

BÁO CÁO ĐỒ ÁN KỸ THUẬT MÁY TÍNH

ĐỀ TÀI: HỆ THỐNG BÁO CHÁY TỰ ĐỘNG

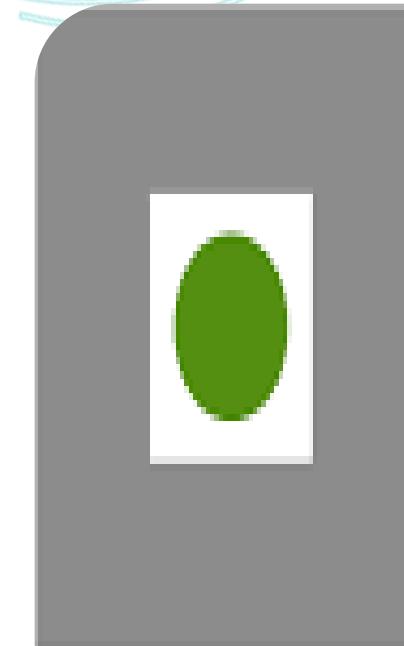
Thực hiện: Nhóm 11

- Lê Hồng Anh - 102200008
- Lê Đức Huy - 102200020
- Huỳnh Bá Thuận - 102200034
- Hồ Ngọc Tùng - 102200038

ĐỒNG HƯỚNG DẪN



Giảng viên hướng dẫn:
TS. Huỳnh Hữu Hưng



enclave
— Reliable Engineering —
— Kỹ thuật надежности —

Anh Vương Nhật Quang (Kerry)
Doanh nghiệp: Enclave

NỘI DUNG TRÌNH BÀY

①

Tổng quan đề tài

②

Tiến trình công việc

③

Kết luận



Tổng quan đề tài

Giới thiệu về mục đích
và giải pháp giải quyết
bài toán



Mục đích

Báo cháy tự động và nhanh chóng

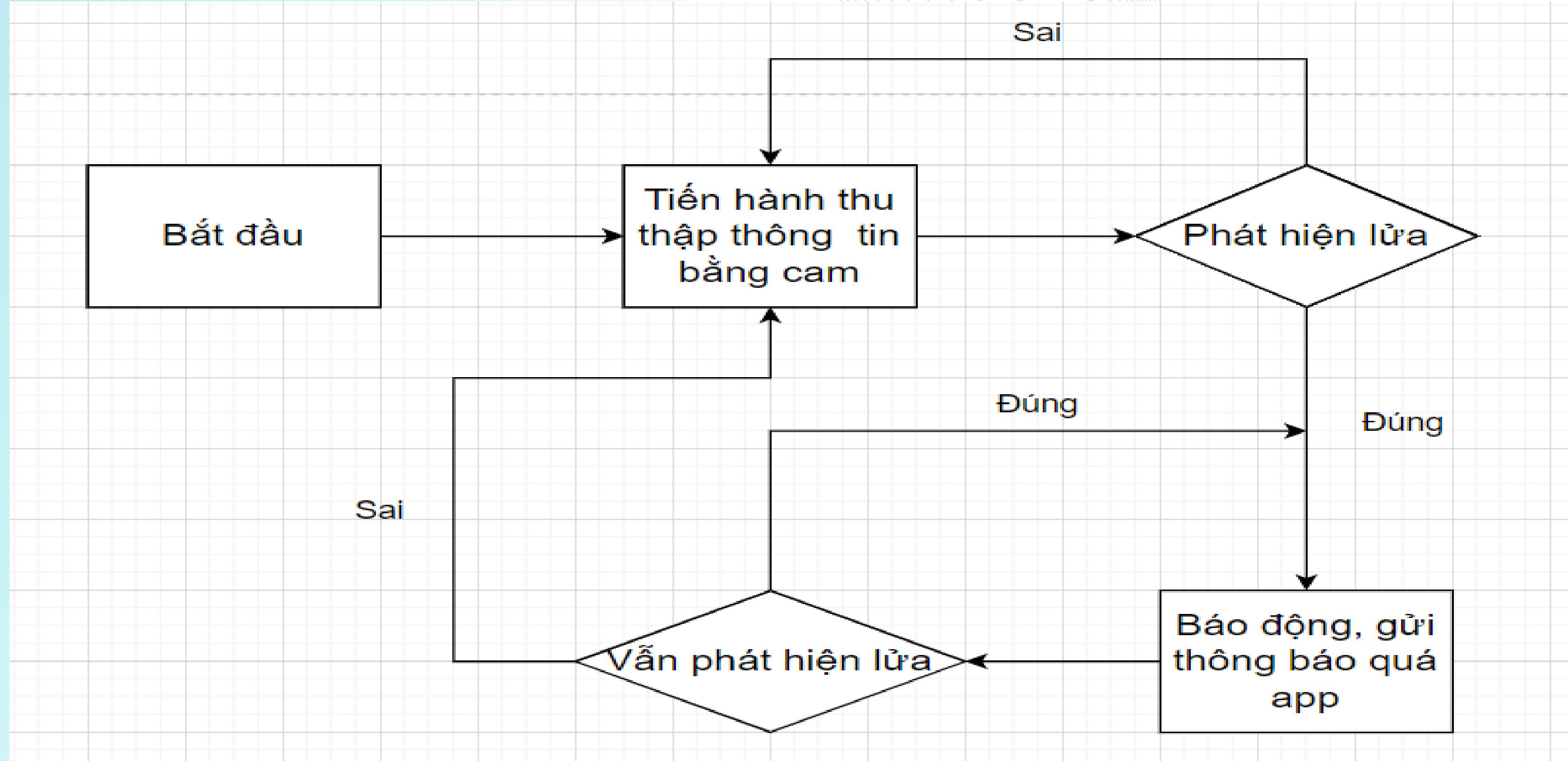
Giảm thiểu thiệt hại về người khi xảy ra hỏa hoạn

Nhanh chóng và tối ưu nguồn lực trong việc cứu hỏa

Giải pháp

- Cần có các thiết bị phần cứng để thu thập dữ liệu môi trường, hình ảnh.
- Cơ sở để lưu trữ, nhận các dữ liệu từ thiết bị phần cứng.
- Phát hiện cháy, báo cháy
- Có ứng dụng để có thể theo dõi và nhận được tín hiệu.
- Hệ thống chạy trong thời gian thực

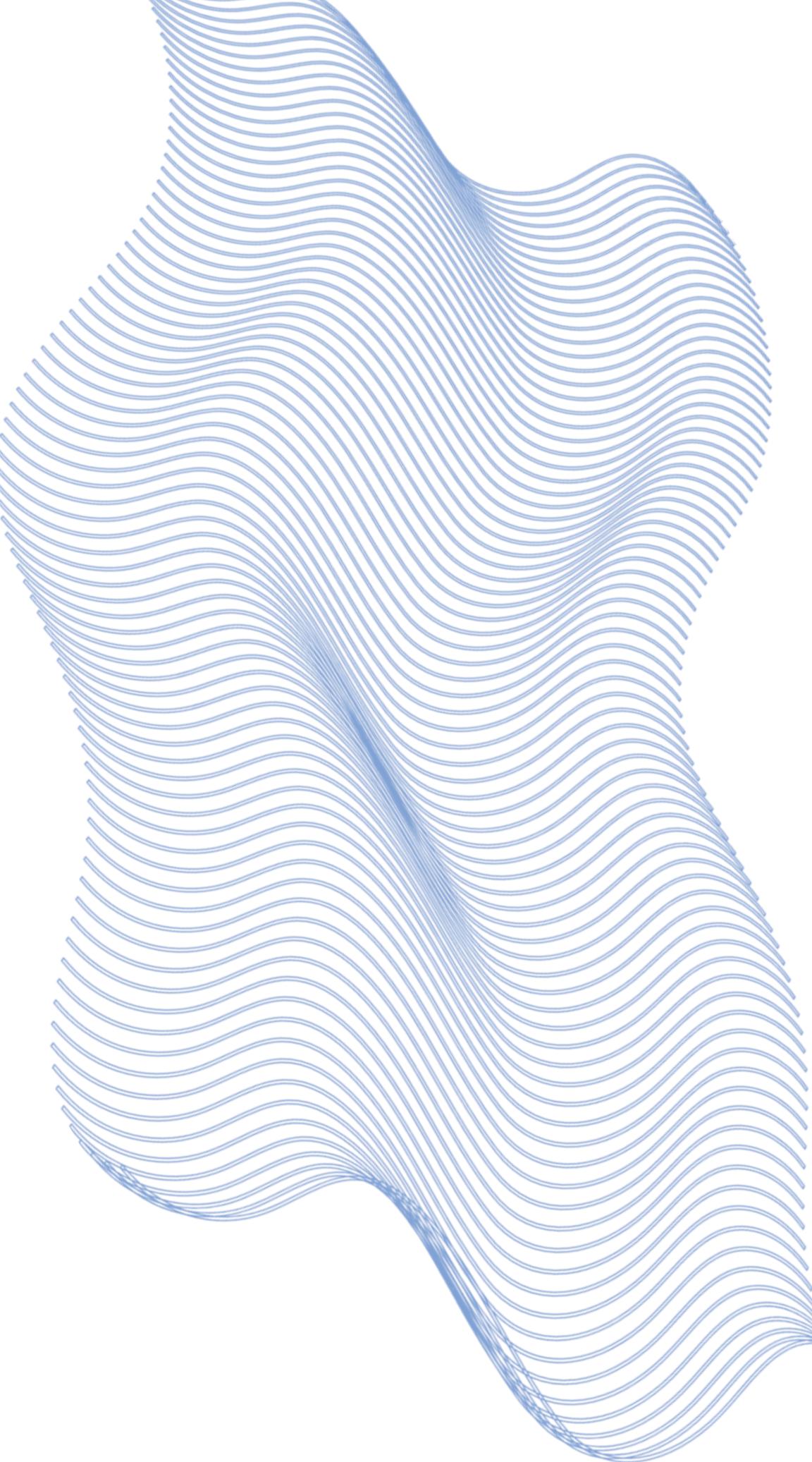
Sơ đồ tổng quan hệ thống



Tiến trình công việc

- Lắp đặt và cài đặt phần cứng
- Xây dựng Server + Firebase
- Xây dựng mô hình
- Xây dựng App Adroid
- Thủ nghiệm và hiệu chỉnh hệ thống

Lắp đặt và cài đặt phần cứng



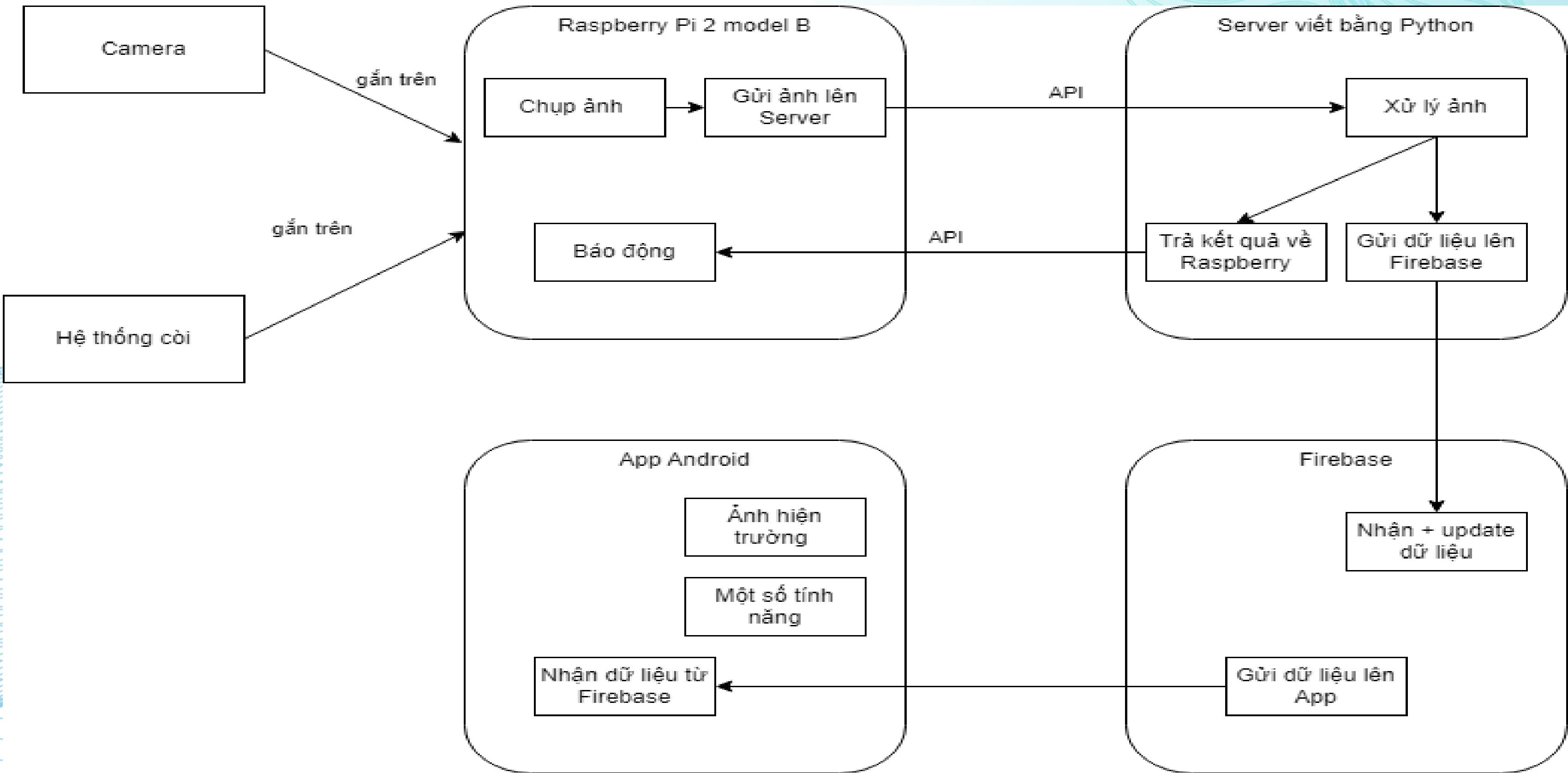
Linh kiện

- Raspberry Pi 2 Model
- WebCam
- Còi báo động



| Physical Pins | | | | |
|-----------------|-----|------|------|-----------------|
| Function | BCM | pin# | pin# | BCM |
| 3.3 Volts | | 1 | 2 | 5 Volts |
| GPIO/SDA1 (I2C) | 2 | 3 | 4 | 5 Volts |
| GPIO/SCL1 (I2C) | 3 | 5 | 6 | GND |
| GPIO/GCLK | 4 | 7 | 8 | 14 TX UART/GPIO |
| GND | | 9 | 10 | 15 RX UART/GPIO |
| GPIO | 17 | 11 | 12 | 18 GPIO |
| GPIO | 27 | 13 | 14 | GND |
| GPIO | 22 | 15 | 16 | 23 GPIO |
| 3.3 Volts | | 17 | 18 | 24 GPIO |
| MOSI (SPI) | 10 | 19 | 20 | GND |
| MISO(SPI) | 9 | 21 | 22 | 25 GPIO |
| SCLK(SPI) | 11 | 23 | 24 | 8 CE0_N (SPI) |
| GND | | 25 | 26 | 7 CE1_N (SPI) |
| RESERVED | | 27 | 28 | RESERVED |
| GPIO | 5 | 29 | 30 | GND |
| GPIO | 6 | 31 | 32 | 12 GPIO |
| GPIO | 13 | 33 | 34 | GND |
| GPIO | 19 | 35 | 36 | 16 GPIO |
| GPIO | 26 | 37 | 38 | 20 GPIO |
| GND | | 39 | 40 | 21 GPIO |

Sơ đồ kết nối hệ thống



Xây dựng server + firebase

Server

- Xây dựng server bằng Flask
- Kết nối với Firebase Realtime Database và Firebase Storage
- Đăng ký tài khoản và xác thực đăng nhập
- Nhận và xử lý dữ liệu từ Raspberry
- Đẩy ảnh lên Firebase Realtime Database và Firebase Storage

Cấu trúc Realtime Database

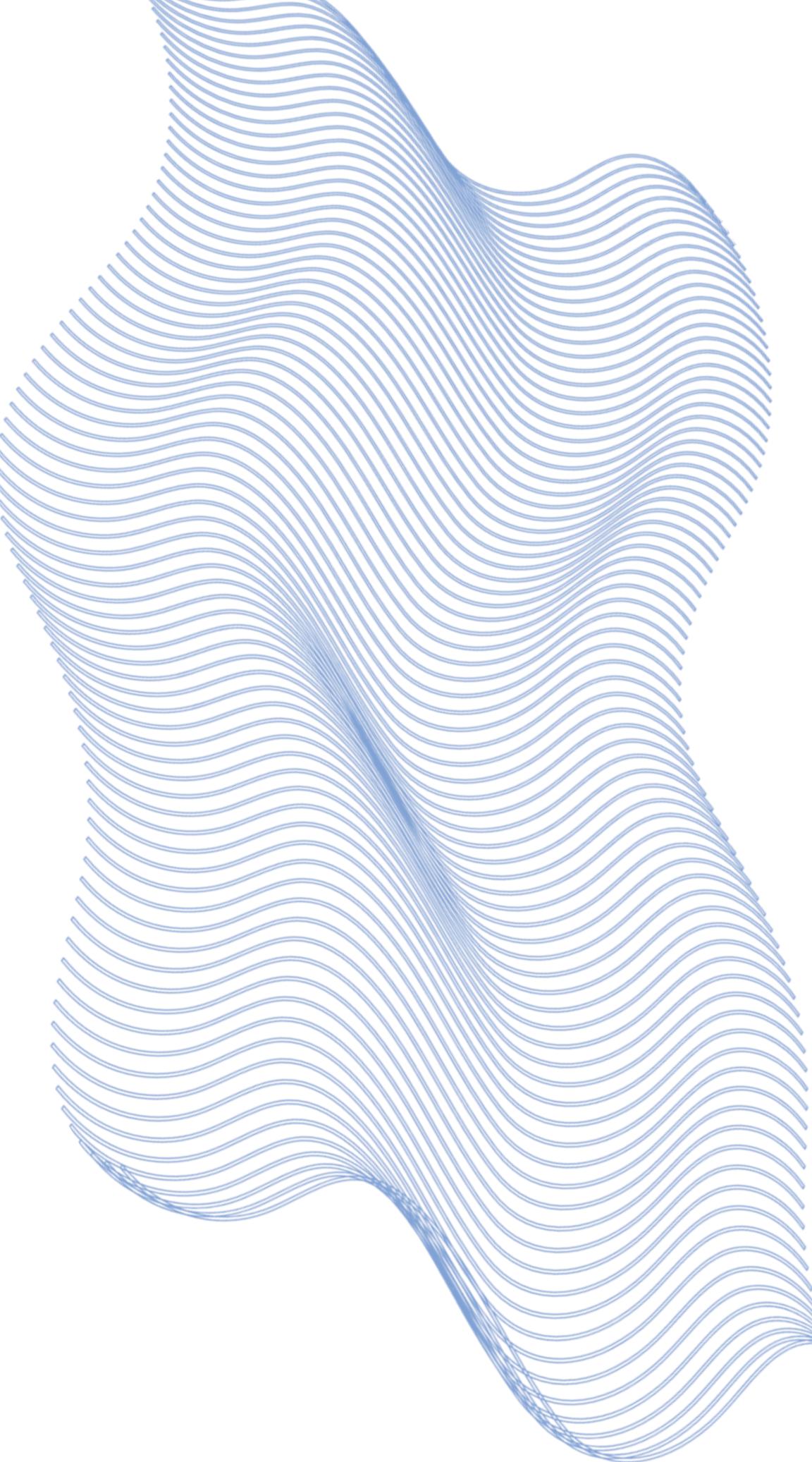
Devices - Thiết bị:

- token: Mã thiết bị
- name: Tên thiết bị
- date: Tem đánh dấu thời gian
- image_url: Hình ảnh khi có cảnh báo cháy

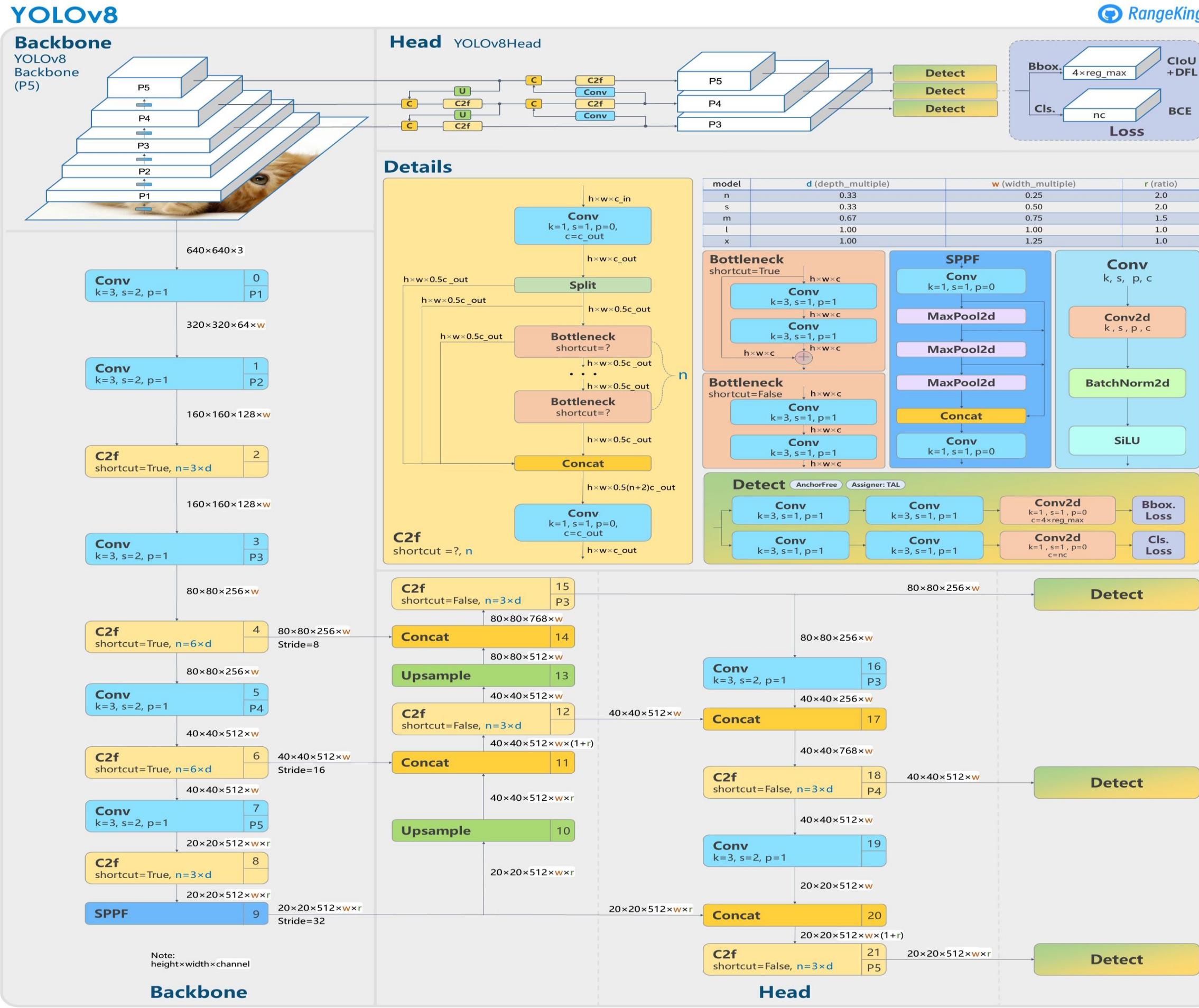
Cấu trúc Storage Database

- Modal
- Link ảnh tải lên từ Raspberry
- Date : Thời gian ảnh được gửi lên
- Ext: có thể là đuôi jpg, png, ...

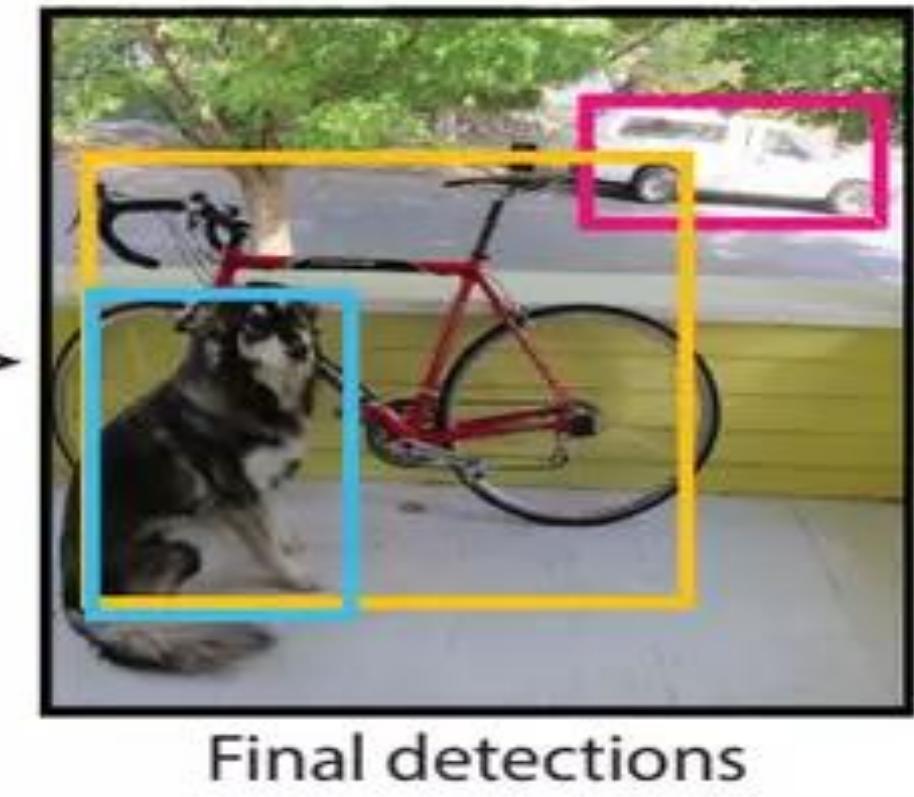
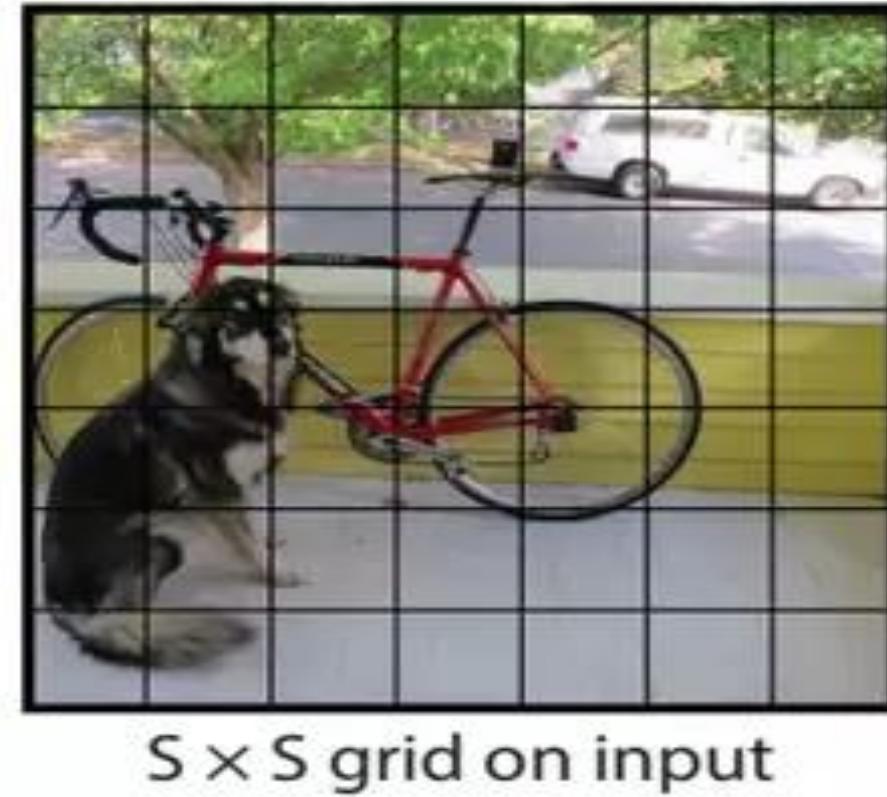
Xây dựng mô hình



Mô hình YOLOv8

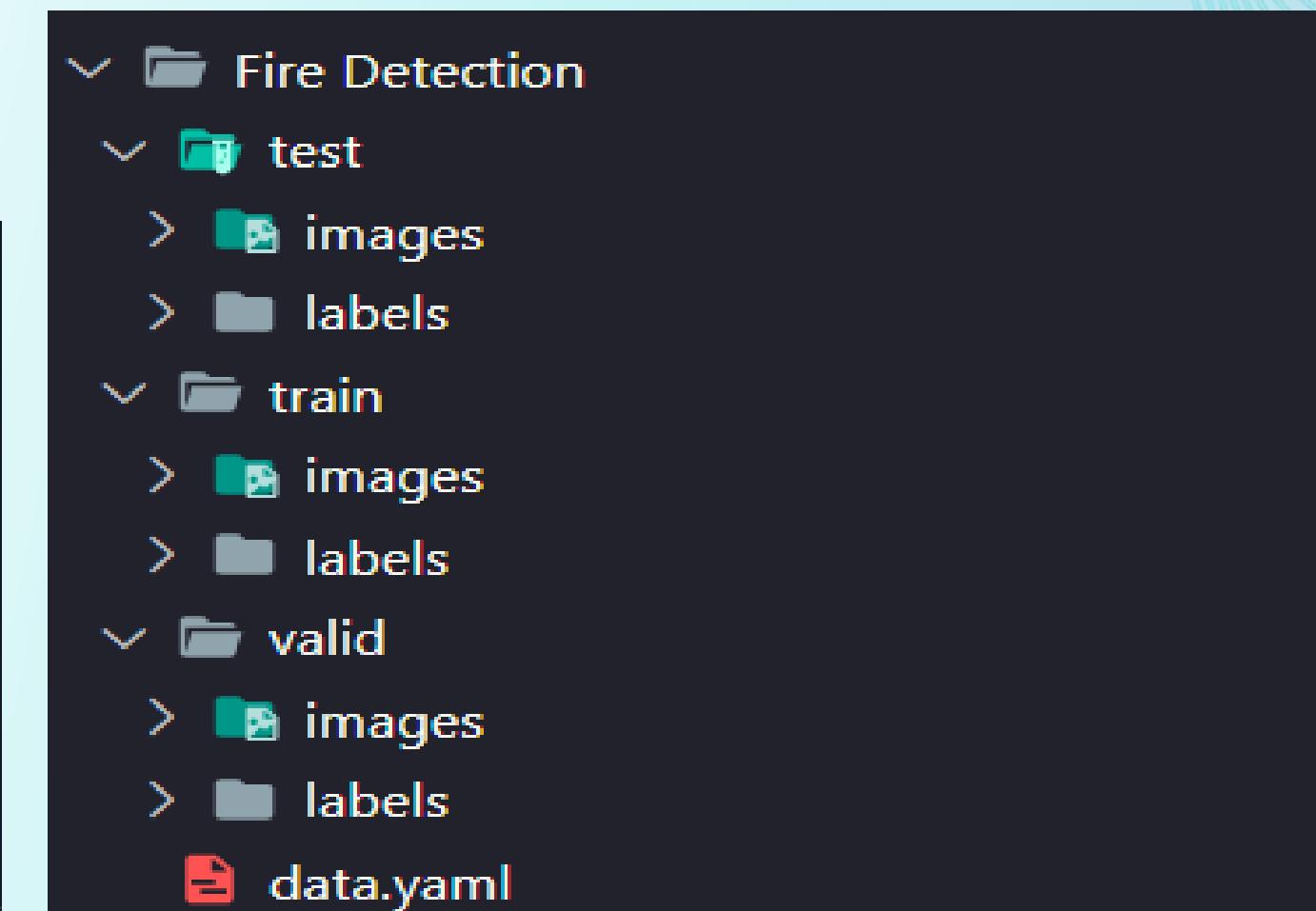
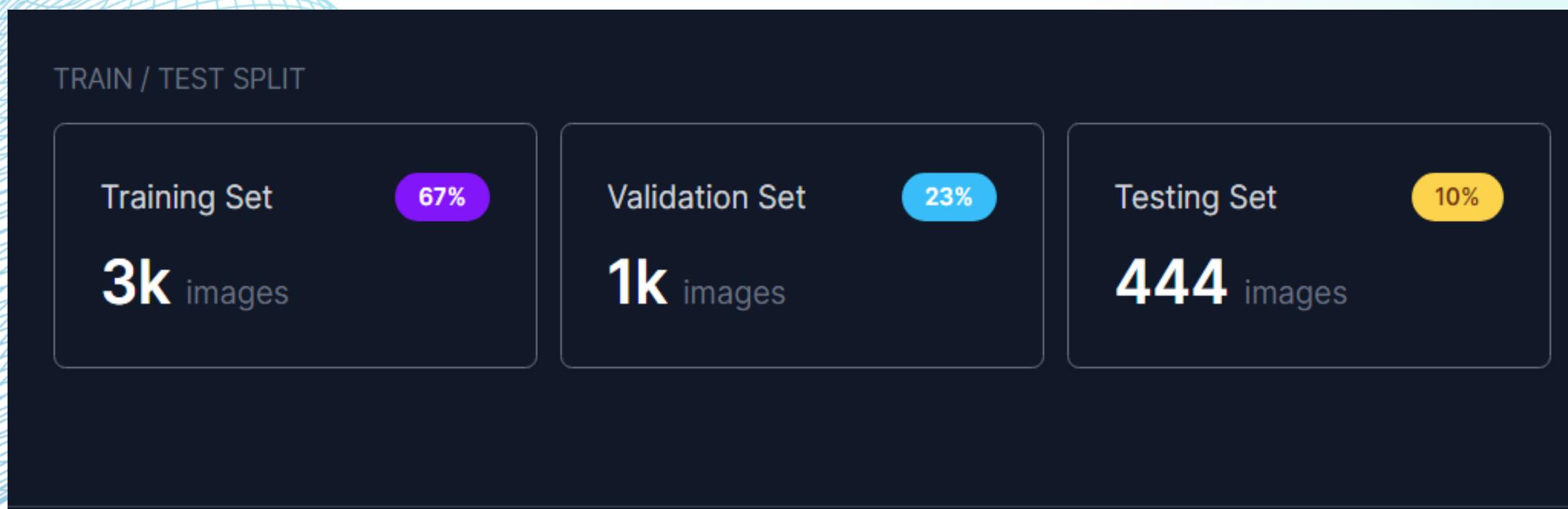


Mô hình YOLOv8



Thu thập dữ liệu

- Tập dữ liệu trên có 4521 hình ảnh có cùng kích thước là 416x416 và gán nhãn 8473 đối tượng trong 3 lớp: nf, fire, smoke.
- Trong đó, tập dữ liệu đã được chia sẵn thành 3 tập: Train, Val, Test và file data.yaml nhằm mục đích huấn luyện mô hình. Phân chia dữ liệu: Train: 2959(67%), Val: 1032(23%), Test: 444 (10%).
- Mỗi tập lại chứa 2 tập con là: images chứa danh sách ảnh và labels chứa danh sách nhãn ứng với mỗi ảnh.

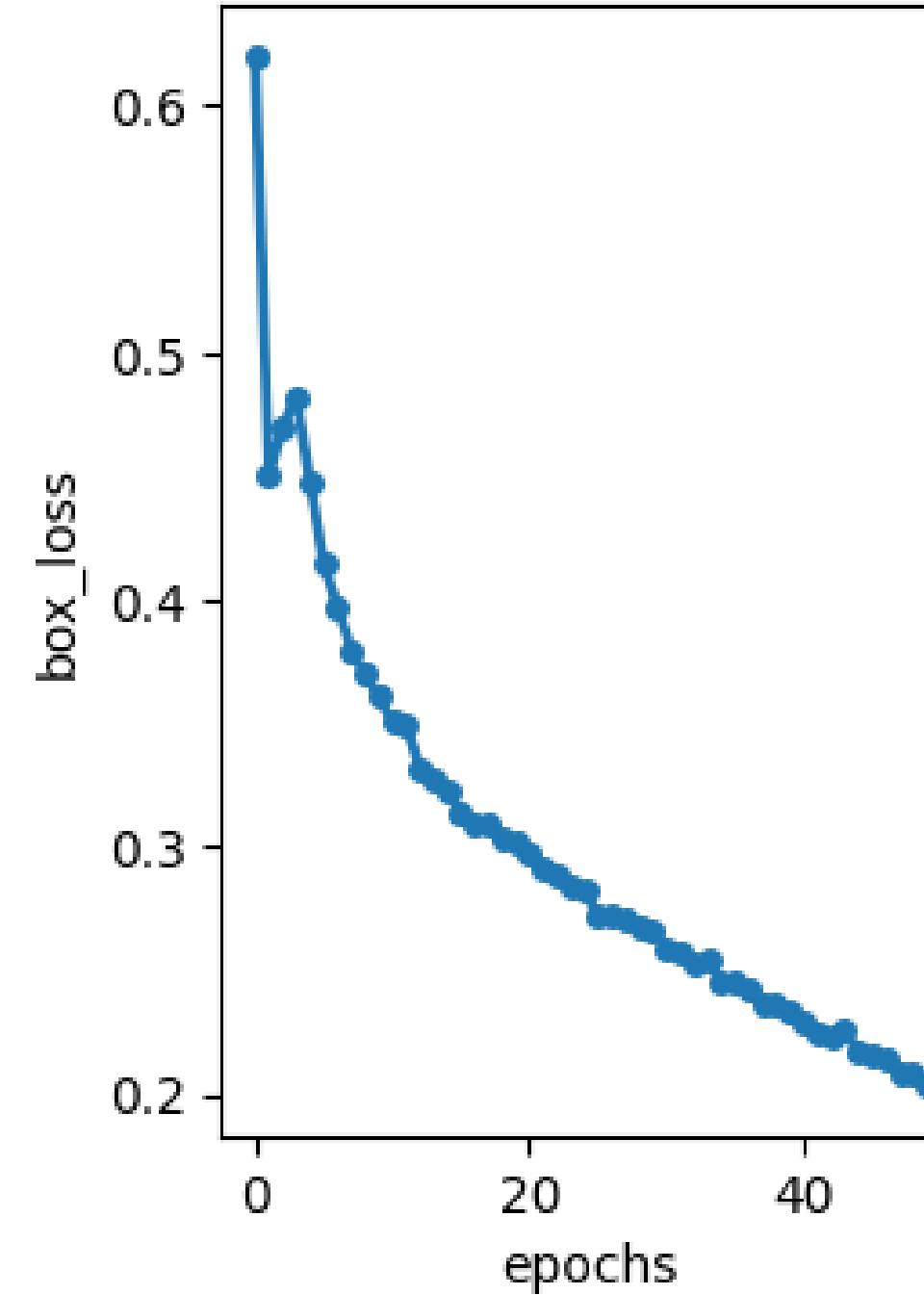


Xây dựng Model

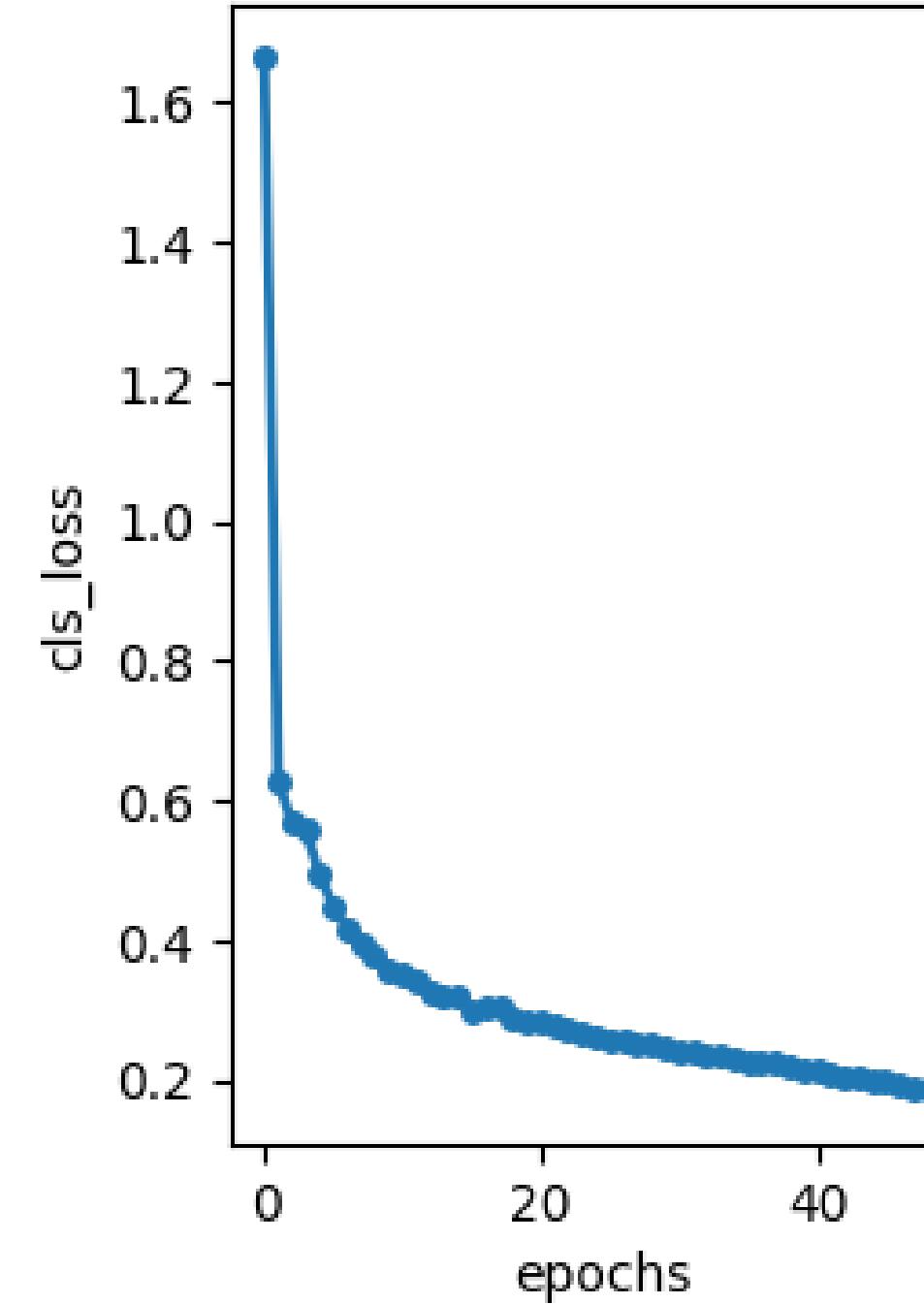
- Sử dụng pre-trained model yolov8.
- Train lại model với tập dữ liệu lửa.
- Training setting:
 - Epochs: 50
 - Patience: 40
 - Batch: 16
 - Lr0: 0.001
 - ImgSz: 416
 - Optimizer: Adam

Kết quả trên tập train

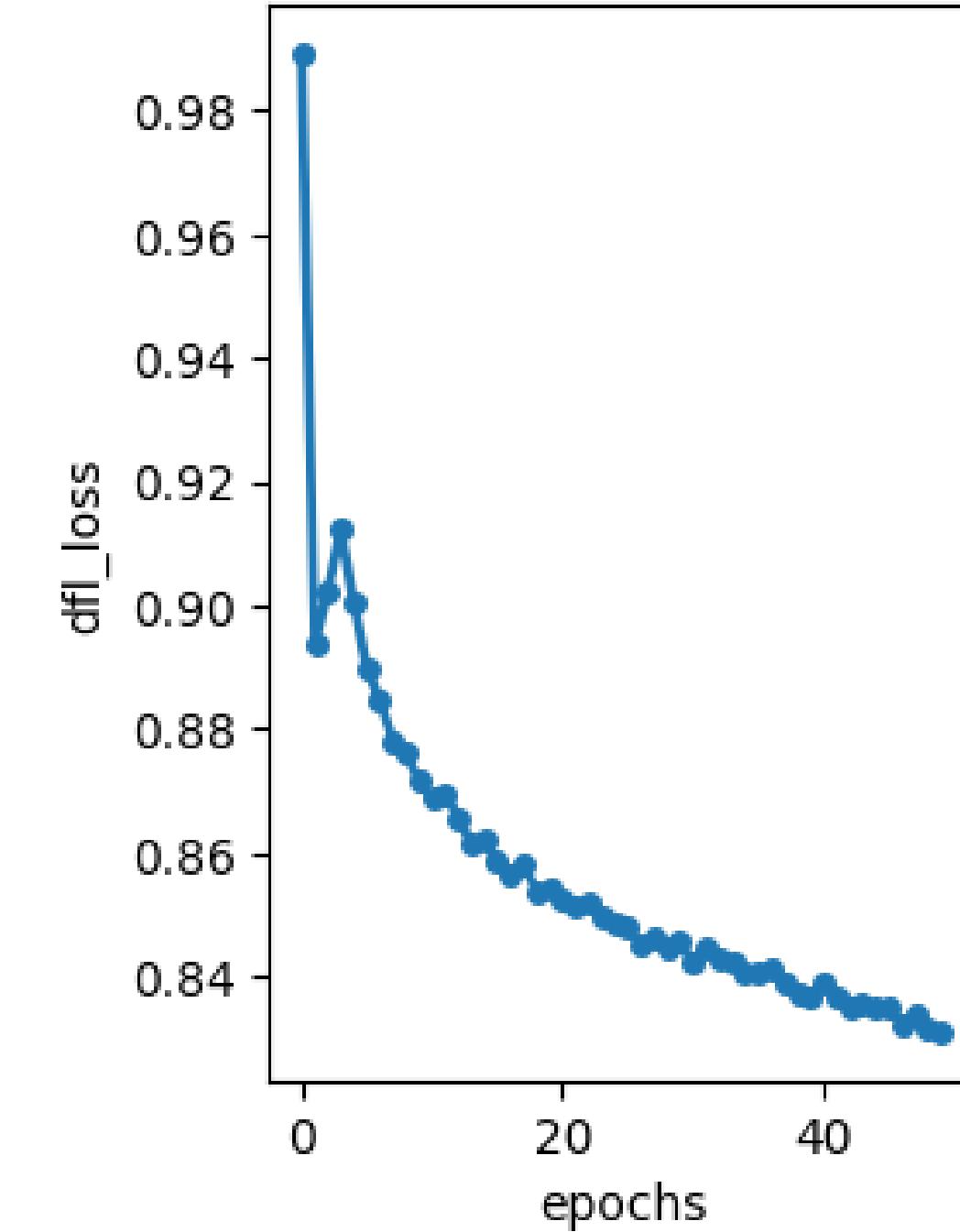
Train/box_loss



Train - Loss
Train/cls_loss



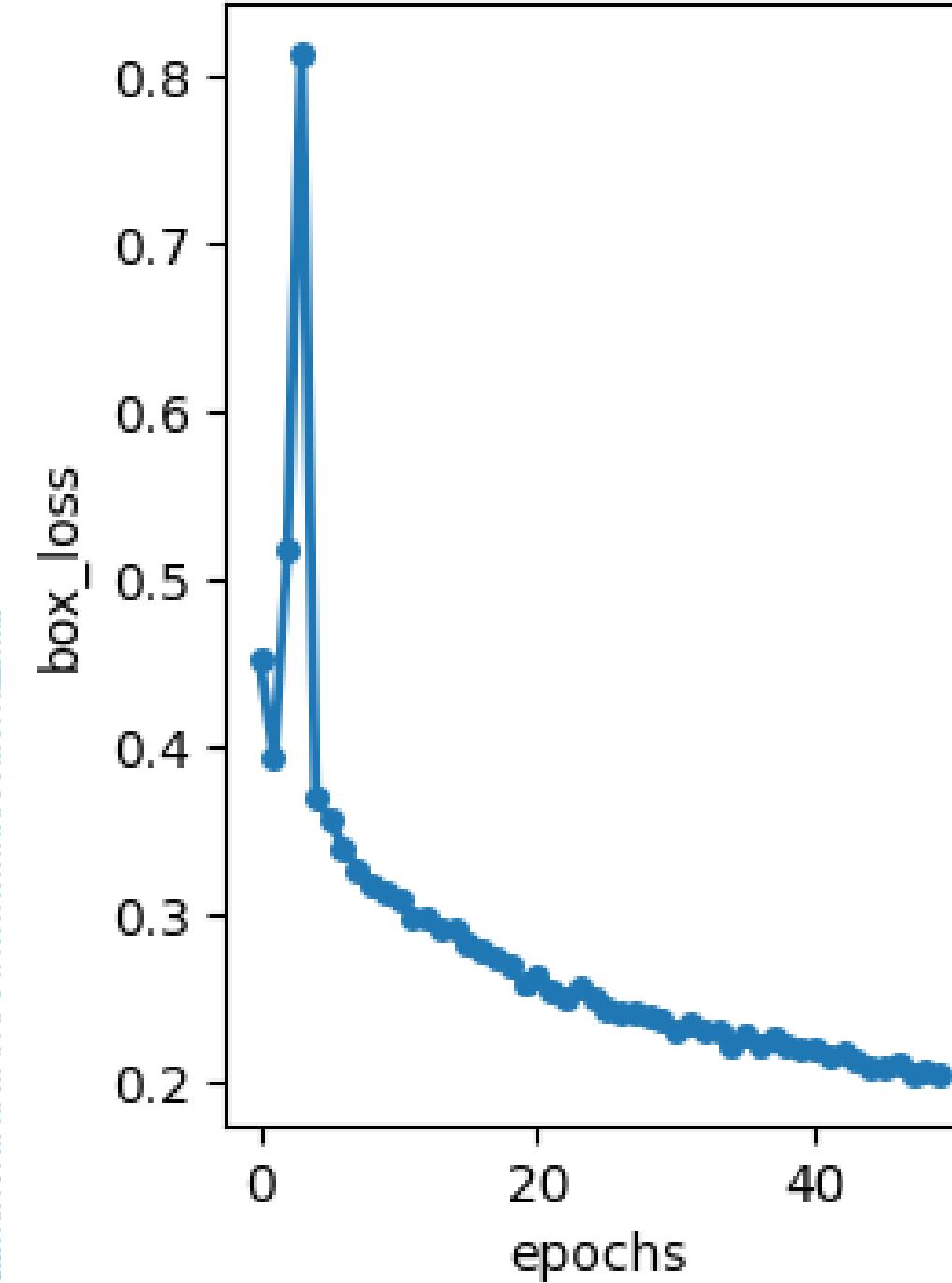
Train/dfl_loss



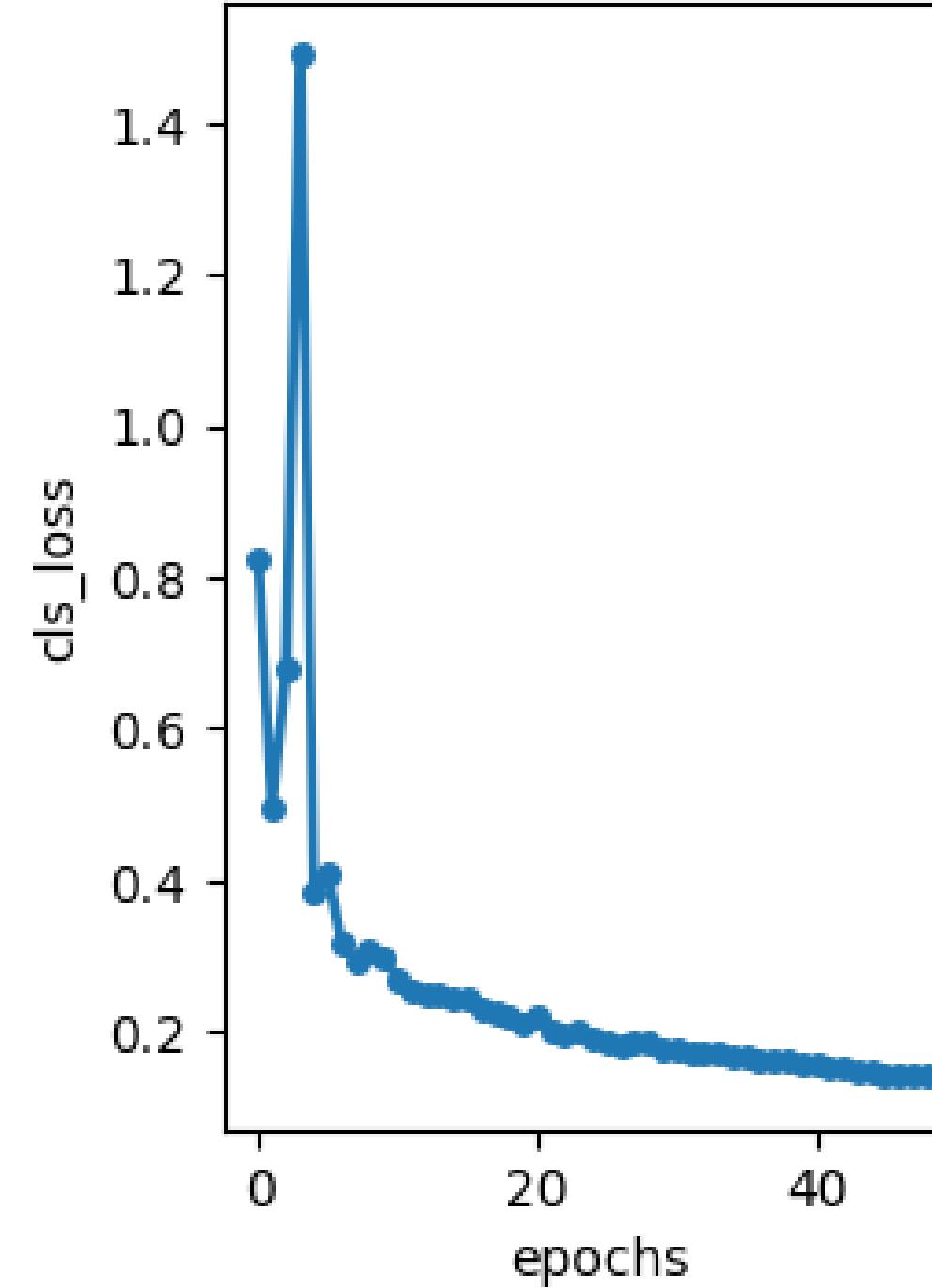
Đồ thị biểu diễn các hàm loss Yolov8 trên tập train

Kết quả trên tập val

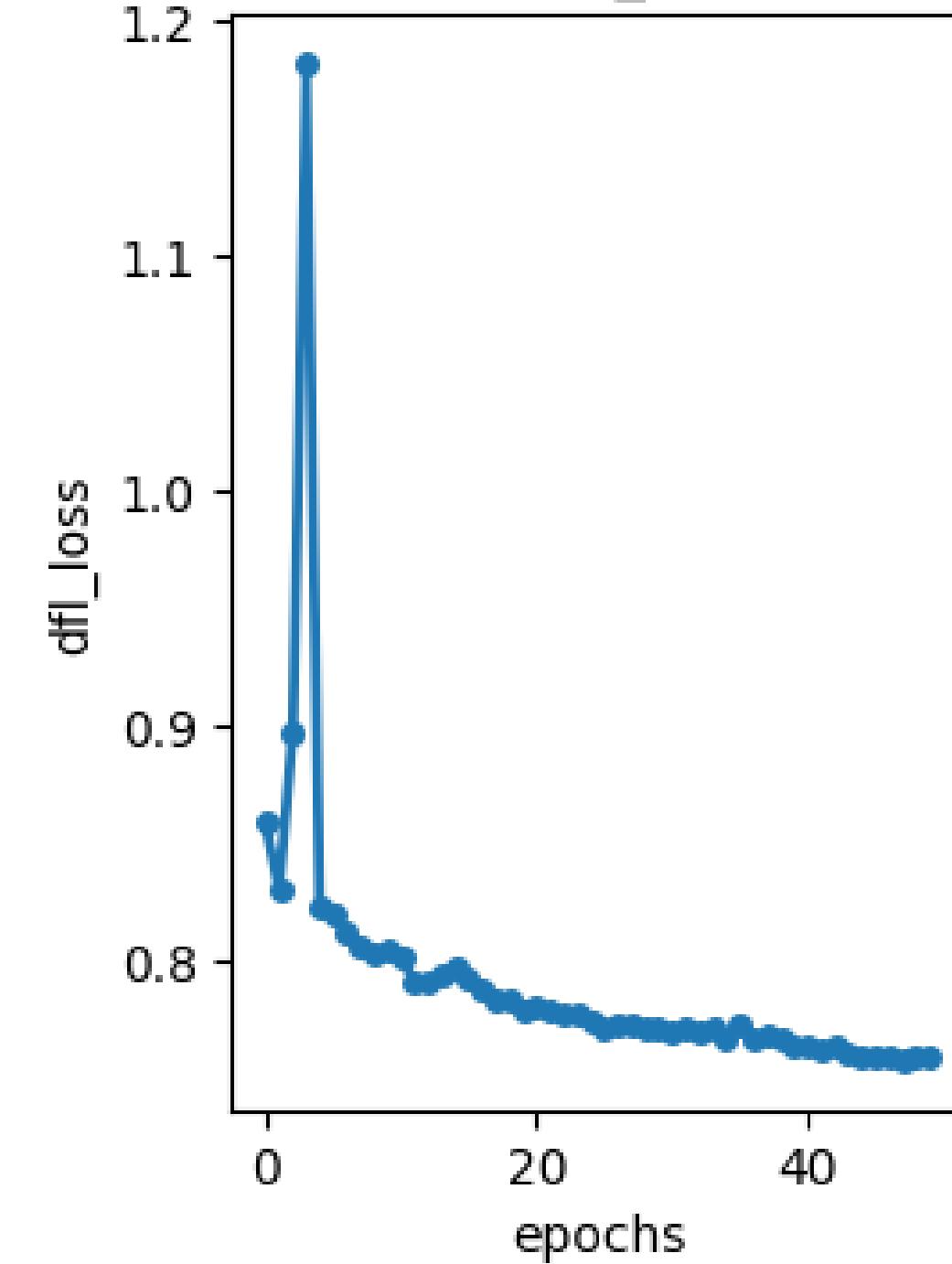
Val/box_loss



Val - Loss
Val/cls_loss

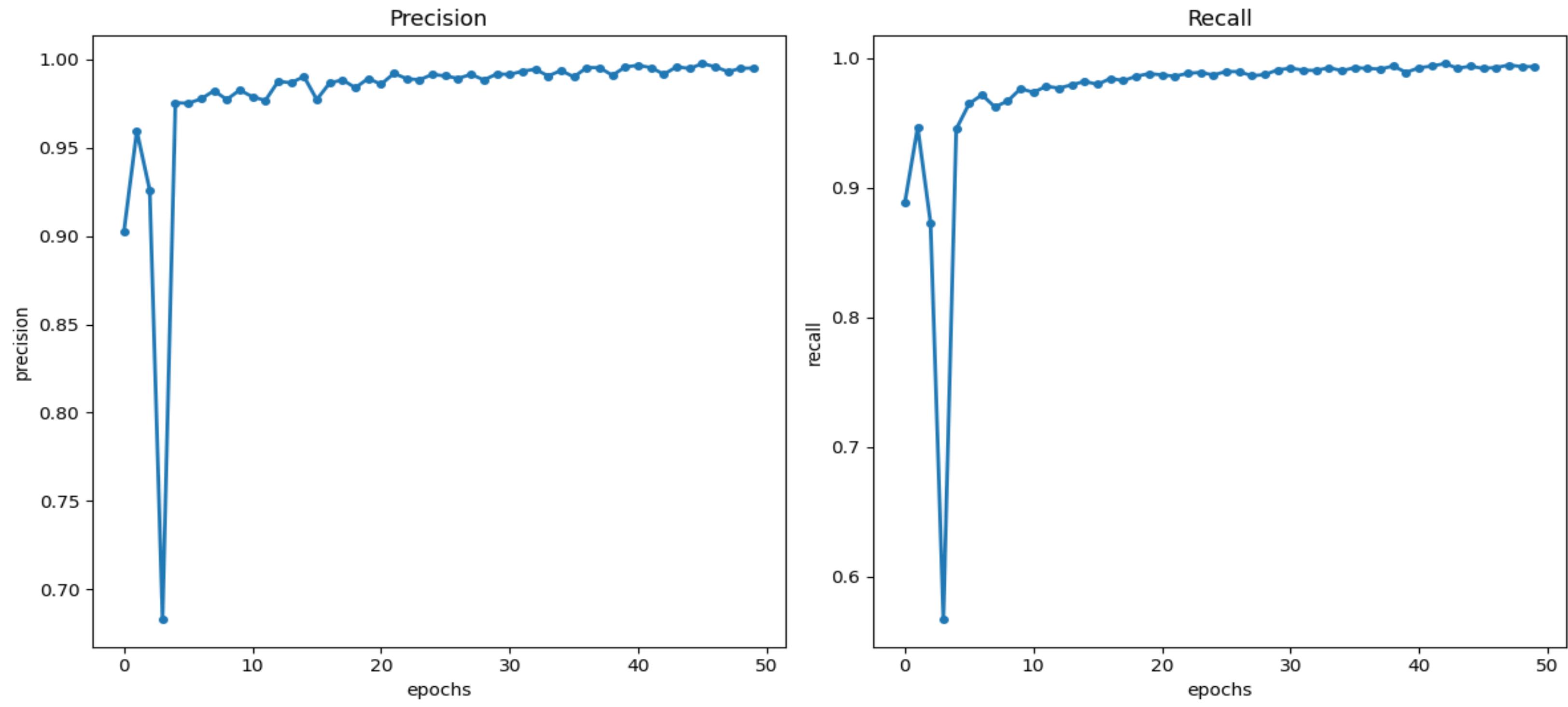


Val/dfl_loss



Đồ thị biểu diễn các hàm loss Yolov8 trên tập val

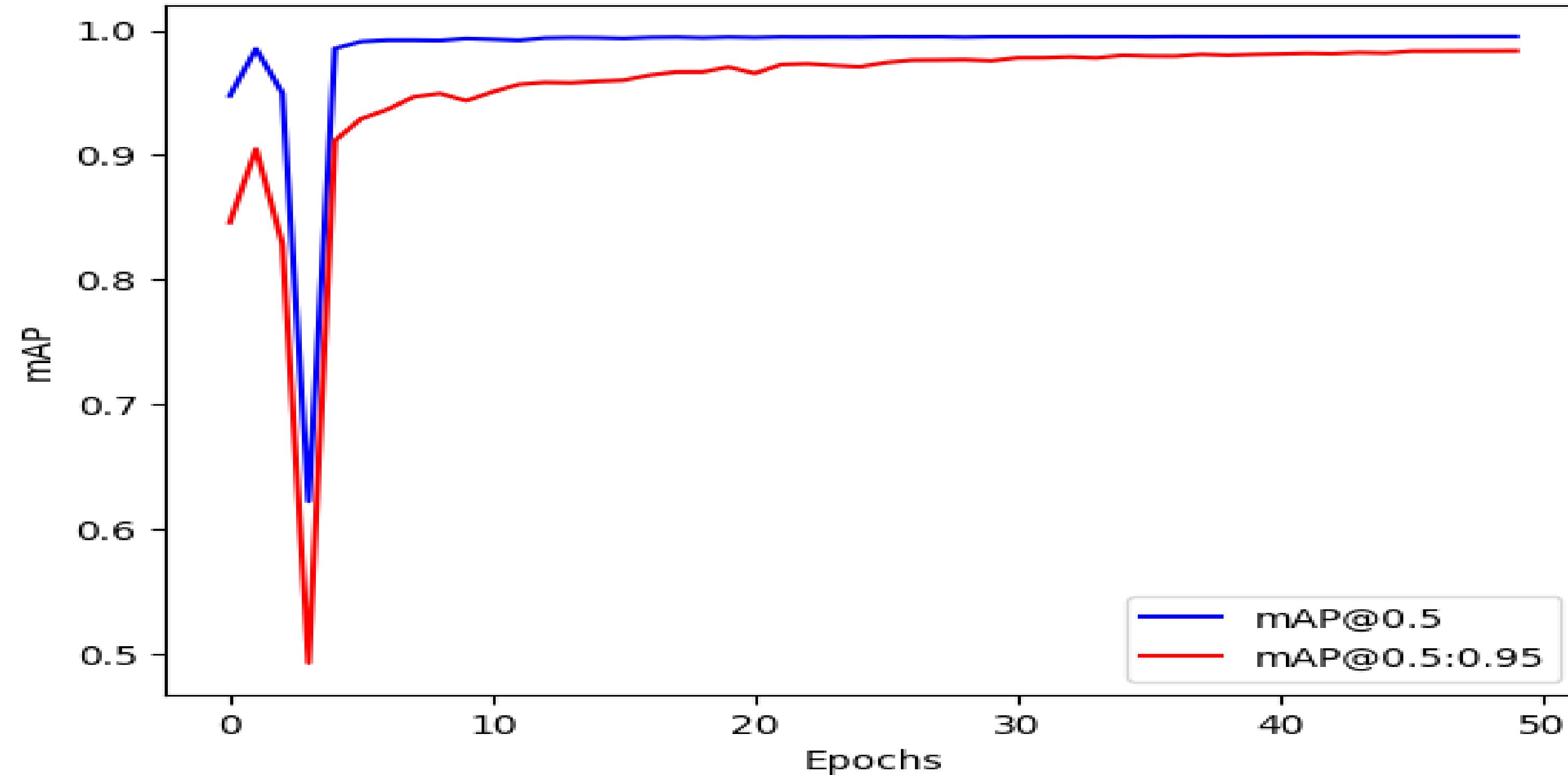
Đánh giá kết quả



Đồ thị các chỉ số Precision và Recall

Đánh giá kết quả

mAP vs mAP50-95



Đồ thị chỉ số mAP và mAP50-95

Kết quả nhận diện bằng ảnh trên tập kiểm thử



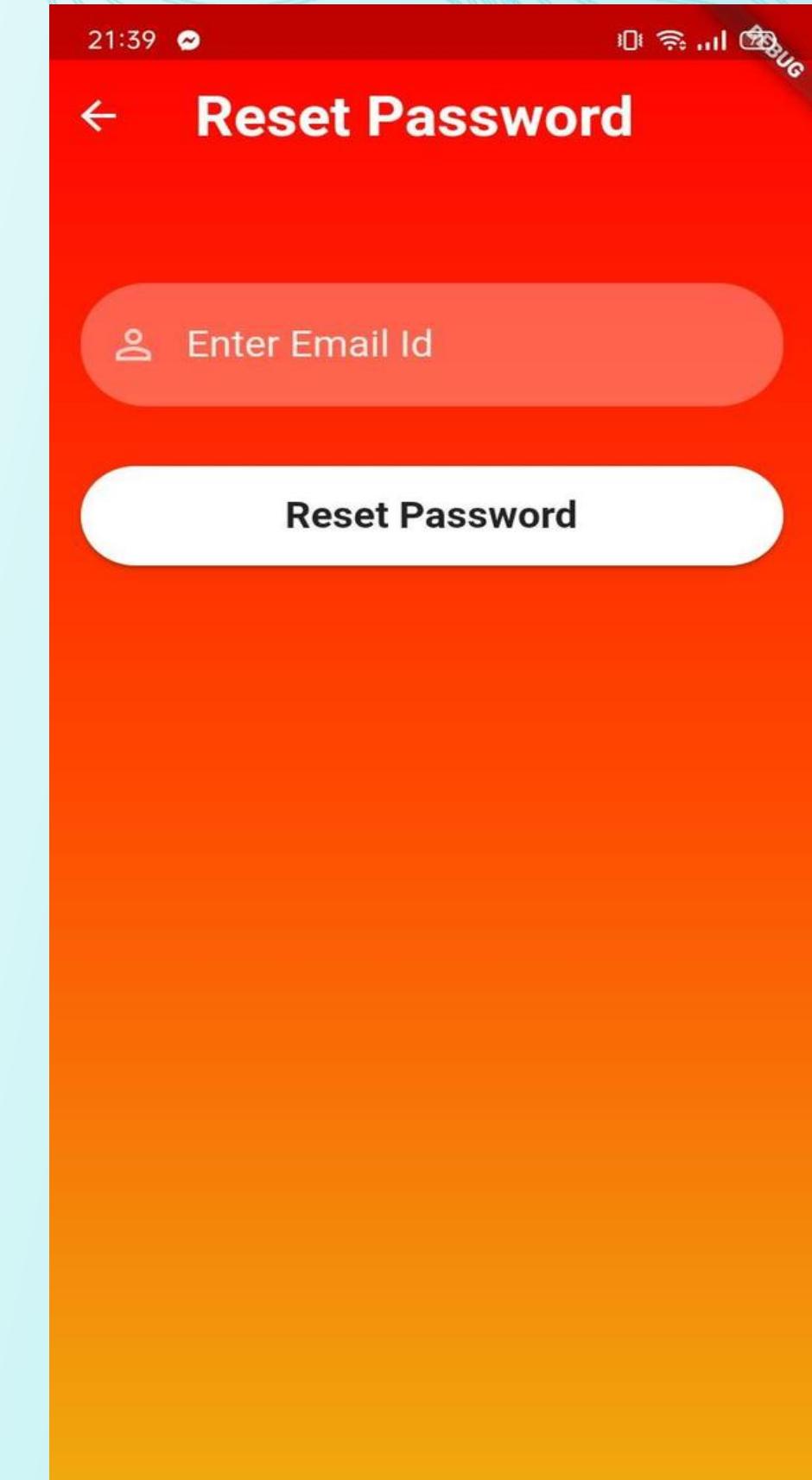
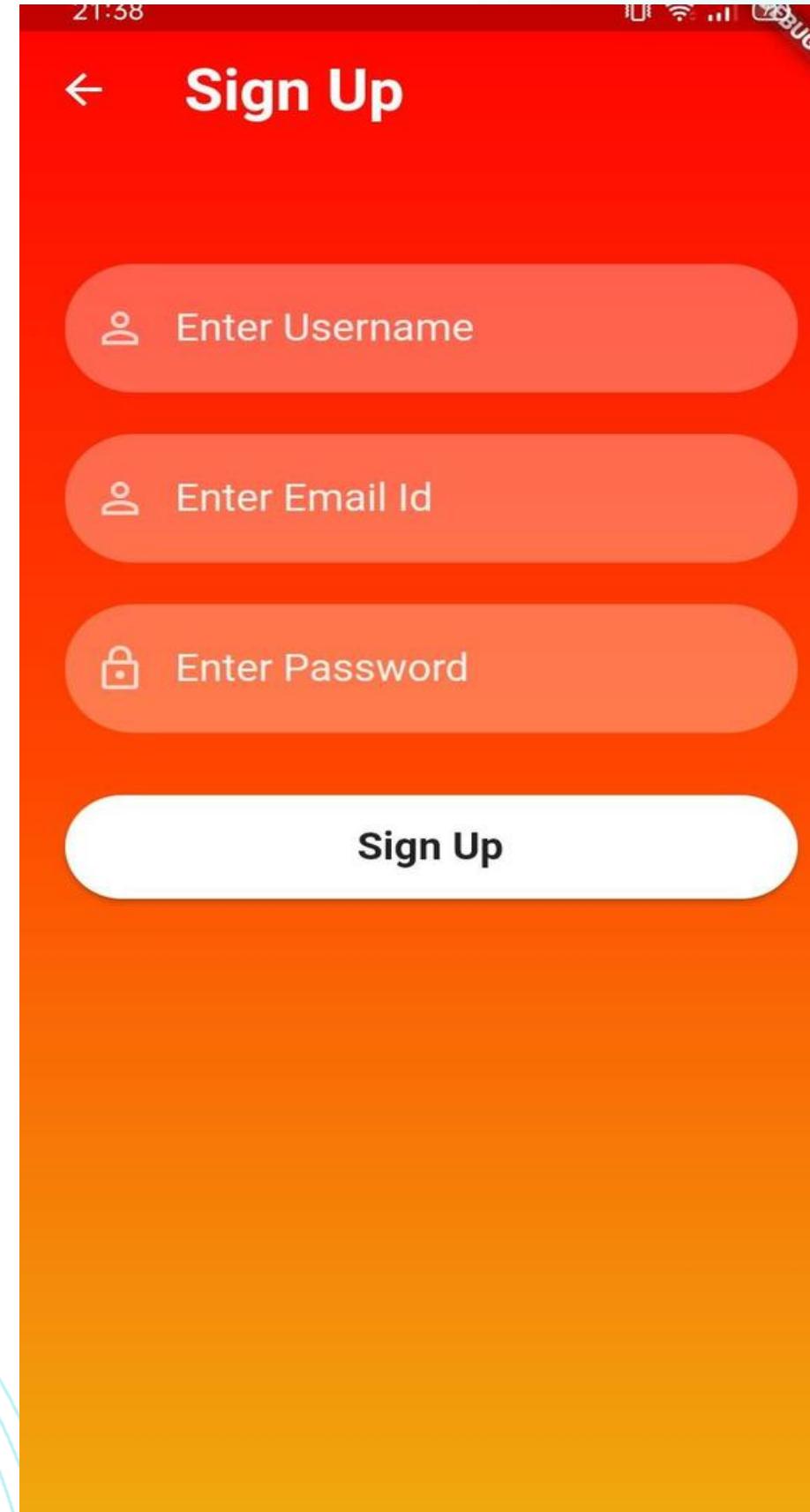
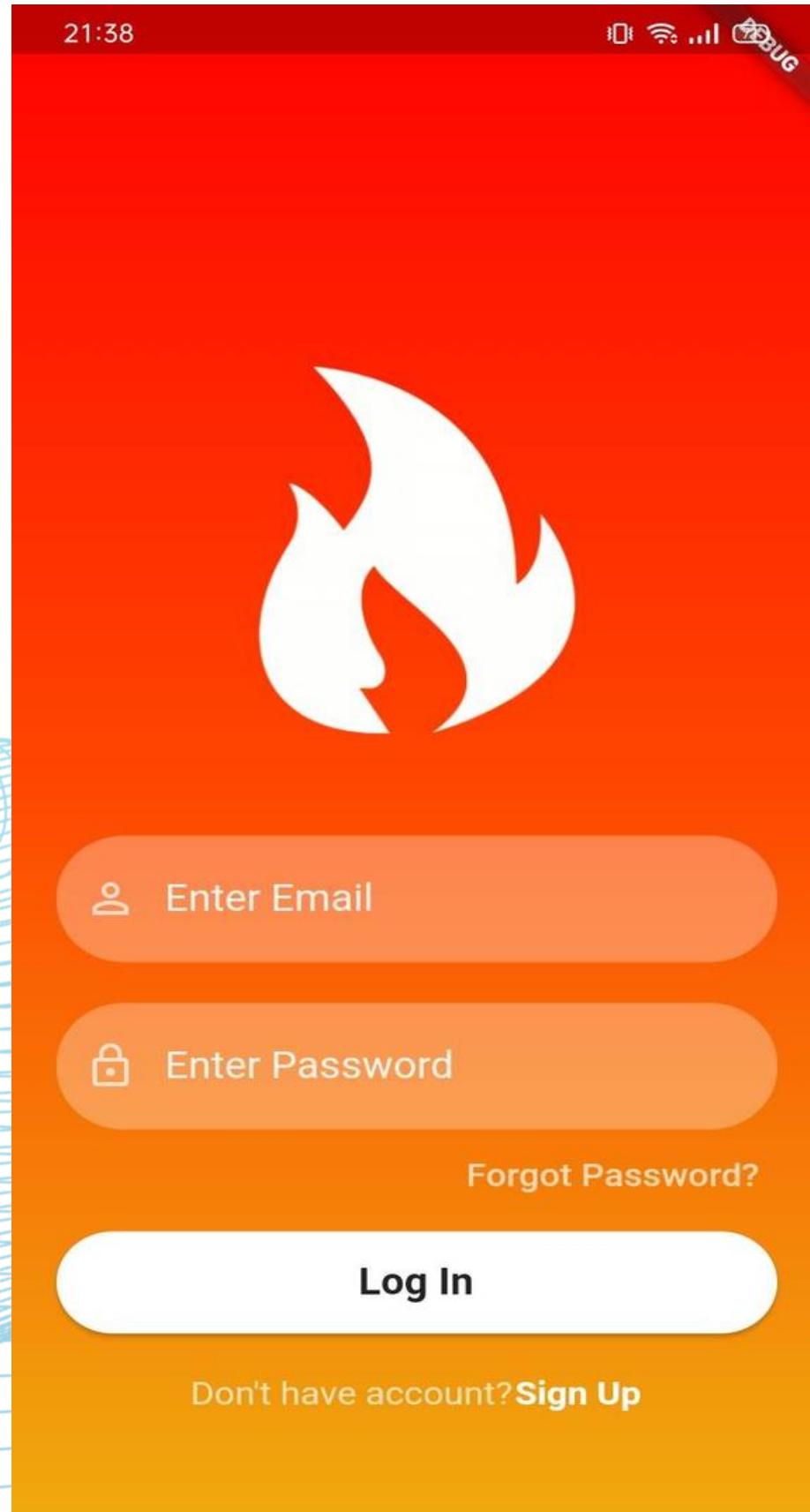
Kết quả nhận diện bằng ảnh thực tế



Xây dựng app android



Giao diện ứng dụng (đăng nhập / đăng ký)



Lịch sử báo cháy / Chi tiết báo cháy

Fire Alert [\[→ Log out\]](#)

Notifications from device:
Raspberry

| | Date |
|--|--------------------|
| | 24/3/2023 10:40:50 |
| | 24/3/2023 10:40:53 |
| | 24/3/2023 10:40:56 |
| | 24/3/2023 10:40:59 |
| | 24/3/2023 10:41:02 |
| | 24/3/2023 10:40:05 |
| | 24/3/2023 10:40:08 |
| | 24/3/2023 10:40:11 |
| | 24/3/2023 10:40:14 |

Fire Alert - Details

Detected on:
24/3/2023 10:40:05

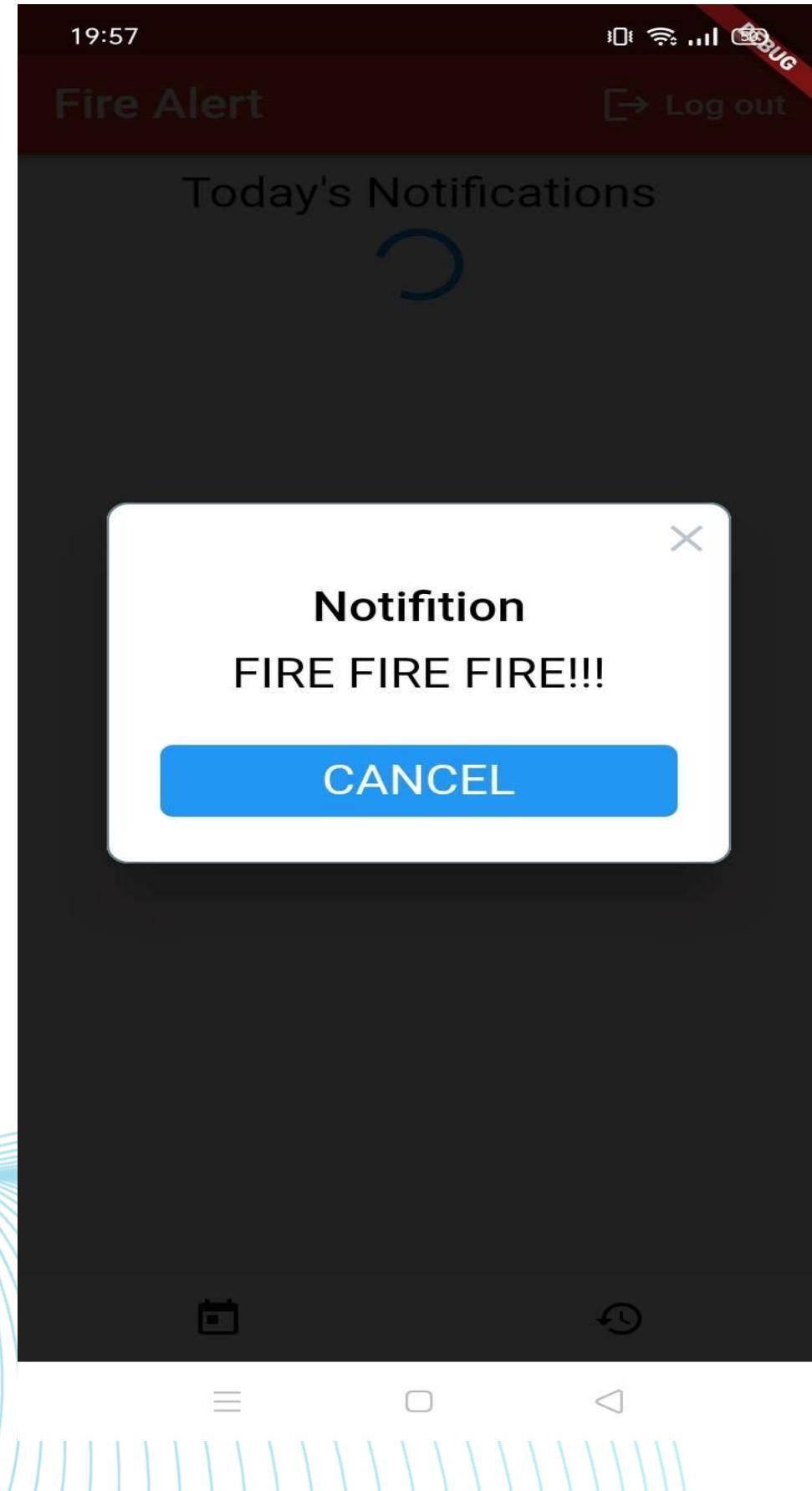


Fire Alert [\[→ Log out\]](#)

History fire

| | Date |
|--|---------------------|
| | 2023-06-08 17:30:22 |
| | 2023-06-08 17:30:24 |
| | 2023-06-08 17:30:34 |
| | 2023-06-08 17:30:40 |
| | 2023-06-08 17:30:50 |
| | 2023-06-08 17:30:55 |

Báo cháy



Kết luận

Đánh giá kết quả
và hướng phát triển
của dự án



Nhận xét

Ưu điểm

- Thực hiện đầy đủ các chức năng đề ra.
- Khả năng mở rộng tốt.

Khuyết điểm

- Ứng dụng di động còn sơ sài, thiếu hụt vài chức năng quan trọng.
- Một vài thiết bị bị hỏng hóc, khi tiến hành còn một chút trực trặc.
- Tốc độ xử lý khung hình không cao.
- Chất lượng camera không được tốt, bị ám xanh -> Độ chính xác phát hiện giảm

Hướng phát triển

- Tăng độ hoàn thiện ứng dụng Android
- Làm thêm mô hình nhận diện người và tăng độ chính xác.
- Thêm các tính năng như kích hoạt hệ thống chữa cháy nước.
- Cải thiện về khả năng phát hiện có lửa.

**Thanks for
listening**

