



TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
PBL5 - ĐỒ ÁN KỸ THUẬT MÁY TÍNH

GÂY THÔNG MINH HỖ TRỢ NGƯỜI KHIẾM THỊ

Giảng viên hướng dẫn: TS. Ninh Khánh Duy

STT NHÓM: 06	
HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN ĐỒ ÁN
Đinh Hoàng Phi Hùng	21N10B
Huỳnh Văn Thái	21N10B
Nguyễn Phan Bảo Lộc	21N10B
Hoàng Thị Hồng Thắm	21N10B

ĐÀ NẴNG, 06/2024

TÓM TẮT ĐỒ ÁN

Đề tài "Thiết bị gây thông minh hỗ trợ người khiếm thị" tập trung vào giải quyết các thách thức mà người khiếm thị đối diện hàng ngày. Trong đó, vấn đề nhận diện mệnh giá tiền Việt Nam, định vị vị trí và cảnh báo vật cản được ưu tiên. Thiết bị này nhằm giúp họ di chuyển một cách an toàn và độc lập hơn trong môi trường hàng ngày.

Thiết bị sử dụng công nghệ học máy và trí tuệ nhân tạo để nhận diện mệnh giá của tiền Việt Nam và sử dụng module gps để xác định vị trí hiện tại của người sử dụng. Kết quả sẽ được thông báo bằng âm thanh và hiển thị vị trí thông qua ứng dụng GPS Tracker để người thân có thể dễ dàng theo dõi và hỗ trợ kịp thời người khiếm thị.

Ngoài ra, thiết bị cũng được trang bị tính năng cảnh báo khi phát hiện vật cản trong quá trình di chuyển, giúp người khiếm thị tránh được các tai nạn không mong muốn.

Với sự hỗ trợ của thiết bị này, người khiếm thị có thể tăng cường sự tự tin và độc lập trong cuộc sống hàng ngày, đồng thời giảm bớt các rủi ro và trở ngại trong việc di chuyển.

Cuối cùng xin chân thành cảm ơn thầy TS. Ninh Khánh Duy đã nhiệt tình hướng dẫn, gợi ý những giải pháp tuyệt vời để chúng em có thể hoàn thành được dự án này.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên	Nhiệm vụ	Hoàn thành
Đinh Hoàng Phi Hùng (nhóm trưởng)	Phân công công việc, đảm bảo tiến độ đề án Tìm hiểu, triển khai model Huấn luyện, đánh giá, hiệu chỉnh model Thu thập dữ liệu Xây dựng app Ghép nối, thử nghiệm sản phẩm Viết báo cáo	✓
Huỳnh Văn Thái	Tìm hiểu, xây dựng phần cứng Thực hiện giải pháp GPS Thực hiện giải pháp phát hiện vật cản Ghép nối và thử nghiệm sản phẩm Viết báo cáo	✓
Nguyễn Phan Bảo Lộc	Xây dựng ứng dụng mobile app Xây dựng cơ sở dữ liệu Thực hiện giải pháp GPS Ghép nối và thử nghiệm sản phẩm Viết báo cáo	✓
Hoàng Thị Hồng Thắm	Tìm hiểu, triển khai model Huấn luyện, đánh giá, hiệu chỉnh model Thu thập dữ liệu Triển khai model trên phần cứng Ghép nối, thử nghiệm sản phẩm Viết báo cáo	✓

MỤC LỤC

1. Giới thiệu	1
1.1. Thực trạng sản phẩm	1
1.2. Các vấn đề cần giải quyết.....	1
1.3. Đề xuất giải pháp tổng quan	1
2. Giải pháp.....	2
2.1. Tổng quan hệ thống	2
2.2. Giải pháp phần cứng	2
2.3. Giải pháp truyền thông.....	4
2.4. Giải pháp nhận diện mệnh giá tiền.....	5
2.4.1. Mô hình SSD MobilenetV2	6
2.4.2. TensorFlow và TensorFlow Lite	11
2.5. Giải pháp GPS.....	12
2.5.1. Quá trình thực hiện giải pháp	12
2.5.2. Tổng quan về hệ thống của giải pháp GPS.....	13
2.6. Giải pháp phát hiện vật cản.....	14
2.7. Giải pháp về ứng dụng di động	15
2.7.1. Phát biểu bài toán	15
2.7.2. Công nghệ sử dụng	15
2.7.3. Sơ đồ hoạt động	16
2.7.4. Cơ sở dữ liệu.....	18
3. Kết quả.....	20
3.1. Nhận diện mệnh giá	20
3.1.1. Tập dữ liệu	20
3.1.2. Huấn luyện với mô hình SSD MobileNetV2.....	22
3.1.3. Kết quả khi tiến hành tối ưu thời gian	28
3.2. GPS	31
3.3. Phát hiện vật cản	32
3.4. Ứng dụng di động	33
3.5. Mô hình hoàn chỉnh	36
4. Kết luận và hướng phát triển.....	36
4.1. Kết luận.....	36
4.2. Hướng phát triển	37
5. Danh mục tài liệu tham khảo	38

DANH MỤC HÌNH ẢNH

Hình 1. Sơ đồ tổng quan hệ thống	2
Hình 2. Sơ đồ lắp đặt phần cứng.....	2
Hình 3. Các dịch vụ của Firebase	4
Hình 4. Sơ đồ hoạt động của giải pháp nhận diện mệnh giá tiền.....	5
Hình 5. Quá trình thực hiện giải pháp nhận diện mệnh giá tiền	6
Hình 6. Cách hoạt động của SSD.....	6
Hình 7. Kiến trúc mạng của SSD.....	7
Hình 8. So sánh giữa Depthwise Separable và Full Convolution MobileNet	8
Hình 9. Tích chập 2 chiều thông thường	9
Hình 10. Tích chập điểm.....	9
Hình 11. So sánh với VGG16 và GoogleNet.....	9
Hình 12. So sánh giữa MobileNetV1 và MobileNetV2.....	10
Hình 13. Mạng SSD MobileNetV2.....	10
Hình 14. Hệ thống TensorFlow	11
Hình 15. Tổng quan mô hình trong hệ thống.....	12
Hình 16. Tổng quan về hệ thống của giải pháp GPS	13
Hình 17. Sơ đồ hoạt động của giải pháp phát hiện vật cản.....	14
Hình 18. Nguyên lý hoạt động của cảm biến siêu âm	15
Hình 19. Sơ đồ hoạt động trang đăng nhập.....	16
Hình 20. Sơ đồ hoạt động trang đăng ký	17
Hình 21. Sơ đồ hoạt động trang chính	18
Hình 22. Đăng ký, xác thực người dùng bằng Authentication Firebase.....	18
Hình 23. Lưu thông tin người dùng trên Realtime Database	19
Hình 24. Lưu tọa độ thiết bị trên Realtime Database.....	19
Hình 25. Đồ thị mất mát của batch-size là 16.....	23
Hình 26. Độ chính xác trên tập test của batch-size 16.....	23
Hình 27. Đồ thị mất mát khi đã tăng cường về độ sáng	25
Hình 28. Hình ảnh nhận diện sai sau khi được tăng cường về độ sáng trên tập hiệu chỉnh	25
Hình 29. Đồ thị mất mát sau khi thay đổi dữ liệu và tăng cường về độ tương phản	27
Hình 30. Kết quả sau khi thay đổi dữ liệu và tăng cường dữ liệu về độ tương phản.....	27
Hình 31. Độ chính xác khi test với tập test mới.....	27
Hình 32. Ma trận nhầm lẫn khi test với tập test mới	28
Hình 33. Kết quả nhận diện với TensorFlow và TensorFlow Lite (1).....	29
Hình 34. Kết quả nhận diện với TensorFlow và TensorFlow Lite (2).....	29

Hình 35. Kết quả với hình ảnh nhận diện sai.....	30
Hình 36. Kết quả nhận diện khi hình ảnh đưa vào không phải là tiền.....	30
Hình 37. Thông tin về dữ liệu được lưu trữ trên Firebase từ GPS Module	31
Hình 38. Bản đồ hiển thị lịch sử đường đi cũng như vị trí hiện tại	32
Hình 39. Dữ liệu về khoảng cách đến vật cản	32
Hình 40. Chức năng đăng kí, đăng nhập.....	33
Hình 41 Chức năng hiển thị đường đi và vị trí	34
Hình 42. Chức năng hiển thị, chỉnh sửa thông tin cá nhân và đăng xuất.....	35
Hình 43. Mô hình hoàn chỉnh của đề tài.....	36

DANH MỤC BẢNG BIỂU

Bảng 1. Bảng đề xuất giải pháp tổng quan	1
Bảng 2. Bảng linh kiện sử dụng	3
Bảng 3. Bảng giá các linh kiện	4
Bảng 4. Kết quả khi huấn luyện với batch size là 8 và 16	23
Bảng 5. So sánh kết quả trước và sau tăng cường dữ liệu về độ sáng	24
Bảng 6. So sánh kết quả với TensorFlow và TensorFlow Lite	28
Bảng 7. So sánh kết quả trước và sau khi tối ưu thời gian	31

1. Giới thiệu**1.1. Thực trạng sản phẩm**

Hiện nay, rất nhiều các thiết bị được tạo ra nhằm mục đích hỗ trợ cho người khiếm thị nhờ khoa học và công nghệ phát triển nhanh chóng. Nhưng hầu hết các thiết bị đó vẫn chưa đáp ứng đầy đủ nhu cầu và tiện ích của người sử dụng. Vì vậy, nhóm chúng em đã đề xuất và phát triển một thiết bị (gậy) thông minh hỗ trợ người khiếm thị, tập trung vào giải quyết các thách thức mà người khiếm thị thường gặp phải. Thiết bị bao gồm các chức năng là nhận diện mệnh giá của tiền Việt Nam, định vị vị trí của người dùng và cảnh báo khi phát hiện vật cản trong quá trình di chuyển. Mục tiêu là tạo ra một công cụ hữu ích giúp người khiếm thị di chuyển an toàn và độc lập hơn, thuận tiện cho người thân trong việc theo dõi vị trí người dùng.

1.2. Các vấn đề cần giải quyết

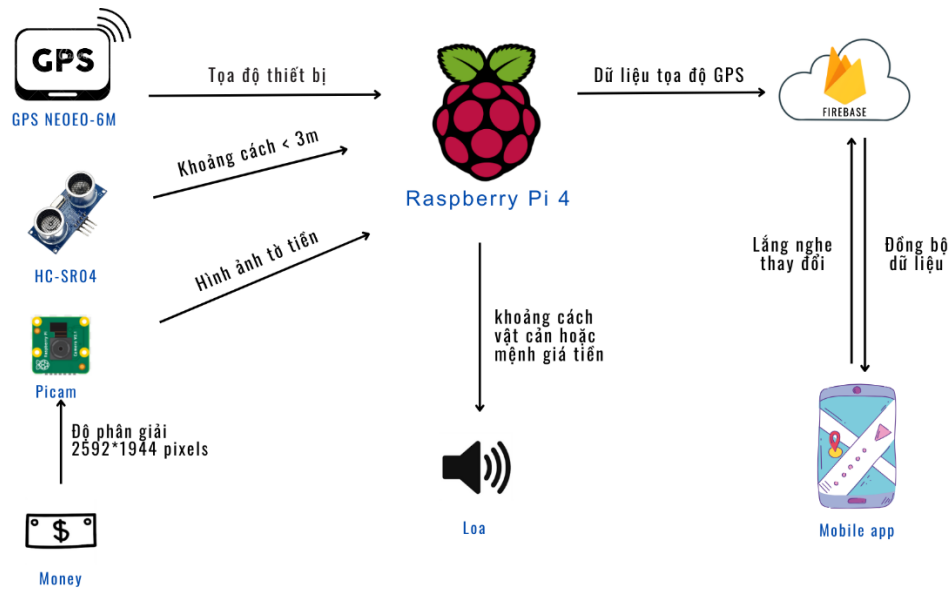
- Model để nhận diện mệnh giá tiền Việt Nam.
- Các thiết bị phần cứng để thu thập dữ liệu, lấy vị trí gps, phát hiện vật cản.
- Hệ thống để nhận và lưu trữ dữ liệu vị trí theo thời gian thực.
- Ứng dụng app để người thân theo dõi vị trí của người khiếm thị.

1.3. Đề xuất giải pháp tổng quan*Bảng 1. Bảng đề xuất giải pháp tổng quan*

Vấn đề	Giải pháp đề xuất
Phần cứng	Raspberry Pi 4 Pi Camera V1 Module NEO-6M GPS Module Cảm biến siêu âm HC-SR04
Nhận diện mệnh giá	Xây dựng và huấn luyện model, sử dụng model SSD MobileNetV2
Nhận và lưu trữ dữ liệu vị trí theo thời gian thực	Sử dụng Firebase
Ứng dụng app	Dùng Flutter để xây dựng app Người thân có thể đăng nhập để sử dụng ứng dụng Có chức năng xem lịch sử vị trí, vị trí hiện tại

2. Giải pháp

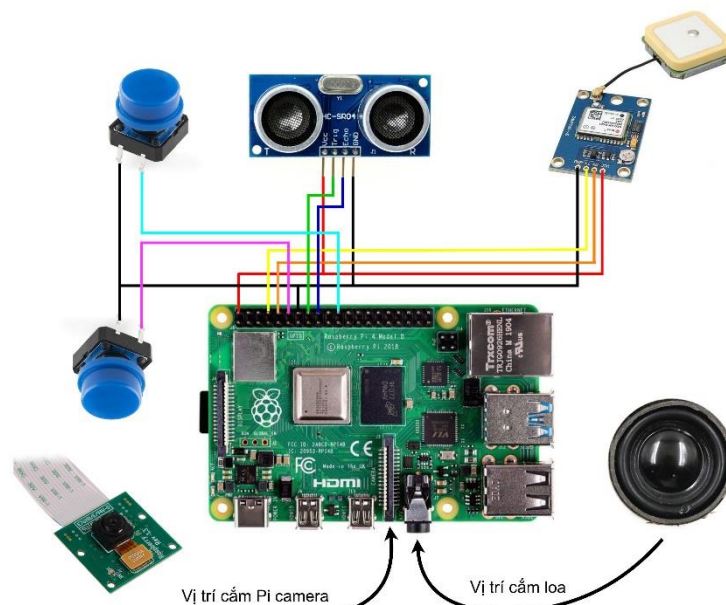
2.1. Tổng quan hệ thống



Hình 1. Sơ đồ tổng quan hệ thống





Hệ thống bao gồm PiCamera thực hiện chụp ảnh, cảm biến siêu âm phát tín hiệu để đo khoảng cách và Raspberry Pi4 xử lý nhận diện mệnh giá tiền, thực hiện đo khoảng cách và thông báo kết quả bằng loa. Ngoài ra GPS thông qua mạng không dây sẽ lưu tọa độ lên Cloud Firebase và ứng dụng di động sẽ lấy dữ liệu đó để hiển thị lên ứng dụng.

2.2. Giải pháp phần cứng



Hình 2. Sơ đồ lắp đặt phần cứng

Bảng 2. Bảng linh kiện sử dụng

Tên linh kiện	Hình ảnh	Thông số kỹ thuật
Raspberry Pi 4		Wi-Fi: 802.11b/g/n/ac (2.4 GHz và 5 GHz) Bluetooth 5.0 Nguồn 5V-3A cổng USB-C 1 cổng Ethernet 2 cổng Micro HDMI 1 jack audio/video 3.5mm 4 cổng USB 2.0 Khe cắm thẻ nhớ MicroSD
Pi CameraV1		Ống kính tiêu cự cố định Độ phân giải 5 megapixel Kích thước tối đa 3280 x 2464 Hỗ trợ video 1080p30, 720p60 và 640x480p90 Kích thước 25mm x 23mm x 9mm
Cảm biến siêu âm HC-SR04		Điện áp làm việc: 3.3V - 5VDC Khoảng cách phát hiện: 2cm - 450cm Độ chính xác: $\pm 3\text{mm}$ Góc đo được bao phủ: $<15^\circ$ Dòng điện tĩnh: $< 2\text{mA}$ Tần số hoạt động: 40Hz
Cảm biến GPS NEO-M6		Nguồn hoạt động: 3.3-5.5VDC Dòng hoạt động bình thường: 50 mA. Dòng hoạt động tiết kiệm: 30 mA. Giao tiếp UART/TTL Baud rate: Gồm nhiều mức khác nhau 1200, 2400, 4800, 19200, 38400, 9600 (mặc định), 57600, 115200,... Kích cỡ module : 39(W)X 25.5(H)mm.

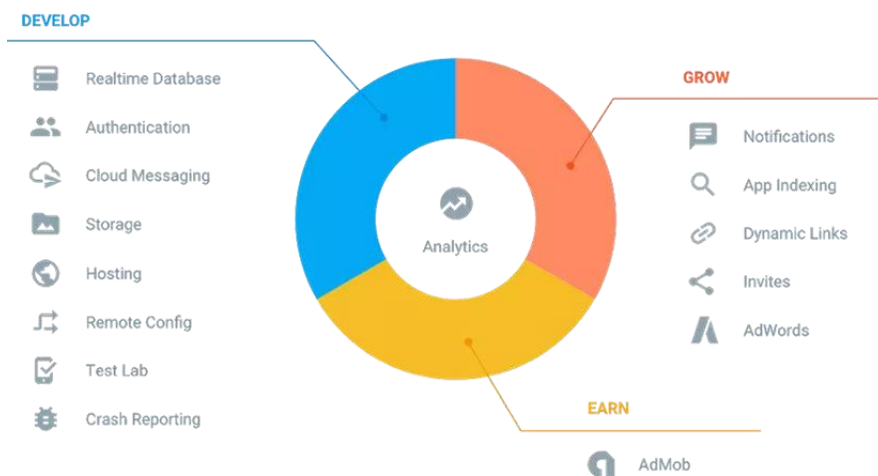
Bảng 3. Bảng giá các linh kiện

Tên linh kiện	Đơn giá	Ghi chú
1 x Raspberry Pi 4	2.134.000	Mua
1 x Pi Camera	85.000	Mua
1 x GPS NEO_M6	99.000	Mua
1 x Cảm biến siêu âm	20.000	Mua
1 x Thẻ nhớ 64GB	80.000	Mượn
1 x Vỏ hộp Raspberry Pi 3	140.000	Mua
1 x Cáp Micro HDMI	65.000	Mua
1 x Loa + mạch khuếch đại	30.000	Mượn
1 x Mạch sạc + 2 pin	30.000	Mua
	Thành tiền: 2.683.000	

2.3. Giải pháp truyền thông

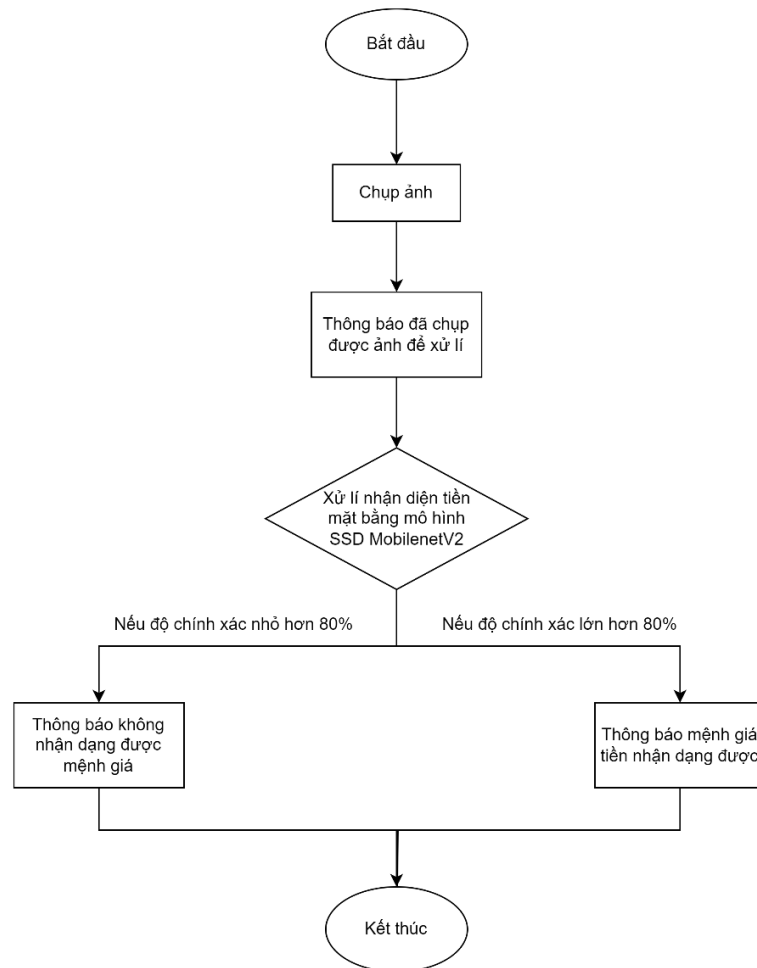
Việc giao tiếp giữa thiết bị phần cứng và phần mềm sẽ sử dụng giải pháp truyền thông là **Firestore**. Đây là một nền tảng hỗ trợ phát triển ứng dụng, đóng vai trò như một Server hay cơ sở hạ tầng Backend phức tạp. Nó cung cấp nhiều dịch vụ rộng lớn như:

- Realtime Database: Lưu trữ và đồng bộ ứng dữ liệu theo thời gian thực.
- Authentication: Quản lý việc đăng nhập, đăng ký và xác thực người dùng.
- Storage Database: Lưu trữ tệp tin, hình ảnh, ...
- Ngoài ra, còn có nhiều tính năng khác như thông báo đẩy, chức năng trò chuyện, ...

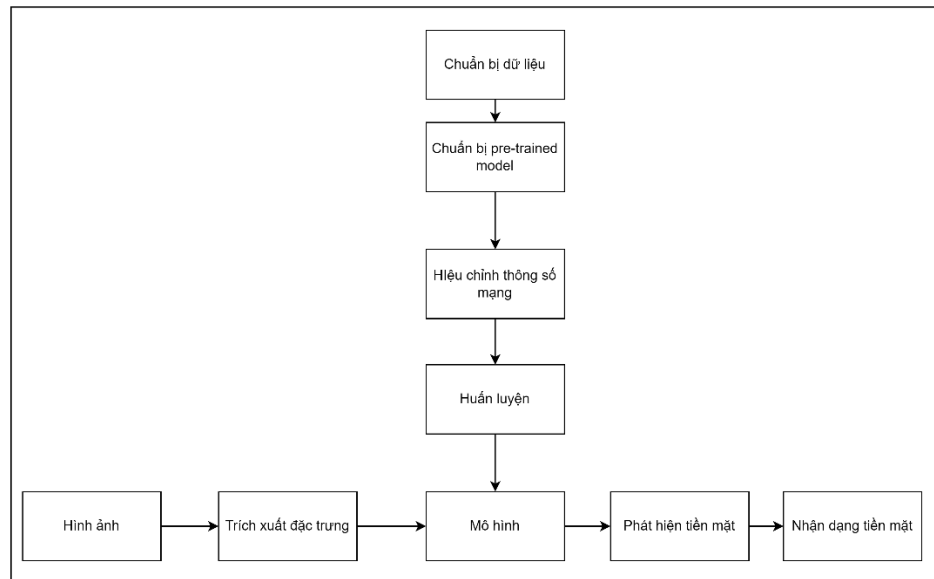
*Hình 3. Các dịch vụ của Firebase*

Trong phạm vi đồ án, nhóm sẽ sử dụng 2 dịch vụ chính được Firebase hỗ trợ là Authentication và Realtime Database. Authentication để quản lý người dùng và xác thực người dùng. Realtime Database dùng để lưu trữ dữ liệu theo thời gian thực bao gồm thông tin người dùng và dữ liệu tọa độ của thiết bị phần cứng.

2.4. Giải pháp nhận diện mệnh giá tiền



Hình 4. Sơ đồ hoạt động của giải pháp nhận diện mệnh giá tiền

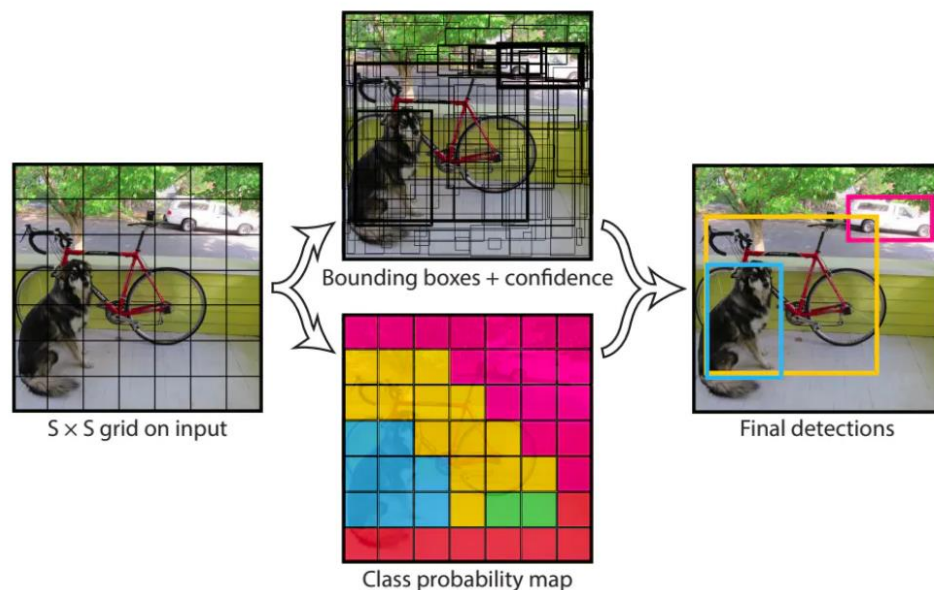


Hình 5. Quá trình thực hiện giải pháp nhận diện mệnh giá tiền

2.4.1. Mô hình SSD MobilenetV2

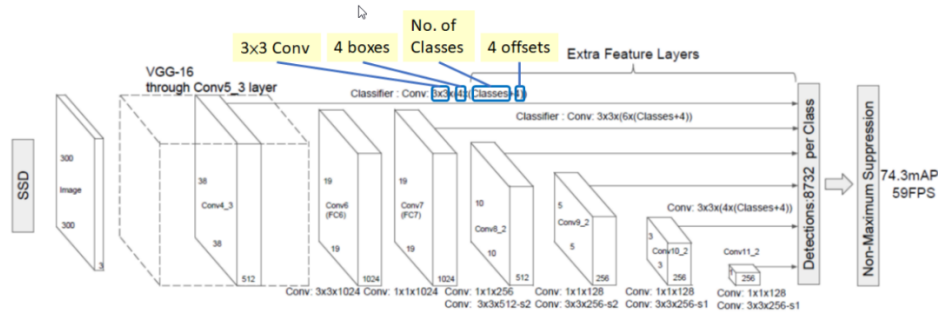
2.4.1.1. Single Shot Detection (SSD)

- SSD là một mạng Object Detect nhận đầu vào là 1 bounding box của vật thể và nhãn của vật thể chứa trong bounding box đó.
- SSD có tốc độ xử lý nhanh là nhờ vào việc chỉ sử dụng 1 mạng duy nhất: gói gọn quá trình phát hiện và phân loại đối tượng chỉ trong 1 bước.



Hình 6. Cách hoạt động của SSD

2.4.1.1.1. Kiến trúc mạng



Hình 7. Kiến trúc mạng của SSD

- Trích xuất feature map (dựa vào mạng cơ sở VGG16) để tăng hiệu quả trong việc phát hiện.
- Áp dụng các bộ lọc tích chập để có thể phát hiện được các đối tượng.

2.4.1.1.2. MultiBox

- Hàm lỗi của MultiBox là sự kết hợp của 2 thành phần tương ứng với 2 chức năng của SSD:

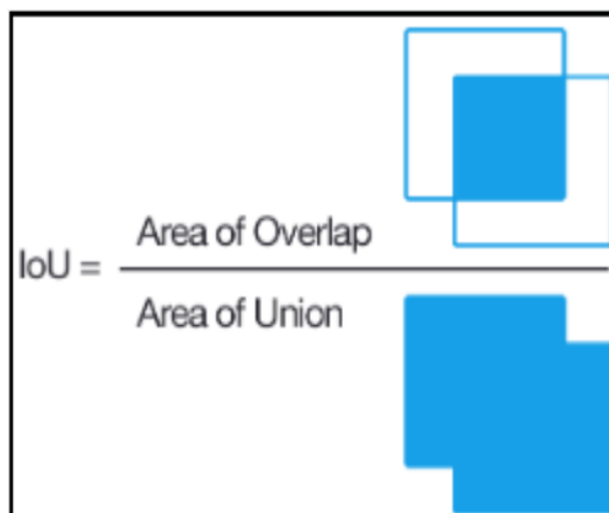
$$multibox_loss = confidence_loss + \alpha * location_loss$$

- **Confidence Loss:** Tính toán tỉ lệ rơi vào class mà bounding box được tính toán. Độ đo cross-entropy được sử dụng để đo thành phần này.
- **Location Loss:** Ước lượng sự sai lệch thực tế của bounding box so với tập dữ liệu mẫu. L2-Norm được sử dụng ở đây.
- **Giá trị alpha:** giúp cân bằng được sự ảnh hưởng của location loss. Mục tiêu của là tìm những giá trị tham số có thể tối thiểu được hàm Loss tốt nhất, theo đó đưa ra được những dự đoán càng gần với dữ liệu mẫu.

2.4.1.1.3. MultiBox Prior and IoU

Priors:

- Là những bounding box được tính toán trước với kích thước cố định tuân theo phân phối gần với phân phối của bounding box mẫu. Trong thực tế, những priors này được lựa chọn khi tỉ lệ Intersection/Union (IoU) lớn hơn ngưỡng 0.5



- Gồm 11 priors cho mỗi ô của các feature map (8x8, 6x6, 4x4, 3x3, 2x2) và 1 prior cho feature map 1x1, tạo tổng cộng 1420 priors cho mỗi hình ảnh. Điều này giúp bao phủ hoàn toàn ảnh đầu vào ở các tỉ lệ khác nhau, dễ dàng nhận diện đối tượng với nhiều kích thước khác nhau.

2.4.1.2. MobileNetV2

2.4.1.2.1. Tổng quan

- Là một kiến trúc CNN được các nhà nghiên cứu tại Google phát triển vào năm 2017.
- Tổng tham số tiêu chuẩn là 4,2 triệu, ít hơn đáng kể so với một số kiến trúc CNN khác như AlexNet, VGGNet,...
- Sử dụng các kỹ thuật như Depthwise Separable Convolution để giảm lượng tính toán và số lượng tham số, đồng thời vẫn duy trì độ chính xác trong việc nhận dạng đối tượng.

Table 4. Depthwise Separable vs Full Convolution MobileNet

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7%	4866	29.3
MobileNet	70.6%	569	4.2

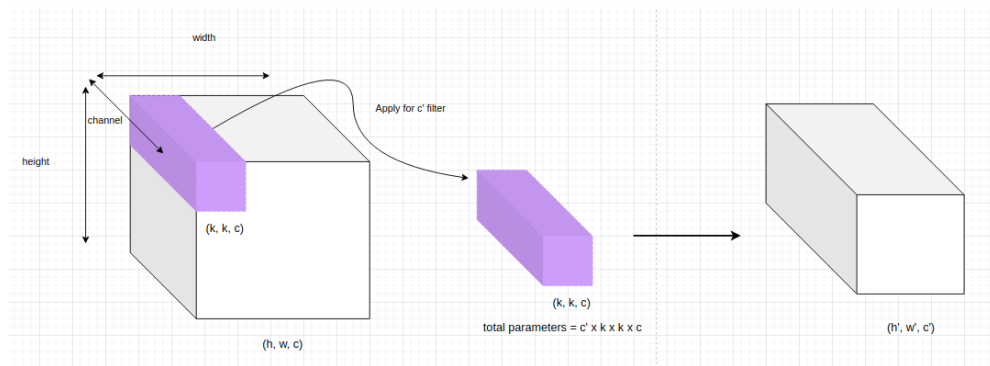
Hình 8. So sánh giữa Depthwise Separable và Full Convolution MobileNet

2.4.1.2.2. Tích chập 2 chiều

Tích chập 2 chiều thông thường: tính toán trên toàn bộ chiều sâu c (channel). Do đó số lượng tham số của mô hình sẽ gia tăng đáng kể phụ thuộc vào độ sâu của layer trước đó.

- Số lượng tham số khi dẫn về những layers cuối cùng sẽ là một con số không hề nhỏ, tạo ra những mô hình cồng kềnh làm chiếm dụng bộ nhớ và ảnh hưởng tới tốc độ tính toán.

- Alexnet và VGGNet là những mô hình điển hình có số lượng tham số rất lớn do chỉ áp dụng những tích chập 2 chiều thông thường.

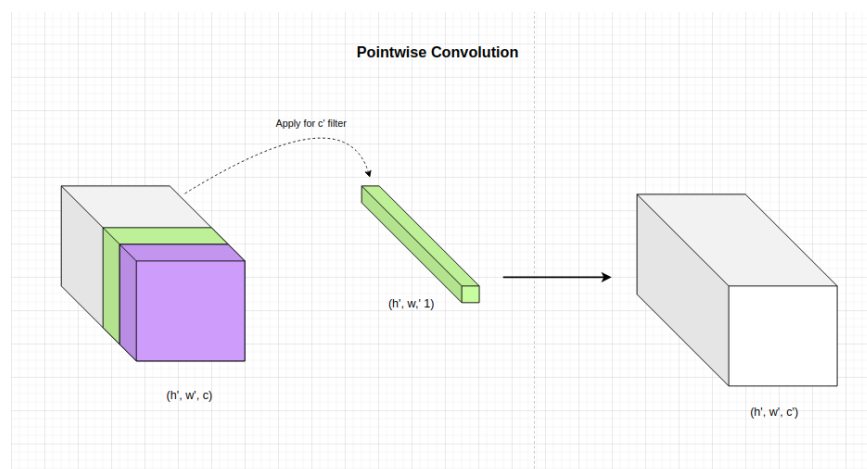


Hình 9. Tích chập 2 chiều thông thường

2.4.1.2.3. Tích chập tách biệt chiều sâu (Depthwise Separable Convolution).

Chia làm 2 bước tuần tự:

- Tích chập chiều sâu (Depthwise Convolution):** chia khối input tensor3D thành những lát cắt ma trận theo độ sâu. Thực hiện tích chập trên từng lát cắt
- Tích chập điểm (Pointwise Convolution):** một phép tích chập với bộ lọc có kích thước là 1×1 .
- Giảm số lượng tính toán, giảm số lượng params, đồng thời có thể thực hiện trích xuất đặc trưng một cách tách biệt trên các channel khác nhau.



Hình 10. Tích chập điểm

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Hình 11. So sánh với VGG16 và GoogleNet

Nhận xét: MobileNet đạt được độ chính xác tương đương nhưng lại thực hiện số phép tính toán và có số lượng tham số ít hơn nhiều so với GoogleNet và VGG 16

2.4.1.2.4. MobileNetV2

MobileNet V2 tiếp tục sử dụng Depthwise Separable Convolutions, ngoài ra còn có thêm:

- Linear bottlenecks: giữ cho thông tin truyền qua các lớp phi tuyến tính không bị mất, giúp cải thiện hiệu suất của mô hình.
- Inverted Residual Block (shortcut connections giữa các bottlenecks): giúp mô hình học được các biểu diễn phức tạp hơn trong khi vẫn giữ được số lượng tham số thấp.

Model	Params	Multiply-Adds	mAP	Mobile CPU
MobileNetV1 + SSDLite	5.1M	1.3B	22.2%	270ms
MobileNetV2 + SSDLite	4.3M	0.8B	22.1%	200ms

Hình 12. So sánh giữa MobileNetV1 và MobileNetV2

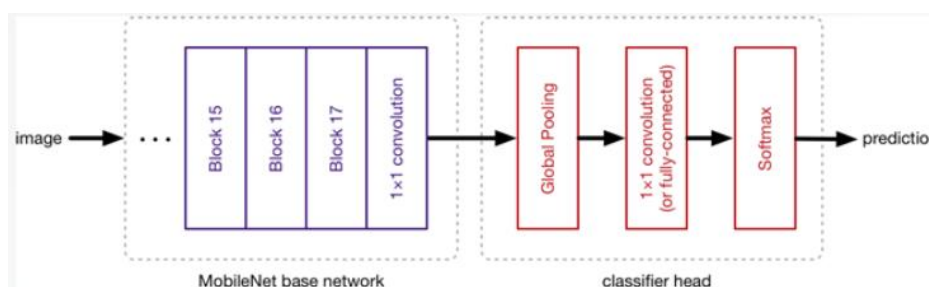
Nhận xét: MobileNetV2 có hiệu suất vượt trội hơn MobileNetV1 và mô hình tương đương với kích thước model và chi phí tính toán tương đương cùng với thời gian xử lý được tăng tốc hơn rất nhiều

2.4.1.2.5. SSD MobileNetV2

Khi dùng MobileNetV2 làm mô hình cơ sở trong SSD, chúng ta có một mô hình Object Detection vừa nhỏ gọn vừa chính xác và các cấu trúc tích chập có thể phân tách theo chiều sâu mới lạ.

Về bản chất, mạng cơ sở MobileNetV2 hoạt động như một trình trích xuất đặc trưng cho lớp SSD, sau đó lớp này sẽ phân loại đối tượng quan tâm.

Mạng MobileNetV2 sử dụng các tích chập phân tách theo chiều sâu và sau đó sử dụng mạng Single Shot Detector (SSD) để phát hiện



Hình 13. Mạng SSD MobileNetV2

“320x320” (px) là kích thước hình ảnh đầu vào mà mô hình mong đợi.

SSD MobileNet V2 320x320 (px) mang lại sự cân bằng tốt giữa tốc độ và độ chính xác, giúp phù hợp với các ứng dụng thời gian thực trên các thiết bị có hạn chế về tài nguyên.

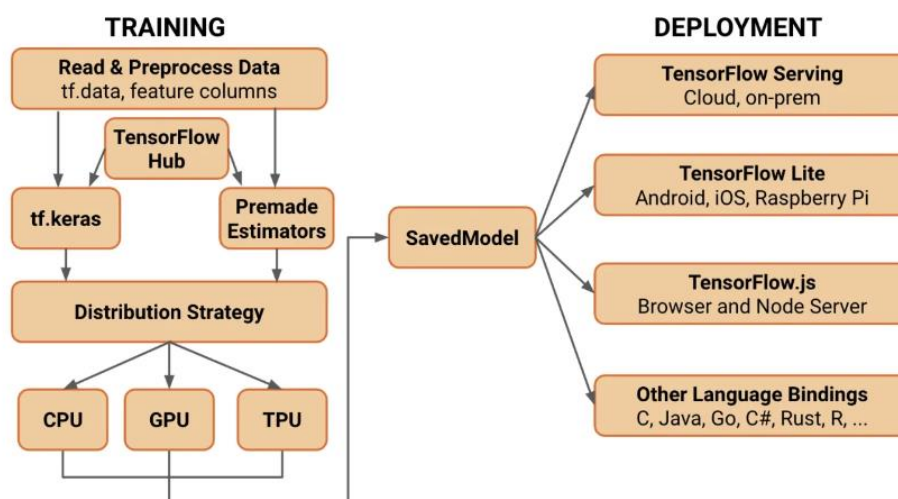
2.4.2. TensorFlow và TensorFlow Lite

2.4.2.1. TensorFlow

TensorFlow là thư viện phần mềm mã nguồn mở và miễn phí được Google phát triển dành riêng cho các ứng dụng học máy và trí tuệ nhân tạo (AI). Mặc dù nó có thể được sử dụng cho nhiều nhiệm vụ học máy khác nhau, nhưng dịch vụ này đặc biệt xuất sắc trong việc đào tạo và triển khai mạng lưới nơ-ron (neural networks). Những thuật toán phức tạp này được lấy cảm hứng từ cấu trúc và chức năng của bộ não con người và chúng là nguồn cung cấp năng lượng cho nhiều hệ thống thông minh mà chúng ta tương tác ngày nay.

Chức năng cốt lõi của TensorFlow nằm ở khả năng xây dựng và đào tạo các mô hình học máy thông qua các công cụ sau:

- **Lập trình luồng dữ liệu:** Sử dụng biểu đồ luồng dữ liệu để biểu diễn các tính toán trong mô hình học máy, xác định luồng dữ liệu giữa các hoạt động, cho phép xây dựng các mô hình phức tạp theo từng bước.
- **Hoạt động với Tensor:** Làm việc với các mảng đa chiều (tensor) và cung cấp thư viện các phép toán cho tensor, cho phép tính toán hiệu quả trên các tập dữ liệu lớn.
- **Tự động phân biệt:** Cung cấp khả năng phân biệt tự động để tính toán độ dốc, giúp điều chỉnh các tham số mô hình mà không cần thực hiện thủ công.
- **Tối ưu hóa thuật toán:** Tích hợp nhiều thuật toán tối ưu hóa để điều chỉnh các tham số mô hình dựa trên dữ liệu huấn luyện, giảm thiểu sai sót và cải thiện độ chính xác.



Hình 14. Hệ thống TensorFlow

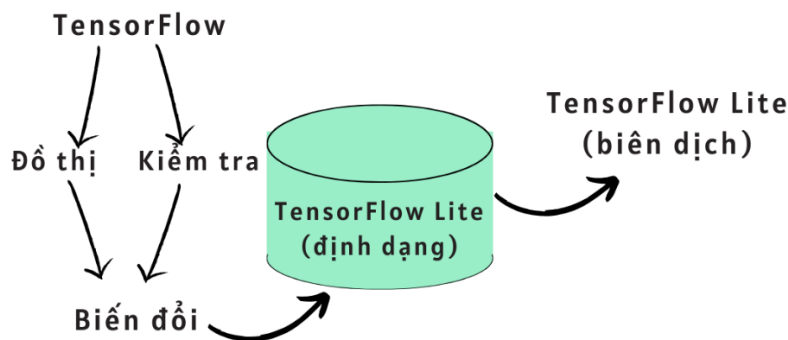
2.4.2.2. TensorFlow Lite

TensorFlow Lite là một phiên bản nhỏ gọn của thư viện machine learning TensorFlow, được tối ưu để chạy trên các thiết bị có tài nguyên hạn chế như điện thoại di động, máy tính nhúng và IoT. Với TensorFlow Lite, người dùng có thể chạy các mô hình machine learning trực tiếp trên các thiết bị của mình mà không cần kết nối mạng, giúp đạt được thời gian phản hồi nhanh và bảo mật dữ liệu trên thiết bị.

- **Tối ưu hóa và triển khai model với TensorFlow Lite**

TensorFlow Lite cung cấp các công cụ tối ưu hóa để giảm kích thước của model và tăng hiệu suất của nó trên các thiết bị nhúng. Các kỹ thuật tối ưu hóa bao gồm nén model, định dạng dữ liệu công ty và việc sử dụng kernel tối ưu cho các thiết bị có tài nguyên hạn chế. Điều này giúp công việc triển khai model trên các thiết bị tăng cường hiệu suất và hiệu quả hơn.

Triển khai trong hệ thống như sau:



Hình 15. Tổng quan mô hình trong hệ thống

⇒ Sử dụng Tensorflow ở giai đoạn huấn luyện, Tensorflow Lite ở giai đoạn dự đoán cho chức năng nhận diện mệnh giá trực tiếp trên Raspberry Pi 4.

2.5. Giải pháp GPS

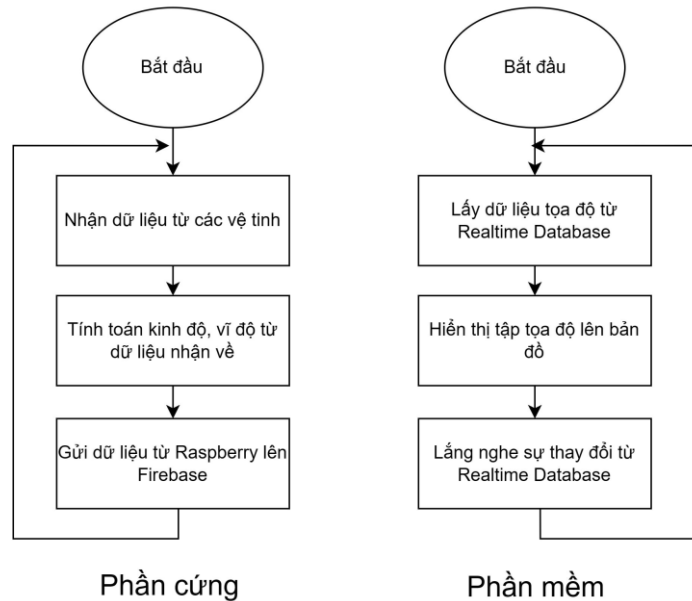
2.5.1. Quá trình thực hiện giải pháp

GPS (Global Positioning System) hay còn gọi là Hệ thống định vị toàn cầu. Đây là hệ thống vệ tinh được phát triển và vận hành bởi Bộ Quốc phòng Hoa Kỳ, cung cấp dịch vụ định vị, dẫn đường và đo thời gian miễn phí cho người dùng trên toàn thế giới. Trong phạm vi đề án thì nhóm sử dụng GPS để định vị vị trí của thiết bị bằng dữ liệu về kinh độ và vĩ độ.

Về phần cứng, thiết bị sẽ sử dụng module định vị GPS NEO-6M để nhận tín hiệu từ vệ tinh truyền về trong điều kiện thời tiết quang đăng, ít mây (tránh sử dụng trong nhà).

Về phần mềm, ứng dụng Android sẽ sử dụng một API google map để có thể vẽ bản đồ lên ứng dụng. Ở đây nhóm lấy API từ nhà cung cấp bản đồ trực tuyến Mapbox và sử dụng Mapbox_gl trên ứng dụng để vẽ bản đồ từ API đó.

2.5.2. Tổng quan về hệ thống của giải pháp GPS



Hình 16. Tổng quan về hệ thống của giải pháp GPS

Thiết bị phần cứng:

Cách hoạt động của module định vị Module GPS NEO-6M:

- **Thu tín hiệu từ vệ tinh:** Module nhận tín hiệu từ ít nhất 3 vệ tinh để tính toán vị trí chính xác (hệ thống cần kinh độ và vĩ độ).
- **Giải mã tín hiệu:** Tín hiệu nhận được từ vệ tinh chứa thông tin về vị trí và thời gian của vệ tinh.
- **Tính toán vị trí:** Dựa trên thời gian truyền tín hiệu từ vệ tinh đến module và vị trí của vệ tinh, module tính toán vị trí hiện tại (kinh độ, vĩ độ).
- **Cung cấp dữ liệu:** Dữ liệu định vị được gửi qua giao tiếp UART để các hệ thống xử lý và sử dụng.

Sau khi lấy được dữ liệu về vị trí từ module, hệ thống sẽ lấy thông tin về kinh độ và vĩ độ để đẩy lên Firebase

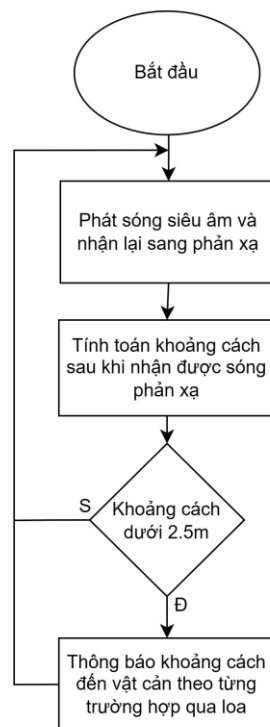
Ứng dụng Android:

- **Mã thiết bị:** Mỗi người dùng được cấp một mã thiết bị duy nhất khi đăng nhập vào ứng dụng. Mã này có vai trò như ID để xác định thiết bị tương ứng của họ.
- **Kết nối với Firebase Realtime Database:** Sau khi đăng nhập, ứng dụng kết nối với Firebase Realtime Database, một cơ sở dữ liệu đám mây lưu trữ thông tin theo thời gian thực.
- **Lấy tọa độ thiết bị:** Ứng dụng truy vấn Firebase Realtime Database để lấy tập tọa độ được liên kết với mã thiết bị của người dùng. Tập tọa độ này đại diện cho

vị trí hiện tại và trước đây của thiết bị.

- **Hiển thị tọa độ trên bản đồ:** Ứng dụng hiển thị các điểm dữ liệu vị trí từ tập tọa độ lên bản đồ tích hợp trong ứng dụng. Nhờ vậy, người dùng có thể dễ dàng theo dõi vị trí của thiết bị theo thời gian.
- **Cập nhật đồng bộ:** Khi thiết bị phần cứng gửi dữ liệu vị trí mới, Firebase Realtime Database tự động cập nhật thông tin trong cơ sở dữ liệu. Nhờ tính năng đồng bộ thời gian thực, ứng dụng sẽ tự động cập nhật để hiển thị vị trí mới nhất của thiết bị lên bản đồ

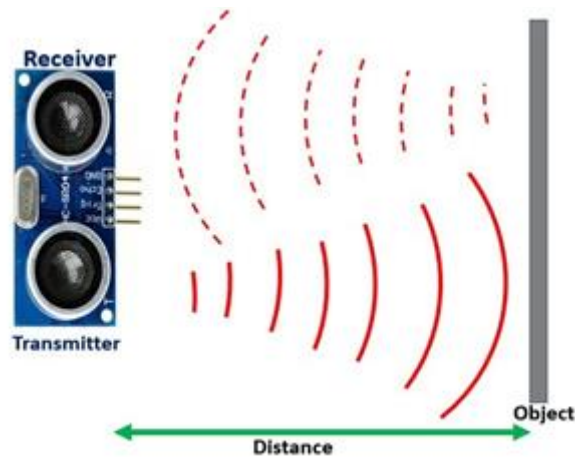
2.6. Giải pháp phát hiện vật cản



Hình 17. Sơ đồ hoạt động của giải pháp phát hiện vật cản

Hệ thống sử dụng cảm biến siêu âm HC-SR04 cho chức năng phát hiện vật cản.

Cảm biến siêu âm HC-SR04 bao gồm một bộ phát sóng siêu âm (chân trig) và một bộ thu sóng siêu âm (chân echo). Khi hoạt động, bộ phát sóng sẽ phát ra một chùm sóng siêu âm, sóng này sẽ dội lại khi gặp vật cản và được bộ thu sóng nhận lại. Thời gian từ lúc phát sóng đến lúc nhận sóng phản hồi được sử dụng để tính khoảng cách đến vật cản.



Hình 18. Nguyên lý hoạt động của cảm biến siêu âm

Tính khoảng cách: $d = V \cdot t / 2$ ($V = 340 \text{ m/s} = 34000 / 10^6 \text{ cm/micro giây}$) ($t/2$ vì thời gian xung chạm tới vật cản rồi phản xạ về ($s = 2d$))

Sau khi phát hiện được vật cản và tính toán được khoảng cách đến vật cản, hệ thống sẽ phát ra 1 tín hiệu âm thanh cảnh báo phù hợp thông qua loa.

2.7. Giải pháp về ứng dụng di động

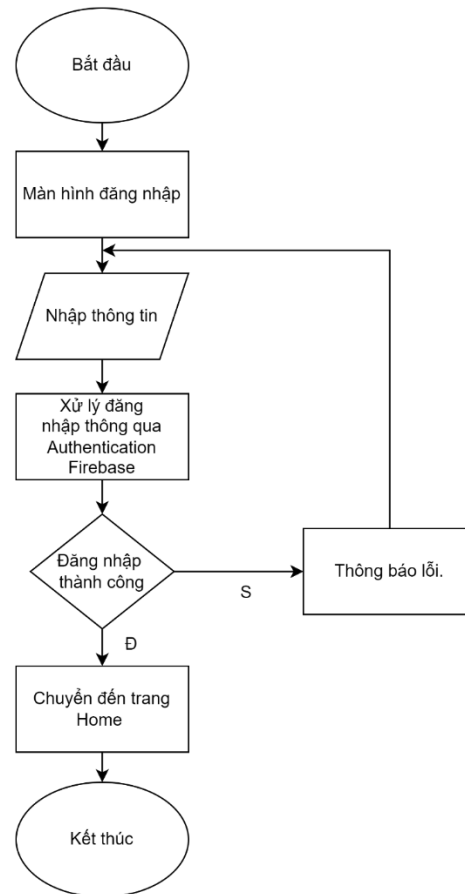
2.7.1. Phát biểu bài toán

Xây dựng ứng dụng GPS Tracker dành cho người thân hoặc người quản lý để giúp họ có thể theo dõi vị trí của người khiếm thị. Ứng dụng tập trung vào chức năng theo dõi vị trí cũng như lịch sử đường đi của người khiếm thị, nhằm giúp người thân hoặc người quản lý có khả năng theo dõi và cung cấp sự hỗ trợ khi cần thiết.

2.7.2. Công nghệ sử dụng

- IDE: Android Studio
- Framework: Flutter
- Database: Lưu trữ trên Realtime Database Firebase
- Device: điện thoại Android

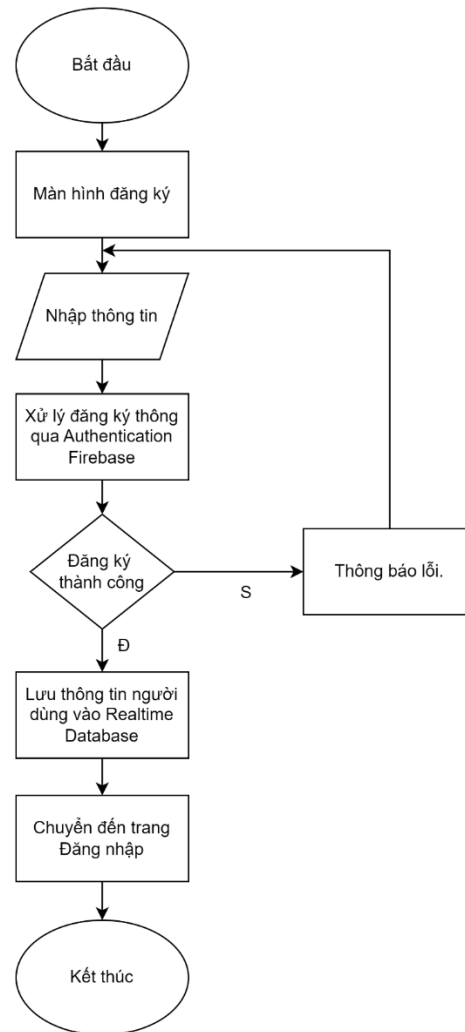
2.7.3. Sơ đồ hoạt động



Hình 19. Sơ đồ hoạt động trang đăng nhập

Nguyên lí hoạt động:

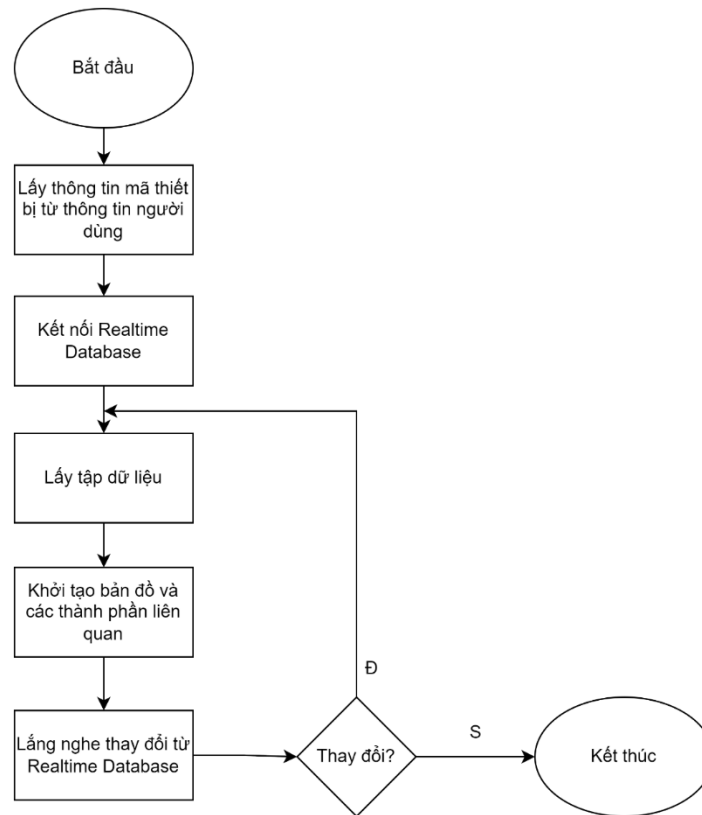
Khi người dùng đăng nhập ứng dụng, hệ thống sẽ kiểm tra thông tin người dùng thông qua dịch vụ Authentication được cung cấp từ Firebase. Nếu thông tin hợp lệ sẽ chuyển người dùng đến trang Home, nếu không sẽ thông báo lỗi.



Hình 20. Sơ đồ hoạt động trang đăng ký

Nguyên lý hoạt động:

Khi người dùng đăng kí tài khoản, hệ thống sử dụng dịch vụ Authentication được Firebase cung cấp để thực hiện quá trình đăng kí người dùng cũng như kiểm tra xác thực người dùng. Nếu đăng ký thành công sẽ chuyển người dùng đến trang đăng nhập, nếu không sẽ thông báo lỗi.



Hình 21. Sơ đồ hoạt động trang chính

Nguyên lý hoạt động:

Sau khi người dùng đăng nhập vào ứng dụng, hệ thống sẽ lấy ra mã thiết bị tương ứng của người dùng và kết nối với dịch vụ Realtime Database từ Firebase để lấy tập dữ liệu tọa độ tương ứng. Từ đó khởi tạo bản đồ và hiển thị các điểm tọa độ đó lên trên bản đồ. Đồng thời sẽ lắng nghe sự thay đổi từ Realtime Database, khi thiết bị phần cứng gửi tọa độ vị trí mới thì ứng dụng sẽ thực hiện lại quá trình lấy tập dữ liệu và khởi tạo bản đồ.

2.7.4. Cơ sở dữ liệu

Sử dụng Firebase để lưu trữ dữ liệu bao gồm tài khoản, thông tin người dùng cũng như tọa độ thiết bị phần cứng.

- Sử dụng dịch vụ Authentication để đăng ký, xác thực tài khoản người dùng

Search by email address, phone number or user UID					Add user
Identifier	Providers	Created ↓	Signed in	User UID	
thaihuynh31220030913...	📧	18 May 2024	18 May 2024	stqHDis6P2dnAvHdxQ6pSdyje...	
npbl2009@gmail.com	📧	18 May 2024	8 Jun 2024	3sXuiqqTMneG9QtnH8yUzBN...	
hung01092003@gmail....	📧	17 May 2024	17 May 2024	DLgGOYvtiyNcp4T0eLXr655mj...	
phihung25052007@gm...	📧	15 May 2024	15 May 2024	7TUueVB0FScmLQKgiSQCNi1...	

Hình 22. Đăng ký, xác thực người dùng bằng Authentication Firebase

- Sử dụng dịch vụ Realtime Database để lưu trữ thông tin người dùng và dữ liệu tọa độ thiết bị phân cứng.



Hình 23. Lưu thông tin người dùng trên Realtime Database



Hình 24. Lưu tọa độ thiết bị trên Realtime Database

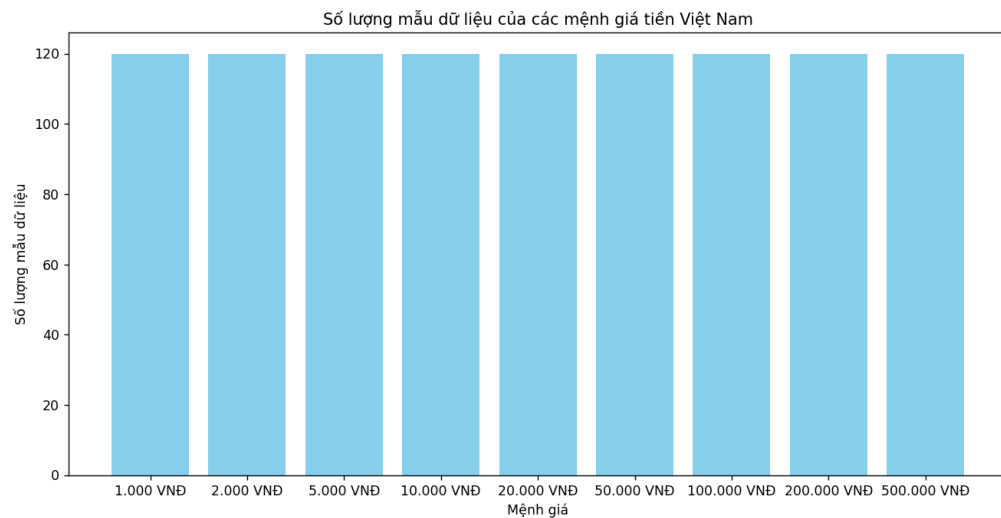
3. Kết quả

3.1. Nhận diện mệnh giá

3.1.1. Tập dữ liệu

3.1.1.1. Thu thập dữ liệu lần 1

- Tập dữ liệu bao gồm ảnh chụp của 9 mệnh giá tiền Việt Nam bằng Pi Camera V1 với độ phân giải là 2592*1944 pixels



- Nhóm thu thập mỗi mệnh giá là 120 ảnh với đầy đủ các trường hợp là chiều ngược xuôi, mặt trước, mặt sau của mỗi tờ tiền. Tổng tập dữ liệu thu được 1080 mẫu dữ liệu.



- Sau khi tổng kết lại có thể thấy ảnh được chụp trong điều kiện ánh sáng được đảm bảo.
- Dùng **chương trình labelImg** để gắn nhãn cho tất cả tiền ở trong dataset để cho ra tọa độ x, y, weight, height bằng hình chữ nhật



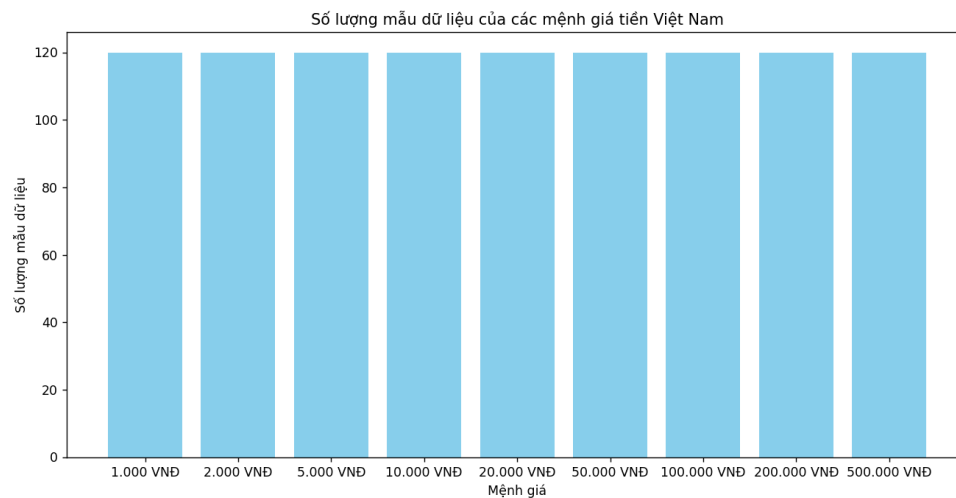
- Sau khi gắn nhãn hết tất cả mẫu dữ liệu, chia tập dữ liệu ra thành 3 phần là train, test và validation với tỉ lệ 6/2/2 với đều các trường hợp về mệnh giá và các mặt của tờ tiền, ta có:
 - **Tập train:** 648 mẫu dữ liệu (72 mẫu mỗi mệnh giá)
 - **Tập validation:** 216 mẫu dữ liệu (24 mẫu mỗi mệnh giá)
 - **Tập test:** 216 mẫu dữ liệu (24 mẫu mỗi mệnh giá)
 - Và thu thập thêm **một tập ảnh** gồm có 71 hình được chụp trong điều kiện ánh sáng khác nhau để kiểm thử thêm nhằm đánh giá mô hình chuẩn xác hơn
- Từ ảnh đầu vào và file csv là file tổng hợp label và các thành phần quan trọng của từng mẫu sẽ được trích xuất đặc trưng thành định dạng record trong TensorFlow.

```
!python generate_tfrecord.py --image_dir=/content/gdrive/MyDrive/data/split_data/train \
--csv_input=/content/gdrive/MyDrive/data/split_data/train_labels.csv \
--output_path=/content/gdrive/MyDrive/data/tfrecord_data/train.record
!python generate_tfrecord.py --image_dir=/content/gdrive/MyDrive/data/split_data/validation \
--csv_input=/content/gdrive/MyDrive/data/split_data/test_labels.csv \
--output_path=/content/gdrive/MyDrive/data/tfrecord_data/validation.record
```

- Sau khi trích xuất đặc trưng dữ liệu trong tập huấn luyện và kiểm thử, bước tiếp theo là tiến hành huấn luyện model.

3.1.1.2. Thu thập dữ liệu lần 2

- Thay thế những ảnh bị tương đồng trong tập train và validation thành những ảnh được chụp trong độ sáng tối ở tất cả các mệnh giá để tập dữ liệu đa dạng về độ sáng nhằm cải thiện được khả năng nhận dạng ở các điều kiện ánh sáng khác nhau:



- Một vài hình ảnh được thêm vào tập dữ liệu



- Sử dụng tập 71 ảnh được chụp ở các điều kiện ánh sáng khác nhau làm tập test để đánh giá lại model
- Sau khi trích xuất đặc trưng tập dataset mới để train và validation thì sẽ tiến hành huấn luyện model.

3.1.2. Huấn luyện với mô hình SSD MobileNetV2

3.1.2.1. Huấn luyện với tập dữ liệu lần 1

Ảnh sẽ được resize về 320*320 và chọn nhiều batch-size cùng với tham số huấn luyện mặc định để huấn luyện và đánh giá độ chính xác trên tập test với num_steps là 1000

Mô hình đánh giá độ mất mát dựa trên các tiêu chí:

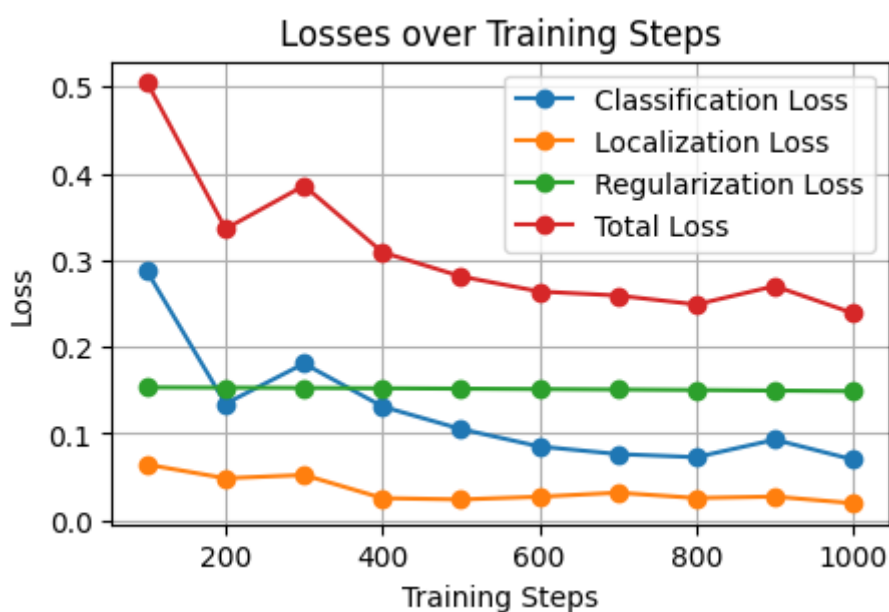
- **Classification Loss** (Loss của phân loại): Được sử dụng trong các bài toán phân loại, nơi mà mô hình cần phải dự đoán lớp của đối tượng.
- **Localization Loss** (Loss của vị trí): Được sử dụng trong các bài toán localization, nơi mà mô hình cần phải dự đoán vị trí chính xác của đối tượng trong hình ảnh hoặc video.
- **Regularization Loss** (Loss của điều chuẩn): Được sử dụng để giảm overfitting bằng cách kiểm soát độ phức tạp của mô hình.

- **Total Loss (Tổng Loss):** Là tổng hợp của tất cả các loại loss trên.

Bảng 4. Kết quả khi huấn luyện với batch size là 8 và 16

Batch size	Accuracy	Total Loss	Classification Loss	Localization Loss	Regularization Loss
8	75.93%	0.31	0.12	0.04	0.17
16	71.75%	0.24	0.08	0.02	0.15

Nhận xét: Dù độ chính xác trên batch-size 8 cao hơn nhưng các thông số về mất mát lại cao hơn so với batch-size 16. Và batch-size 16 có được khả năng nhận diện tốt ở nhiều mệnh giá hơn nên chọn batch-size 16 để tiếp tục huấn luyện và tăng cường dữ liệu để tăng độ chính xác của mô hình.



Hình 25. Đồ thị mất mát của batch-size là 16

	precision	recall	f1-score	support
1000	1.00	0.33	0.50	24
2000	0.56	1.00	0.72	24
5000	1.00	0.04	0.08	24
10000	0.67	1.00	0.80	24
20000	0.47	1.00	0.64	24
50000	1.00	1.00	1.00	24
100000	1.00	0.88	0.93	24
200000	1.00	0.21	0.34	24
500000	0.89	1.00	0.94	24
accuracy			0.72	216
macro avg	0.84	0.72	0.66	216
weighted avg	0.84	0.72	0.66	216

Hình 26. Độ chính xác trên tập test của batch-size 16

Tiến hành tăng cường dữ liệu với batch-size 16 về ‘random_crop_image’ và ‘random_adjust_brightness’ để cắt ảnh ngẫu nhiên và tăng cường về độ sáng và tăng cường số lần huấn luyện lên thành 2000 num_steps.


```
data_augmentation_options {  
  random_crop_image {  
    min_object_covered: 0.1  
    min_aspect_ratio: 0.5  
    max_aspect_ratio: 2.0  
    min_area: 0.5  
    max_area: 1.0  
    overlap_thresh: 0.0  
  }  
}  
}  
type: replicas: true
```

- **Random_crop_image:** Cắt ngẫu nhiên ảnh để tạo ra các phiên bản mới.
 - **min_object_covered:** Xác định tỷ lệ tối thiểu của đối tượng cần được bao phủ trong hình ảnh cắt.
 - **min_aspect_ratio, max_aspect_ratio:** Tỷ lệ khung hình (chiều rộng / chiều cao) tối thiểu/tối đa của hình ảnh cắt.
 - **min_area, max_area:** Tỷ lệ diện tích tối thiểu/tối đa của hình ảnh cắt so với hình ảnh gốc.
 - **overlap_thresh:** Ngưỡng chồng lấn tối thiểu cần thiết giữa đối tượng và hình ảnh cắt để giữ lại hình ảnh cắt.
- **Random_adjust_brightness:** Điều chỉnh độ sáng ngẫu nhiên.

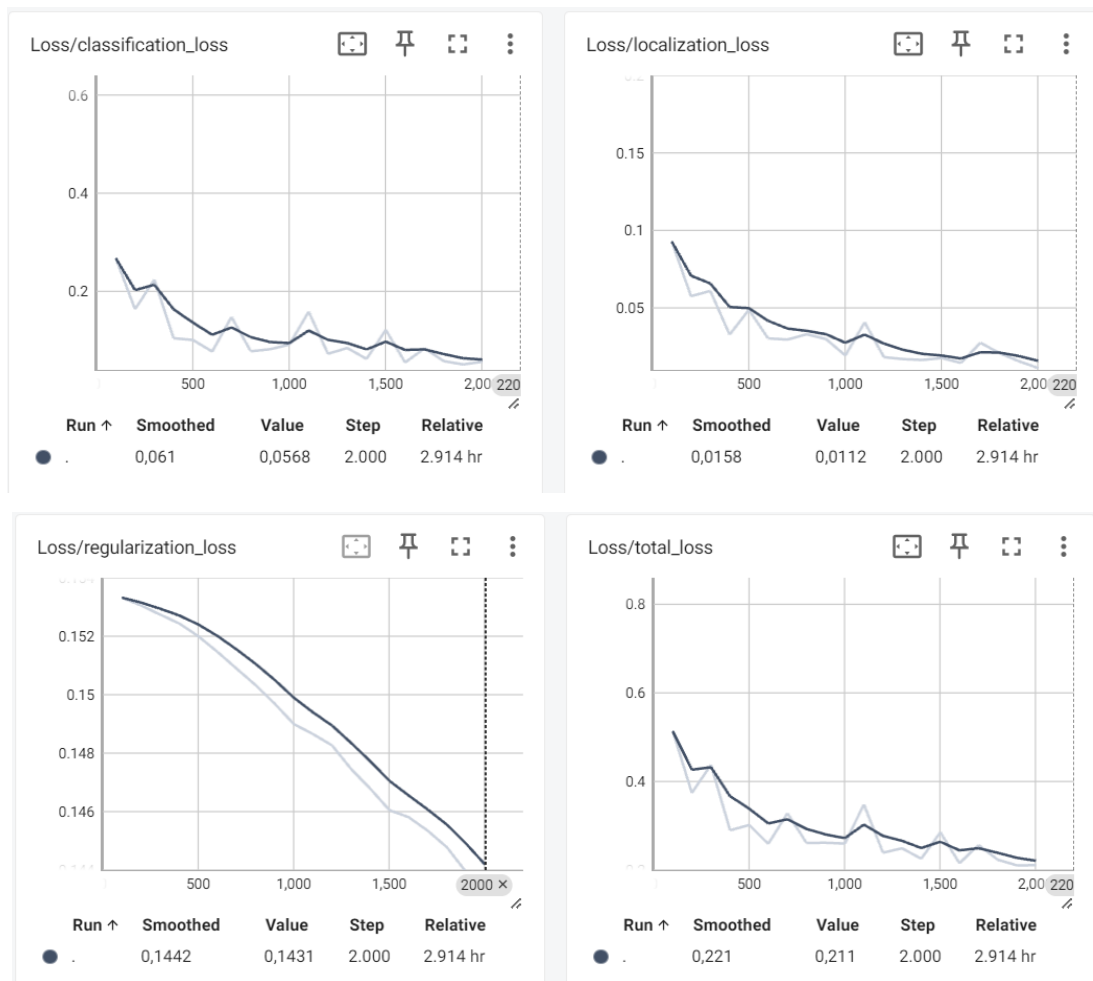
Các hình ảnh sau khi được tăng cường dữ liệu



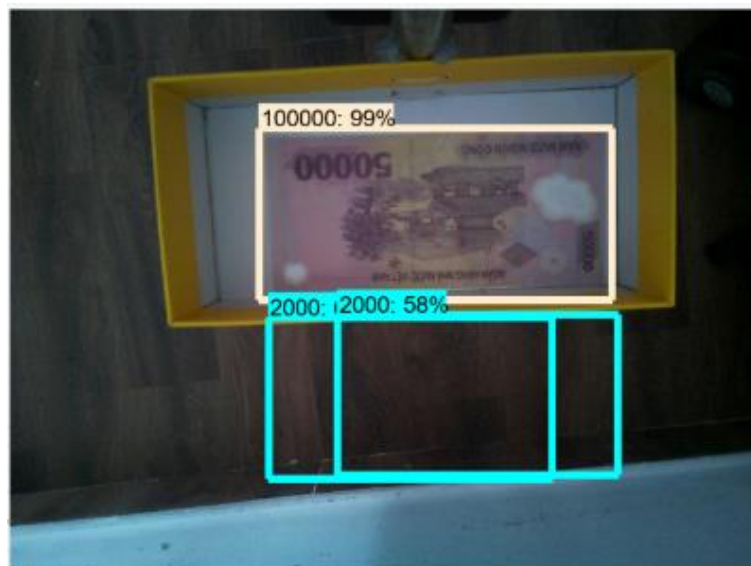
Vì data trong tập test được chụp với điều kiện ánh sáng như tập huấn luyện và tập kiểm thử nên cần phải kiểm tra thêm với tập ảnh được chụp trên thực tế

Bảng 5. So sánh kết quả trước và sau tăng cường dữ liệu về độ sáng

Batch size 16	Độ chính xác trên tập test	Độ chính xác trên tập ảnh thực tế	Classification Loss	Localization Loss	Regularization Loss
Khi chưa tăng cường độ sáng	100% (216/216)	85.92% (61/71)	0.06	0.01	0.14
Khi đã tăng cường độ sáng	100% (216/216)	92.96% (66/71)	0.02	0.007	0.14



Hình 27. Đồ thị mất mát khi đã tăng cường về độ sáng



Hình 28. Hình ảnh nhận diện sai sau khi được tăng cường về độ sáng trên tập hiệu chỉnh

Nhận xét:

- Sau khi đã tăng cường dữ liệu để huấn luyện nhưng vẫn không thể đạt được độ chính

xác đáng kể ở tập hiệu chỉnh.

- Các thông số mất mát cũng cho thấy được mô hình không bị overfitting hay underfitting

Nguyên nhân: Có thể do tập dữ liệu huấn luyện vẫn chưa được đa dạng về độ sáng, những trường hợp sai vẫn là những hình có độ sáng tối.

Giải pháp:

- Thêm các ảnh có độ sáng tối hơn để tập dữ liệu huấn luyện được đa dạng về độ sáng (**Kết quả là tập dữ liệu ở mục 3.1.1.2**)
- Tăng cường thêm về độ tương phản ('**random_adjust_contrast**')

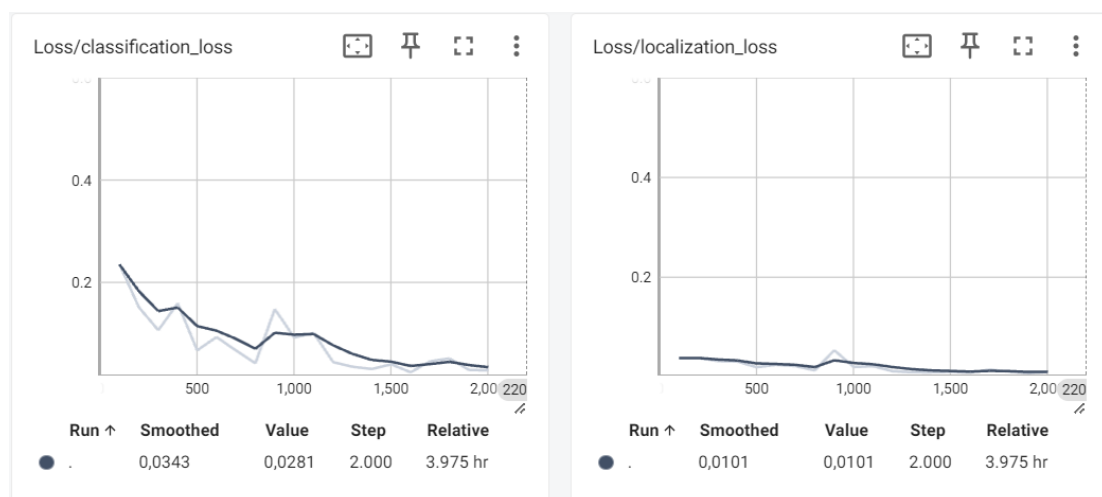
3.1.2.2. Huấn luyện với tập dữ liệu lần 2

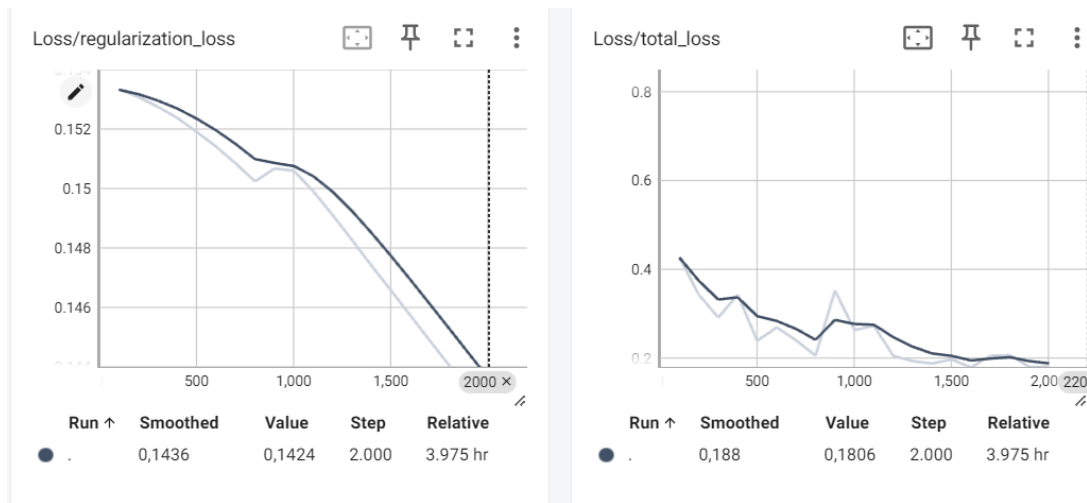
Các hình ảnh sau khi được tăng cường dữ liệu:



Huấn luyện lại mô hình với tập dataset mới và tiến hành hiệu chỉnh dựa trên tập hiệu chỉnh.

Kết quả thu được là độ chính xác được cải thiện thành **98% (70/71 trên tập ảnh mới)** và đã giải quyết được việc nhận dạng mệnh giá tiền ở các điều kiện ánh sáng khác nhau. Các ảnh được chụp ở độ sáng không tốt đã được nhận dạng một cách chính xác





Hình 29. Đồ thị mất mát sau khi thay đổi dữ liệu và tăng cường về độ tương phản

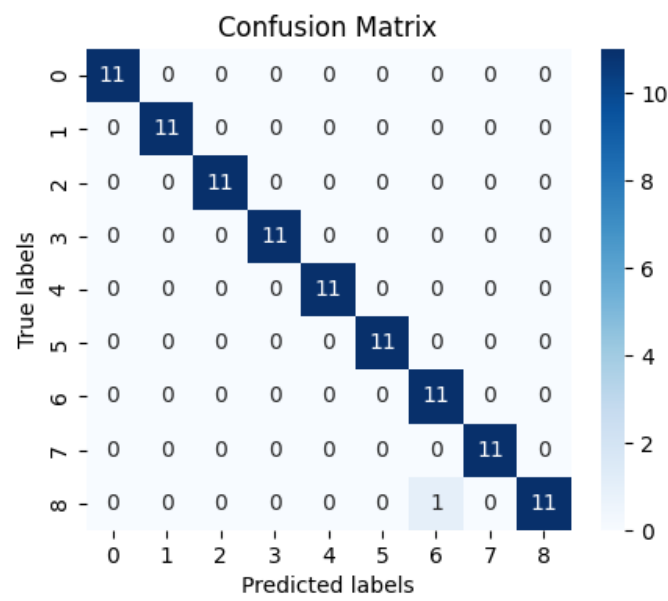


Hình 30. Kết quả sau khi thay đổi dữ liệu và tăng cường dữ liệu về độ tương phản

Kiểm tra lại với **tập test khác**, ở ngoài thực tế gồm 100 mẫu được chụp bằng Pi Camera V1 5MP ở nhiều điều kiện ánh sáng và địa điểm khác nhau, nhóm thu được kết quả:

	precision	recall	f1-score	support
1000	1.00	1.00	1.00	11
2000	1.00	1.00	1.00	11
5000	1.00	1.00	1.00	11
10000	1.00	1.00	1.00	11
20000	1.00	1.00	1.00	11
50000	1.00	1.00	1.00	11
100000	0.92	1.00	0.96	11
200000	1.00	1.00	1.00	11
500000	1.00	0.92	0.96	12
accuracy			0.99	100
macro avg	0.99	0.99	0.99	100
weighted avg	0.99	0.99	0.99	100

Hình 31. Độ chính xác khi test với tập test mới



Hình 32. Ma trận nhầm lẫn khi test với tập test mới

Nhận xét:

Độ chính xác đạt **xấp xỉ 100%**. Mô hình đã nhận dạng được hình ảnh được chụp ở mọi điều kiện ánh sáng, tuy nhiên ở ảnh bị che tay nhiều thì mô hình vẫn chưa nhận diện chính xác.

Giải pháp:

Tăng num_steps để mô hình có khả năng được học được nhiều đặc trưng hơn và xử lý được nhiều hơn về các trường hợp ngoại lệ như che tay hay bị gấp lại.

3.1.3. Kết quả khi tiến hành tối ưu thời gian

3.1.3.1. Dùng TensorFlow Lite

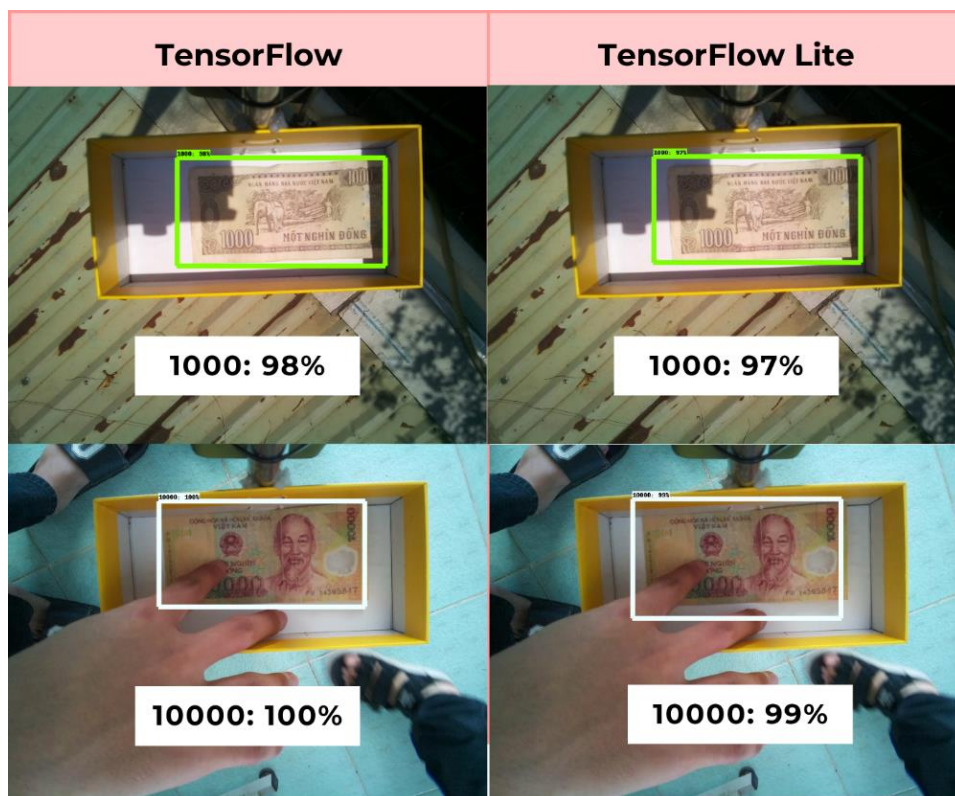
Sau khi thực hiện export sang TensorFlow Lite, tiến hành dự đoán trên cùng tập test với TensorFlow ở Raspberry Pi4, ta thu được kết quả:

Bảng 6. So sánh kết quả với TensorFlow và TensorFlow Lite

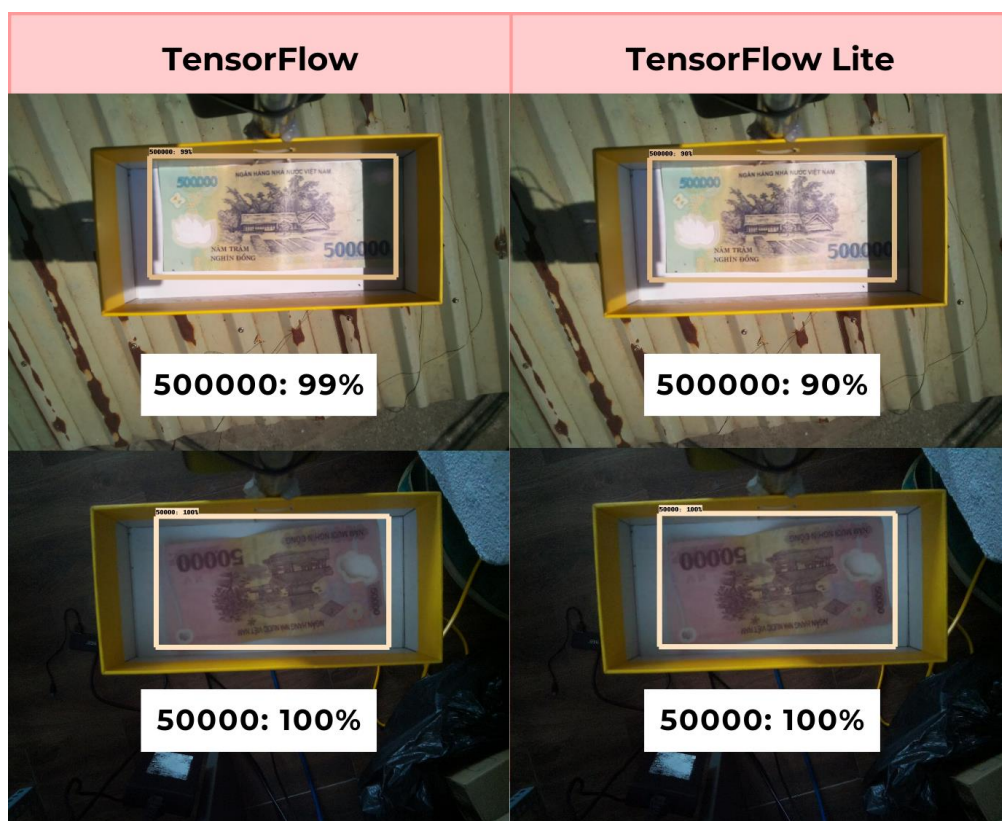
	Độ chính xác	Thời gian
TensorFlow	99% (99/100)	1m23s
TensorFlow Lite	99% (99/100)	40s

- **Kết luận:** TensorFlow Lite không làm thay đổi nhiều đến độ chính xác khi nhận diện, trong khi thời gian thực hiện nhận diện giảm đi đáng kể so với Tensorflow.

Kết quả nhận diện ở cả hai phiên bản như sau:



Hình 33. Kết quả nhận diện với TensorFlow và TensorFlow Lite (1)

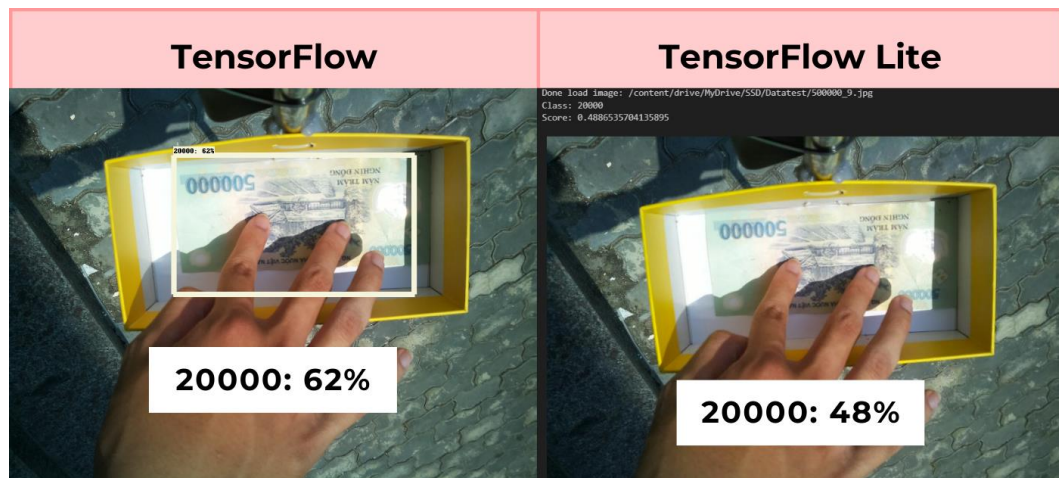


Hình 34. Kết quả nhận diện với TensorFlow và TensorFlow Lite (2)

Trong đó, kết quả nhận diện sai ở cả hai phiên bản là đối với tờ tiền đưa vào bị dính tay

Báo cáo đồ án PBL5 - Kỹ thuật máy tính

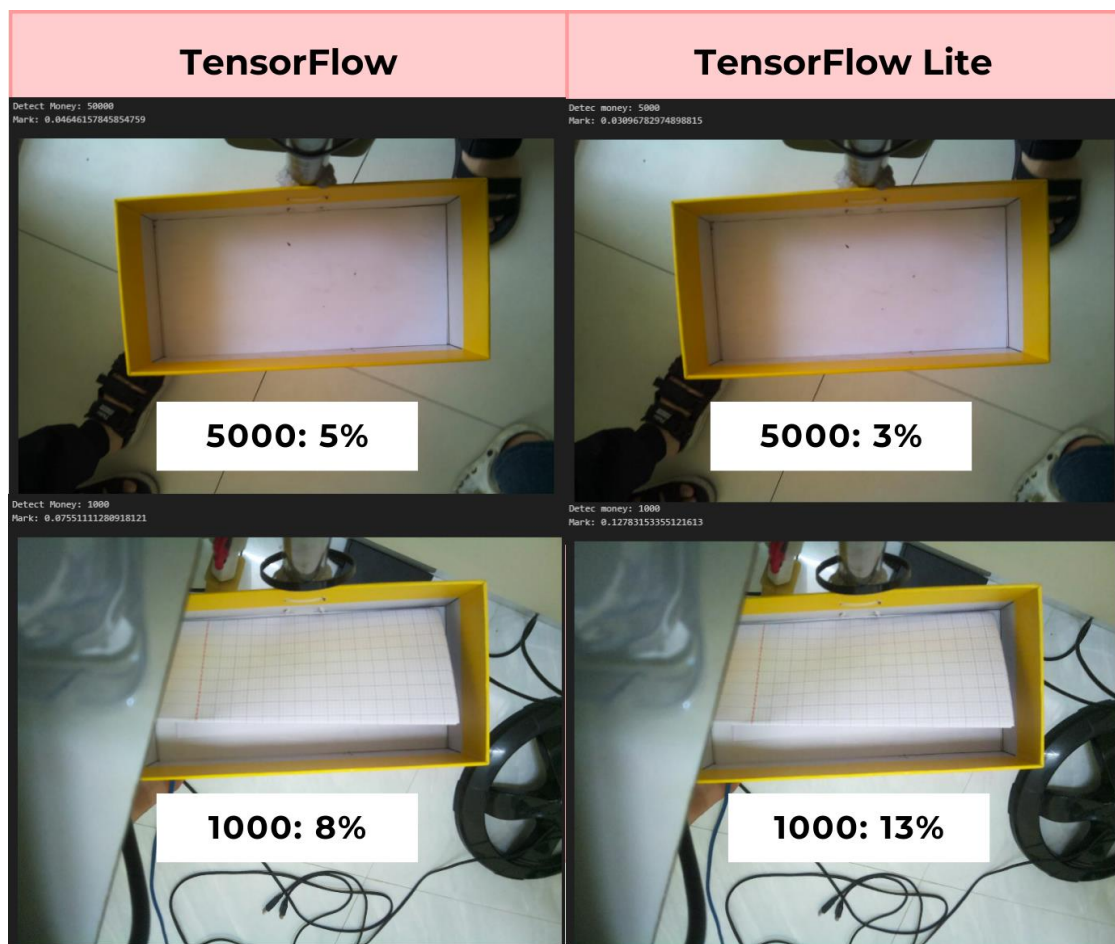
của người dùng, làm che mất một phần lớn tờ tiền, thì có sự chênh lệch lớn về độ chính xác của kết quả nhận diện.



Hình 35. Kết quả với hình ảnh nhận diện sai

Kết quả khi test với hình ảnh **không có/không phải** tờ tiền thì độ chính xác khi nhận diện rất thấp.

- **Kết luận:** Với hình ảnh đưa vào là **không có/không phải** tờ tiền thì mô hình không nhận diện được.



Hình 36. Kết quả nhận diện khi hình ảnh đưa vào không phải là tiền

Xử lý với trường hợp hình ảnh đưa vào có độ chính xác khi nhận diện thấp, kết quả sẽ phát ra âm thanh với nội dung là: “Nhận diện mệnh giá không thành công, xin vui lòng thử lại.”

3.1.3.2. Load model ngay khi khởi động Raspberry

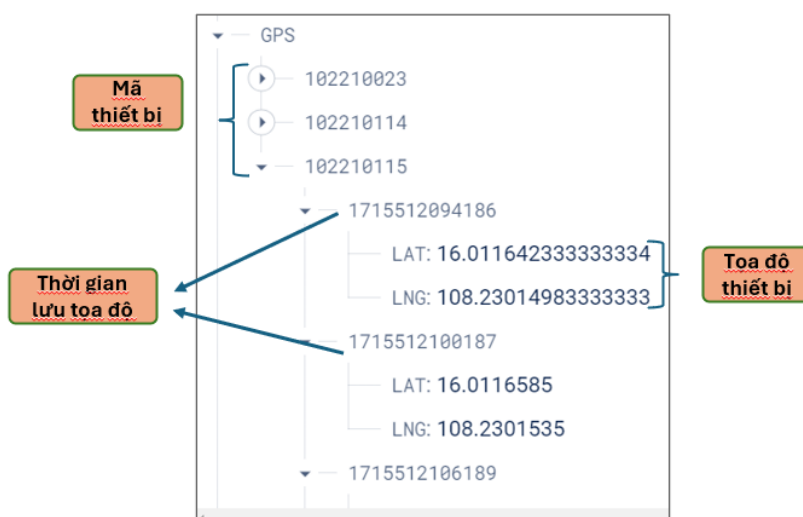
Để giảm thời gian xử lý nhận dạng hình ảnh, thay vì phải load model lại sau mỗi lần chụp ảnh, nhóm thực hiện load model ngay sau khi khởi động Raspberry Pi 4. Mỗi ảnh sau khi được chụp sẽ được xử lý với model được load ngay từ đầu chương trình để giảm thời gian nhận dạng mệnh giá.

Bảng 7. So sánh kết quả trước và sau khi tối ưu thời gian

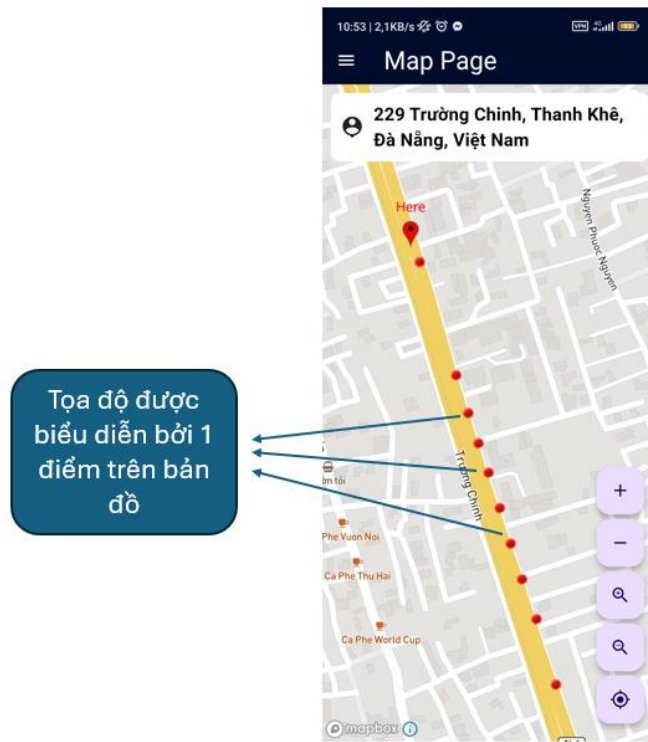
	Thời gian chụp ảnh	Thời gian xử lý	Tổng thời gian thực hiện nhận diện
Trước khi tối ưu	8s	32s	40s
Sau khi tối ưu	8s	17s	25s

3.2. GPS

Phần cứng đẩy dữ liệu lên Firebase thành công, bao gồm dữ liệu tọa độ vị trí và thời gian lấy tọa độ. Phần mềm lấy dữ liệu từ Firebase và hiển thị lên bản đồ thành công, với mỗi tọa độ vị trí là một điểm trên bản đồ.



Hình 37. Thông tin về dữ liệu được lưu trữ trên Firebase từ GPS Module



Hình 38. Bản đồ hiển thị lịch sử đường đi cũng như vị trí hiện tại

3.3. Phát hiện vật cản

Đã phát hiện được vật cản với khoảng cách trong 3m, kết quả sẽ được thông báo nếu khoảng cách đến vật cản dưới 2.5m và được qui chiếu về dưới 2.5m, dưới 2m, dưới 1.5m, dưới 1m và cảnh báo cách vật cản rất gần (dưới 0.5m)

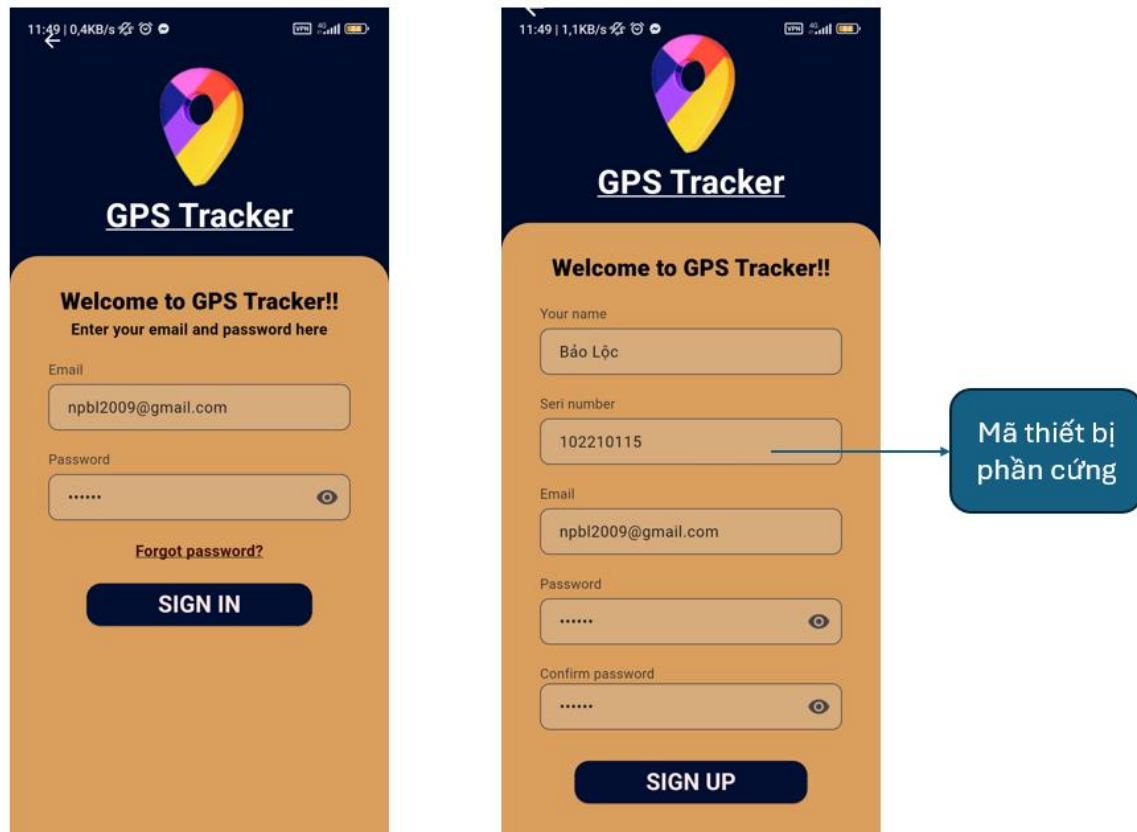
```
geany_run_script_UHF4N2.sh
File Edit Tabs Help
theadthdocs.io/en/stable/api_input.html#distancesensor-hc-sr04 for more info
warnings.warn(PWMSoftwareFallback(
Distance to nearest object is: 129.64202488786768 cm
Distance to nearest object is: 145.63644770666295 cm
Distance to nearest object is: 230.06076414288435 cm
Distance to nearest object is: 269.76657514517865 cm
Distance to nearest object is: 237.90404918682037 cm
Distance to nearest object is: 212.02632998101626 cm
Distance to nearest object is: 185.3677800898128 cm
Distance to nearest object is: 166.6880171282081 cm
Distance to nearest object is: 145.22420961024912 cm
Distance to nearest object is: 120.37366162758039 cm
Distance to nearest object is: 95.6444560560492 cm
Distance to nearest object is: 82.328422364915 cm
Distance to nearest object is: 67.65513863416368 cm
Distance to nearest object is: 50.190207137977694 cm
Distance to nearest object is: 38.24445020423363 cm
Distance to nearest object is: 25.90869842098934 cm
Distance to nearest object is: 16.411088970185233 cm
Distance to nearest object is: 103.76535262521114 cm
Distance to nearest object is: 130.0671695608616 cm
Distance to nearest object is: 203.95218542406406 cm
Distance to nearest object is: 213.25724317760697 cm
```

Hình 39. Dữ liệu về khoảng cách đến vật cản

Kết quả được thông báo bằng âm thanh với khoảng cách tương ứng sau khi qui chiếu

3.4. Ứng dụng di động

- Chức năng đăng ký, đăng nhập



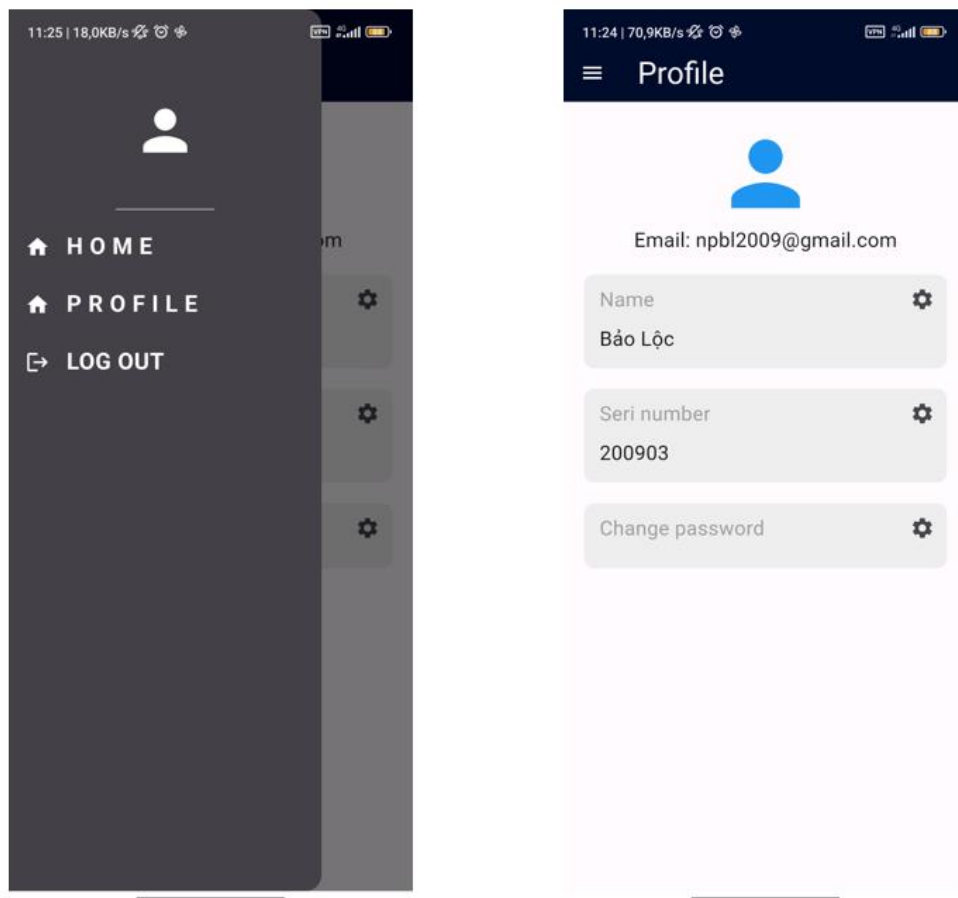
Hình 40. Chức năng đăng kí, đăng nhập

- Chức năng hiển thị thông tin vị trí



Hình 41 Chức năng hiển thị đường đi và vị trí

- Chức năng hiển thị, chỉnh sửa thông tin cá nhân và đăng xuất



Hình 42. Chức năng hiển thị, chỉnh sửa thông tin cá nhân và đăng xuất

3.5. Mô hình hoàn chỉnh



Hình 43. Mô hình hoàn chỉnh của đề tài

4. Kết luận và hướng phát triển

4.1. Kết luận

Thiết kế: Sản phẩm thiết kế gọn gàng, trọng lượng nhẹ, dễ dàng di chuyển, có thể cải tiến mở rộng trong tương lai cả về phần cứng lẫn phần mềm

Chức năng:

- Các chức năng hoạt động tốt, đáp ứng yêu cầu đặt ra.
- Chức năng nhận diện mệnh giá tiền hoạt động tốt, độ chính xác cao. Tuy nhiên, với những tờ tiền không phẳng, bị mất hình dạng hay trong điều kiện chụp hình quá tối hoặc bị che khuất quá nhiều thì chưa thể nhận diện được. Thời gian thực hiện nhận diện nhanh đáp ứng được yêu cầu là nhận dạng trực tiếp.
- Chức năng phát hiện vật cản hoạt động tốt, thời gian nhanh, nhưng không ổn định do chất lượng về thiết bị cảm biến còn hạn chế.
- Chức năng định vị GPS và theo dõi thông qua app GPS Tracker đáp ứng đầy đủ các chức năng, gần như không xảy ra lỗi. Về cơ bản thì thân thiện với người dùng, dễ sử dụng.

4.2. Hướng phát triển

- Cần phải tinh gọn và làm bắt mắt thiết kế, đảm bảo chất lượng có thể sử dụng trong các điều kiện thời tiết khác nhau và dùng được thời gian dài.
- Huấn luyện thêm mô hình nhận diện mệnh giá tiền trên tập dữ liệu mới để có thể nhận diện tốt hơn các trường hợp tờ tiền đưa vào không phẳng hay bị mất hình dạng để đa dạng khả năng nhận diện.
- Đầu tư thêm về chất lượng thiết bị để tăng độ ổn định cũng như độ chính xác khi phát hiện vật cản và định vị vị trí.
- Phát triển thêm các tính năng mới cho app GPS Tracker như là tìm đường đi từ vị trí hiện tại đến vị trí người sử dụng thiết bị, ...

5. Danh mục tài liệu tham khảo

- [1] Qian Zhang, *Currency Recognition Using Deep Learning*, New Zealand: Auckland, 2018.
- [2] Duy Hung, “Object Detection với mô hình SSD-MobileNet,” *Panda ML*, November 2021. [Online]. Available: <https://www.pandaml.com/object-detection-ssd/>. [Accessed: March 22, 2024].
- [3] Vu Huu Long, “Overfitting”, *Machine Learning cơ bản*, Mar 4, 2017. [Online]. Available: <https://machinelearningcoban.com/2017/03/04/overfitting/>. [Accessed: April 12, 2024].
- [4] Pham Dinh Khanh, “Model SSD trong Object Detection”, *Khoa học dữ liệu – Khanh’s blog*, October 2019. [Online]. Available: <https://phamdinhkhanh.github.io/2019/10/05/SSDModelObjectDetection.html>. [Accessed: March 9, 2024].