảnh từ ESP32-CAM. Trong mỗi vòng lặp: Lấy hình ảnh từ ESP32-CAM.

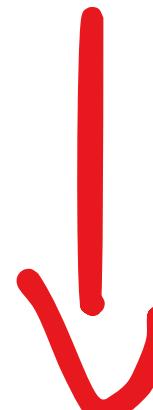
Sử dụng hàm detect\_objects để nhận diện đối tượng trên hình ảnh.

Lưu hình ảnh đã phát hiện đối tượng với một dấu thời gian duy nhất.

Hiển thị hình ảnh đó trực tiếp.

Đặt  
môi

trường  
và cấu hình



Tải - load  
mô hình  
YOLOv3

# KẾT QUẢ



# THẢO LUẬN





Kết luận

# TƯ LIỆU THAM KHẢO

[https://www.hackster.io/mjrobot/esp32-cam-tinyml-image-classification-fruits-vs-veggies-4ab970.](https://www.hackster.io/mjrobot/esp32-cam-tinyml-image-classification-fruits-vs-veggies-4ab970)

[https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/.](https://how2electronics.com/esp32-cam-based-object-detection-identification-with-opencv/)

[https://www.youtube.com/watch?v=npJsmbFZiMg.](https://www.youtube.com/watch?v=npJsmbFZiMg)

[https://www.youtube.com/watch?v=A1SPJSVra9I.](https://www.youtube.com/watch?v=A1SPJSVra9I)

[https://www.youtube.com/watch?v=hrUomPXdYZg.](https://www.youtube.com/watch?v=hrUomPXdYZg)

[https://forum.arduino.cc/t/problems-uploading-to-esp32-camera-with-arduino-ide-solved/1042295.](https://forum.arduino.cc/t/problems-uploading-to-esp32-camera-with-arduino-ide-solved/1042295)

# THANK'S FOR WATCHING

Contact: dat20026969@gmail.com

Sống cho hết đời thanh xuân để trọn  
vẹn hết những thứ mà ta cần.



**fit@hcmus**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG - HCM  
**KHOA CÔNG NGHỆ THÔNG TIN**

