Các bước cài đặt :

Phần 1: công cụ chuẩn bị

<https://www.makesense.ai/> : công cụ cắt ảnh và dán nhãn label

googlecolab : công cụ train data

visual studio code

data ảnh khói và lửa

Phần 2 : chuẩn bị data

Bộ dữ liệu ảnh lửa và khói chúng ta cần thu thập trên mạng , càng nhiều càng tốt , tối thiểu 1000 tấm

A screenshot of a computer

Description automatically generated

Xử lý dữ liệu

Chúng ta sẽ cắt ảnh bằng makesence.ai

A screenshot of a computer screen

Description automatically generated

Xuất file có định dạng như dưới

A screenshot of a computer

Description automatically generated

Tạo file data có cấu trúc thư mục như dưới , sau đó bỏ image và label chúng ta đã chuẩn bị ở trên để tiến hành train

A screenshot of a chat

Description automatically generated

Phần 4: training data

Cài đặt thư viện : !pip install ultralytics

# Download pretrain model Yolov8n - Nhanh

!wget https://github.com/ultralytics/assets/releases/download/v0.0.0/yolov8n.pt

# Create training yaml file

%cd /content/drive/MyDrive

!echo 'train: /content/drive/MyDrive/data/train' >> data/mydataset.yaml

!echo 'val: /content/drive/MyDrive/data/train' >> data/mydataset.yaml

!echo 'nc: 2' >> data/mydataset.yaml

!echo "names: ['lửa\_cháy','Khói']" >> data/mydataset.yaml

# Train với CLI

!yolo task=detect mode=train model=yolov8n.pt data=data/mydataset.yaml epochs=120 imgsz=640

Sau khi train file train sẽ được lưu tại

Results saved to **runs/detect/train5**

**Chúng ta sẽ lấy file đã được huấn luyện thường có tên best.pt**

**Đưa vào visual studio code để tiến hành sử dụng**

*from* ultralytics *import* YOLO

*import* cv2

*import* math

*import* pyaudio

*import* wave

*import* threading

*import* cvzone

*import* time

*# Hàm để phát âm thanh cảnh báo*

def play\_notification\_sound():

    global notification\_playing

    CHUNK = 1024

    FORMAT = pyaudio.paInt16

    CHANNELS = 2

    RATE = 44100

    p = pyaudio.PyAudio()

    wf = wave.open("baodong.wav", 'rb')

    stream = p.open(*format*=p.get\_format\_from\_width(wf.getsampwidth()),

*channels*=wf.getnchannels(),

*rate*=wf.getframerate(),

*output*=True)

    data = wf.readframes(CHUNK)

*while* data:

*if* not notification\_playing:

*break*  *# Dừng nếu cảnh báo đã ngừng*

        stream.write(data)

        data = wf.readframes(CHUNK)

    stream.stop\_stream()

    stream.close()

    p.terminate()

*# Hàm để dừng phát âm thanh cảnh báoq*

def stop\_notification\_sound():

    global notification\_playing

    notification\_playing = False

*# Hàm để nhận diện và báo động chạy song song*

def detect\_and\_alert(*cap*, *model*):

    global notification\_playing

    is\_fire\_smoke\_detected = False  *# Biến để theo dõi xem có lửa hoặc khói được phát hiện không*

*while* True:

        ret, frame = *cap*.read()

*if* not ret:

*break*  *# Kết thúc nếu không còn khung hình nào được đọc được nữa*

        frame = cv2.resize(frame, (640, 480))

        result = *model*(frame, *stream*=True)

        fire\_smoke\_detected\_this\_iteration = False  *# Biến để theo dõi xem có lửa hoặc khói được phát hiện trong khung hình này không*

*for* info *in* result:

            boxes = info.boxes

*for* box *in* boxes:

                confidence = box.conf[0]

                confidence = math.ceil(confidence \* 100)

                Class = int(box.cls[0])

*if* confidence > 50:

                    x1, y1, x2, y2 = box.xyxy[0]

                    x1, y1, x2, y2 = int(x1), int(y1), int(x2), int(y2)

                    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 5)

                    cvzone.putTextRect(frame, f'{classnames[Class]} {confidence}%', [x1 + 8, y1 + 100],

*scale*=1.5, *thickness*=2)

                    fire\_smoke\_detected\_this\_iteration = True  *# Đặt cờ báo hiệu rằng đã phát hiện được lửa hoặc khói*

*# Kiểm tra xem có phát hiện lửa hoặc khói không*

*if* fire\_smoke\_detected\_this\_iteration:

            is\_fire\_smoke\_detected = True

*# Chỉ phát cảnh báo nếu cảnh báo hiện tại đã kết thúc*

*if* not notification\_playing:

                notification\_playing = True

                alert\_thread = threading.Thread(*target*=play\_notification\_sound)

                alert\_thread.start()

*else*:

            is\_fire\_smoke\_detected = False

            stop\_notification\_sound()

        cv2.imshow('frame', frame)

*if* cv2.waitKey(1) & 0xFF == ord('q'):

*break*

*# Nếu không còn lửa hoặc khói được phát hiện, tắt âm thanh cảnh báo*

*if* not is\_fire\_smoke\_detected:

        stop\_notification\_sound()

*# Khởi tạo webcam*

cap = cv2.VideoCapture(0)  *# Sử dụng webcam, 0 là số ID của webcam, nếu có nhiều webcam, bạn có thể thử các số khác nhau*

*# Khởi tạo mô hình YOLO*

model = YOLO('best.pt')

*# Đọc các lớp*

classnames = ['FIRE', 'smoqqke']

*# Biến để theo dõi trạng thái của âm thanh cảnh báo*

notification\_playing = False

*# Khởi động một luồng riêng để nhận diện và báo động chạy song song*

detection\_thread = threading.Thread(*target*=detect\_and\_alert, *args*=(cap, model))

detection\_thread.start()

*# Chờ cho đến khi luồng nhận diện và báo động kết thúc trước khi giải phóng tài nguyên*

detection\_thread.join()

*# Giải phóng tài nguyên và đóng cửa sổ*

cap.release()

cv2.destroyAllWindows()

trên là code nhận diện ngọn lửa và phát thông báo cháy ngọn lửa , để có thể sử dụng trên các nền tảng web hay platform khác , chỉ cần gom lại tạo thành API KEY để sử dụng , rất đơn giản