**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**BÁO CÁO BTL THUỘC HỌC PHẦN:**

**CÔNG CỤ PHÁT TRIÊN PHẦN MỀM**

**Nghiên cứu xây dựng hệ thống chấm công tự động**

**bằng công cụ Emgu,OpenCV trên Visual Studio**

|  |  |
| --- | --- |
| GVHD: | Ths Nguyễn Thái Cường |
| Nhóm - Lớp: | 3 – 20222IT6003004 |
| Thành viên: | Nguyễn Phúc Hưng |
|  | Nguyễn Khánh Thọ  Nguyễn Quang Hưng  Đặng Việt Anh Đinh Anh Quân |
|  | *…* |
|  |  |
|  |  |

Hà Nội, Năm 2023

**MỤC LỤC**

**Lời nói đầu**

Là một sinh viên trường đại học Công nghiệp Hà Nội, báo cáo bài tập lớn là minh chứng cho những kiến thức đã có sau thời gian học tập. Trong quá trình hoàn thành bài tập lớn, ngoài sự cố gắng của nhóm thì nhóm cũng nhận được sự hướng dẫn tận tình của thầy **Ths**.Nguyễn Thái Cường. Qua đây, nhóm xin chân thành cảm ơn khoa Công nghệ thông tin, trường đại học Công nghiệp Hà Nội đã trang bị kiến thức cho em trong suốt quá trình học tập. Đặc biệt, nhóm cảm ơn tới thầy ThS. Nguyễn Thái Cường đã luôn giúp đỡ, hướng dẫn, chỉ bảo tận tình để giúp nhóm hoàn thành tốt bài tập lớn của mình. Nhóm đã cố gắng hoàn thành bài tập lớn nhưng vẫn rất mong nhận được sự đóng góp ý của thầy cô và các bạn để bài tập lớn cuả nhóm được hoàn thiện hơn.

**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

* 1. **Đặt vấn đề**

Trong thời kỳ phát triển mạnh mẽ của công nghệ thông tin, con người đang dần áp dụng các ứng dụng khoa học máy tính để phục vụ các công việc hằng ngày. Trong đó, những phần mềm bảo mật an ninh được thiết kế dựa trên hệ thống thị giác máy tính để thực hiện việc phát hiện, nhận dạng và sàng lọc xuất hiện ngày càng nhiều với những đối tượng ngày càng phong phú, đa dạng. Hiện nay, các ứng dụng nhận dạng khuôn mặt được xây dựng và phát triển trên rất nhiều thiết bị cũng như hệ thống nhằm giúp con người thuận tiện hơn trong việc quản lý, bảo mật, giám sát hoặc tìm kiếm. Các ứng dụng nhận diện khuôn mặt, nụ cười trong máy ảnh hoặc smartphone dần trở nên quen thuộc với mọi người.

Nhận thấy sự cần thiết trong việc nghiên cứu công nghệ nhận diện giúp con người có thể thuận tiện hơn trong nhiều công việc trong cuộc sống,kèm theo đó là tiền đề để nâng cao và phát triển mạnh về hệ thống thị giác máy tính, các ứng dụng về nhận dạng hình ảnh khuôn mặt nhằm mục đích tăng cáo tính bảo mật của hệ thống.

Đề tài với tên gọi “ nghiên cứu về thư viện OpenCV – nghiên cứu về hệ thống tự động chấm công bằng cách nhận dạng khuôn mặt” được thực hiện với mong muốn khai thác những khía cạnh về nhận dạng tự động và tự động vận hành nhằm giảm gánh nặng cho con người cũng như tiết kiệm thời gian. Có thể nói đây là phương hướng nghiên cứu có thể mang lại nhiều thuận lợi cũng nhợi ích cho xã hội.

* 1. **Một số nghiên cứu trước đây**
* Có rất nhiều nghiên cứu sử dụng các phương pháp khác nhau để giải quyết bài toán nhận dạng khuôn mặt, hai cách phổ biến nhất là thường được sử dụng là haarCascade và nhận dạng dựa trên xét tổng thể của khuôn mặt (Appearance based face recognition). Nhìn chung, các phương pháp đều có ưu nhược điểm nhằm giải quyết bài toán nhận dạng mặt người
  + Những năm 1960, Woody Bledsoe, Helen Chan Wolf, và Charles Bisson đã cho ra đời hệ thống nhận dạng khuôn mặt bán tự động đầu tiên. Hệ thống yêu cầu người quản trị xác định vị trí các đặc trưng như: mắt, tai, mũi và miệng trên các tấm ảnh. Sau đó, hệ thống sẽ tính khoảng cách và tỷ lệ đến một điểm tham chiếu chung. Cuối cùng là so sánh với kho dữ liệu lưu trữ.
  + Năm 1970, Goldstein, Harmon và Lesk sử dụng 21 dấu hiệu chủ quan như: màu tóc và độ dày môi để nhận dạng tự động. Giống như ở giải pháp trước, các phép đo và các vị trí được tính bằng tay nên đòi hỏi nhiều thời gian.
  + Năm 1988, Kirby và Sirovich áp dụng phương pháp phân tích thành phần chính (PCA), phương pháp này sử dụng chuẩn đại số tuyến tính cho vấn đề nhận dạng Nguyễn Hoàng Phúc – 1111540 Nghiên cứu thư viện OpenCv - Ứng dụng nhận dạng khuôn mặt người Trang 2 khuôn mặt. Đây được coi là cột mốc quan trọng vì nó cho thấy ít hơn một trăm giá trị cần thiết để nhận dạng một khuôn mặt được chuẩn hoá.
  + Năm 1991, Turk và Pentland sử dụng thuật toán eigenfaces để nhận dạng khuôn mặt tự động trên thời gian thực. Dù phương pháp này một phần nào bi hạn chế bởi yếu tố môi trường nhưng nó cũng góp phần đáng kể cho sự phát triển của công nghệ nhận dạng khuôn mặt.
  + Năm 2001, Paul Viola và Michael J. Jones đã cho ra đời đặc trưng Haar-Like, đó là những đặc trưng ảnh số cũng với phương pháp tích phân ảnh (integral images) để nâng cao khả năng rút trích đặc trưng trên thời gian thực.
  + Tháng 7-2003, Trần Phước Long và Nguyễn Văn Lượng dùng mạng Neural để dò tìm khuôn mặt trên ảnh, kết hợp phương pháp phân tích thành phần chính và biến đổi Cosine rời rạc để rút ra Vector đặc trưng làm đầu vào cho hai bộ nhận dạng SVM và HMM. Hệ thống tiếp cận từ các phương pháp học mạnh, nhưng nhiều trường hợp tối ưu cần thử nghiệm nhiều lần với nhiều bộ tham số khác nhau.
  + Năm 2005, Trần Lê Hồng Dũ sử dụng các đặc trưng lòi lõm để phát hiện khuôn mặt trên ảnh nhưng còn hạn chế về độ sáng, quan hệ hình học và các đặc trưng cùng mức.
  + Năm 2007, Lê Hồng Chuyên đã sử dụng mạng Neural để phát hiện khuôn mặt trên ảnh. Hệ thống đã nhận dạng được những khuôn mặt ở các tư thế: thẳng đứng, không thẳng đứng hoặc bị che mất một phần. Nhưng kết quả dò tìm không cao trong môi trường ảnh có nhiều người.
  + Năm 2013, Mã Trường Thành thực hiện đề tài “Điều khiển Robot Pioneer P3- DX bám sát đối tượng” đã sử dụng đặc trưng Haar-Like, bộ phân tầng Cascades of Boosted Classifiers và thuật toán Adaboost để nhận dạng đối tượng.
  + Năm 2013, Châu Ngân Khánh đã sử dụng đặc trưng Haar-Like, bộ phân tầng Cascades of Boosted Classifiers và thuật toán Adaboost để nhận dạng đối tượng bằng cách so khớp SIFT. Nhưng cơ sở dữ liệu không bao quát được hết sự thay đổi của khuôn mặt con người trên thời gian thực…

**1.3 Phạm vi đề tài.**

* Đề tài tập trung nghiên cứu gói thư viện mã nguồn mở OpenCV để giải quyết bài toán quan sát, phát hiện và nhận dạng đối tượng theo thời gian thực bằng đặc trưng HaarLike Features – Cascades of Boosted Classifiers.
* Triển khai nghiên cứu bằng việc thu thập ảnh, huấn luyện và giải quyết bài toán thực tế về nhận dạng khuôn mặt người bằng cách lưu các hoạt động của đối tượng dưới dạng ảnh và video thông qua Webcam.

**1.4 Phương pháp nghiên cứu**

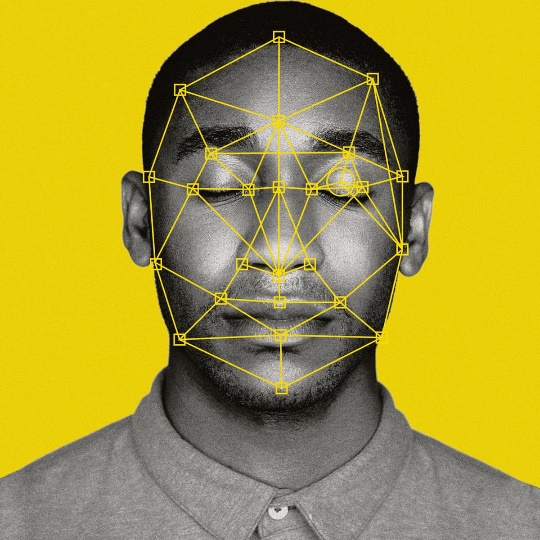
* + Nghiên cứu gói thư viện OpenCv – cụ thể là gói thư viện Emgu với ngôn ngữ được sử dụng là C# (C Sharp).
  + Sử dụng các hàm trong thư viện OpenCV bằng việc kết hợp đặc trưng HaarLike với thuật toán Adaboost và mô hình Cascades of Boosted Classifiers để thực hiện dò tìm đối tượng trên mặt phẳng ảnh theo thời gian thực để triển khai giải quyết bài toán đặt ra.

**1.5 Bố Cục và nội dung**

* Bài báo cáo nghiên cứu về việc nhận dạng khuôn mặt người với OpenCV với những nội dung sau:
  + **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**: Giới thiệu tổng quan về đề tài, xác định vấn đề, bối cảnh và nội dung bài toán cần giải quyết. Trình bày về lịch sử giải quyết vấn đề, các nghiên cứu và thành quả của những người đi trước trong cùng lĩnh vực, phạm vi và phương pháp nghiên cứu đề tài.
  + **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT:** Giới thiệu về bài toán dò tìm khuôn mặt, đặc trưng Haar-Like Features, thuật toán AdaBoost và bộ phân tầng Cascade of Boosted Classifier. Tổng quan về gói thư viện OpenCV để nghiên cứu về nhận dạng mặt người, thư viện Emgu nền tảng của OpenCV trên nền ngôn ngữ C# và hệ thống gửi tin nhắn GSMComm.
  + **CHƯƠNG 3,4,5: NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU:** Thể hiện cách giải quyết bài toán thông qua hệ thống thực nghiệm. Xây dựng một bộ huấn luyện nhằm nhận dạng khuôn mặt người trên mặt phẳng ảnh. Nêu lên các thiết kế về chương trình và trình bày về thực nghiệm chương trình kèm theo hướng dẫn

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1 Giới thiệu về bài toán dò tìm khuôn mặt**



Hình 1.Dò tìm khuôn mặt

Bài toán dò tìm khuôn mặt (face detection) là một trong những bài toán quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nó đặt ra mục tiêu nhận diện và phát hiện khuôn mặt trong một hình ảnh hoặc một đoạn video.

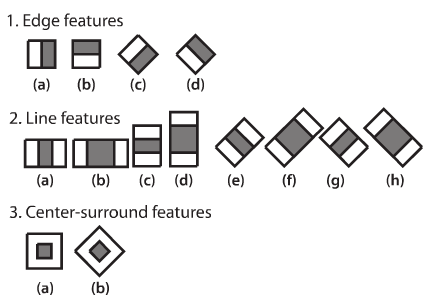
Bài toán này thường được giải quyết thông qua việc sử dụng các thuật toán học máy và deep learning để học và tìm kiếm các đặc trưng của khuôn mặt, ví dụ như hình dạng, màu sắc, kích thước và đặc điểm hình học khác.

Các ứng dụng của bài toán dò tìm khuôn mặt rất đa dạng, từ nhận dạng khuôn mặt để mở khóa điện thoại, nhận diện khuôn mặt để đăng nhập vào các ứng dụng, đến giám sát an ninh và phát hiện tội phạm. Tuy nhiên, bài toán này còn đối mặt với nhiều thách thức như sự biến đổi về ánh sáng, góc nhìn và vật liệu, cũng như sự đa dạng của các khuôn mặt trong thực tế.

**2.2 Đặc trưng Haar-Like Features**

Đặc trưng Haar-Like Features là một trong những phương pháp được sử dụng trong bài toán dò tìm khuôn mặt. Phương pháp này được đề xuất bởi Viola và Jones vào năm 2001 và đã trở thành một trong những phương pháp phổ biến trong lĩnh vực nhận dạng khuôn mặt.

Trong bài báo của tác giả *Paul Viola và Michael Jones* đã trình bày một phương pháp mới và nhanh hơn để xử lý hình ảnh và phát hiện khuôn mặt bằng cách sử dụng các đặc điểm hình chữ nhật như hình dưới đây. Các **đặc điểm hình chữ nhật** tương tự như nhân được sử dụng để phát hiện các đặc điểm khác nhau của khuôn mặt như mắt và các nốt như trong hình minh họa.



Hình 2. Đặc trưng Haar-Like

Các tính năng hình chữ nhật được chạy lần lượt trên hình ảnh và tổng số pixel nằm trong phần màu trắng được trừ cho tổng số pixel nằm trong phần màu đen.

Trong đó:

Edge features: Đặc trưng cạnh

Line features: Đặc trưng đường

Center-suround features: Đặc trưng trung tâm

Trong hình minh họa bên dưới, đặc điểm hình chữ nhật đầu tiên đang tính toán cho sự khác biệt về cường độ giữa vùng mắt và vùng má trên khuôn mặt. Và đặc điểm hình chữ nhật thứ hai là đo sự chênh lệchvề cường độ giữa hai vùng mắt và sống mũi. Bộ lọc Haar chỉ có thể nhìn cụ thể vào một vùng trong cửa sổ để nhận diện.



Hình 3. Ví dụ

* 1. **Trích đặc trưng Haar-Like từ ảnh**

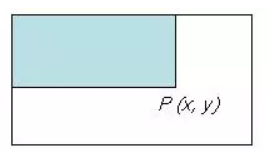
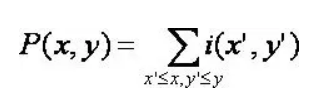
Đặc trưng Haar-like sẽ diễn đạt được tri thức về các đối tượng trong ảnh (bởi vì nó biểu diễn mối liên hệ giữa các bộ phận của đối tượng), điều mà bản thân từng điểm ảnh không diễn đạt được – dựa trên việc co giãn của khung cửa sổ tìm kiếm. Để tính giá trị các đặc trưng Haar-Like, ta tính sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức:

****

Vì vậy chúng ta sẽ thấy rằng, với tổng các pixel trên ảnh *(pixel vùng trắng và pixel vùng đen)* sẽ cho ta các giá trị của đặc trưng Haar – Like. Nhưng để tính toán những giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, dẫn đến không thể đáp ứng được cho các ứng dụng đòi hỏi tính thời gian thực*(run-time)* trực tiếp từ Camera *(hay Webcam trên Laptop)*. Do đó Viola và Jones đề xuất một khái niệm gọi là ***“tich phân ảnh”*** *(Integral Image)* để tính toán nhanh cho các đặc trưng cơ bản của Haar-like.

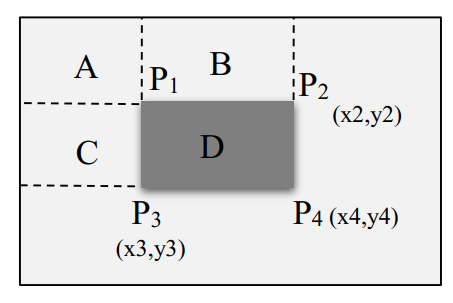
Tích phân ảnh được định nghĩa là 1 mảng 2 chiều có kích thước bằng kích thước ảnh cần tìm.

Mỗi phần tử trong mảng có giá trị bằng tổng các phần từ ở phái trên và bên trái của nó. Bắt đầu từ vị trí trên bên trái và kết thúc tại vị trí dưới bên phải của ảnh, tính lần lượt từ trái qua phải và từ trên xuống dưới.

Hình 4. Hình minh họa tích phân ảnh và công thức tính tích phân ảnh

Sau khi tính tích phân ảnh, ta tính tổng điểm ảnh theo các sau:



Hình 5.Tính nhanh tổng điểm ảnh trong vùng D

Tính điểm D:  
▪Tổng = A + B + C + D  
▪D =Tổng – (A+B) – (A+C) + A

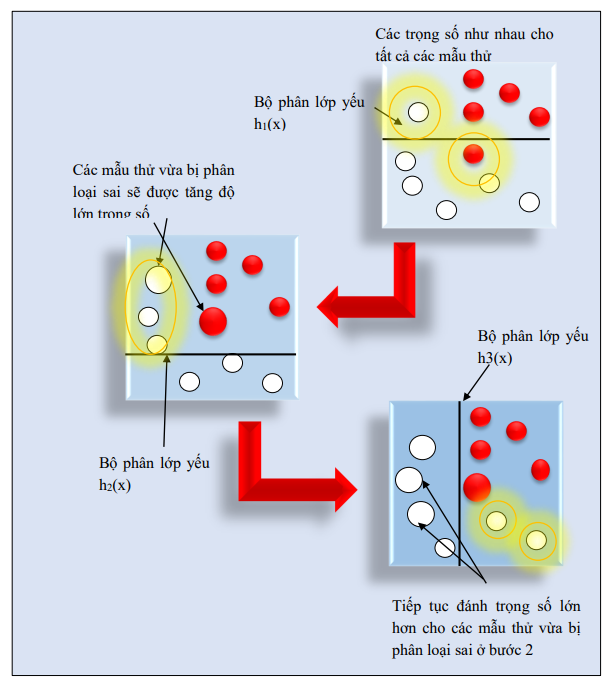
Với A + B + C + D chính là giá trị tại điểm P4 trên đảo hàm ảnh , tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1.  
 Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

D= P4-P2-P3+P1

D= (x4,y4)-(x2,y2)-(x3,y3)+(x1,y1)

**2.4 Thuật toán AdaBoost**

Thuật toán AdaBoost (Adaptive Boosting) là một trong những thuật toán quan trọng và phổ biến nhất được sử dụng trong bài toán dò tìm khuôn mặt. AdaBoost kết hợp nhiều bộ phân loại yếu thành một bộ phân loại mạnh. Mỗi bộ phân loại yếu được huấn luyện để phân loại một tập con của dữ liệu và nhận được một trọng số ứng với độ chính xác của nó. AdaBoost sẽ tăng trọng số của các dữ liệu bị phân loại sai và giảm trọng số của các dữ liệu được phân loại đúng. Sau đó, các bộ phân loại yếu sẽ được sử dụng để tạo ra một bộ phân loại mạnh bằng cách kết hợp dự đoán của chúng theo một quy tắc đa số.



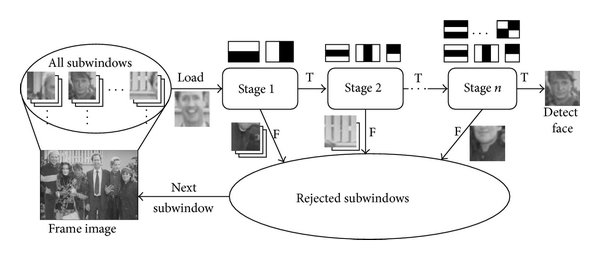
Hình 6. Kỹ thuật Adaboost

**2.5 Bộ phân tầng Cascade of Boosted Classifier**

Bộ phân tầng Cascade of Boosted Classifier là một phương pháp dò tìm khuôn mặt cải tiến dựa trên AdaBoost. Nó bao gồm nhiều giai đoạn, mỗi giai đoạn sử dụng một bộ phân loại được huấn luyện để loại bỏ các vùng không phải là khuôn mặt. Ở mỗi giai đoạn, hình ảnh được chia thành nhiều vùng nhỏ hơn, các vùng này được xem xét bởi các bộ phân loại với độ chính xác khác nhau. Các vùng bị loại bỏ không phải là khuôn mặt sẽ không được xem xét trong các giai đoạn tiếp theo. Các khuôn mặt được phát hiện trong các giai đoạn sau sẽ được xác định chính xác hơn vì chỉ cần xem xét các vùng nhỏ hơn và đặt ưu tiên cho các vùng có khả năng cao hơn chứ không cần phân tích toàn bộ hình ảnh.

Bộ phân tầng Cascade of Boosted Classifier có nhiều ưu điểm, bao gồm tăng tốc độ dò tìm, cải thiện độ chính xác và giảm độ phức tạp tính toán. Tuy nhiên, cần phải cân nhắc kỹ lưỡng trong việc lựa chọn các tham số và bộ phân loại để đạt được hiệu quả cao nhất.

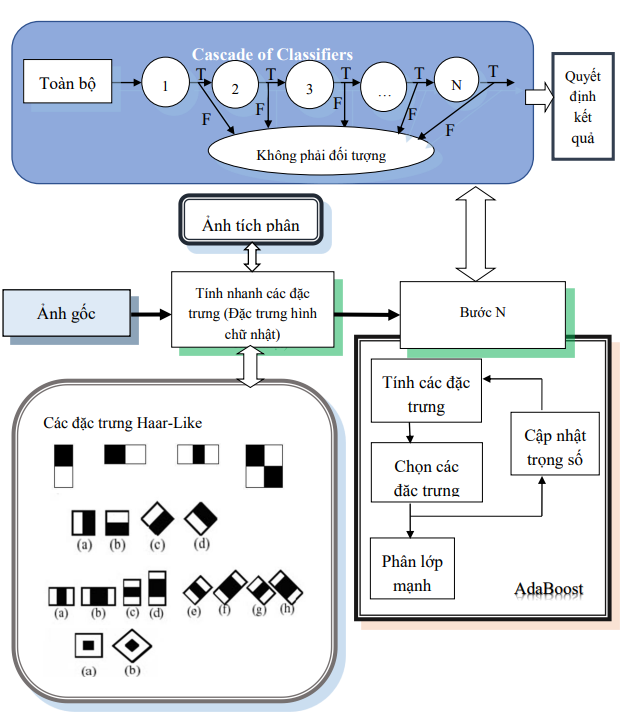
Để được nhận dạng là khuôn mặt, các cửa sổ ảnh nhỏ sẽ phải đi qua hết các stage của bộ phân loại. Các bộ lọc ở mỗi stage đằng sau sẽ được huấn luyện bằng các mẫu không có đối tượng mà các stage trước nó nhận định sai, do đó sự kết hợp các lớp lọc này sẽ giúp hệ thống giảm thiếu khả năng nhận định sai.



Hình 7.Bộ phân tầng Haar Cascade

* **Bước 1:** Hình ảnh (đã được gửi đến bộ phân loại) được **chia thành các phần nhỏ**(hoặc các cửa sổ con như trong hình minh họa).
* **Bước 2:** Chúng tôi đặt N bộ dò theo cách xếp tầng trong đó mỗi bộ phát hiện sự kết hợp của các loại **đặc trưng khác nhau** từ các hình ảnh (ví dụ: đường thẳng, cạnh, hình tròn, hình vuông) được truyền qua. Giả sử khi việc trích xuất đối tượng địa lý được thực hiện, mỗi phần phụ được gán một **giá trị tin cậy**.
* **Bước 3:** Hình ảnh (hoặc hình ảnh phụ) có độ tin cậy cao nhất được phát hiện dưới dạng khuôn mặt và được gửi đến **bộ tích lũy** trong khi phần còn lại bị từ chối. Do đó, Cascade tìm nạp khung hình / hình ảnh tiếp theo nếu còn lại và bắt đầu lại quá trình.

**2.6 Định vị đối tượng**

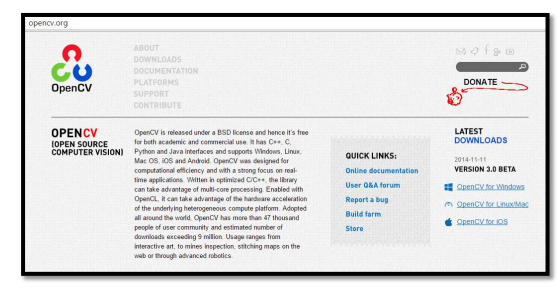


Hình 8. Dò tìm đối tượng với Haar-Like Features và Cascade of Boosted Classifiers

* Image Kernels là 1 ma trận nhỏ dùng để sử dụng các hiệu ứng chỉnh sửa. Trong mỗi ô của ma trận sẽ chứa giá trị tờ 0 đến 255 để biểu thị giá trị sang tối của từng pixel ảnh trong ảnh chính. Từ đó, người dùng sẽ áp dụng các công thức tính toán để làm mờ, làm rõ nét bằng các phép nhân ma trận. Hơn thế nữa, chúng ta có thể dùng phép tích phân ảnh để sử dụng trong việc nhận dạng khuôn mặt.
* Trình tự hoạt động: Sau khi quét ảnh từ camera, hệ thống sẽ chuyển hình ảnh quét được thành ảnh đen trắng. Sau đó chuyển hình ảnh đen trắng đã có thành dạng Image Kernels. Sau đó hệ thống trích các đặc trưng Haar-Like từ ảnh thông qua tích phân ảnh và sử dụng các hình chữ nhật đặc trưng để phát hiện gương mặt trong khung hình. Sau đó được đưa qua bộ điều chỉnh AdaBoost để loại bỏ bớt các đặc trưng không cần thiết. Các đặc trưng còn lại sẽ được đưa qua bộ phân lớp để quyết định xem có phải mặt người hay không.

**2.7 Tổng quan về thư viện OpenCV để nghiên cứu về nhận dạng mặt người**

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở cung cấp nhiều công cụ và thuật toán để xử lý ảnh và thị giác máy tính. OpenCV được viết bằng C++, nhưng cũng có thể sử dụng các ngôn ngữ khác như Python và Java thông qua các giao diện lập trình ứng dụng (API).



Hình 9. Thư viện OpenCV

OpenCV cung cấp nhiều tính năng để hỗ trợ nhận dạng mặt người, bao gồm:

1. Xử lý ảnh: OpenCV cung cấp các tính năng để xử lý ảnh như đọc, ghi, chuyển đổi, cắt, phóng to, thu nhỏ ảnh, làm mịn ảnh, tăng độ tương phản và cân bằng sáng tối.
2. Dò tìm khuôn mặt: OpenCV cung cấp các thuật toán để dò tìm khuôn mặt như Haar Cascades, LBP (Local Binary Pattern), và HOG (Histogram of Oriented Gradients).
3. Nhận dạng khuôn mặt: OpenCV cung cấp các thuật toán để nhận dạng khuôn mặt như Eigenfaces, Fisherfaces, và Local Binary Patterns Histograms (LBPH).
4. Ghi chú và vẽ hình trên ảnh: OpenCV cung cấp các công cụ để ghi chú và vẽ hình trên ảnh để đánh dấu khuôn mặt đã được tìm thấy hoặc hiển thị kết quả của thuật toán.

OpenCV cũng có thể được sử dụng để kết hợp nhiều thuật toán nhận dạng khuôn mặt và tạo ra một hệ thống hoàn chỉnh để nhận dạng khuôn mặt trong thời gian thực.

Việc sử dụng OpenCV để nghiên cứu về nhận dạng khuôn mặt là rất phổ biến vì thư viện này cung cấp nhiều tính năng và thuật toán cho bài toán này và có thể được sử dụng trên nhiều nền tảng khác nhau như Windows, Linux, MacOS, Android và iOS.

**2.8 Thư viện EmguCV**

Emgu CV là gói thư viện được viết hoàn toàn bằng ngôn ngữ C#. Nó bao gồm các chức năng của OpenCV, cho phép gọi các hàm của OpenCV từ Visual Studio bằng các ngôn ngữ .Net như: C#, VB, VC++, v.v... Vì Visual Studio/.Net không thể gọi trực tiếp các hàm của OpenCV nên Emgu là giải pháp cho vấn đề này. Emgu có thể chạy trên Windows, Linux, iOS, Mac OS X, Android và Windows Phone.



Hình 10. Thư viện EmguCV

**Chương 3: Mô tả Usecase**

3.1 Danh sách Usecase

|  |  |  |
| --- | --- | --- |
| **Tên UC** | **Mã UC** | **Chức năng** |
| Đăng nhập | UC01 | Đăng nhập vào hệ thống |
| Thêm tài khoản | UC02 | Thêm tài khoản |
| Cập nhật tài khoản | UC03 | Sửa thông tin tài khoản |
| Xóa tài khoản | UC04 | Xóa tài khoản |
| Tìm tài khoản | UC05 | Tìm kiếm thông tin tài khoản |
| Thêm nhân viên | UC06 | Thêm thông tin nhân viên |
| Cập nhật nhân viên | UC07 | Sửa thông tin nhân viên |
| Xóa nhân viên | UC08 | Xóa thông tin nhân viên |
| Tìm kiếm nhân viên | UC09 | Tìm kiếm thông tin nhân viên |
| Điểm danh | UC10 | Điểm danh |
| Thông báo nhân viên vắng | UC11 | Thông báo tên của nhân viên vắng mặt |
| Xuất danh sách theo ngày | UC12 | Xuất danh sách vắng mặt và đi làm theo ngày |
| Gửi báo cáo qua email | UC13 | Gửi file báo cáo thông qua email |
| Lấy lại mật khẩu | UC14 | Lấy lại mật khẩu |

3.2 Mô tả chi tiết UC

### UC01: Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Login | **Code** | UC01 |
| **Description** | Cho phép actor đăng nhập vào hệ thống | | |
| **Actor** | Admin/Staff | **Trigger** | Actor bấm nút login |
| **Pre-condition** | Điền đủ vào ô tài khoản và mật khẩu | | |
| **Post condition** | Chuyển tới trang menu với role tương ứng | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Login thành công** | | | |
| 1 | Actor nhập tên đăng nhập/mật khẩu và click vào button Login trên trang đăng nhập ([SC01](#_SC01:_My_Profile)). |  |  |
|  |  | 2 | Kiểm tra tên đăng nhập/mật khẩu là chính xác sau đó chuyển tới trang default role tương ứng ([SC03](#_SC03:_My_Group), [SC02](#_SC02:_List_users)...) |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS01 | “Tên Đăng Nhập và/hoặc Mật Khẩu của bạn không chính xác. Vui lòng kiểm tra và thử lại.”  Message thông báo khi actor nhập sai tên đăng nhập/mật khẩu |

### UC02: Thêm tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Thêm tài khoản | **Code** | UC02 |
| **Description** | Cho phép actor thêm tài khoản staff/admin trong công ty | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Thêm mới]** trên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Thêm thành công tài khoản | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thêm tài khoản thành công** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn left menu **Quản lý tài khoản/Thêm mới** |  |  |
|  |  | 2 | Hệ thống Load trang Thêm mới thông tin tài khoản |
| 3 | Nhập vào các thông tin được yêu cầu. |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Lưu thông tin người dùng mới vào bảng Account trên CSDL, thông báo thành công và chuyển sang trang Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message: N/A**

### UC03: Cập nhật tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Cập nhật thông tin tài khoản | **Code** | UC03 |
| **Description** | Cho phép actor cập nhật thông tin tài khoản trong hệ thống. | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Cập nhật]** ở bên phải tài khoản cần sửa trong danh sách tài khoản của hệ thống được hiển thị trên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Cập nhật tài khoản thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Cập nhật thông tin tài khoản thành công** | | | |
| 1 | Actor click button **[Cập nhật]** ở bên phải tài khoản cần sửa trong danh sách tài khoản của hệ thống được hiển thị trên màn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống lấy thông tin của tài khoản cần sửa trong bảng Account trên CSDL rồi hiển thị lên trang Thông tin tài khoản |
| 3 | Cập nhật những thông tin cần thiết rồi ấn button **[Cập nhật]** |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Hệ thống update thông tin tài khoản trong bảng WorkerList trên CSDL. Hệ thống thông báo thành công và chuyển về màn hình Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC04: Xóa tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xóa tài khoản | **Code** | UC04 |
| **Description** | Cho phép actor xóa tài khoản trong công ty | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Xóa]** ở bên phải tài khoản cần sửa trong danh sách tài khoản của hệ thống được hiển thị trên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xóa tài khoản thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xóa tài khoản thành công** | | | |
| 1 | Actor click button **[Xóa]** ở bên phải tài khoản cần xóa của hệ thống được hiển thị trên màn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống hiển thị thông báo “Xác nhận xóa ?” |
| 3 | Actor click **[Xác nhận]** |  |  |
|  |  | 4 | Hệ thống xóa thông tin tài khoản đó ra khỏi bảng Account và những bảng liên quan trên CSDL. Thông báo thành công và chuyển sang trang Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS02 | “Xác nhận xóa  Message thông báo khi actor nhập xóa |

### UC05: Tìm kiếm tài khoản

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Tìm kiếm tài khoản | **Code** | UC05 |
| **Description** | Cho phép actor tìm kiếm tài khoản bằng tên đăng nhập hoặc email | | |
| **Actor** | Admin | **Trigger** | Actor click button **[Tìm kiếm]** bên cạnh thanh tìm kiếmtrên màn hình Quản lý tài khoản |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Tìm kiếm tài khoản thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Tìm kiếm tài khoản thành công** | | | |
| 1 | Actor nhập tên đăng nhập hoặc email của tài khoản cần tìm vào thanh tìm kiếm rồi click button **[Tìm kiếm]** trênmàn hình Quản lý tài khoản |  |  |
|  |  | 2 | Hệ thống tìm tài khoản có tên đăng nhập/ email trùng với tên đăng nhập/ email actor nhập trong bảng Account trên CSDL rồi hiển thị lên list tài khoản trên màn hình Quản lý tài khoản |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC06: Thêm nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Quản lý nhân viên | **Code** | UC06 |
| **Description** | Cho phép actor quản lý nhân viên trong công ty | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Thêm mới]** trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Thêm thành công nhân viên | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thêm nhân viên thành công** | | | |
| 1 | Từ màn hình bất kỳ, actor lựa chọn left menu **Quản lý nhân viên /Thêm mới** |  |  |
|  |  | 2 | Hệ thống Load trang Thêm mới thông tin người dùng |
| 3 | Nhập vào các thông tin được yêu cầu. |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Lưu thông tin người dùng mới vào bảng WorkerList trên CSDL, thông báo thành công và chuyển sang trang Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC07: Cập nhật thông tin nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Cập nhật thông tin nhân viên | **Code** | UC07 |
| **Description** | Cho phép actor cập nhật thông nhân viên trong phòng ban. | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Cập nhật]** ở bên phải nhân viên cần sửa trong danh sách nhân viên tại phòng ban được hiển thị trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Cập nhật thông tin nhân viên thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Cập nhật thông tin nhân viên thành công** | | | |
| 1 | Actor click button **[Cập nhật]** ở bên phải nhân viên cần sửa trong danh sách nhân viên tại phòng ban được hiển thị trên màn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống lấy thông tin của nhân viên cần sửa trong bảng WorkerList trên CSDL rồi hiển thị lên trang Thông tin nhân viên |
| 3 | Cập nhật những thông tin cần thiết rồi ấn button **[Cập nhật]** |  |  |
|  |  | 4 | Validation các thông tin nhập vào. |
| 5 | Sửa lại những thông tin chưa đúng (nếu hệ thông check validation chưa đúng). |  |  |
|  |  | 6 | Hệ thống update thông tin nhân viên trong bảng WorkerList trên CSDL. Hệ thống thông báo thành công và chuyển về màn hình Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC08: Xóa nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xóa nhân viên | **Code** | UC08 |
| **Description** | Cho phép actor xóa nhân viên trong công ty | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Xóa]** ở bên phải nhân viên cần sửa trong danh sách nhân viên tại phòng ban được hiển thị trên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Xóa nhân viên thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xóa nhân viên thành công** | | | |
| 1 | Actor click button **[Xóa]** ở bên phải nhân viên cần sửa trong danh sách nhân viên tại phòng ban được hiển thị trên màn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống hiển thị thông báo “Xác nhận xóa ?” |
| 3 | Actor click **[Xác nhận]** |  |  |
|  |  | 4 | Hệ thống xóa thông tin nhân viên đó ra khỏi bảng WorkerList và những bảng liên quan trên CSDL. Thông báo thành công và chuyển sang trang Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC09: Tìm kiếm nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Tìm kiếm nhân viên | **Code** | UC09 |
| **Description** | Cho phép actor tìm kiếm nhân viên trong phòng ban bằng mã ID | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Tìm kiếm]** bên cạnh thanh tìm kiếmtrên màn hình Quản lý nhân viên |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Tìm kiếm nhân viên thành công | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Tìm kiếm nhân viên thành công** | | | |
| 1 | Actor nhập mã Id của nhân viên cần tìm vào thanh tìm kiếm rồi click button **[Tìm kiếm]** trênmàn hình Quản lý nhân viên |  |  |
|  |  | 2 | Hệ thống tìm nhân viên có mã ID trùng với ID actor nhập trong bảng WorkerList trên CSDL rồi hiển thị lên list nhân viên trên màn hình Quản lý nhân viên |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC10: Điểm danh nhân viên

**Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Điểm danh nhân viên | **Code** | UC10 |
| **Description** | Cho phép actor điểm danh nhân viên trong phòng ban. | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Start Điểm danh]** |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống | | |
| **Post condition** | Actor có được danh sách nhân viên đã điểm danh trong ngày | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Danh sách nhân viên đã điểm danh trong ngày** | | | |
| 1 | Actor bấm “Start điểm danh” |  |  |
|  |  | 2 | Hệ thống lấy list dữ liệu “image” trong bảng WorkerList trong CSDL. |
|  |  | 3 | Hệ thống quét khuôn mặt rồi so sánh trong dữ liệu vừa lấy. |
|  |  | 4 | Hệ thống thêm tên nhân viên và thời gian điểm danh vào danh sách nhân viên đã điểm danh được hiển thị trên màn hình. |
| 5 | Actor bấm “End điểm danh” |  |  |
|  |  | 7 | Hệ thống hiển thị thông báo kết thúc điểm danh. |
|  |  | 8 | Hệ thống thêm danh sách nhân viên đã điểm danh vào bảng Attendance trong csdl. |
|  |  | 9 | Hệ thống đưa ra thông báo cập nhật dữ liệu thành công |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC11: Thông báo nhân viên vắng mặt

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Thông báo nhân viên vắng mặt | **Code** | UC11 |
| **Description** | Cho phép actor có danh sách nhân viên vắng mặt | | |
| **Actor** | Staff | **Trigger** | Actor bấm nút End khi kết thúc điểm danh |
| **Pre-condition** | Không có | | |
| **Post condition** | Không có | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Thông báo nhân viên vắng mặt** | | | |
|  |  | 1 | Use case bắt đầu khi hệ thống điểm danh đạt đến giới hạn thời gian quy định nhưng chưa đạt đủ nhân số đã được đưa ra trong CSDL. |
|  |  | 2 | Hệ thống truy xuất dữ liệu từ bảng Attendance và bảng WorkerList để lấy ra danh sách nhân viên vắng mặt |
|  |  | 3 | Sau đó hiển thị danh sách vào bảng Vắng mặt trong trang Attendance |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

**N/A**

### UC12: Xuất danh sách theo ngày

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Xuất danh sách theo ngày | **Code** | UC12 |
| **Description** | Cho phép actor có thông tin danh sách nhân viên nghỉ/ đi làm theo ngày | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Export]** trong trang Report |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin ngày-tháng-năm | | |
| **Post condition** | Không có | | |

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Xuất danh sách theo ngày** | | | |
| 1 | Người dùng truy cập vào hệ thống và chọn chức năng "Report". |  |  |
|  |  | 2 | Hệ thống truy xuất dữ liệu từ bảng Attendance trong cơ sở dữ liệu sao cho dữ liệu ngày tháng năm trong bảng Attendance trùng với ngày/tháng/năm mà người quản lý muốn xuất. |
|  |  | 3 | Hệ thống hiển thị danh sách nhân viên đi làm và nghỉ của ngày hôm đó vào bảng trong trang Report |
| 4 | Người dùng bấm Export |  |  |
|  |  | 5 | Hệ thống xuất danh sách báo cáo vào file Excel có tên:”Ngày\_Tháng\_Năm\_Phòng ban” vào folder Report. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

##### **N/A**

### UC13: Gửi báo cáo qua email

##### **Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Gửi báo cáo qua email | **Code** | UC13 |
| **Description** | Cho phép actor gửi báo cáo danh sách nhân viên nghỉ/ đi làm theo ngày qua email | | |
| **Actor** | Staff | **Trigger** | Actor click button **[Send]** trong trang Report |
| **Pre-condition** | Actor đã đăng nhập vào hệ thống, điền thông tin ngày-tháng-năm | | |
| **Post condition** | Không có | | |

##### **Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Gửi danh sách theo ngày qua email** | | | |
| 1 | Người dùng truy cập vào hệ thống và chọn chức năng "Report". |  |  |
|  |  | 2 | Hệ thống truy xuất dữ liệu từ bảng Attendance trong cơ sở dữ liệu sao cho dữ liệu ngày tháng năm trong bảng Attendance trùng với ngày/tháng/năm mà người quản lý muốn xuất. |
|  |  | 3 | Hệ thống hiển thị danh sách nhân viên đi làm và nghỉ của ngày hôm đó vào bảng trong trang Report |
| 4 | Người dùng bấm Send |  |  |
|  |  | 5 | Hệ thống xuất danh sách báo cáo vào file Excel có tên:”Ngày\_Tháng\_Năm\_Phòng ban” vào folder Report. Sau đó gửi file excel vừa xuất đến 1 hoặc nhiều email đã nhập trước. |

##### **Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

##### **System Message**

|  |  |
| --- | --- |
| MS03 | “Gửi thành công”  Message thông báo khi hệ thống đã gửi thành công tới các email |
| MS04 | “Gửi thất bại”  Message thông báo khi hệ thống không gửi được file tới các email |

### UC14: Lấy lại mật khẩu

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Register | **Code** | UC14 |
| **Description** | Cho phép actor đăng ký tài khoản vào hệ thống | | |
| **Actor** | Staff | **Trigger** | Actor bấm nút Cant login |
| **Pre-condition** | Điền đủ vào ô tài khoản và mật khẩu, email | | |
| **Post condition** | Chuyển tới trang đăng nhập | | |

**Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | | **System** | |
| **Main Flow: Đăng ký thành công** | | | |
| 1 | Người dùng truy cập vào trang đăng ký trên hệ thống. |  |  |
| 2 | Người dùng nhập thông tin email, tên đăng nhập và họ và tên, và bấm nút Recovery |  |  |
|  |  | 3 | Hệ thống kiểm tra tính hợp lệ của thông tin người dùng. Nếu thông tin hợp lệ, hệ thống sẽ gửi 1 mã mật khẩu mới qua email |
|  |  | 4 | Hệ thống chuyển đến trang Đăng nhập. |

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **Rule No.** | **Rule** | **Description** |
|  |  |  |

**System Message**

|  |  |
| --- | --- |
| MS05 | “Vui lòng điền đủ thông tin cá nhân để lấy lại tài khoản.”  Message thông báo khi actor không nhập đủ thông tin được yêu cầu |
| MS06 | “Lấy mật khẩu thành công”  Message thông báo khi actor điền đủ thông tin và hợp lệ sau khi hệ thống kiểm tra |

**Chương 4: Thiết kế giao diện**

4.1 Giao diện dành cho admin

4.2 Giao diện dành cho quản lý