

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



**BÁO CÁO ĐỒ ÁN**  
**“SMART CAMERA – CAMERA GIÁM SÁT AN**  
**NINH**  
**SỬ DỤNG RASPBERRY PI”**

**GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN KHÁNH THUẬT**

STT	Họ và tên	MSSV
1	Mai Ngọc Phương Trinh	20520823
2	Trần Thảo Nhi	21521236
3	Trần Huỳnh Ngọc	21522384

**TP. HỒ CHÍ MINH, 2024**

## MỤC LỤC

Chương 1: Giới thiệu đề tài.....	5
1.1.Lý do chọn đề tài.....	5
1.2.Định hướng phát triển.....	5
Chương 2: Cơ sở lý thuyết .....	5
2.1. Raspberry pi 4 .....	6
2.2. Camera raspberry pi rev 1.3 .....	7
2.3. Cảm biến chuyển động PIR .....	7
Chương 3: Phân tích thiết kế: .....	8
3.1. Raspberry pi .....	8
3.2. Ứng dụng di động .....	8
Chương 4: Hiện thực đề tài.....	10
4.1. Cấu hình Raspberry pi 4 .....	10
4.2. Thực hiện liên kết với firebase.....	11
4.2.1. Kết nối ứng dụng với firebase.....	11
4.2.2. Kết nối raspberry pi 4 với firebase .....	12
4.3. Xử lý và tải hình ảnh/video, giá trị cảm biến lên firebase .....	13
4.4.Server xử lý nhận diện hình ảnh.....	15
4.5. Xử lý nhận hình ảnh/video, giá trị cảm biến từ firebase xuống ứng dụng di động..	17
4.5. Thông báo trên ứng dụng di động.....	19
Chương 5: Thực nghiệm đề tài (demo).....	19
Chương 6: Kết luận và hướng phát triển .....	20
6.1. Kết luận.....	20
6.2. Hướng phát triển .....	20
TÀI LIỆU THAM KHẢO .....	20

## LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Khánh Thuật vì sự hướng dẫn và đóng góp quý báu của thầy trong quá trình thực hiện đồ án của chúng em.

Chúng em xin cảm ơn thầy đã luôn sẵn sàng lắng nghe, giải đáp các câu hỏi và định hướng chúng em trong quá trình nghiên cứu và phát triển đồ án. Chúng em rất biết ơn sự hỗ trợ của thầy trong việc phản hồi và đánh giá bài làm của chúng em, giúp chúng em hiểu rõ hơn về cách cải thiện và phát triển kỹ năng của mình. Sự kiên nhẫn và kiến thức sâu rộng của thầy đã giúp chúng em tự tin hơn trong việc thực hiện dự án này.

Cuối cùng, chúng em xin bày tỏ lòng biết ơn chân thành đến thầy vì sự tận tâm và kiên nhẫn của thầy trong việc hướng dẫn và hỗ trợ chúng em trong từng bước của quá trình nghiên cứu và viết đồ án. Thầy đã luôn sẵn lòng lắng nghe và đồng hành cùng chúng em, giúp chúng em hiểu rõ hơn về lĩnh vực chúng em quan tâm. Những kiến thức và kỹ năng mà chúng em đã học được từ thầy sẽ có tầm ảnh hưởng lớn trong sự nghiệp học tập và sự phát triển của chúng em trong tương lai.

Chân thành cảm ơn thầy và mong rằng thầy sẽ tiếp tục có những thành công và đạt được những mục tiêu cao trong sự nghiệp giảng dạy và nghiên cứu.

Nhóm sinh viên thực hiện  
Mai Ngọc Phương Trinh  
Trần Thảo Nhi  
Trần Huỳnh Ngọc

## DANH MỤC HÌNH ẢNH

Hình 1.	Raspberry pi 4.....	6
Hình 2.	Camera raspberry pi rev 1.3.....	7
Hình 3.	Cảm biến chuyển động PIR .....	8
Hình 4.	Giao diện xem lại video của ứng dụng.....	9
Hình 5.	Giao diện các thông báo của ứng dụng .....	10
Hình 6.	Code import các thư viện.....	11
Hình 7.	Xử lý đăng nhập và đăng ký .....	12
Hình 8.	Xác thực bằng Firebase Authentication .....	12
Hình 9.	Kết nối raspberry với firebase.....	13
Hình 10.	Code xử lý cảm biến PIR.....	13
Hình 11.	Code xử lý nhận diện khuôn mặt trong khung hình .....	14
Hình 12.	Code mã hóa và xử lý nhận diện khuôn mặt chủ nhà.....	14
Hình 13.	Code cập nhật trạng thái của PIR từ Raspberry lên Firebase .....	15
Hình 14.	Tính toán vector.....	16
Hình 15.	Huấn luyện mô hình nhận diện .....	16
Hình 16.	Server nhận yêu cầu POST và giải mã hình ảnh.....	16
Hình 17.	Dự đoán danh tính khuôn mặt.....	17
Hình 18.	Trả về kết quả nhận diện.....	17
Hình 19.	Code lấy video từ Firebase xuống ứng dụng.....	18
Hình 20.	Code lấy trạng thái của cảm biến PIR .....	18
Hình 21.	Code gửi thông báo đến ứng dụng.....	19
Hình 22.	Code lưu lại các thông báo trên firebase .....	19

## Chương 1: Giới thiệu đề tài

Smart Camera, hay camera thông minh, là thiết bị giám sát được tích hợp nhiều công nghệ hiện đại như trí tuệ nhân tạo (AI), học máy sâu (machine learning), và có khả năng kết nối internet. So với camera truyền thống, Smart camera sở hữu nhiều tính năng vượt trội hơn như nhận diện được đối tượng trong quá trình giám sát, kết nối với ứng dụng để gửi thông báo cho người dùng,....

### 1.1. Lý do chọn đề tài

- Nhu cầu về an ninh ngày càng tăng cao: như đã biết xã hội ngày càng phát triển, đời sống con người ngày càng được nâng cao, kéo theo nhu cầu về an ninh cũng ngày càng tăng cao. Con người luôn mong muốn bảo vệ bản thân, tài sản và những người thân yêu khỏi những nguy cơ tiềm ẩn như trộm cắp, đột nhập, xâm hại,... nên cần có những thiết bị để giúp phòng ngừa và góp phần ngăn chặn những thiệt hại đáng tiếc xảy ra.
- Bên cạnh đó, camera giám sát truyền thống còn nhiều hạn chế như:
  - o Khả năng ghi hình thụ động, lưu trữ dữ liệu hạn chế
  - o Không thể phân biệt được người và vật, khó khăn trong việc theo dõi và quản lý
  - o Phụ thuộc vào con người để theo dõi và xử lý sự cố,...
- Ưu điểm vượt trội của Smart camera:
  - o Tích hợp trí tuệ nhân tạo giúp phát hiện chuyển động và âm thanh thông minh, nhận dạng khuôn mặt và vật thể, phân tích dữ liệu hình ảnh,...
  - o Kết nối internet: xem hình ảnh và video trực tiếp mọi lúc mọi nơi, nhận thông báo khi có sự cố, điều khiển camera từ xa.
  - o Lưu trữ dữ liệu linh hoạt: lưu trữ trên thẻ nhớ, đám mây hoặc kết nối với hệ thống lưu trữ NAS.
- Hoạt động hiệu quả: hoạt động 24/7, ghi hình rõ nét cả ban ngày và ban đêm, chịu được mọi điều kiện thời tiết.
- Raspberry Pi là một thiết bị nhỏ gọn và tiêu thụ ít năng lượng, dễ dàng triển khai ở nhiều vị trí khác nhau.
- Với sự phát triển của công nghệ việc xây dựng hệ thống giám sát mang lại sự an toàn và tiện lợi.

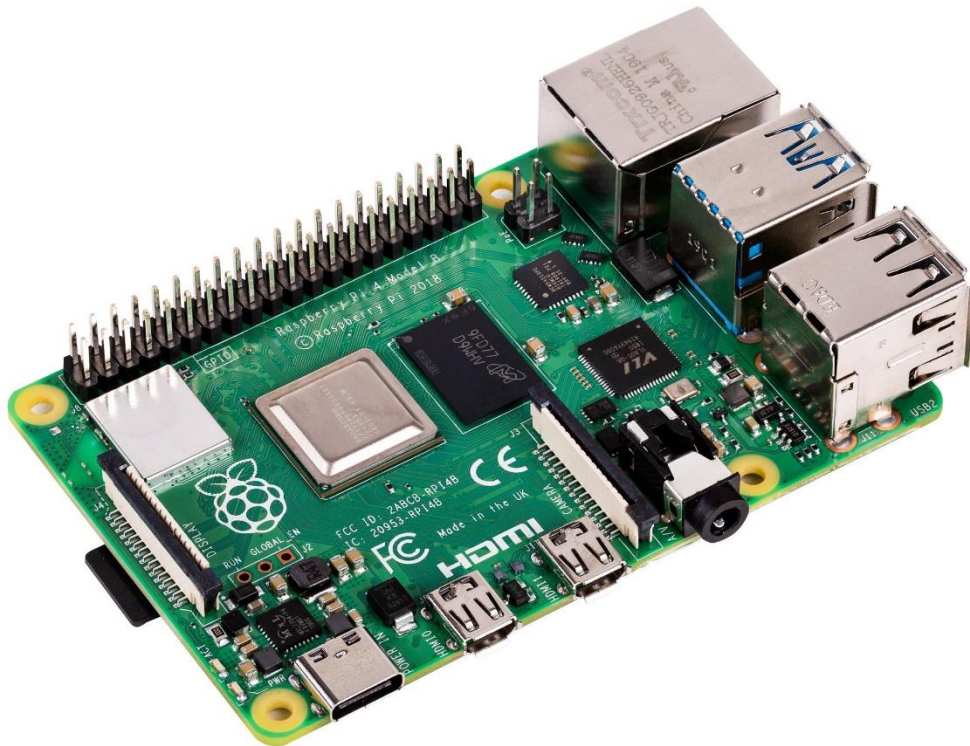
### 1.2. Định hướng phát triển

- Xây dựng một Smart camera để giám sát an ninh trong một khu vực nhỏ như nhà ở, văn phòng, quán ăn; có chi phí lắp đặt thấp, tiện lợi và dễ thi công.
- Mục tiêu: Smart camera có các chức năng như nhận diện khuôn mặt và vật thể, phát hiện chuyển động, kết nối với thiết bị điện tử như smart phone để gửi thông báo về cho người dùng khi có người qua lại trong khu vực của camera.

## Chương 2: Cơ sở lý thuyết

## 2.1. Raspberry pi 4

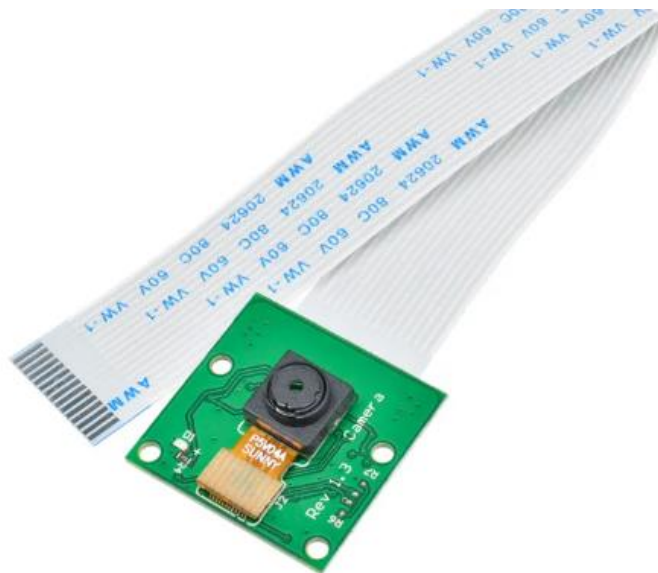
- Cấu trúc Raspberry Pi 4 được thiết kế với một số thành phần chính bao gồm:
  - o Bộ xử lý: Raspberry Pi 4 được trang bị bộ vi xử lý ARM Cortex-A72, cung cấp khả năng xử lý tính toán mạnh mẽ và hiệu suất đáng kể so với các phiên bản trước đó.
  - o Bộ nhớ RAM: Có các tùy chọn RAM từ 2GB đến 8GB, giúp Raspberry Pi 4 có thể xử lý các ứng dụng đa nhiệm và đòi hỏi nhiều tài nguyên mạnh mẽ.
  - o Kết nối: Raspberry Pi 4 có các cổng kết nối như HDMI, USB, Ethernet, Wi-Fi, Bluetooth, GPIO (General Purpose Input/Output), và các cổng khác, cung cấp tính linh hoạt cho việc kết nối với các thiết bị và mạng.
- Nguyên lý hoạt động: Raspberry Pi 4 chạy trên hệ điều hành Linux như Raspbian, được cài đặt trên một thẻ nhớ microSD. Bộ xử lý chính của Raspberry Pi 4 xử lý các tác vụ từ hệ điều hành và các ứng dụng được cài đặt. Dữ liệu được truyền qua các cổng và giao tiếp khác nhau để tương tác với các thiết bị ngoại vi và mạng.
- Sử dụng Raspberry Pi 4 như một trung tâm điều khiển để kết nối và điều khiển các cảm biến chuyển động cũng như camera. Các cảm biến chuyển động được kết nối với Raspberry Pi thông qua các giao tiếp cụ thể như GPIO (General Purpose Input/Output) và các giao thức như I2C hoặc SPI. Khi cảm biến chuyển động phát hiện có sự thay đổi trong môi trường, Raspberry Pi sẽ nhận được tín hiệu và kích hoạt một loạt các hành động.



Hình 1. Raspberry pi 4

## 2.2. Camera raspberry pi rev 1.3

- Cấu trúc: camera Raspberry thường được thiết kế với một cảm biến hình ảnh và một ống kính. Cảm biến hình ảnh chịu trách nhiệm cho việc chuyển đổi ánh sáng thành dữ liệu số, trong khi ống kính được sử dụng để thu nhận và tập trung ánh sáng vào cảm biến.
- Nguyên lý hoạt động: Khi ánh sáng đi qua ống kính, nó được lấy mẫu và chuyển đổi thành tín hiệu điện bởi cảm biến hình ảnh. Cảm biến này chứa hàng loạt các pixel, mỗi pixel đại diện cho một điểm ảnh trên hình ảnh cuối cùng. Thông qua quá trình chuyển đổi analog-to-digital, tín hiệu điện được biến đổi thành dữ liệu số, sau đó được xử lý để tạo ra hình ảnh hoặc video.
- Các Tính Năng Chính
  - o Độ phân giải: Camera Raspberry thường có nhiều tùy chọn độ phân giải, từ các model cơ bản đến các model cao cấp hỗ trợ độ phân giải cao như 1080p hoặc thậm chí 4K.
  - o Khả năng điều chỉnh: Một số camera Raspberry cho phép điều chỉnh tiêu cự hoặc góc nhìn của ống kính, tạo điều kiện tối ưu cho các ứng dụng cụ thể.
  - o Kết nối: Camera Raspberry thường được kết nối với Raspberry Pi thông qua các cổng và giao tiếp như CSI (Camera Serial Interface) hoặc USB.



Hình 2. Camera raspberry pi rev 1.3

## 2.3. Cảm biến chuyển động PIR

- Cấu Trúc của Cảm Biến Chuyển Động PIR



- Bộ Lọc Quang Học: Loại bỏ các tín hiệu không mong muốn và chỉ cho phép các tín hiệu sóng hồng ngoại từ vật thể di chuyển.
- Cặp Cảm Biến Nhiệt: Gồm hai cảm biến nhiệt được đặt ở hai phía của cảm biến, nhận biết sự thay đổi nhiệt độ.
- Mạch Điện Tử: Xử lý tín hiệu từ cặp cảm biến nhiệt và tạo ra tín hiệu kích hoạt khi phát hiện sự chuyển động.
- Nguyên lý hoạt động: Cảm biến chuyển động PIR hoạt động dựa trên nguyên lý của sóng hồng ngoại. Nó bao gồm một bộ lọc quang học và một cặp cảm biến nhiệt. Khi một vật thể di chuyển qua trường quét của cảm biến, sự thay đổi nhiệt độ của vật thể sẽ tạo ra một tín hiệu điện trong cặp cảm biến nhiệt, và sau đó được phát hiện bởi bộ lọc quang học.



Hình 3. Cảm biến chuyển động PIR

## Chương 3: Phân tích thiết kế:

### 3.1. Raspberry pi

Khi Raspberry Pi nhận được tín hiệu từ cảm biến chuyển động, nó sẽ kích hoạt camera để ghi lại hình ảnh hoặc video của khu vực được giám sát. Quá trình này được thực hiện thông qua việc sử dụng các thư viện và giao thức phần mềm để điều khiển camera và xử lý hình ảnh.

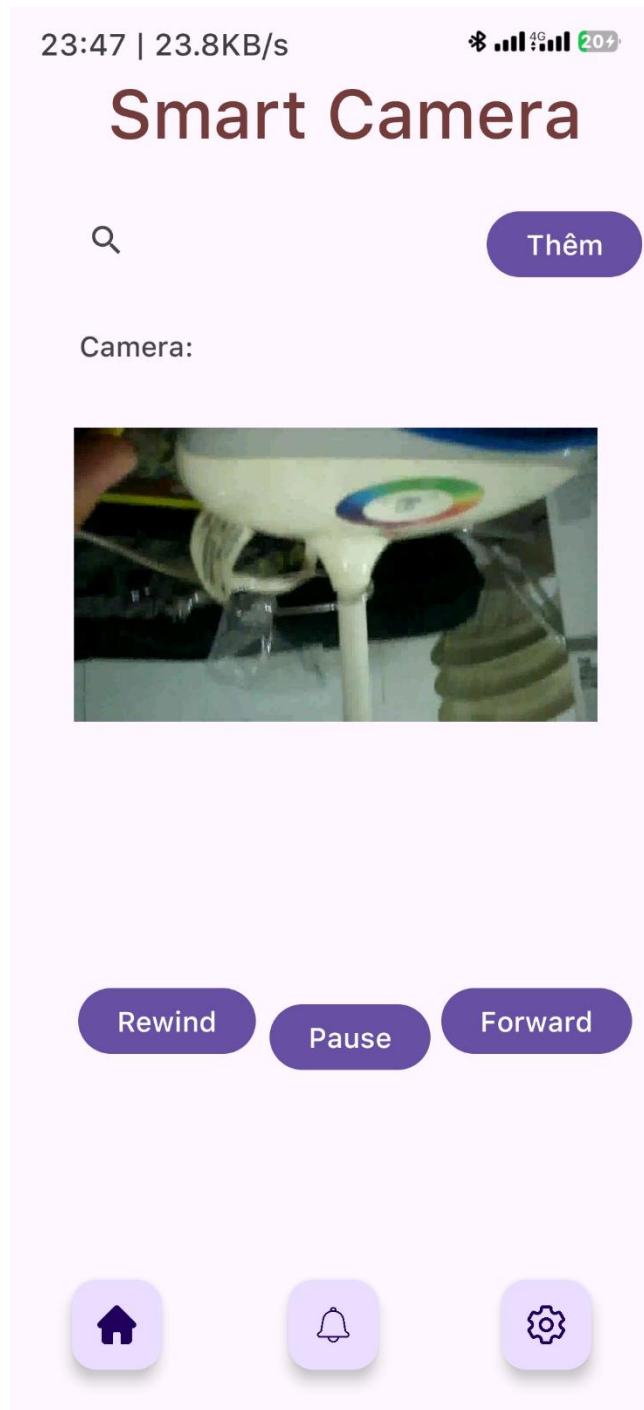
- Xử lý đọc giá trị cảm biến từ raspberry nếu phát hiện chuyển động sẽ bật camera lên và ghi hình.
- Xử lý quá trình ghi hình bằng camera

Kết nối và tải dữ liệu lên firebase

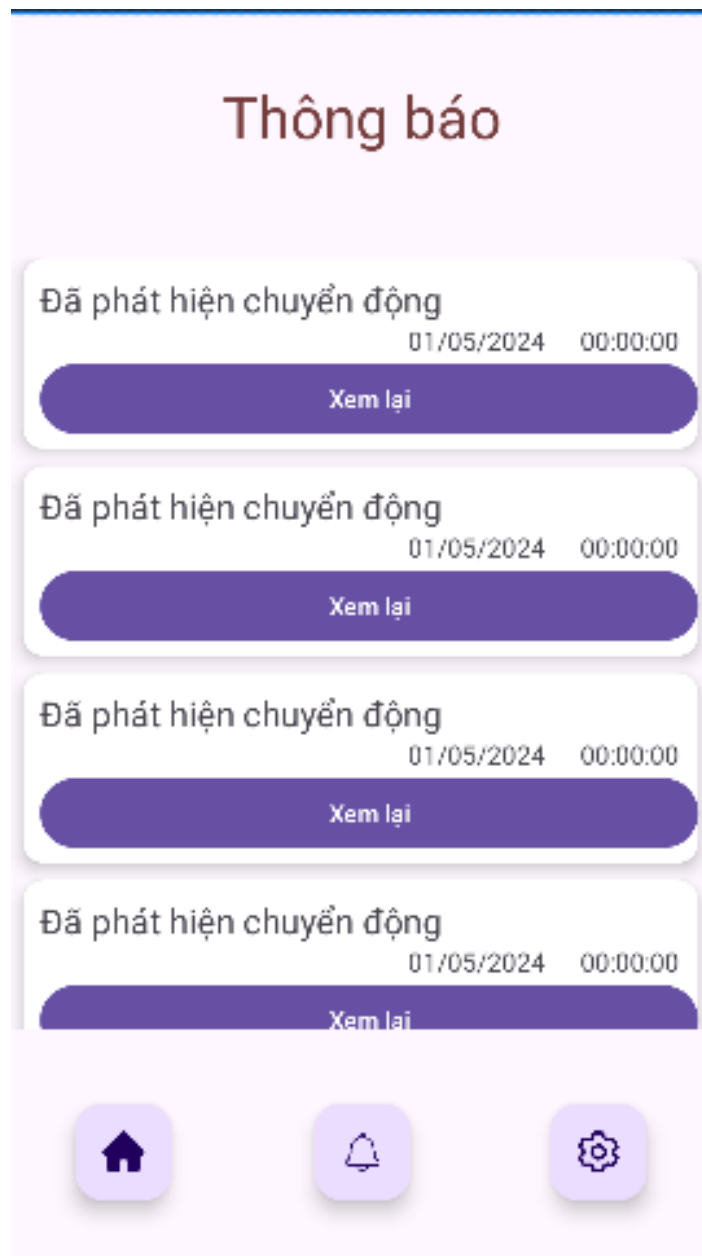
### 3.2. Ứng dụng di động

Giao diện chính ứng dụng di động





Hình 4. Giao diện xem lại video của ứng dụng



Hình 5. Giao diện các thông báo của ứng dụng

Ứng dụng kết nối và lấy dữ liệu từ firebase:

- Người dùng có thể xem trực tiếp hình ảnh từ camera.
- Khi có chuyển động sẽ gửi thông báo đến người dùng và cho phép người dùng xem lại chuyển động (video ngắn)

## Chương 4: Hiện thực đề tài

### 4.1. Cấu hình Raspberry pi 4

Cài đặt hệ điều hành:

- Tải và cài đặt hệ điều hành Raspbian (Raspberry Pi OS) từ trang chủ Raspberry Pi.
- Sử dụng Raspberry Pi Imager để ghi hệ điều hành lên thẻ nhớ SD.

Cấu hình hệ điều hành:

- Khởi động Raspberry Pi và thực hiện cấu hình ban đầu bao gồm việc thiết lập mật khẩu, cập nhật hệ thống, và cài đặt các gói phần mềm cần thiết.
- Kích hoạt các giao diện cần thiết như camera và GPIO trong raspi-config.

Cài đặt các gói phần mềm và thư viện cần thiết:

- Cài đặt OpenCV cho xử lý ảnh và video.
- Cài đặt gpiozero cho điều khiển cảm biến và LED.
- Cài đặt dlib cho nhận diện khuôn mặt.
- Cài đặt Pyrebase cho kết nối với Firebase.

```
import cv2
import time
from gpiozero import MotionSensor, LED
from picamera2 import Picamera2
from datetime import datetime
import pyrebase
import requests
import base64
import dlib
```

Hình 6. Code import các thư viện

## 4.2. Thực hiện liên kết với firebase

### 4.2.1. Kết nối ứng dụng với firebase

- Đăng ký tài khoản
  - Khởi tạo xác thực bằng Firebase Authentication
  - Xử lý đăng ký: thực hiện xác thực với email và password được nhập từ người dùng qua phương thức createUserWithEmailAndPassword của FirebaseAuth, đăng ký người dùng với Firebase và kiểm tra nếu thành công thì chuyển đến trang đăng nhập

```

mAuth.createUserWithEmailAndPassword(email2, password2).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //FirebaseUser user = mAuth.getCurrentUser();
            // FirebaseDatabase.getInstance().getReference("users")
            //     .child(user.getUid()).child("username").setValue(username);
            Toast.makeText(getApplicationContext(), text: "Sign up successfully", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(getApplicationContext(), registerpage.this, log_in.class);
            startActivity(intent);
        }
        else {
            Toast.makeText(getApplicationContext(), text: "Sign up failed", Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

Hình 7. Xử lý đăng nhập và đăng ký

- Đăng nhập
  - o Khởi tạo xác thực bằng Firebase Authentication

```
mAuth = FirebaseAuth.getInstance();
```

Hình 8. Xác thực bằng Firebase Authentication

- o Xử lý đăng nhập: thực hiện đăng nhập với email và password được nhập từ người dùng qua phương thức signInWithEmailAndPassword, đăng nhập người dùng với Firebase và kiểm tra nếu thành công thì chuyển đến trang home

#### 4.2.2. Kết nối raspberry pi 4 với firebase

Tạo dự án trên Firebase:

- Truy cập trang chủ Firebase và tạo một dự án mới có tên SmartCamera
- Lấy các thông tin cấu hình Firebase (API key, Auth domain, Database URL, v.v.).

Cài đặt và cấu hình Pyrebase:

- Sử dụng Pyrebase để kết nối Raspberry Pi với Firebase.
- Cấu hình thông tin Firebase trong mã Python:

```

firebaseConfig = {
  'apiKey': "AIzaSyA0P0mvkBLIYW1N3BSeqfF0PMzm8WvjuYk",
  'authDomain': "smartcamera-e705c.firebaseio.com",
  'databaseURL': "https://smartcamera-e705c-default-rtdb.firebaseio.com",
  'projectId': "smartcamera-e705c",
  'storageBucket': "smartcamera-e705c.appspot.com",
  'messagingSenderId': "692217413026",
  'appId': "1:692217413026:web:f1dab3b0572405d7e246fb",
  'measurementId': "G-RQVNB1Y805"
}

firebase = pyrebase.initialize_app(firebaseConfig)
storage = firebase.storage()
db = firebase.database()

```

Hình 9. Kết nối raspberry với firebase

#### 4.3. Xử lý và tải hình ảnh/video, giá trị cảm biến lên firebase

Phát hiện chuyển động:

- Sử dụng cảm biến chuyển động (PIR) để phát hiện chuyển động.
- Khi phát hiện chuyển động, kích hoạt camera để quay video và lưu trữ lại.

```

pir.wait_for_motion()
print("Phat hien chuyen dong!")
led.on()

current_time = datetime.now().strftime("%d%m%Y%H%M%S")
video_name = f"{current_time}.h264"
camera.start_and_record_video(video_name, duration=10)
print(f"Video {video_name} da luu.")
led.off()

```

Hình 10. Code xử lý cảm biến PIR

Xử lý video và nhận diện khuôn mặt:

- Mở video đã quay và xử lý từng khung hình.
- Sử dụng thư viện dlib để nhận diện khuôn mặt trong từng khung hình.

```

detector = dlib.get_frontal_face_detector()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = detector(gray)

for face in faces:
    x1 = face.left()
    y1 = face.top()
    x2 = face.right()
    y2 = face.bottom()
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 3)

cv2.imshow('Video', frame)

```

Hình 11. Code xử lý nhận diện khuôn mặt trong khung hình

- Nếu phát hiện khuôn mặt, mã hóa hình ảnh và gửi tới server để nhận diện.

Gửi kết quả nhận diện đến server:

- Mã hóa hình ảnh thành dạng base64 và gửi POST request đến server để nhận diện khuôn mặt.

```

if len(faces) > 0:
    encoded_image_data = encode_image_to_base64(frame)
    data = {
        'image': encoded_image_data,
        'w': w,
        'h': h
    }
    response = requests.post(url, data=data)

    print("Response from server:")
    print(response.text)

```

Hình 12. Code mã hóa và xử lý nhận diện khuôn mặt chủ nhà

- Nếu nhận diện được chủ nhà (Ngoc, Trinh, Nhi), không tải video lên Firebase. Nếu không, tải video lên Firebase và cập nhật trạng thái chuyển động.

Tải video và cập nhật trạng thái lên Firebase:

- Nếu phát hiện người lạ, tải video lên Firebase Storage.

- Cập nhật trạng thái chuyển động trong cơ sở dữ liệu thời gian thực của Firebase.

```

if upload_to_firebase:
    try:
        storage.child(video_name).put(video_name)
        print("Video đã tải lên Firebase.")
    except Exception as e:
        print(f"Không thể tải video lên Firebase: {e}")

    motion_data = {
        "status": "on"
    }
    try:
        db.child("PIR").update(motion_data)
        print("Lưu tín hiệu phát hiện chuyển động lên firebase.")
    except Exception as e:
        print(f"Không thể lưu tín hiệu lên fb: {e}")

else:
    motion_data = {
        "status": "off"
    }
    try:
        db.child("PIR").update(motion_data)
        print("Không phải người lạ lưu off lên firebase.")
    except Exception as e:
        print(f"Không thể lưu tín hiệu lên fb: {e}")

```

Hình 13. Code cập nhật trạng thái của PIR từ Raspberry lên Firebase

#### 4.4. Server xử lý nhận diện hình ảnh

##### Xử lý dữ liệu

- Thu thập mỗi chủ nhà 200 hình ảnh gương mặt từ các góc mặt khác nhau để đảm bảo dữ liệu tốt để sử dụng làm tập huấn luyện
- Sử dụng MTCNN để phát hiện và căn chỉnh hình ảnh khuôn mặt, sau đó lưu trữ hình ảnh đã căn chỉnh và thông tin vào thư mục đầu ra.

##### Tính toán embeddings



- Chạy quá trình forward pass để tính toán embeddings cho các hình ảnh. Embeddings là các vector đặc trưng đại diện cho các khuôn mặt.

```
for i in range(nrof_batches_per_epoch):
    start_index = i*args.batch_size
    end_index = min((i+1)*args.batch_size, nrof_images)
    paths_batch = paths[start_index:end_index]
    images = facenet.load_data(paths_batch, False, False, args.image_size)
    feed_dict = { images_placeholder:images, phase_train_placeholder:False }
    emb_array[start_index:end_index,:] = sess.run(embeddings, feed_dict=feed_dict)
```

Hình 14. Tính toán vector

#### Huấn luyện mô hình

- Sử dụng SVM (Support Vector Machine) với kernel tuyến tính để huấn luyện mô hình phân loại trên các embeddings.

```
print('Training classifier')
model = SVC(kernel='linear', probability=True)
model.fit(emb_array, labels)
```

Hình 15. Huấn luyện mô hình nhận diện

#### Phân loại hình ảnh:

- Sử dụng mô hình đã được huấn luyện để phân loại các hình ảnh mới.

#### Server nhận yêu cầu:

- Server nhận yêu cầu POST và giải mã chuỗi base64 thành một hình ảnh. Sử dụng MTCNN để phát hiện các khuôn mặt trong hình ảnh.

```
if request.method == 'POST':
    name="Unknown"
    f = request.form.get('image')
    print(type(f))
    w = int(request.form.get('w'))
    h = int(request.form.get('h'))

    decoded_string = base64.b64decode(f)
    frame = np.fromstring(decoded_string, dtype=np.uint8)
    frame = cv2.imdecode(frame, cv2.IMREAD_ANYCOLOR)
```

Hình 16. Server nhận yêu cầu POST và giải mã hình ảnh

- Nếu tìm thấy khuôn mặt, cắt và thay đổi kích thước khuôn mặt theo kích thước yêu cầu. Tiền xử lý hình ảnh và tính toán embeddings.
- Sử dụng mô hình phân loại để dự đoán danh tính của khuôn mặt.

```
predictions = model.predict_proba(emb_array)
best_class_indices = np.argmax(predictions, axis=1)
best_class_probabilities = predictions[
    np.arange(len(best_class_indices)), best_class_indices]
best_name = class_names[best_class_indices[0]]
```

Hình 17. Dự đoán danh tính khuôn mặt

Server trả về kết quả:

- Nếu xác suất dự đoán cao hơn 0.5, trả về tên của người dự đoán.
- Nếu không, trả về "Unknown".

```
if best_class_probabilities > 0.5:
    name = class_names[best_class_indices[0]]
else:
    name = "Unknown"
```

Hình 18. Trả về kết quả nhận diện

#### 4.5. Xử lý nhận hình ảnh/video, giá trị cảm biến từ firebase xuống ứng dụng di động

Tạo file tạm thời trên máy để tải video về và đổi định dạng video từ .h264 thành .mp4. Sau đó video sẽ được hiển thị lên VideoView của ứng dụng.

```
// Tạo file tạm thời để lưu video tải về
File localFile = File.createTempFile( prefix: "video", suffix: ".h264");
File convertedFile = new File(getExternalFilesDir( type: null), child: "video" + videoNumber + ".mp4");

tnhi1603
videoRef.getFile(localFile).addOnSuccessListener(new OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
    tnhi1603
    @Override
    public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
        // Chuyển đổi file h264 thành mp4
        String[] command = {"-i", localFile.getAbsolutePath(), convertedFile.getAbsolutePath()};
        int rc = FFmpeg.execute(command);

        if (rc == 0) {
            // Chuyển đổi thành công
            videoView.setVideoURI(Uri.fromFile(convertedFile));
            videoView.start();
            playPauseButton.setText("Pause");
        } else {
            Log.e(TAG, msg: "FFmpeg process failed with return code: " + rc);
        }
    }
}
```

Hình 19. Code lấy video từ Firebase xuống ứng dụng

Đọc trạng thái của cảm biến chuyển động PIR trên firebase realtime và điều chỉnh gửi thông báo đến người dùng

```
// Lắng nghe sự thay đổi của PIR
tnhi1603
pirRef.addValueEventListener(new ValueEventListener() {
    2 usages tnhi1603
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        String status = snapshot.getValue(String.class);
        if (status != null && status.equals("on") && !pirStatusOn) {
            sendNotification();
        }
    }
}
```

Hình 20. Code lấy trạng thái của cảm biến PIR

Gọi hàm sendNotification() nếu phát hiện chuyển động để gửi thông báo đến người dùng

```
private void sendNotification() {
    if (!isActivityVisible) { // Only send notification if the activity is not visible
        Intent intent = new Intent( packageContext: this, notification.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        PendingIntent pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0, intent, flags: PendingIntent.FLAG_UPDATE_CURRENT

        NotificationCompat.Builder builder = new NotificationCompat.Builder( context: this, CHANNEL_ID)
            .setSmallIcon(R.drawable.smartcam)
            .setContentTitle("SmartCam Notification")
            .setContentText(message)
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)
            .setContentIntent(pendingIntent)
            .setAutoCancel(true);

        NotificationManager notificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.notify( id: 0, builder.build());
        saveNotification( title: "Đã phát hiện chuyển động");
    } else {
        Log.d(TAG, msg: "Activity is visible, no need to show notification");
    }
}
```

Hình 21. Code gửi thông báo đến ứng dụng

#### 4.5. Thông báo trên ứng dụng di động

Thông báo sẽ được lưu lại trên firebase và hiển thị xuống phần log của thông báo.

```
public void saveNotification(String title) {
    String date = new SimpleDateFormat( pattern: "dd/MM/yyyy", Locale.getDefault()).format(new Date());
    String time = new SimpleDateFormat( pattern: "HH:mm:ss", Locale.getDefault()).format(new Date());

    NotificationData notification = new NotificationData(title, date, time);
    notificationsRef.push().setValue(notification);
}
```

Hình 22. Code lưu lại các thông báo trên firebase

## Chương 5: Thực nghiệm đề tài (demo)

### 5.1. Kịch bản demo 1

Người lạ hoặc các vật không xác định khi đi vào vùng quan sát của camera sẽ tiến hành nhận diện khuôn mặt. Nếu không nhận diện được khuôn mặt sẽ quay lại video và thông báo đến người dùng.

Video:

[https://drive.google.com/file/d/1FoOq4Jtp2PCvOX655sBRzgAHsbmNg3sC/view?usp=drive\\_link](https://drive.google.com/file/d/1FoOq4Jtp2PCvOX655sBRzgAHsbmNg3sC/view?usp=drive_link)

### 5.2. Kịch bản demo 2

Chủ nhà khi di chuyển vào vùng quan sát của camera sẽ tiến hành nhận diện khuôn mặt. Nếu dữ liệu khớp với dữ liệu khuôn mặt của chủ nhà sẽ tiến hành quay lại video nhưng không gửi thông báo đến người dùng.

