ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC AN GIANG

ĐỒ ÁN

CHỦ ĐỀ

NHẬN DẠNG ẢNH SỬ DỤNG THUẬT TOÁN Violes – Jones

Giáo viên hướng dẫn: Nguyễn Văn Hòa

Nhóm đăng kí: 14

Sinh Viên thực hiện:

Đặng Thị Phương Thanh -DTH185374

Phan Hoàng Trung -DTH185409

Trần Minh Trí -DTH185413

Phương Thái Ngọc -DTH185327

Long Xuyên, **Ngày 31 Tháng 05 Năm 2021**

Mục lục

LỜI CẢM ƠN

Được sự phân công của Giảng viên hướng dẫn về việc hoàn thành đồ án Trí Tuệ Nhân tạo. Với chủ đề tự chọn, nhóm chúng tôi đã nghiên cứu về phần mềm nhận diện khung mặt trên cở sở thư viện OpenCV.

Trước tiên nhóm chúng tôi xin được bày tỏ sự trân trọng và lòng biết ơn đối với thầy giáo giảng viên – Khoa Công nghệ thông tin. Trong suốt thời gian học và làm đồ án, thầy đã dành rất nhiều thời gian quý báu để tận tình chỉ bảo, hướng dẫn, định hướng cho em thực hiện đồ án.

Nhóm chúng tôi xin được cảm ơn các thầy cô giáo Trường Đại học An Giang đã giảng dạy trong quá trình học tập, thực hành, làm bài tập, giúp em hiểu thấu đáo hơn các nội dung học tập và những hạn chế cần khắc phục trong việc học tập, nghiên cứu và thực hiện bản đồ án này.

Mặc dù đã có nhiều cố gắng để thực hiện đồ án một cách hoàn chỉnh nhất. Song do chưa có kinh nghiệm quản lí thực tiễn và cách tiếp cận với môi trường bên ngoài còn hạn chế nên mặt kiến thức lẫn kinh nghiệm khó có thể tránh khỏi những thiếu sót nhất định mà bản thân chưa thấy được. Nhóm chúng tôi rất mong sự góp ý của Thầy để đồ án được chỉnh chu và hoàn thiện hơn.

Chân Thành Cảm Ơn.

Nhận xét của Giảng Viên

LỜI NÓI ĐẦU

1. Đặt vấn đề.

Công nghệ thông tin ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lí được các công việc một cách tự động, phục vụ cho lợi ích của con người. Trong những năm gần đây, một trong những bài toán nhận được nhiều sự quan tâm và tốn nhiều công sức nhất của lĩnh vực công nghệ thông tin, đó chính là bài toán nhận dạng. Tuy mới xuất hiện chưa lâu nhưng nó đã rất được quan tâm vì tính ứng dụng thực tế của bài toán cũng như sự phức tạp của nó.Bài toán nhận dạng có rất nhiều lĩnh vực như: nhận dạng vất chất(nước, lửa, đất, đá, gỗ..) nhận dạng chữ viết, nhận dạng giọng nói, nhận dang hình dáng, nhận dạng khuôn mặt.. trong đó phổ biến và được ứng dụng nhiều hơn cả là bài toán nhận diện khuôn mặt. Để nhận dạng được khuôn mặt, bước đầu tiên để nhận dạng là phát hiện ra khuôn mặt, điều này thực sự là quan trọng và hết sức khó khăn. Cho đến tận bây giờ, các nhà nghiên cứu vẫn chưa đạt được sự ưng ý trong việc giải quyết các khó khăn của bài toán và cho kết quả hoàn toàn đúng. Tuy nhiên, những gì đã đạt được cũng đủ để chúng ta áp dụng rộng rãi và đem lại những lợi ích to lớn trong cuộc sống.

2. Lí do chọn đề tài

Với sự hấp dẫn của bài toán và những thách thức còn đang ở phía trước, với niềm đam mê công nghệ hiện đại và những ứng dụng thực tế tuyệt với của nó, với khát khao khám phá và chinh phục những chi thức mới mẻ.. Nhóm chúng tôi đã chọn đề tài nghiên cứu: TÌM HIỂU VỀ PHẦN MỀM NHẬN DIỆN KHUÔN MẶT làm để tài nghiên cứu cho môn học.

3. Mục tiêu của đề án

- Phát hiện và đưa ra vùng khuôn mặt của ảnh từ camera.
- Trích xuất đặc trưng khuôn mặt từ vùng khuôn mặt.
- Phân loại các đặc trưng để nhận diện khuôn mặt.
- Xác định danh tính khuôn mặt và hiện thông tin.

Phần 1 : Giới thiệu khái quát lịch sử của AI nhận diện khuôn mặt

I. "Bình minh" của công nghệ Nhận diện khuôn mặt – những năm 1960

Những người tiên phong đầu tiên về công nghệ nhận diện khuôn mặt là Woody Bledsoe, Helen Chan Wolf và Charles Bisson. Năm 1964 và 1965, Bledsoe cùng với Wolf và Bisson bắt đầu sử dụng máy tính để nhận diện khuôn mặt con người. Do kinh phí của dự án được bắt nguồn từ một cơ quan tình báo giấu tên, phần lớn công việc đều không được công bố. Tuy nhiên, sau này người ta tiết lộ rằng hầu hết các công việc đều liên quan đến việc đánh dấu bằng tay các điểm khác nhau trên khuôn mặt như tâm mắt, miệng, v.v. Khoảng cách giữa các điểm mốc cũng được tự động tính toán và so sánh giữa các hình ảnh khác nhau để xác định danh tính.

II. Bắt đầu quá trình nâng cao độ chính xác của công nghệ – những năm 1970

Tiếp tục từ công việc ban đầu của Bledsoe, vào năm 1970, Goldstein, Harmon và Lesk đã mở rộng phạm vi của công việc bằng cách phân tích những đặc điểm cụ thể như màu tóc, độ dày của môi để phục vụ cho công việc nhận diện khuôn mặt.

III. Chương trình FERET – những năm 1990/2000

Cơ quan Dự án Nghiên cứu Tiên tiến Quốc phòng (DAPRA) và Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) đã triển khai chương trình Công nghệ Nhận diện Khuôn mặt (FERET) vào đầu những năm 1990 để khuyến khích thị trường của công nghệ này. Dự án liên quan đến việc tạo ra một cơ sở dữ liệu về hình ảnh khuôn mặt. Trong bộ thử nghiệm, có tới 2.413 ảnh tĩnh đại diện cho 856 người. Chương trình này

là bước ngoặt lớn cho công nghệ nhận diện khuôn mặt và những ngành nghề liên quan tới nó.

IV. Truyền thông xã hội – 2010 – Hiện tại

Trở lại những năm 2010, Facebook đã bắt đầu triển khai chức năng nhận diện khuôn mặt giúp xác định những người xuất hiện trong ảnh đăng Facebook. Tính năng này lập tức gây tranh cãi với các phương tiện truyền thông tin tức, làm dấy lên một loại các bài viết liên quan đến quyền riêng tư cá nhân. Tuy nhiên, người dùng Facebook nói chung dường như không bận tâm tới điều này và mỗi ngày, hơn 350 triệu ảnh được đăng tải và gắn thẻ mỗi ngày.

V. Iphone X – 2017

Công nghệ nhận diện khuôn mặt ngay sau đó đã được ứng dụng nhiều hơn vào các thiết bị thông minh trong cuộc sống. Cụ thể, công nghệ này đã được sử dụng thay cho khóa mở điện thoại. FaceID ngay lập tức trở thành từ khóa hot nhất, và là chìa khóa khiến Iphone X bán chạy hơn bao giờ hết.

VI. Tương lai của công nghệ nhận diện khuôn mặt

Bước vào năm 2021, công nghệ nhận diện khuôn mặt đang phát triển ở tốc độ chóng mặt và việc sử dụng công nghệ này ngày càng trở nên phổ biến hơn. Một số ngành nghề sử dụng công nghệ này bao gồm:

- Bán lẻ
- Khách sạn và Khu nghỉ dưỡng
- Máy ATM
- Quảng cáo kỹ thuật số
- An toàn xe buýt
- Các hãng hàng không
- Trải nghiệm khách hàng được cá nhân hóa
- Cửa hàng không cần nhân viên

Sử dụng công nghệ này một cách chính xác có thể tạo ra những khác biệt to lớn cho cuộc sống của chúng ta.

Phần 2 : Thư Viện OPENCV

1. OPENCV là gì ?

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần.

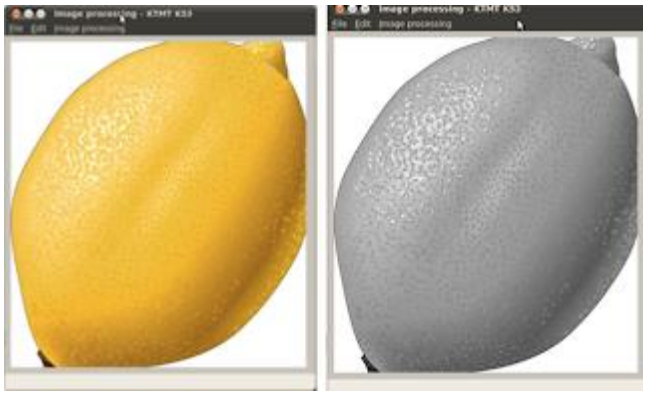
2. Ứng dụng.

Opencv có rất nhiều ứng dụng:

- Nhận dạng ảnh

Hình 1. Nhận dạng ảnh

- Xử lý ảnh



Hình 2. Xử lí ảnh, chuyển đổi màu

- Thực tế ảo

Hình 1. Ứng dụng trong kinh thực tế ảo

- Các ứng dụng khác

Hình 2. Nhận diện bảng số xe trong OpenCV

Hình 5. ứng dụng trong Robot

3. Cài đặt.

Cài đặt trên Windows: https://acodary.wordpress.com/2018/07/21/opencv-cai-dat-opencv-python-tren-windows/

Cài đặt trên Ubuntu: https://aicurious.io/posts/2018-09-14-cai-dat-opencv-tren-ubuntu-18-04/

Cài đặt OpenCV trên MacOS: https://viblo.asia/p/huong-dan-cai-dat-opencv-4-len-macos-phan-1-Qbq5Q323ZD8

4. Tổ chức thư viện OPENCV .

❖ **modules**: các modules chính của OpenCV

- **calib3d** Camera Calibration and 3D Reconstruction. Công dụng là tìm tọa độ trong không gian 3D của vật thể. Ứng dụng như dùng để hiệu chuẩn camera khi ống kính bị méo góc (fisheye), chống rung cho camera (calibrate). Tìm tọa độ 3D của bàn cờ vua trong không gian 3 chiều,…
- **core** Project chính của OpenCV vì cv::Mat được khai báo trong đây.
- **dnn** Deep neural network hay còn gọi là Deep Learning. Bản thân nó chỉ dùng để kết nối tới các framework khác như Caffe, TensorFlow, Torch, Darknet,… chứ không implement code.
- **features2d** Dùng để lấy các đặc trưng 2D của vật thể bằng các giải thuật như SIFT, FAST, KAZE, AKAZE,… Từ đó tìm ra các keypoint của vật thể để training nhận dạng.
- **flann** Tìm kiếm vật thể bằng cách so khớp các keypoint 2D của vật thể
- **highgui** Vẽ các GUI như cửa sổ cv::imshow(), trackbar.
- **imgcodecs** Đọc ảnh *.jpg, *.png, *.webp,… bằng cách gọi các project trong nhóm 3rdparty
- **imgproc** Chứa các hàm xử lý, biến đổi,… hình ảnh như dilate, erode, blur,…

- **js** Js chính là java script, dùng để nhúng vào website
- **ml** các thuật toán của Machine Learning như SVM, KNN,…
- **objdetect** Detect object dùng Cascade
- **photo** Xử lý ảnh chụp tương tự lightroom: chỉnh tone màu, HDR,…
- **shape** Tìm các hình cơ bản như tam giác, tứ giác, vuông,…
- **stitching** Ghép nhiều ảnh thành ảnh panorama
- **superres** Super Resolution: nâng cao resolution của ảnh bằng phương pháp nội suy
- **video** Xử lý video
- **videoio** Load file video
- **videostab** Video stabilization: sửa video quay bị rung
- **viz** Đồ họa 3D tương tự OpenGL

❖ **3rdparty:** các project bên thứ 3 viết ra được tổ chức OpenCV include vào

- **ippicv** Intell IPP, dùng để tăng tốc độ xử lý ảnh
- **ittnotify** thu thập data trong quá trình sử dụng, tìm hiểu thêm ITT
- **libjasper** định dạng hiếm gặp, không quan trọng
- **libjpeg-turbo** đọc ảnh jpeg, từ version 3.4 trở đi libjpeg thay bằng libjpeg-turbo **Bắt buộc phải có**
- **libpng** đọc ảnh png
- **libtiff** đọc ảnh tiff
- **libwebp** Webp là định dạng ảnh mới, nhẹ hơn jpeg, hay dùng trên website

- **openexr** định dạng ảnh EXR, ít khi sử dụng
- **protobuf** là 1 giao thức văn bản mới (như XML, JSON,…) được Google tạo ra, chỉ include khi sử dụng DNN
- **zlib** Nén và giải nén file

❖ **contrib:** các project do người dùng đóng góp

- **aruco** AR – thực tế tăng cường
- **bgsegm** Background segment: tách nền sử dụng thuật toán Background subtraction
- **bioinspired** làm tăng độ tương phản & xác định chuyển động trong 1 chuỗi hình ảnh liên tiếp nhau.
- **ccalib** Custom calibration: có thêm option từ calib3d
- **cnn_3dobj** Convolutional Neural Networks để phân lớp 3D object
- **cvv** debug hình ảnh trong các quá trình xử lý. Tuy nhiên cần có QT trong quá trình make
- **dnns_easily_fooled** Đây không phải là thư viện, đây chỉ là 1 chứng minh cho việc DNN (Deep Neural Network) dễ dàng bị đánh lừa bởi ảnh trừu tượng
- **dnn_objdetect** Phát hiện vật thể bằng Deep Neural Network
- **dpm** Deformable Part-based Models dùng để phát hiện vật thể
- **face** nhận diện khuôn mặt
- **freetype** vẽ text lên ảnh với font bất kỳ tùy chọn
- **fuzzy** phục hồi ảnh
- **hdf** Dùng để đọc ghi file HDF. File HDF có cấu trúc phức tạp, không giới hạn dung lượng, khả năng đọc ghi nhanh, thường dùng trong Big Data.

- **hfs** tách ảnh khỏi nền
- **img_hash** Dùng để kiểm tra ảnh có giống nhau hay không bất kể định dạng và kích thước. Giống như mắt người sẽ nhận dạng ảnh giống nhau bởi nội dung chứ không quan tâm đến giá trị nhị phân lưu trữ.
- **line_descriptor** tìm đường thẳng trong ảnh
- **matlab** sử dụng opencv với matlab
- **optflow** xác định hướng chuyển động của vật thể
- **ovis**
- **phase_unwrapping**
- **plot** vẽ biểu đồ
- **reg** Image registration
- **rgbd** Color data with depth processing sử dụng dữ liệu từ máy Kinect. Hình ảnh được truyền về kèm theo kênh Depth (ngoài RGB) để tính toán độ sâu của vật thể.
- **saliency** tìm vật thể nổi bật nhất trong ảnh
- **sfm** Tạo bản đồ 3D (hay 3D model) bằng ảnh chụp từ nhiều góc độ của vật thể
- **stereo** Xử lý ảnh stereo. Ảnh stereo là ảnh tạo ra sự khác biệt giữa mắt trái và mắt phải để cảm nhận được hình ảnh 3D bên trong. (Mình đã cố gắng thử nhưng không thể nào nhìn được 3D, không biết người khác thì sao)
- **structured_light** Scan vật thể 3D bằng cách tạo ra các vân ánh sáng, từ đó xác định được hình khối của vật thể cần scan.
- **surface_matching** Tìm kiếm vật thể 3D bằng dữ liệu từ các máy scan 3D trong môi trường thực tế. Giống như tìm vật thể 2D trong ảnh.

- **text** phát hiện vị trí text trong ảnh và đọc text
- **tracking** bám theo vật thể di chuyển
- **xfeatures2d** nâng cao của project feature2D
- **ximgproc** nâng cao của project imgproc
- **xobjdetect** nâng cao của project objdetect
- **xphoto** nâng cao của project photo

5. Hàm cho một số chức năng cụ thể .

- ***cv2.CascadeClassifier(cv2.data.haarcascades+ "haarcascade_frontalface_default.xml")***

  Bộ phân loại Haar Cascade của OpenCV hoạt động trên phương pháp tiếp cận *cửa sổ trượt* . Trong phương pháp này, một cửa sổ (kích thước mặc định 20 x 20 pixel) được trượt trên hình ảnh (từng hàng) để tìm các đặc điểm trên khuôn mặt. Sau mỗi lần lặp, hình ảnh được thu nhỏ (thay đổi kích thước) theo một yếu tố nhất định (được xác định bởi tham số ' scaleFactor '). Kết quả đầu ra của mỗi lần lặp được lưu trữ và thao tác trượt được lặp lại trên hình ảnh nhỏ hơn, đã thay đổi kích thước. Có thể có kết quả dương tính giả trong các lần lặp đầu tiên sẽ được thảo luận chi tiết hơn ở phần sau của bài viết này. Quá trình thu nhỏ và mở rộng cửa sổ này tiếp tục cho đến khi hình ảnh quá nhỏ so với cửa sổ trượt. Giá trị của scaleFactor càng nhỏ thì độ chính xác càng lớn và chi phí tính toán càng cao.

- ***cv2.VideoCapture(0)***: để chụp 1 video cap.read() có chức năng trả về giá trị bool(Đúng/ sai), nếu khung đọc đúng trả về 'True' và tập dữ liệu , đọc sai trả về False

- ***cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)***: có chức năng xử lí hình ảnh, chuyển đổi từ kiểu màu RGB chuyển sang kiểu màu Gray để máy có thể đọc được.

- ***face_cascade.detectMultiScale(gray,1.3,5)***

Truyền vào 3 tham số: hình ảnh được chuyển sang kiểu màu Gray, tham số scaleFactor, tham số MinMultiScale.

Hình ảnh kiểu màu Gray: giá trị các thông số màu của ảnh khuôn mặt trả về kiểu ma trận mãng 2 chiều.

Tham số ScaleFactor: tính toán và điều độ so sánh chi tiết của các ảnh khi hình ảnh truyền vào có các kính thước khác nhau (nếu thông số = 1 có nghĩa là hình ảnh nguyên bản ban đầu, thông số càng tăng thuật toán sẽ giảm độ chính xác).

Tham số MinMultiScale: Tham số này sẽ ảnh hưởng đến chất lượng của các khuôn mặt được phát hiện. Giá trị cao hơn dẫn đến ít phát hiện hơn nhưng với chất lượng cao hơn. 3 ~ 6 là một giá trị tốt cho nó.

- ***cv2.rectangle(frame, (x,y), (x+w, y+h), (0,0,255), 2)***:

  tạo ra khung hình tứ giác và các thông số truyền vào bao gồm: hình ảnh dạng RGB, toạ độ ban đầu, toạ độ được tịnh tiến(Kích thước khung ảnh khuôn mặt), màu sắc viền khung, độ dày viền khung

- ***cv2.imwrite('dataSetimg/img.'+str(id)+'.'+str(index)+'.jpg',gray[y: y+h,x: x+w]:*** tạo đường dẫn địa chỉ vào trong File.

- ***cv2.imshow("HienAnh",frame):*** hiển thị khung camera

- ***cv2.waitKey(1):*** điều kiện tắt camera

- ***cv2.destroyAllWindows():*** đóng camera

- ***cv2.face.LBPHFaceRecognizer_create()***

- ***cv2.FONT_HERSHEY_SIMPLEX:*** một loại font chữ.

- ***cv2.putText(frame, ""+str(profile[1]), (x+10 ,y+h+30),fontface,1,(0,255,0),2):*** được sử dụng để vẽ một chuỗi văn bản lên trên ảnh.

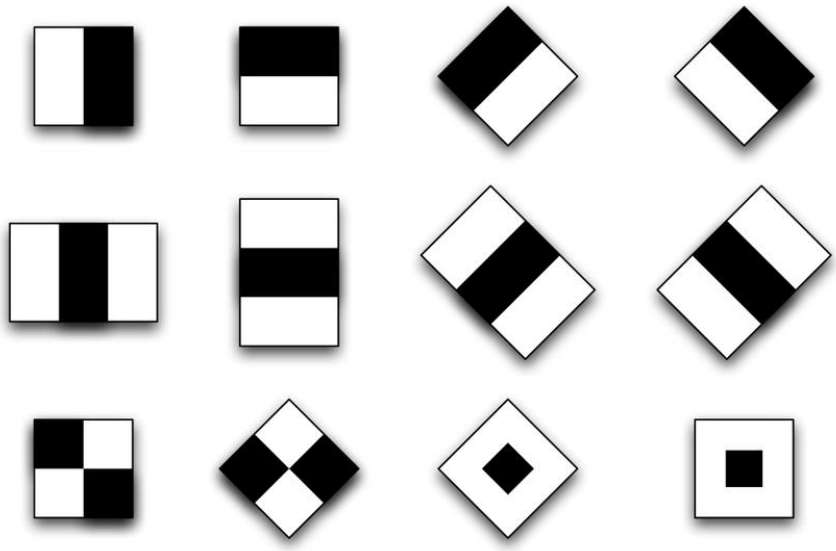
Phần 4 : Thuật toán Viola – Jones

1. Giới thiệu thuật toán Viola -Jones

- Thuật toán Viola-Jones lần đầu tiên được công bố nghiên cứu vào năm 2001 bởi Paul Viola và Michael Jones, thuật toán có khả năng phát hiện các vật thể trong hình ảnh, bất kể vị trí và tỷ lệ của chúng trong một hình ảnh. Hơn nữa, thuật toán này có thể chạy trong thời gian thực, giúp phát hiện các đối tượng trong video stream.
- Cụ thể, Viola và Jones tập trung vào việc phát hiện khuôn mặt trong ảnh, nhưng thuật toán này cũng có thể được sử dụng để huấn luyện máy dò tìm các vật thể tùy ý, như xe hơi, tòa nhà, dụng cụ nhà bếp và thậm chí là một trái chuối….
- Mặc dù khung Viola-Jones chắc chắn đã mở ra cánh cửa để phát hiện đối tượng, nhưng giờ đây nó đã vượt xa các phương pháp khác, chẳng hạn như sử dụng Histogram of Oriented Gradients (HOG) + Linear SVM và Deep Learning

2. Phân tích thành phần chính .

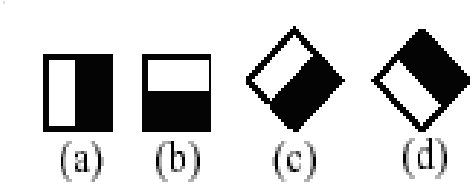
➢ *Đặc trưng Haar – Like*



Hình 6. Các Haar-Like

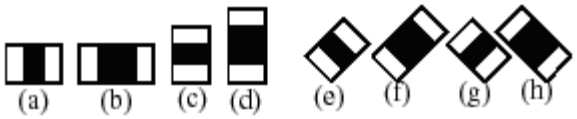
Để sử dụng các đặc trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-Like cơ bản được mở rộng ra và được chia làm 3 tập đặc trưng như sau:

Đặc trưng cạnh(edge feature):



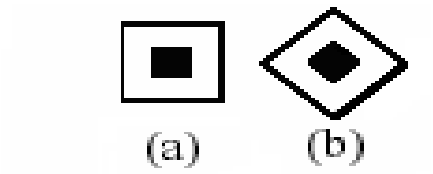
Hình 7. Đặc trưng cạnh

Đặc trưng đường(line feature):



Hình 8. Đặc trưng đường

Đặc trưng xung quanh tâm(center-surround features):

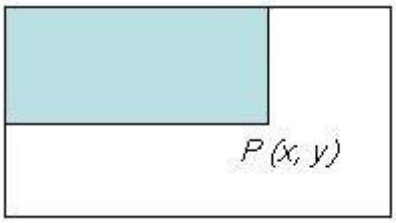


Hình 9. Đặc trưng xung quanh tâm

Dùng các đặc trưng trên, ta có thể tính được các giá trị của đặc trưng Haar-Like là sự chênh lệch giữa tổng của các pixel của vùng đen và vùng trắng như trong công thức sau:

$$f(x) = \text{Tổng}_{\text{vùng đen}}(\text{các mức xám của pixel}) - \text{Tổng}_{\text{vùng trắng}}(\text{các mức xám của pixel})$$

Viola và Joines đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó.

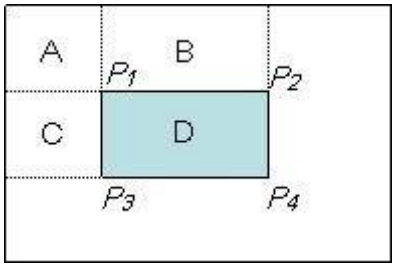


$$P(x,y) = \sum_{x' \le x, y' \le y} i(x', y')$$

Hình 10. Công thức tính Integral Image

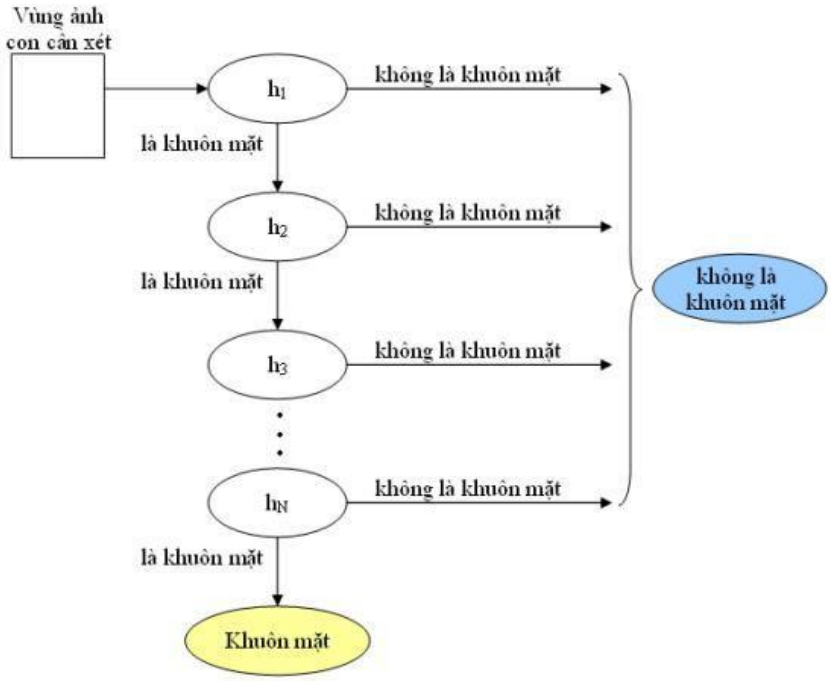
Với A + B + C + D chính là giá trị tại điểm P4 trên Integral Image, tương tự như vậy A+B là giá trị tại điểm P2, A+C là giá trị tại điểm P3, và A là giá trị tại điểm P1. Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

$$D = \underbrace{(x4,y4)}_{A+B+C+D} - \underbrace{(x2,y2)}_{(A+B)} - \underbrace{(x3,y3)}_{(A+C)} + \underbrace{(x1,y1)}_{A}$$



Hình 11. Công thức tính Integral Image

➢ *AdaBoost*



Hình 12. Mô hình phân tầng trong AdaBoost

- Trong đó, h(k) là các bộ phân loại yếu, được biểu diễn như sau:

$$h_k = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ 0 & \text{ngược lại} \end{cases}$$

Hình 13. Công thức tính xử lí trong AdaBoost

x: cửa sổ con cần xét

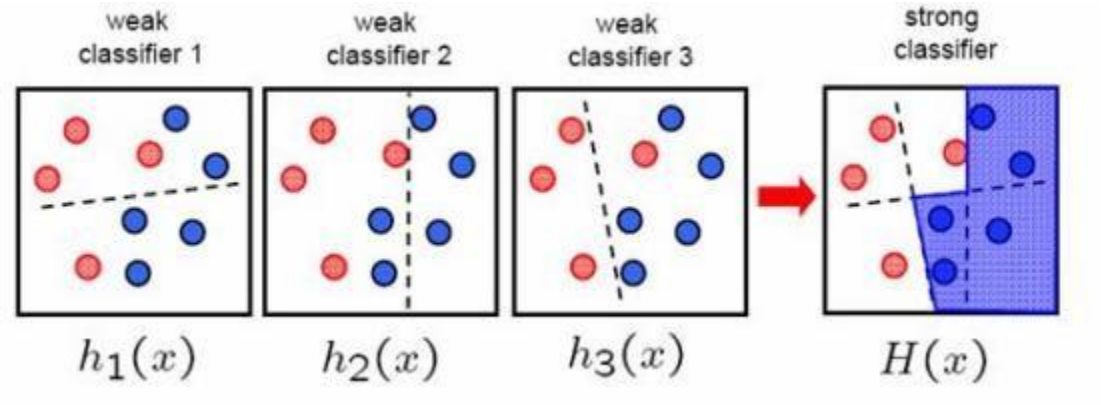
θk : ngưỡng

f_k: giá trị của đặc trưng Haar-like

p_k: hệ số quyết định chiều của phương trình

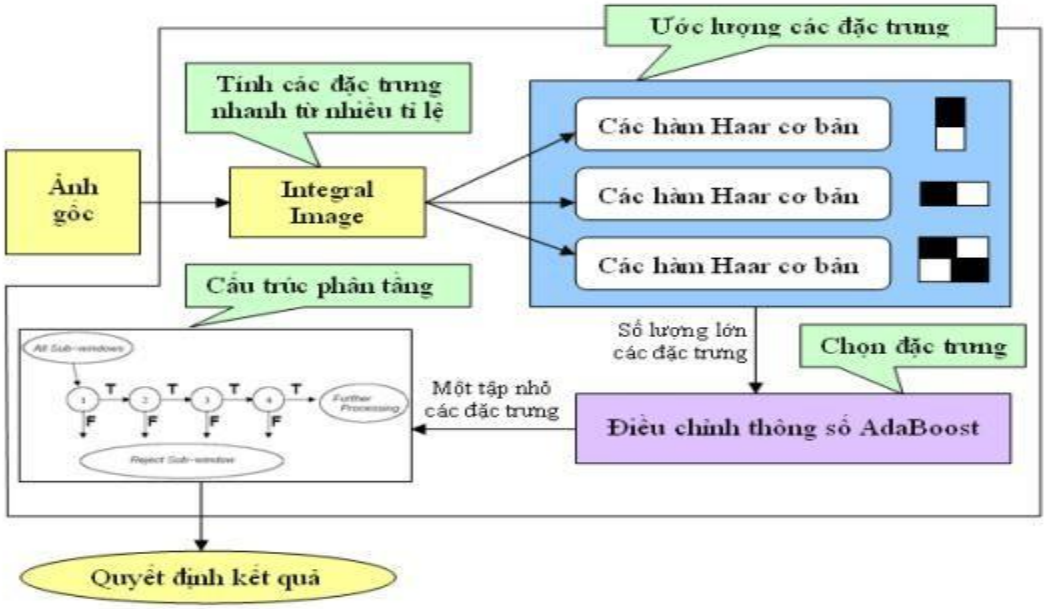
Hình 14. Chú thích công thức tính trong AdaBoost

- Đây là hình ảnh minh họa việc kết hợp các bộ phân loại yếu thành bộ phân loại mạnh



Hình 15. Hình ảnh minh hoạ phân loại vùng

➢ *Sơ đồ Nhận diện khuôn mặt*



Hình 16. Sơ đồ xử lí ảnh trong thuật toán

Phần 5 : Mô phỏng chương trình

1. Công cụ sử dụng

Ngôn ngữ lập trình Python (khuyến khích python 3.x)
Thư viện opencv version: 4.5.1.48
Thư Viện opencv-contrib-pyhon:4.5.1.48
Thư viện Pillow = 8.1.0
Thư viện Numpy: 1.19.5
Công cụ Visual Studio Code
SqlLite3

2. Các biến, hàm trong chương trình

- **Cv2.imread():** phương thức tải một ảnh từ một tệp chỉ định, nếu không thể đọc thì nó trả về một ma trận rỗng
  Vd: image = cv2.imread("image2.jpg",cv2.IMREAD_GRAYSCALE)
- **Cv2.imshow():** hiển thị hình ảnh ra của sổ. Cửa số tự động phù hợp với kích thước của hình ảnh.
  Vd: cv2.imshow("ảnh xám",image)
- **Cv2.release():** giải phóng webcam
- **Cv2.destroyAllWindows() :** đóng tắt cả các cửa sỗ windows
- **Cv2.VideoCapture() :** lấy đối tượng quay video cho máy ảnh.
- **cvWaitKey(x):**Chờ bấm phím bất kỳ để kết thúc chương trình. Có thể truyền khoảng thời gian vào thay vì nhấn phím.(x tính bằng mili)
- **cv2.rectangle(image, start_point, end_point, color, thickness):** được sử dụng để vẽ một hình chữ nhật trên bất kỳ hình ảnh nào.
  Vd: cv2.rectangle("image2.jpg", (5,5), (200,200), (0,255,0), 2)
- **cv2.putText(image, text, org, font, fontScale, color, thickness):** được sử dụng để vẽ một chuỗi văn bản lên trên ảnh.
  Vd: cv2.putText(frame, "Ảnh",
  cv2.FONT_HERSHEY_SIMPLEX,(50,50),1,(255,0,0),2)
- **cv2.CascadeClassifier(cv2.data.haarcascades+ "haarcascade_frontalface_default.xml")**
- **cv2.VideoCapture(0):** chụp ảnh trong webcam
- **cv2.cvtColor():** được sử dụng để chuyển đổi một hình ảnh từ không gian màu này sang không gian màu khác
  Cú pháp: cv2.cvtColor(src,code)
  Các thông số:
  Src: là hình ảnh có có không gian màu cần thay đổi
  Code: mã chuyển đổi không gian màu
  (cv2.COLOR_RGB2GRAY,cv2.RGB2HSV…)
- **face_cascade.detectMultiScale(gray,1.1,5)**
  grayImage: Ảnh Xám

scaleFactor: chỉ số thu nhỏ hình ảnh sau mõi lần trượt của cửa sổ trượt trong bộ phân lại haar cascade để tìm những đặc điểm trên khuôn mặt

minNeighbors: là số lần tối thiểu một vùng phải được xác định như một khuôn mặt.

- **cv2.rectangle(frame, (x,y), (x+w, y+h), (0,0,255), 2)**
- **cv2.imshow("HienAnh",frame)**
- **cv2.destroyAllWindows() :** đóng tất các của số.
- **cv2.FONT_HERSHEY_SIMPLEX** :một loại font chữ.

OS

- **os.path.join(path, f):** có chức năng là nối các đường dẫn
- **os.path.exists() :** phương pháp trong python được sử dụng để kiểm tra xem đường dẫn chỉ định có được tồn tại hay không.
- **os.makedirs('recognizer'): đc sử dụng để tạo ra một thư mục** theo các đệ quy.

Image

- **faceImg = Image.open(imagePath).convert('L'):** chưa tìm hiểu đc

Numpy:

- **Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của** Python**. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng "code Python" đơn thuần.**
  - Cài đặt: pip install numpy
  - Nó có thể thao tác với mảng:

+ Khởi tạo mảng một chiều(2 chiều, 3 chiều)

+ Khởi tạo với các hàm có sẵn: ví dụ như zeros,ones,arange, full,eye,random.random..

+ truy cập vào từng phần tử của mảng (các phần tử của mảng được đánh số từ 0).

+ tính toán thống kê với các hàm có sẵn

SqlLite3

- **Sqlite3.connect():** phương thức dùng để kết nối xuống cơ sở dữ liệu.
- **Execute()** thực thi chuỗi truy vấn trong sql (như insert, update, delete…)
- **Connection.commit():** có tác dụng lưu lại những thay đổi khi đã thực thi.
- **Connection.close()** đóng kết nối.

Phần 6 : Kết luận

Cụ thể, Viola và Jones tập trung vào việc phát hiện khuôn mặt trong ảnh, nhưng thuật toán này cũng có thể được sử dụng để huấn luyện máy dò tìm các vật thể tùy ý, như xe hơi, tòa nhà, dụng cụ nhà bếp và thậm chí là một trái chuối.

Mặc dù khung Viola-Jones chắc chắn đã mở ra cánh cửa để phát hiện đối tượng, nhưng giờ đây nó đã vượt xa các phương pháp khác, chẳng hạn như sử dụng Histogram of Oriented Gradients (HOG) + Linear SVM và Deep Learning Nhưng mình vẫn nghĩ rằng điều quan trọng là chúng ta rất quan tâm đến thuật toán này và ít nhất là có sự hiểu biết ở mức độ cao về những gì mà diễn ra bên dưới.

Ngày nay thuật toán nhận diện khuôn mặt vẫn đang luôn tìm kiếm và khắc phục những hạn chế về khi người dùng xoay gốc nghiên thì thuật toán không thể nhận diện được, cùng với việc khi đeo kiếng sẽ giảm đi độ chính xác khi nhận diện.

Phần 7 : Tài liệu tham khảo

- **OpenCV ứng dụng trong nhận diện khuôn mặt:**

https://www.stdio.vn/computer-vision/opencv-voi-python-trong-ung-dung-phat-hien-khuon-mat-trong-buc-anh-dYG31n1

https://ichi.pro/vi/nhan-dien-khuon-mat-trong-10-dong-cho-nguoi-moi-bat-dau-25871442088231

https://stackoverflow.com/questions/22249579/opencv-detectmultiscale-minneighbors-parameter

- **Tìm hiểu RGB chuyển sang thang màu xám (Gray):**

https://qastack.vn/programming/12201577/how-can-i-convert-an-rgb-image-into-grayscale-in-python

- **Tìm hiểu về RGB và Alpha:**

https://vi.phhsnews.com/articles/howto/rgb-cmyk-alpha-what-are-image-channels-and-what-do-they-mean.html

- **Tìm hiểu về thị giác máy tính:**

https://thigiacmaytinh.com/cau-truc-opencv/

- **Tìm hiểu về OpenCV:**

https://topdev.vn/blog/opencv-la-gi-hoc-computer-vision-khong-kho/

https://www.geeksforgeeks.org/

- **Youtube:**

https://www.youtube.com/watch?v=Lusa912ax5g&t=15s

https://www.youtube.com/watch?v=kAH9Bih32WE&ab_channel=NewDevNewDev

https://www.youtube.com/watch?v=kgO4FL1t_9w&ab_channel=NewDevNewDev