

Camera từ lâu đã là 1 vật dụng phổ biến ở các hộ gia đình hay các nơi công sở nhằm mục đích theo dõi và đảm bảo an ninh cho 1 khu vực nhất định. Tuy nhiên, những chiếc camera đó đòi hỏi người dùng phải theo dõi 24/24 nếu không muốn bỏ lỡ những việc xảy ra trong 1 ngày. Việc dành quá nhiều thời gian kiểm tra camera là điều không khả thi, dẫn đến có thể bỏ qua 1 số những rủi ro, ví dụ: không phát hiện kịp thời người lạ vào khu vực của camera vì lúc đó người dùng không có mặt để kiểm tra, dẫn tới những hậu quả có thể xảy ra như mất trộm. Hơn nữa 1 chiếc camera thông thường sẽ luôn truyền hình ảnh với 1 fps cố định dẫn tới việc tiêu tốn tài nguyên khi truyền tải do đem lại giá trị không cao khi không có sự việc gì bất thường xảy ra. Người dùng muốn 1 chiếc camera có thể tiết kiệm chi phí truyền tải, có những tính năng giúp quản lý an ninh, con người được dễ dàng và hiệu quả hơn sẽ có thể phải trả 1 chi phí lớn cho việc mua những chiếc camera có công nghệ mới và chi phí lắp đặt tốn kém. Để giải quyết chúng tôi đã xây dựng 1 nền tảng cho phép chiếc camera chỉ có chức năng ghi hình trở thành 1 chiếc camera thông minh có những chức năng như là cảnh báo người dùng khi có đối tượng lạ vào 1 khu vực nhất định, sử dụng camera làm công cụ điểm danh, truyền hình ảnh với fps thấp khi không có gì bất thường. Người dùng chỉ cần cài đặt các ứng dụng hoặc truy cập vào website được cung cấp để sử dụng những chức năng kể trên, cách này giúp người dùng tiết kiệm tối đa chi phí lắp đặt và mua thiết bị mới không cần thiết.

Gần đây, do những tiến bộ trong thị giác máy tính và công nghệ AI, camera có thể tự động phát hiện những người đáng ngờ hoặc các sự kiện bất thường trong thời gian thực [... - ...]. Những sự kiện bất thường đó sẽ được cảnh báo và gửi đến các công ty an ninh hoặc chủ sở hữu của các camera thông qua Internet hoặc các mạng truyền thông khác. Các nền tảng mới [... - ...] cũng cung cấp cho người dùng những công cụ để biến chiếc camera trở nên thông minh hơn. Những công nghệ mới cung cấp nhiều chức năng với độ chính xác cao cho người dùng. Tuy nhiên, chính việc quá nhiều những chức năng dẫn đến nếu 1 người thông thường mua dịch vụ sẽ có thể không sử dụng hết chức năng được cung cấp để có thể quản lý gia đình dẫn tới tổn chi phí nếu muốn chiếc camera của mình trở nên thông minh hơn.

Bài báo này chúng tôi tập trung giới thiệu 1 nền tảng đơn giản, tiết kiệm chi phí giúp người dùng quản lý an ninh 1 cách tự động, thông báo về những sự việc bất thường sẽ được chúng tôi thông báo trên điện thoại, máy tính, website thông qua các app và đường link cho chúng tôi cung cấp. Bài báo được tổ chức như sau. Phần 2 chúng tôi sẽ nói về những related work, phần 3 sẽ nói chi tiết về cách hoạt động của hệ thống của chúng tôi, phần 4 chúng tôi sẽ cung cấp những số liệu về kết quả đạt được và bàn luận về những khó khăn, hạn chế cần khắc phục. Phần cuối kết thúc bài báo này với một cách tóm lược

introduction

Cameras have been popular in households or offices for security monitoring. However, human employment is still required to capture and report any possible abnormal behaviour. It has been proven that humans could not withstand long-term focus, leading to unreliable

observations, false-positive reports or even unnoticed behaviour (strangers entering the area when no human supervision is available). Moreover, normal cameras always use fixed FPS transmission, which leads to high resource wastage due to low chance of occurrence for unusual events. Low transmission costs and accessible security management are generally in-demand, albeit with expensive installation. To solve this, we have built a platform that turns ordinary cameras with basic recording ability into smart cameras that can alleviate human-related faults by auto-detection of strange objects entering a given area, auto roll-calling, and motion-adapted FPS. This platform is available on our website or as a one-time desktop/mobile installation, providing easy and simple access to non-technician clients, and removing the need to buy costly equipment.

Recently, due to advances in computer vision and AI technology, cameras can automatically detect suspicious people or unusual events in real time [... - ...]. Such unusual events will be alerted and sent to the security companies or owners of the cameras via the Internet or other communication networks. New platforms [... - ...] also give users the tools to make the camera smarter. New technologies provide many functions with high precision to users. However, existing platforms often include unnecessary features for small-scale clients, resulting in overspending.

In this paper, we focus on a simple, cost-effective platform to help users manage security automatically, and notify them of unusual events through their phones, computers, and other devices. Computer installation and web-based implementation are available on the link we provide. The paper is organized as follows: Related works will be mentioned in part2. Part 3 will give detail about our system's workflow. Data on achieved results and difficulties, limitations and solution proposals will be discussed in part 4. The last section ends this paper with a summary

related work

Hiện tại, hệ thống đang có 2 chức năng chính là face detection và face recognition. Cụ thể ta có thể chia chúng thành 2 vấn đề chính: Phát hiện khuôn mặt(face detection) và phân loại khuôn mặt (face verification). Để giải quyết 2 bài toán trên, chúng tôi sử dụng 2 mô hình mạng khá nổi tiếng hiện nay: MTCNN (Multi-task Cascaded Convolutional Networks) và FaceNet.

Phần đầu tiên sẽ về face detection, chúng ta cần phát hiện ra các khuôn mặt trong ảnh hoặc frame trong video. Có rất nhiều mô hình mạng có thể giúp ta giải quyết vấn đề này, Yolo, SSD, MTCNN,... Nhưng để phù hợp với giai đoạn đầu của hệ thống, chúng tôi cần những mô hình đơn giản, dễ cài đặt để có thể đưa vào hoạt động ngay. Do đó, MTCNN là một mô hình mạng rất phù hợp. MTCNN bao gồm 3 lớp mạng riêng biệt tương ứng với 2 stage chính là P-Net, R-Net và O-Net.

Tiếp theo là về face verification với mô hình mạng FaceNet. Về cơ bản thì khi ta đưa ảnh khuôn mặt vào FaceNet, nó sẽ trả ra 1 vector 128 features cho 1 khuôn mặt. Sau đó ta phân nhóm các vector đó vào các nhóm để biết vector đó là mặt của ai.

method

1. Architectural overview

The CamAI platform includes 5 components. It aims to create a clear and easy-to-use workflow for users. Here, 5 components are Camera, Source server, AI server, Web application, Desktop application, Mobile application.

Each component has its own distinct features. However, they are closely connected together for the common purpose of the whole system.

The Camera with recording feature, it will transmit the raw streaming to the Source server. At the Source server, the computer will preprocess the raw streaming for the purpose of optimizing the video stream, making it fast, light and for the best use. After preprocessing, streaming video can be transmitted directly to the client. Streaming videos are also passed through an AI server to implement an AI module. After that, the streaming video with results of AI will be transmitted to all clients. The clients are Web application, Desktop application, Mobile application. In addition, Clients can send some image requests to the AI server and receive AI prediction results.

2. Source server

The source server is responsible for receiving frames from camera and preprocess images. This server implements 3 different functionalities: Frame denoise, Motion-adaptive FPS and Low-light enhancement.

a. Frame denoise:

Input frames go through a median filter to remove noise. Median filter is available in OpenCV

b. Motion-adaptive FPS:

Input frames go through an Optical Flow module to detect motions in the video. When there's no motion, streaming FPS is 1 FPS. As soon as any motion (above a given threshold) is detected, streaming FPS will be locked at 60FPS until 10 seconds have passed since the last detected motion.

c. Low-light enhancement:

Input frames go through a module that implements a technique for low-light illumination based on this [paper, cite at the end]. The workflow is as follow:

1. Dark/bright channel extraction
2. Atmospheric light estimation
3. Estimating transmission map using dark/bright channel prior
4. Refine transmission map using guided filter
5. Recover scene radiance using estimated transmission map and atmospheric light

3. AI server

Overview: AI server is built on the FastAPI framework. All clients will communicate with the server by using API. APIs are considered as an abstraction of your server (the user doesn't see your code and doesn't install anything, they simply call the API) and a simplification of the integration with third parties (with our platform, they are website, desktop and mobile application).

FastAPI is currently the modern, fast, high-performance for building APIs with Python. We chose FastAPI for server because some reasons:

- Simple syntax based on standard Python type hints
- Fast, low latency when used with uvicorn: according to Tectempower - an independent website that benchmarks web server by running a variety of tests on them
- Specially, the advantage of FastAPI over other frameworks (such as Flask or Django) is that FastAPI supports Asynchronous requests easily and strongly.
- Moreover, FastAPI supports creating API documentation (Swagger), that is useful for client developers.

3.1. AI server image processing

In this system, we currently provide 2 functions for still images: face detection and face recognition. Users are able to choose the function they want the server to perform and upload their own image to the server through the user interface of the web application, desktop application or mobile application. The server will return the result to the client device and the application will show the users the result.

3.1.a Face detection

We use MTCNN for this task for short response time. After users submit their image to the server by pressing the submit button, the client application will send the image to the server through an API. The server will then process the request image and send back the detection result of MTCNN in the form of a json message to the application.

3.1.b Face recognition

Similar to face detection, users also submit their image, the application will send the image to the server. We currently use FaceNet for this task. We want to focus on recognizing Vietnamese people, the dataset contains images of 14 Vietnamese people, their images are fed into Facenet to get the corresponding feature vectors called face embeddings and those are saved into a compressed numpy array file for later use. The image that the user sent to the server is extracted to get the embedding and classified by SVM. The result name is sent to the client application in a json message.

3.2. AI server streaming processing

About the input of the AI server, it will receive the data streaming from the Source server. This streaming video has been preprocessed by Source server. In this system, we currently provide 3 simple functions for streaming: real-time face detection, real-time gender recognition and realtime age recognition.

At the AI server, streaming data will be received continuously. The feature of the AI server is splitting video into frames. Then it performs an AI function in each frame.

3.2.a Real-time face detection

We use the `detect_face` function of `cvlib` library python because of the short response time. With each frame, we implement `detect_face` and get the result returned. We extract the coordinates from the result and draw a bounding box for face into this frame. After that, the system will return the streaming response with detect face result. If clients access, it will receive this streaming response through a API of *FastAPI*

Moreover, we have a small function. It is that if the vision of the camera does not have any face currently, when a face appears in the camera vision, the system will send a warning notification to all clients' devices right away. We use *Firebase Cloud Messaging* of

Google in order to send notifications because it is free, fast, easy-to-use and supports multi-language, multi-platform.

3.2.b. Real-time gender recognition

Firstly, in each frame, we also extract the face by using *detect_face* of *cvlib* library for better prediction. After that we use a pre-trained model (*detect_gender* of *cvlib*) to recognize the gender of the person. The result male or female, will be drawn into frame.

After that, the system will return the streaming response with detect face result. If clients access, it will receive this streaming response through a API of *FastAPI*

3.2.c Real-time age recognition

In each frame , we also extract the face by using *detect_face* of *cvlib* library for better prediction. After that we use a pre-trained model to recognize the age of the person. The result of age will be drawn into frame.

After that, the system will return the streaming response with detect face result. If clients access, it will receive this streaming response through a API of *FastAPI*

4. Application

4. Desktop application

Về phía ứng dụng desktop, hiện nay có nhiều ngôn ngữ có thể lập trình được tuy nhiên chúng tôi đã chọn python để lập trình. Bởi vì Python có nhiều framework hỗ trợ việc thiết kế giao diện và hỗ trợ openCV rất mạnh trong việc xử lý ảnh, đây cũng là ngôn ngữ quen thuộc nên không mất nhiều thời gian làm quen, hơn nữa ứng dụng thiết kế cũng nhẹ.

Sau đây chúng tôi sẽ trình bày về cách thiết kế

Ban đầu chúng tôi sử dụng framework pyqt5 để lập trình giao diện. Khi người dùng chọn 1 chức năng, hệ thống sẽ ghi nhận, gửi request tới server và nhận response. Sau đó chúng tôi sẽ xử lý kết quả nhận được bằng openCV và hiển thị kết quả ra màn hình cho người dùng

With desktop application, there are now many programming languages available but we chose python for programming. Because Python has many frameworks that support interface design and support openCV strongly for Computer Vision, this is also a familiar language, so it doesn't take much time to get used to, and the design application is also lightweight.

Initially we used the framework pyqt5 to program the interface. When the user selects a function, the system will record, send the request to the server and receive the response. We will then process the results received with openCV and display the results to the user's screen

4.2 Mobile Application

- Hiện nay, smartphone rất phổ biến, và để người dùng nhận thông tin cập nhật một cách nhanh và dễ dàng nhất thì việc phát triển một mobile app là điều cần thiết. Vì độ phổ biến của android lớn hơn so với ios, nên chúng tôi quyết định phát triển ứng dụng android trước tiên. Ứng dụng được viết bằng ngôn ngữ java, đơn giản và dễ tiếp cận, đặc biệt còn được hỗ trợ bởi Android Studio IDE chuyên để phát triển các ứng dụng dành cho Android. Ngoài ra, lập trình Android bằng Java được hỗ trợ nhiều thư viện hữu ích.
- Nowadays, smartphone is popular, and to let user receives update notification as fast and easy as possible, we developed a mobile application. Initially, we created an Android app because of its popularity over iOS. It is written in Java for an easy approach, strong support from Android Studio IDE and useful libraries.
- Ứng dụng có giao diện chọn chức năng theo Navigation Bar được cung cấp sẵn bởi Android Studio. Ngoài ra, app có tích hợp fab menu để người dùng có thể dễ dàng chọn hành động thực hiện, đồng thời giúp giao diện gọn gàng hơn.
- The application has an interface to select functions according to the Navigation Bar provided by Android Studio. In addition, the app has a built-in fab menu so that users can easily choose the action to take, while helping to make the interface cleaner.
-
- Để gửi các yêu cầu lên server, chúng tôi lựa chọn thư viện HttpURLConnection, mặc dù có nhiều thư viện dễ sử dụng hơn như Volley,... nhưng với HttpURLConnection, chúng tôi có thể tùy biến sâu hơn để có thể gửi request nhiều thứ hơn lên server như ảnh, file,...

To send requests to the server, we choose the HttpURLConnection library, although there are more easy-to-use libraries like Volley,... but with HttpURLConnection, we can customize it more deeply to be able to send more requests with

different data types to the server like photos, files, etc.

- Ngoài ra, để nhận thông báo trực tiếp từ server khi có báo động hoặc thông tin khẩn cấp, chúng tôi sử dụng service của Firebase. Với ưu điểm được phát triển bởi google, app có thể nhận thông báo một cách dễ dàng và nhanh nhất, giúp người dùng tránh việc miss các thông tin quan trọng.

Moreover, to receive notifications directly from the server when there is an alarm or emergency information, we use Firebase's service. With the advantage of being developed by google, the app can receive notifications easily and quickly, so that users avoid missing important information.

- Ứng dụng của chúng tôi còn cho phép người dùng quan sát camera của mình một cách real time tại bất cứ đâu. Hiện tại, chúng tôi deploy dòng stream lên web và nhúng trực tiếp video vào ứng dụng.

Our application also allows users to view their live streaming video in real time anywhere. We embed the video directly into the app.

experiment

discussion

1. Streaming server:

a. Problems

- i. Motion-dependent FPS: motion detection using optical flow proves to be slow on low-end devices.
- ii. Low-light enhancement: Very slow when utilizing guided filter
- iii. Denoising: Using gauss/median filter does not reliably remove noise

b. Proposed solution

- i. Use frame-differencing on motion-dependent FPS module

Discussion sketch 2

1/ Primary discovery

No new discovery has been made in this project. This platform only utilizes latest available technologies in computer vision, application libraries and frameworks

2/ Consistency with previous research

No new research was done yet in this project

3/ Explanation

No new research was done yet in this project

4/ Meaning of research

No new research was done yet in this project

5/ Possible mistakes of research; Pros and cons of research

No new research was done yet in this project

6/ Does the conclusion matches the data ?

No new research was done yet in this project

discussion sketch 3:

Paragraph 1: This paragraph provides a “big picture” perspective for readers to remind them of the importance of your study.

- *Summarize the major gap in understanding that your work is attempting to fill. What was the overarching hypothesis?*

There hasn't been any overreaching hypothesis yet. We will continue to research in the future

Paragraph 2: This paragraph provides a critical analysis of your major finding(s).

No major finding (yet)

conclusion

NOTE 2/10

Intro quan trọng => nêu ra thực trạng hiện nay cần thiết, thêm tài liệu tham khảo, hiện tại chưa nói chi tiết, thêm bài báo thống kê số lượng camera. Hiện nay có ng giải quyết nhưng có vấn đề, mình giải quyết tốt hơn = cách nào (cải thiện như nào) , quan tâm đến tài liệu tham khảo khoảng 3 năm đổ lại, hội thảo, ISI. Có thể tiền xử lí gì đó, thêm tài liệu tham khảo vào chỗ existing platform. hiện nay có công nghệ mới gì đó. Thêm 1 khổ để nói đến cống hiến, đâu là cái mới

Discuss: nhắc lại cống hiến

Related: hệ thống camera như nào, đường truyền, giao thức truyền, cái gì ít quá thì ko đưa vào (ví dụ MTCNN).

Kiến trúc: thay = kiến trúc đề xuất, method nên để chung vào kiến trúc

method: Trong mỗi khối thì nó là gì, ý nghĩa nó trong bài báo là gì, tại sao sử dụng pi, denoise,... Trong hệ thống thì gửi gì. NÓI ý nghĩa chức năng này là gì trong hệ thống. Thêm link đến docs opencv. Kịch bản xử lí, mục tiêu, nhận giữ liệu sao, trả dữ liệu sao.

Thêm ý nghĩa mục tiêu, sử dụng công nghệ gì, tại sao sử dụng(app)

Những cấu hình gì cần thay đổi, tham số quan trọng. GỒM tất cả các thư viện, ng dùng chỉ cần ấn vào là dc

Xem thử fps

cách lấy dữ liệu từ app, web, đt như thế nào

result : sau method, discuss sau result hoặc gộp vào luôn

1. Frame denoise

1. What it is, how it works
2. Explain the filters
3. Meaning

2. Optical flow

1. PC stats
2. Results
3. How optical flow works, how does it detect motion
4. Meaning of optical flow

2.5. >= 10 citations

3. Put in as many reference on how company implement camera that require sepialized tool

4. Related work used for flattering : D

5. Existing platforms: can cite blog/ paper/ (BBC news)

6. Put the contribution everywhere in paper

7. Academic writing rules

8. Link to manual of openCV (mention the function)

9. Paraphrase the optical flow (and explain)

10. Input and output for each module

11. Repeat the main idea of the whole paper in every-part

Chúng tôi xây dựng client trên 3 nền tảng là web app, desktop app và mobile app. Lí do chúng tôi sử dụng cả 3 nền tảng là vì hiện nay, điện thoại thông minh là thứ mà đa số ai cũng sở hữu. Vì vậy để người dùng có thể thao tác với hệ thống bất kì nơi đâu và bất kì lúc nào chúng tôi đã xây dựng mobile app. Tuy nhiên, người dùng không phải lúc nào cũng sẽ muốn quan sát trên điện thoại do hạn chế về màn hình và pin. Do đó, chúng tôi đã xây dựng web app để người dùng có thể quan sát trên màn hình lớn hơn trên máy tính, thích hợp với những người hay sử dụng máy tính để làm việc. Mặc dù web app có thể giúp người dùng vấn đề sử dụng trên máy tính, nhưng nếu người dùng sử dụng trên 1 máy tính có cấu hình yếu và người dùng khi đó không có nhu cầu sử dụng những việc khác ngoài việc theo dõi camera, thì việc sử dụng mở trình duyệt sẽ tiêu tốn bộ nhớ không cần thiết. Nên vì thế, chúng tôi đã xây dựng thêm desktop app giúp người dùng tiết kiệm bộ nhớ khi sử dụng

We build clients on 3 platforms: web app, desktop app and mobile app. The reason we use all 3 platforms is because nowadays, smartphones are something that most people own. So for users to be able to manipulate the system anywhere and anytime we have built mobile app. However, users will not always want to look on the phone due to screen and battery limitations. Therefore, we have built a web app so that users can observe on a larger screen on a computer, suitable for people who often use computers to work. Although the web app can help users with problems using on the computer, but if the user uses it on a computer with a weak configuration and the user then has no need to use other things, monitoring the camera, then using open browser will consume unnecessary memory. Therefore, we have built more desktop apps to help users save memory when using