REAL-TIME OBJECTS DETECTION FOR A WARNING SYSTEM

GVHD: Trần Văn Lộc

Group 13

Our Team

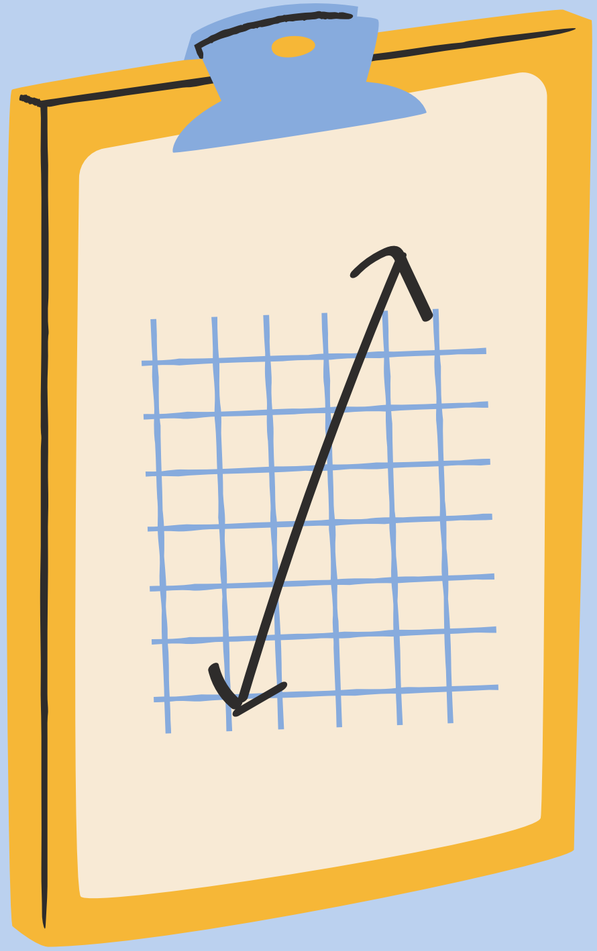
Hồng Thư

Thành Nhân

Khôi Nguyên

Đức Tiến

TABLE OF CONTENTS



- Giới thiệu dự án
- Cơ sở lý thuyết
- Xây dựng mô hình
- Đánh giá

GIỚI THIỆU DỰ ÁN

ABOUT PROJECT

- **Tổng quan và mục tiêu dự án:** Dự án này tập trung vào việc phát hiện và cảnh báo về vật thể xâm nhập vào camera. Mục tiêu là phát triển một hệ thống tự động có khả năng nhận diện các vật thể và thông báo ngay lập tức qua Telegram và Gmail để cảnh báo người dùng

- **Phạm vi dự án:** Dự án của nhóm tập trung vào việc thực hành nhận diện vật thể trên các class đã được định nghĩa sẵn trong mô hình YOLOv4.

- **Các bước thực hiện:**

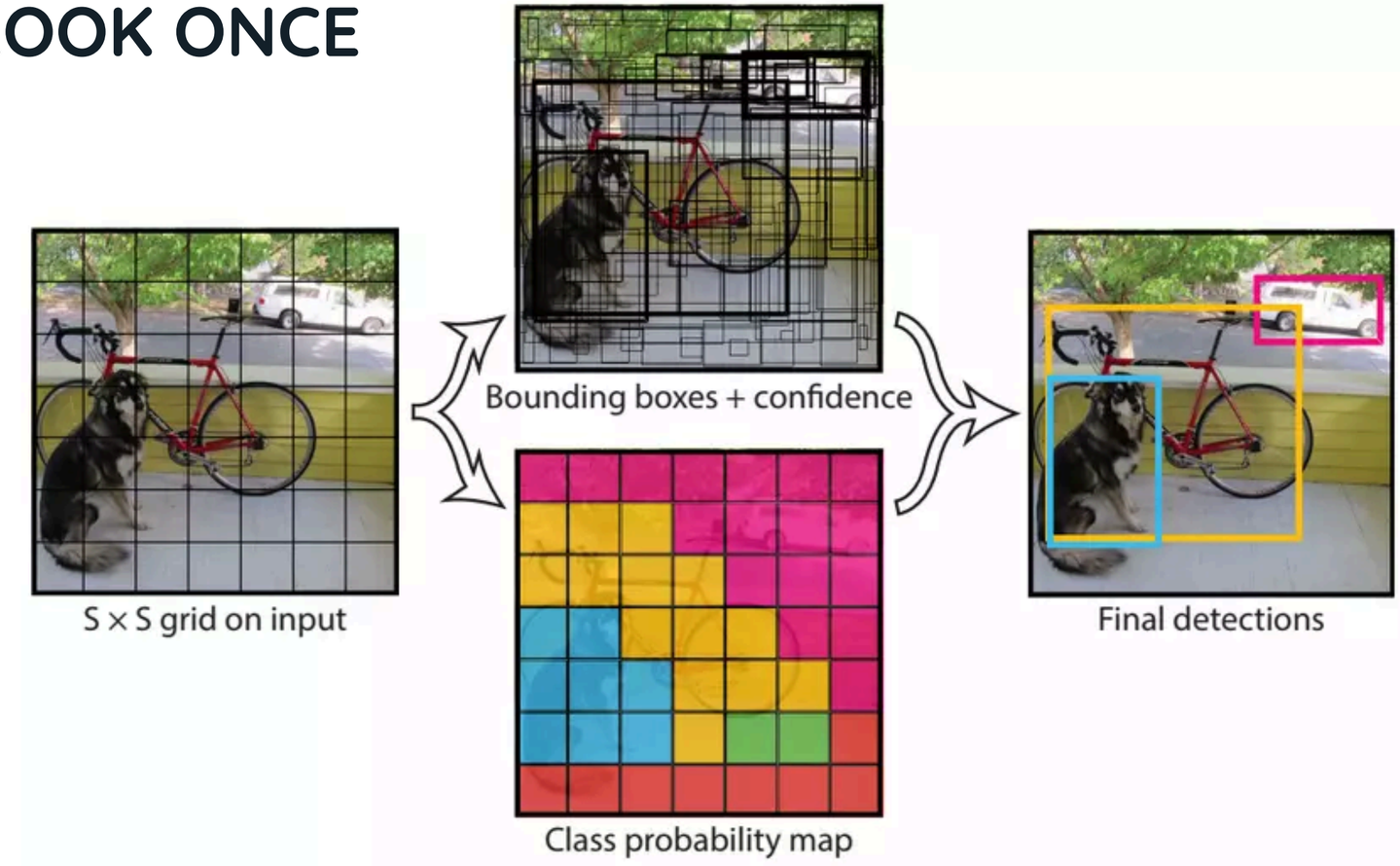
1. Nghiên cứu và phát triển mô hình AI để phát hiện vật thể.
2. Phát triển tính năng kết nối và thông báo qua Telegram và Gmail đồng thời cảnh báo bằng âm thanh.
3. Triển khai và duy trì hệ thống.

- **Công nghệ và phương pháp:** Sử dụng ngôn ngữ lập trình và framework phù hợp như Python và OpenCV, sử dụng API của Telegram và Gmail để gửi thông báo.

CƠ SỞ LÝ THUYẾT

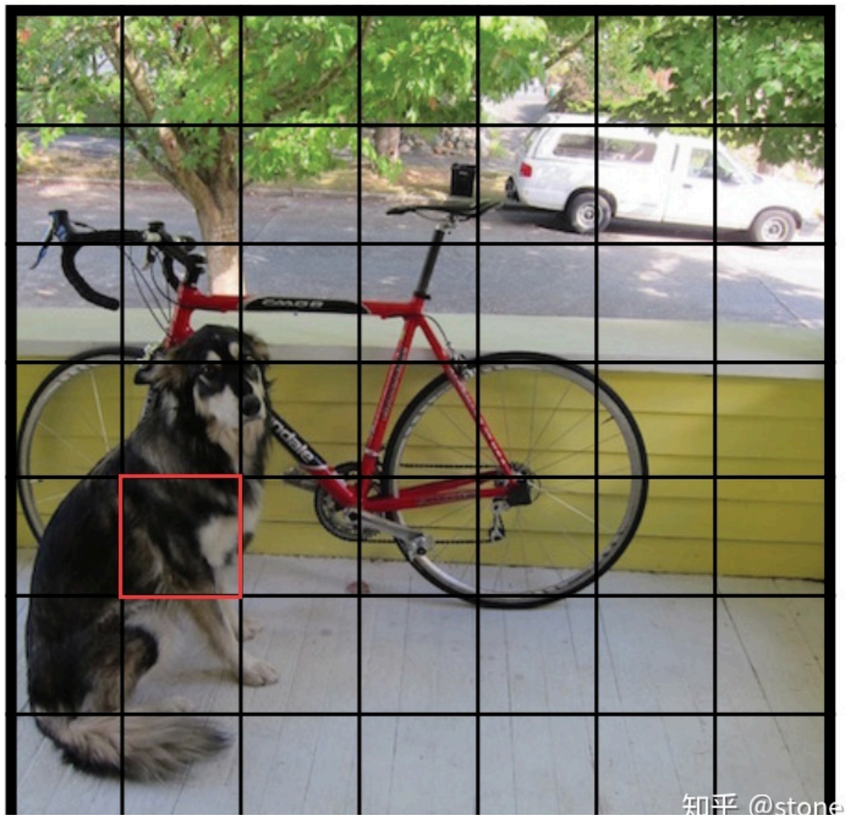
MÔ HÌNH YOLO V4

- **YOLO - YOU ONLY LOOK ONCE**
- **Gồm :**
  - Grid System
  - Bounding box
  - CNN



MÔ HÌNH YOLO V4

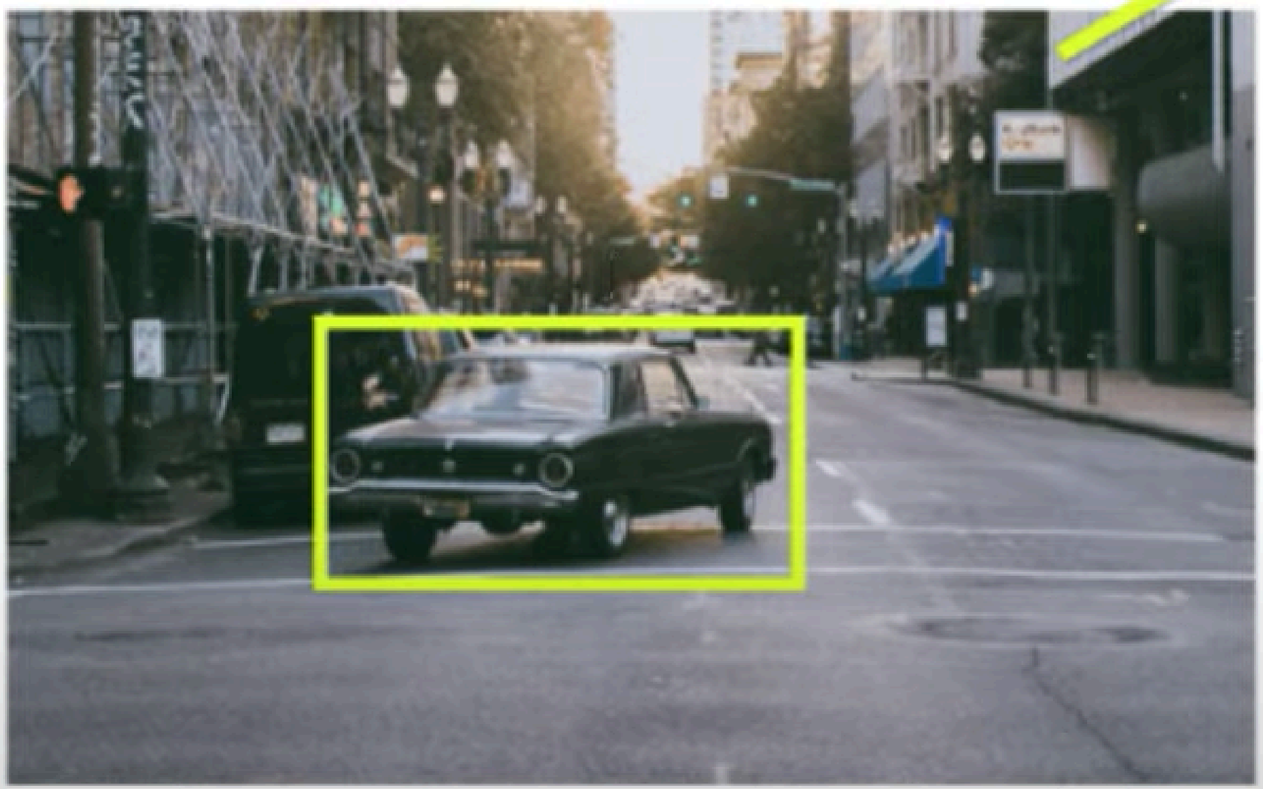
- **Grid system**
  - Chia ảnh gốc thành S x S
  - Mỗi ô lưới dự đoán 1 đối tượng
  - Kích thước ảnh là bội số của grid size

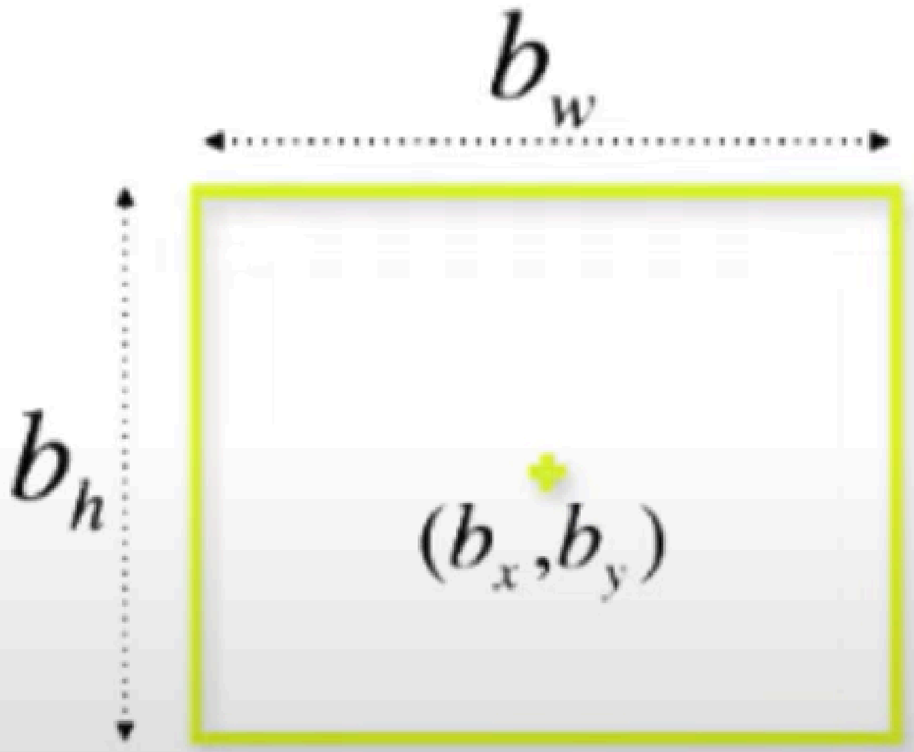


MÔ HÌNH YOLO V4

- Bounding box

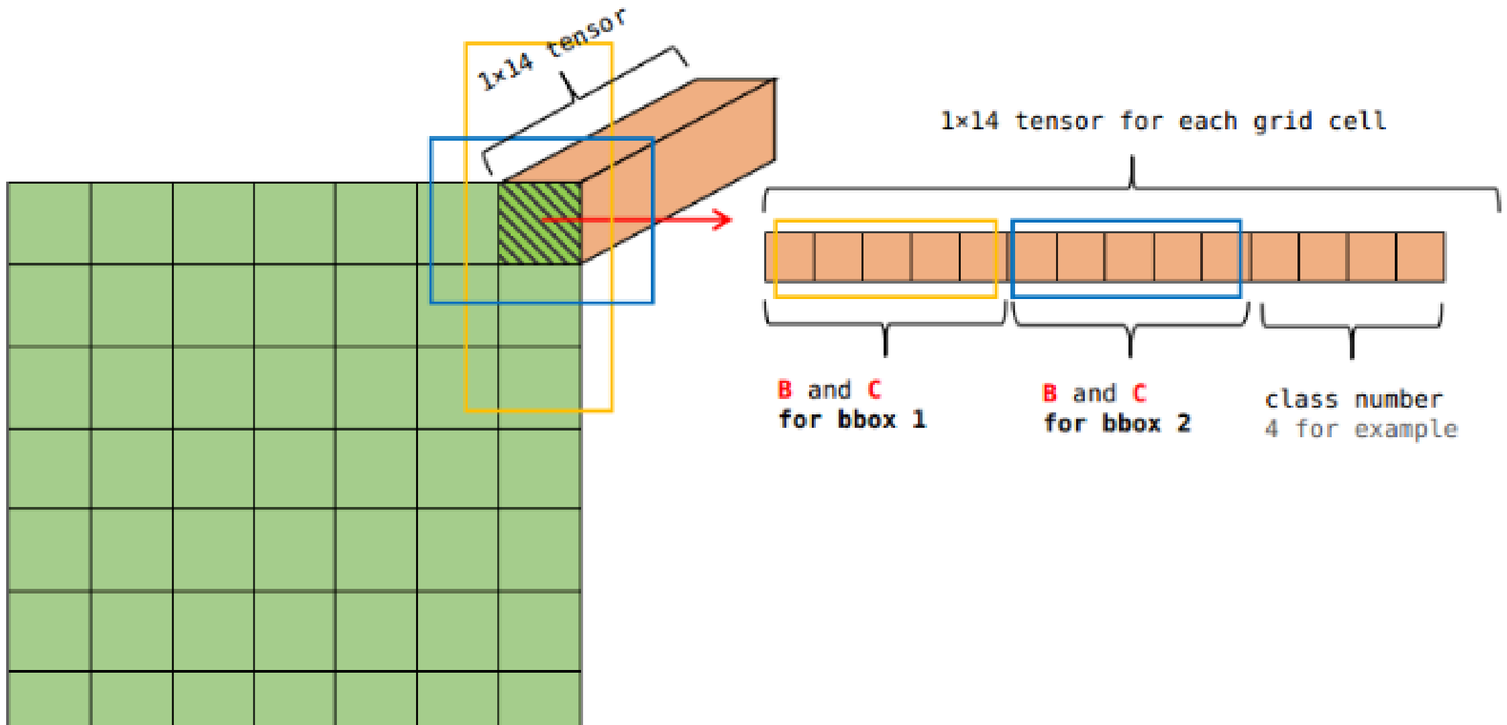
$$y = (p_c, b_x, b_y, b_h, b_w, c)$$

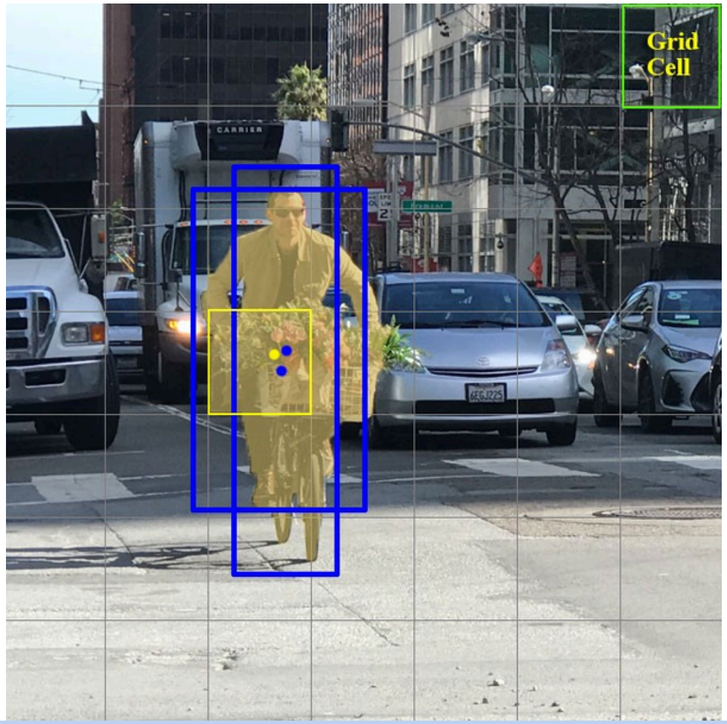




MÔ HÌNH YOLO V4

- **Bounding box**
  - Mỗi ô lưới dự đoán các ô biên => bounding box của đối tượng (hộp ranh giới).
  - Bounding box: Vị trí ở đâu và có chứa đối tượng không?
  - Số chiều output: **5*n_box + number_classes**

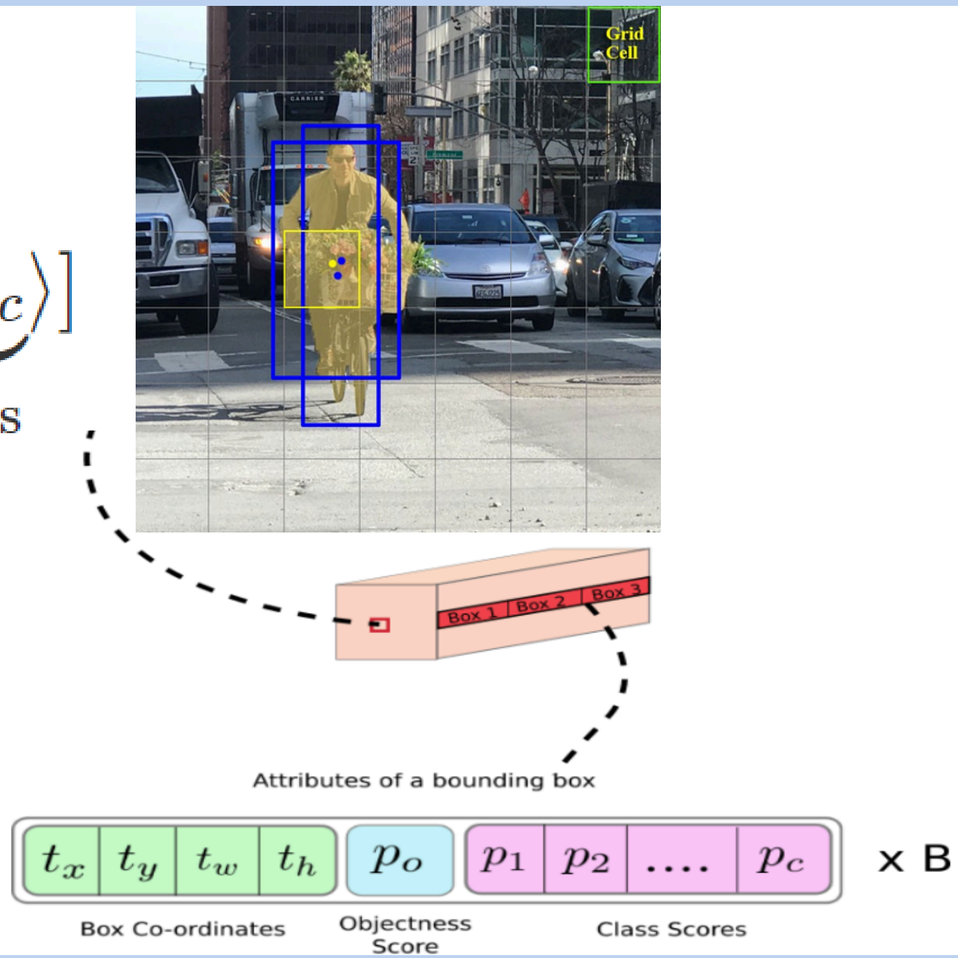




MÔ HÌNH YOLO V4

- Output YOLO:

$$y^T = [p_0, \langle \underbrace{t_x, t_y, t_w, t_h}_{\text{bounding box}} \rangle, \langle \underbrace{p_1, p_2, \ldots, p_c}_{\text{scores of c classes}} \rangle]$$



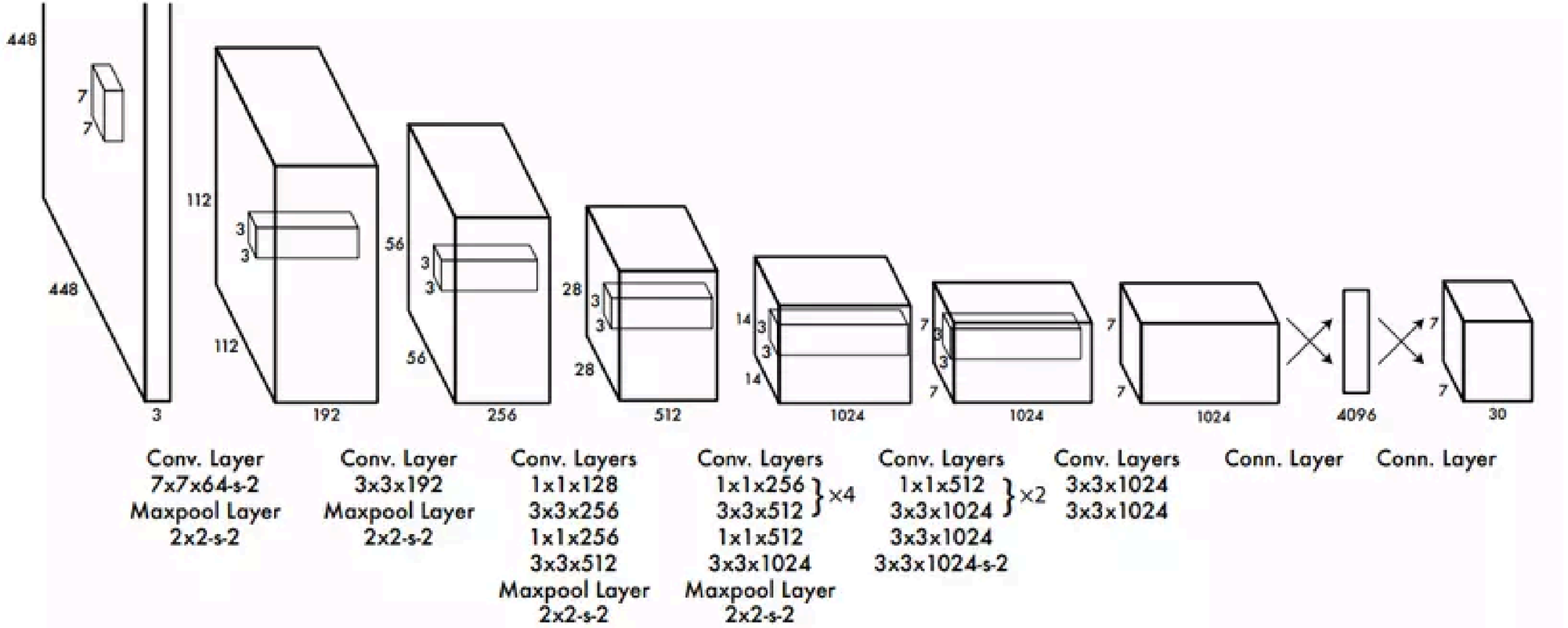
MÔ HÌNH YOLO V4

- Hàm tính IOU

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

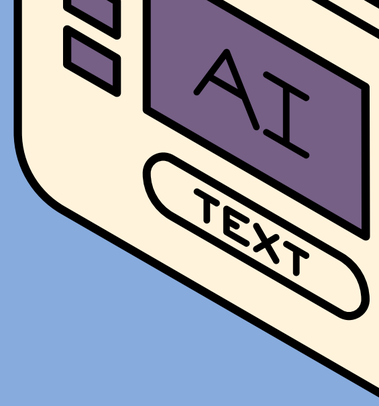
MÔ HÌNH YOLO V4

- **Kiến trúc CNN**
  - Sử dụng hàm kích hoạt linear regression => dự đoán thông tin ở mỗi ô vuông
  - Layer cuối cùng không sử dụng hàm kích hoạt



XÂY DỰNG MÔ HÌNH

Cách thức hoạt động



Đọc ảnh từ Webcam và hiển thị khung hình

Vẽ đa giác xác định khu vực cấm

Xác định vật thể cần phát hiện

Phát hiện vật thể dựa vào YOLO + Xác định toạ độ

Kiểm tra tọa độ của vật thể (nằm trong khu vực cấm không)

Hiển thị cảnh báo và gửi qua Telegram và Gmail

MAIN

- **Sử dụng OpenCV**

DetectObject > main.py > ...

```python
import cv2
import numpy as np
from imutils.video import VideoStream
from yolodetect import YoloDetect

video = VideoStream(src=0).start()
points = []


model = YoloDetect()

def handle_left_click(event, x, y, flags, points):
    if event == cv2.EVENT_LBUTTONDOWN:
        points.append([x, y])

def draw_polygon(frame, points):
    for point in points:
        frame = cv2.circle(frame, (point[0], point[1]), 5, (0,0,255), -1)

    frame = cv2.polylines(frame, [np.int32(points)], False, (255,0,0),thickness=2)
    return frame
detect = False
while True:
    frame=video.read()
    frame = cv2.flip(frame,1)
    #Ve polygon
    frame= draw_polygon(frame,points)
    if detect:
        frame = model.detect(frame= frame , points= points)
    key = cv2.waitKey(1)
    if key == ord('q'):
        break
    elif key ==ord('d'):
        points.append(points[0])
        detect = True
    #Hien anh ra man hifnh
    cv2.imshow("Intrusion Warning", frame)
    cv2.setMouseCallback('Intrusion Warning',handle_left_click,points)
video.stop()
cv2.destroyAllWindows()
```

- **Hàm trợ giúp vẽ phạm vi lên camera**

```python
def handle_left_click(event, x, y, flags, points):
    if event == cv2.EVENT_LBUTTONDOWN:
        points.append([x, y])

def draw_polygon(frame, points):
    for point in points:
        frame = cv2.circle(frame, (point[0], point[1]), 5, (0,0,255), -1)

    frame = cv2.polylines(frame, [np.int32(points)], False, (255,0,0),thickness=2
    return frame
```

YOLODETECT

- **Thư viện Shapely:**

```python
from shapely.geometry import Point
from shapely.geometry.polygon import Polygon
```

```python
def isInside(points, corners):
    polygon = Polygon(points)
    for corner in corners:
        centroid = Point(corner
        print(polygon.contains(centroid))
    return any(polygon.contains(Point(corner)) for corner in corners)
```

(variable) centroid: Any | Point

- **Một số hàm khác:**

```python
def alert(self, img):
    cv2.putText(img, "ALARM!!!!", (10, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

    # New thread to send telegram after 15 seconds
    if (self.last_alert is None) or (
            (datetime.datetime.utcnow() - self.last_alert).total_seconds() > self.alert_telegram_each):
        self.last_alert = datetime.datetime.utcnow()
        cv2.imwrite("alert.png", cv2.resize(img, dsize=None, fx=0.2, fy=0.2))
        asyncio.run(send_telegram("alert.png"))
        asyncio.run(send_email_with_image("Alert", "Có kẻ trộm đột nhập, nguy hiểm!", "khoinguyen09102004@gmail.com", "alert.png")
    return img
```

```python
def draw_prediction(self, img, class_id, x, y, x_plus_w, y_plus_h, points):
    label = str(self.classes[class_id])
    color = (0, 255, 0)
    cv2.rectangle(img, (x, y), (x_plus_w, y_plus_h), color, 2)
    cv2.putText(img, label, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    # Tính toán centroid

    top_left = (x, y)
    top_right = (x_plus_w, y)
    bottom_left = (x, y_plus_h)
    bottom_right = (x_plus_w, y_plus_h)

    cv2.circle(img, (x,y),5,(0,0,255),-1)
    cv2.circle(img, (x_plus_w,y),5,(0,0,255),-1)
    cv2.circle(img, (x,y_plus_h),5,(0,0,255),-1)
    cv2.circle(img, (x_plus_w,y_plus_h),5,(0,0,255),-1)

    if isInside(points, (top_left, top_right, bottom_right, bottom_left)):
        img = self.alert(img)

    return isInside(points, (top_left, top_right, bottom_right, bottom_left))
```

```python
def detect(self, frame, points):
    blob = cv2.dnn.blobFromImage(frame, self.scale, (416, 416), (0, 0, 0), True, crop=False)
    self.model.setInput(blob)
    outs = self.model.forward(self.output_layers)
    # Loc cac object trong khung hinh
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if (confidence >= self.conf_threshold) and (self.classes[class_id] == self.detect_class):
                center_x = int(detection[0] * self.frame_width)
                center_y = int(detection[1] * self.frame_height)
                w = int(detection[2] * self.frame_width)
                h = int(detection[3] * self.frame_height)
                x = center_x - w / 2
                y = center_y - h / 2
                class_ids.append(class_id)
                confidences.append(float(confidence))
                boxes.append([x, y, w, h])
    indices = cv2.dnn.NMSBoxes(boxes, confidences, self.conf_threshold, self.nms_threshold)
```

DATASET

Search

KONSTANTIN VERNER · UPDATED 2 YEARS AGO

152

Human Detection Dataset

CCTV footage of humans

Data Card | Code (2) | Discussion (2) | Suggestions (0)

About Dataset

Context

It is important to detect humans on footage of CCTV, so, let us use this dataset to train a neural network to

Content

see 'footage' folder

Dataset contains CCTV footage images(as indoor as outdoor), a half of them w humans and a half of them is
follow:

0_n.png or 1_n.png

the first digit is a class of image, 0 means a scene without humans, and 1 means a scene with humans.

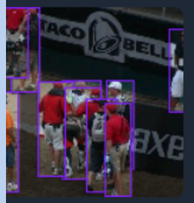
n is just a number of an image in the whole dataset

Sources of dataset:

1) cctv footage from youtube;

roboflow | Workspace | Universe | Documentation | Forum

h 500,000+ Open Source Computer Vision Projects...



le Detection

etection

erview

ges 7261

taset 3

del

Docs

alth Check

Roboflow Universe > Leo Ueno > **People Detection**

20 People Detection Computer Vision Project

TRY THIS MODEL

Drop an image or

browse your device

Images

7261 images

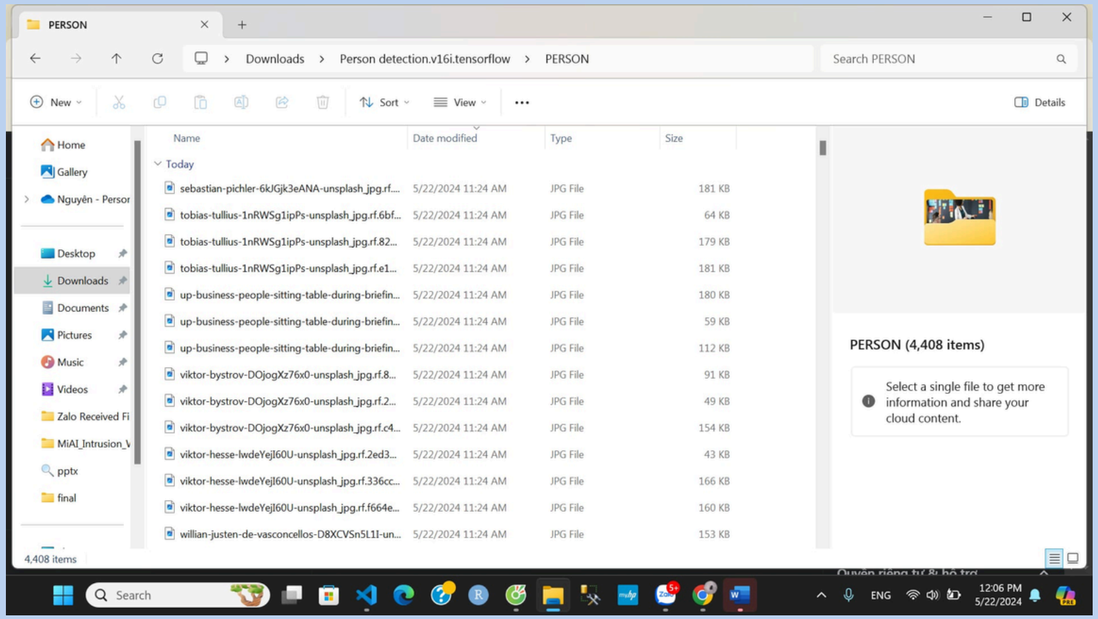
Detect people in all kinds of scenarios using this generalized dataset.

For this dataset, we curated images and annotations from various projec

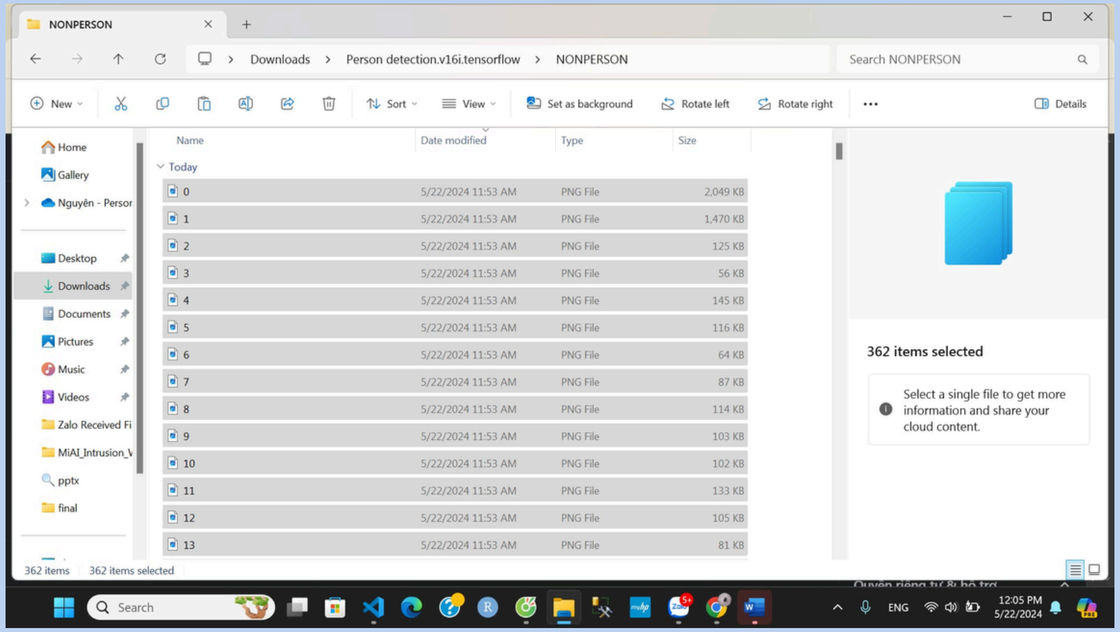
DATASET

Số lượng hình ảnh: 5.074

Person: 4712



Non-Person: 362



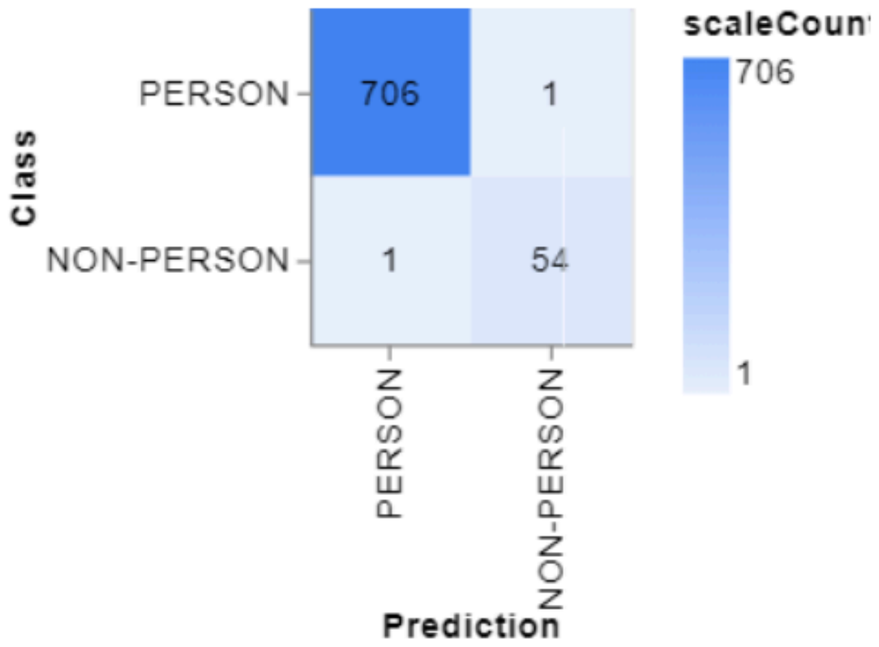
ĐÁNH GIÁ KẾT QUẢ VỀ MÔ HÌNH

Accuracy per class

CLASS	ACCURACY	# SAMPLES
PERSON	1.00	707
NON-PERSON	0.98	55

Precision	0.9985
Recall	0.9985

Confusion Matrix



TELEGRAM

```python
# telegram_utils.py
import telegram
import asyncio

async def send_telegram(photo_path="alert.png"):
    try:
        my_token = "7094401597:AAEbWC3xBgG0mMS0Xv9zcauR3053z9R6Kbw"
        bot = telegram.Bot(token=my_token)
        with open(photo_path, "rb") as photo:
            await bot.sendPhoto(chat_id="6183307421", photo=photo, caption="Có xâm nhập, nguy hiểm!")
        print("Send success")
    except telegram.error.TimedOut as e:
        print("Request timed out:", e)
    except Exception as ex:
        print("Can not send message telegram", ex)
```

EMAIL

```python
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from email.mime.text import MIMEText
import asyncio

async def send_email_with_image(subject, body, to_email, image_path, sender_email="your_email@gmail.com", sender_password="your_password"):
    try:
        # Thông tin tài khoản email của bạn
        sender_email = "pthu7325@gmail.com"
        sender_password = "jjrk impv sptg bghz"

        # Đọc nội dung của hình ảnh
        with open(image_path, "rb") as image_file:
            image_data = image_file.read()

        # Tạo một email đa phần
        msg = MIMEMultipart()
        msg["From"] = sender_email
        msg["To"] = to_email
        msg["Subject"] = subject

        # Thêm nội dung văn bản
        msg.attach(MIMEText(body, "plain"))

        # Thêm hình ảnh như một phần của email
        image = MIMEImage(image_data, name="alert.png")
        msg.attach(image)

        # Kết nối đến máy chủ SMTP của Gmail
        server = smtplib.SMTP("smtp.gmail.com", 587)
        server.starttls()
        server.login(sender_email, sender_password)

        # Gửi email
        server.sendmail(sender_email, to_email, msg.as_string())

        # Đóng kết nối
        server.quit()
        print("Send successssssssssss")
    except Exception as ex:
        print("Can not send email:", ex)
```

CHẠY DEMO

ĐÁNH GIÁ

ĐÁNH GIÁ

ƯU ĐIỂM

- Tốc độ xử lí nhanh
- Độ chính xác cao
- Nhiều loại đối tượng cùng lúc
- Có thể tuỳ chỉnh đối tượng linh hoạt

NHƯỢC ĐIỂM

- Yêu cầu phần cứng cao
- Khả năng phát hiện đối tượng nhỏ, che khuất hạn chế.
- Chưa thực nghiệm trên thiết bị rời

ỨNG DỤNG

Phát hiện kẻ xâm nhập

Cảnh báo nguy hiểm

Hỗ trợ xe tự lái

THANK YOU