|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**  Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png  **ĐỒ ÁN CHUYÊN NGÀNH**  **NHẬN DIỆN KHUÔN MẶT QUA KHUNG ẢNH TRONG VIDEO VÀ THỜI GIAN THỰC**  **Giảng viên hướng dẫn: TS.HÀ MINH TÂN**  **Sinh viên thực hiện: NGÔ TẤN LỢI**  **MSSV: 2000006297**  **Chuyên ngành: TRÍ TUỆ NHÂN TẠO**  **Môn học: ĐỒ ÁN CHUYÊN NGÀNH AI**  **Khóa: 2020**  **Tp.HCM, tháng 9 năm 2023** |

|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**  **KHOA CÔNG NGHỆ THÔNG TIN**  Description: C:\Documents and Settings\Administrator\Desktop\logo dai hoc_khong nen.png  **ĐỒ ÁN CHUYÊN NGÀNH**  **NHẬN DIỆN KHUÔN MẶT QUA KHUNG ẢNH TRONG VIDEO VÀ THỜI GIAN THỰC**  **Giảng viên hướng dẫn: TS. HÀ MINH TÂN**  **Nhóm sinh viên thực hiện:**  **NGUYỄN VIỆT CƯỜNG MSSV** : 2000005818  **NGÔ TẤN LỢI MSSV** : 2000006297  **LÊ DIÊN THANH TÙNG MSSV :** 2000006184  **Chuyên ngành: TRÍ TUỆ NHÂN TẠO**  **Môn học: ĐỒ ÁN CHUYÊN NGÀNH AI**  **Khóa: 2020**  **Tp.HCM, tháng 9 năm 2023** |

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ: 3 NĂM HỌC: 2022 - 2023** |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

BM-ChT-11

Môn thi: Đồ án chuyên ngành AI Lớp học phần: 20DTH1B

Nhóm sinh viên thực hiện :

1.Nguyễn Việt Cường Tham gia đóng góp: 10/10

2.Ngô Tấn Lợi Tham gia đóng góp: 10/10

3.Lê Diên Thanh Tùng Tham gia đóng góp: 10/10

Ngày thi:19/09/2023 Phòng thi:L.510

Đề tài tiểu luận/báo cáo của sinh viên :

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  |  |  |
| Nội dung |  |  |  |
| * Các nội dung thành phần |  |  |  |
| * Lập luận |  |  |  |
| * Kết luận |  |  |  |
| Trình bày |  |  |  |
| **TỔNG ĐIỂM** |  |  |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

**LỜI MỞ ĐẦU**

Trải qua nhiều thập kỷ, ngành Công nghiệp Công nghệ Thông tin (IT) đã trở thành một lĩnh vực vô cùng phát triển và đóng vai trò quan trọng trong xã hội hiện đại. Đặc biệt, trong thời đại ngày nay, nền tảng vững mạnh của IT đã mở ra không gian không giới hạn cho các đổi mới, đóng góp tích cực vào sự phát triển toàn cầu.

Chuyên ngành Trí tuệ Nhân tạo (AI) - một nhánh nổi bật của IT - đang là trung tâm của sự quan tâm và nghiên cứu rộng rãi. Trong lĩnh vực này, Face Detection - hay nhận diện khuôn mặt - nổi lên như một trong những phân khúc quan trọng và tiềm năng nhất.

Lợi ích của nghiên cứu về Face Detection không chỉ dừng lại ở việc đơn thuần nhận diện khuôn mặt. Điều này mở ra cánh cửa cho một loạt ứng dụng sáng tạo trong nhiều lĩnh vực khác nhau. Hãy cùng nhau khám phá những tiềm năng và lợi ích mà việc nghiên cứu về Face Detection mang lại cho xã hội và ngành công nghiệp AI.

**LỜI CẢM ƠN**

Trước khi học kỳ khép lại, em muốn dành chút thời gian để bày tỏ lòng biết ơn chân thành đối với sự tận tâm và sự hướng dẫn của thầy Hà Minh Tân trong suốt khóa học. Sự nhiệt tình của thầy trong việc giảng dạy cùng với niềm đam mê về nội dung môn học thực sự để lại ấn tượng sâu sắc trên hành trình học tập của em .

Sự hướng dẫn của thầy đã có giá trị vô cùng hữu ích, và em thực sự biết ơn sự kiên nhẫn và sẵn sàng giải quyết bất kỳ câu hỏi hoặc vấn đề nào em có trong suốt khóa học. Sự ủng hộ của thầy đã đóng vai trò quan trọng trong sự thành công học tập của em , và em thực sự biết ơn các lời động viên này của thầy. Khi em tiến xa trên hành trình học tập và đi làm sau này, em sẽ mang những bài học em đã học từ thầy và vận dụng nó.

Đồ án chuyên ngành trí tuệ nhân tạo vô cùng bổ ích và có tính thực tế cao, đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ.

Mặc dù em đã cố gắng hết sức nhưng chắc chắn bài tiểu luận khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong các quý thầy cô xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

Em xin chân thành cảm ơn !

**NHẬN XÉT CỦA GIẢNG VIÊN GIẢNG DẠY**

*Tp.HCM, Ngày . . . . tháng . . . . năm . . . .*

**Giảng viên giảng dạy**

(Ký tên và ghi rõ họ tên)

MỤC LỤC

[CHƯƠNG I: GIỚI THIỆU 1](#_Toc145804696)

[1. Supervised learning: 1](#_Toc145804697)

[1.1. Supervised learning là gì? 1](#_Toc145804698)

[1.2. Ứng dụng của Supervised learning: 2](#_Toc145804699)

[1.3. Giới thiệu về đề tài : 3](#_Toc145804700)

[1.3.1. Lý do chọn đề tài: 3](#_Toc145804701)

[1.3.2. Mục đích: 4](#_Toc145804702)

[CHƯƠNG II : CÁC BÀI NGHIÊN CỨU LIÊN QUAN 5](#_Toc145804703)

[CHƯƠNG III : CƠ SỞ LÝ THUYẾT 8](#_Toc145804704)

[2. Deep Learning 8](#_Toc145804705)

[2.1. Deep Learning là gì? 8](#_Toc145804706)

[2.2. Deep Learning thuật toán 8](#_Toc145804707)

[2.3. Các bài toán của DL trong Computer Vision 12](#_Toc145804708)

[3. Computer Vision? 14](#_Toc145804709)

[4. OpenCV 15](#_Toc145804710)

[5. Face Detection ? 16](#_Toc145804711)

[CHƯƠNG IV. XÂY DỰNG MÔ HÌNH 17](#_Toc145804712)

[CHƯƠNG V . KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 24](#_Toc145804713)

[1. Kết luận 24](#_Toc145804714)

[2. Hướng phát triển 24](#_Toc145804715)

[TÀI LIỆU THAM KHẢO 25](#_Toc145804716)

**DANH MỤC CÁC BẢNG HÌNH**

[Hình 1 : Supervised learning 1](#_Toc145805220)

[Hình 2 : Các thuật toán của Deeplearning 8](#_Toc145805221)

[Hình 3 : Mô tả mô hình CNN 9](#_Toc145805222)

[Hình 4 : Mô tả mô hình RNN 10](#_Toc145805223)

[Hình 5: Mô hình Transformer 11](#_Toc145805224)

[Hình 6 : So sánh mô hình Transformer với LSTM 12](#_Toc145805225)

[Hình 7 : Mô tả phân loại hình ảnh 13](#_Toc145805226)

[Hình 8 : Minh họa cho nhận diện đối tượng 14](#_Toc145805227)

[Hình 9 : Quy trình thực hiện 17](#_Toc145805228)

[Hình 10 : kết quả nhận diện khuôn mặt bằng webcam 21](#_Toc145805229)

[Hình 11 : kết quả nhận diện khuôn mặt qua file .mp4 22](#_Toc145805230)

**DANH MỤC CÁC TỪ VIẾT TẮT**

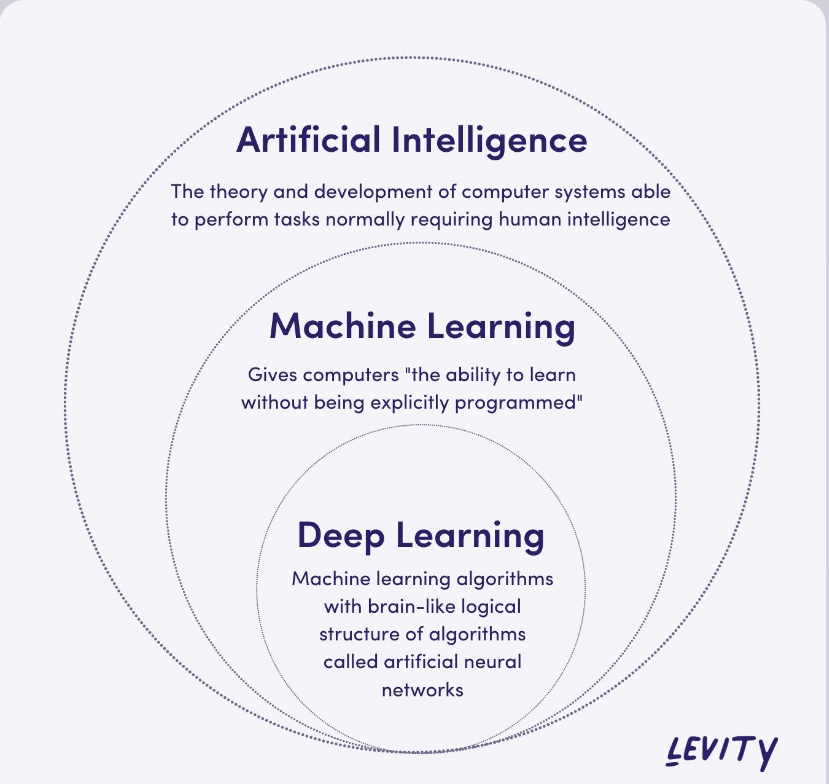
|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Nghĩa của từ |
| 1 | AI | Artificial Inteligence |
| 2 | SVM | Support Vector Machine |
| 3 | NLP | Nature language Processing |
| 4 | CT scan | computerized tomography scan |
| 5 | CNN | Convolutional Neural Network |
| 6 | LSTM | Long Short Term Memory networks |
| 7 | BERT | Bidirectional Encoder Representations from Transformers |
| 8 | GPT | Generative Pre-trained Transformer |
| 9 | DL | Deep Learning |
| 10 | RNN | Recurrent Neural Network |
| 11 | GAN | Generative Adversarial Networks |
| 12 | GNN | Graph Neural Network |
| 13 | Gru | Gated recurrent units |

# CHƯƠNG I: GIỚI THIỆU

## Supervised learning:

### Supervised learning là gì?

* Supervised learning (học có giám sát) là một loại phương pháp trong lĩnh vực máy học, trong đó một mô hình máy tính được đào tạo để tìm hiểu một mối quan hệ giữa đầu vào (input) và đầu ra (output) từ các cặp dữ liệu đã được gán nhãn trước.
* Quá trình này giống như việc giáo dục mô hình bằng cách cung cấp cho nó ví dụ cụ thể về cách mối quan hệ hoạt động. Mục tiêu của quá trình học là để mô hình có thể tổng quát hóa từ các ví dụ đã được đào tạo và sau đó thực hiện dự đoán chính xác cho các dữ liệu mới mà nó chưa từng thấy.
* Các thuật toán phổ biến trong học có giám sát bao gồm Support Vector Machines (SVM), Decision Trees, Random Forests, Neural Networks, và nhiều thuật toán khác.



Hình : Supervised learning

### Ứng dụng của Supervised learning:

* Supervised learning có rất nhiều ứng dụng trong nhiều lĩnh vực khác nhau do khả năng của nó trong việc học từ dữ liệu đã được gán nhãn. Dưới đây là một số ví dụ về ứng dụng của supervised learning:
  + Phân loại: Phân loại là một ứng dụng quan trọng của supervised learning. Ví dụ, phân loại email là spam hoặc không spam, phân loại ảnh là loại động vật, dự đoán bệnh lý dựa trên dữ liệu y tế, phân loại tin tức vào các chủ đề khác nhau.
  + Dự đoán và dự báo: Supervised learning được sử dụng để dự đoán giá cổ phiếu, dự báo thời tiết, dự đoán lưu lượng giao thông, dự đoán nhu cầu sản phẩm trong tương lai dựa trên dữ liệu lịch sử.
  + Nhận dạng văn bản và ngôn ngữ: Phân loại văn bản theo chủ đề, nhận dạng ngôn ngữ tự nhiên (NLP), dịch máy, phân loại cảm xúc từ văn bản (tích cực, tiêu cực, trung lập) là một số ví dụ trong lĩnh vực này.
  + Xử lý ảnh và thị giác máy tính: Nhận dạng khuôn mặt, phát hiện đối tượng trong hình ảnh, phân loại ảnh theo nội dung, và xử lý ảnh y tế là một số ví dụ.
  + Âm nhạc và âm thanh: Phân loại thể loại âm nhạc, dự đoán sở thích người nghe, nhận dạng từ ngữ trong âm thanh là những ứng dụng có thể sử dụng supervised learning.
  + Tư vấn và khuyến nghị: Xây dựng hệ thống gợi ý sản phẩm cho khách hàng dựa trên hành vi mua sắm trước đây, khuyến nghị nội dung cho người dùng trên các nền tảng truyền thông xã hội cũng là những ứng dụng thường thấy.
  + Y tế: Dự đoán chuẩn đoán bệnh dựa trên dữ liệu y tế, phân loại ảnh chụp cắt lớp (CT scan) để phát hiện bất thường, dự đoán tác động của các loại thuốc dựa trên tình trạng bệnh là những ứng dụng quan trọng trong lĩnh vực y tế.
  + Tài chính và ngân hàng: Dự đoán rủi ro tín dụng, phát hiện gian lận trong giao dịch tài chính, dự đoán thị trường tài chính là những ứng dụng trong lĩnh vực này.

### Giới thiệu về đề tài :

#### Lý do chọn đề tài:

Trong thời đại công nghệ ngày nay, Face Detection (Nhận diện khuôn mặt) đóng vai trò không thể thiếu trong nhiều lĩnh vực khác nhau của cuộc sống hàng ngày. Từ các ứng dụng di động cho đến hệ thống an ninh, công nghệ này đã trở thành một phần quan trọng của xã hội hiện đại. Nhiều điện thoại di động và máy tính bảng sử dụng Face Detection để mở khóa thiết bị cho người dùng. Thay vì phải nhập mật khẩu hoặc mã PIN, họ chỉ cần nhìn vào camera. Các dịch vụ xã hội và ứng dụng quản lý ảnh sử dụng Face Detection để nhận diện và gắn nhãn các khuôn mặt trong ảnh. Điều này giúp dễ dàng tìm kiếm và quản lý các bức ảnh.

**Lý do:**

* **Ứng dụng rộng rãi**: Kỹ thuật nhận diện khuôn mặt có nhiều ứng dụng trong thực tế, từ an ninh, nhận diện và phân tích người dùng, đến nhận diện biểu cảm và nhiều lĩnh vực khác.
* **Thú vị và thách thức**: Nhận diện khuôn mặt trong video đòi hỏi kỹ thuật cao, và đây là một thách thức thú vị cho nghiên cứu và phát triển.
* **Phát triển kỹ năng**: Nghiên cứu về nhận diện khuôn mặt trong video đồng nghĩa với việc áp dụng và phát triển các kỹ năng trong lĩnh vực thị giác máy tính và học sâu.

#### Mục đích:

Mục đích chính của đề tài này có thể là cải thiện khả năng nhận diện khuôn mặt trong các đoạn video. Điều này có thể hữu ích trong nhiều tình huống thực tế như giám sát an ninh hoặc phân tích dữ liệu video lớn. Nghiên cứu về nhận diện khuôn mặt trong video có thể đóng góp vào sự phát triển và cập nhật các kỹ thuật và công nghệ liên quan đến thị giác máy tính và trí tuệ nhân tạo. Kết quả của việc nghiên cứu có thể được ứng dụng trong các lĩnh vực như an ninh, giải trí, quảng cáo và nhiều ứng dụng khác.

# CHƯƠNG II : CÁC BÀI NGHIÊN CỨU LIÊN QUAN

**Bài báo 1**

Tiêu đề : Deep Learning Face Detection and Recognition

Tác giả : Samar S. Mohamed, Wael A. Mohamed, A. T. Khalil, and A. S. Mohra

Tạp chí : International Journal of Electronics and Telecommunications

**Hướng nghiên cứu :**

Bài báo tập trung vào việc nghiên cứu và phát triển một phương pháp nhận diện khuôn mặt dựa trên mạng nơ-ron tích chập (CNN). Mục tiêu của nghiên cứu là cải thiện độ chính xác của hệ thống nhận diện khuôn mặt 2D bằng cách sử dụng mạng CNN với 15 tầng để học biểu diễn phân biệt.

**Kết quả đạt được:**

So sánh tỷ lệ the ratio of train ở mức 1000 bức ảnh

|  |  |
| --- | --- |
| The ratio of train to test | Accuracy |
| (50% to50%) | 98.60% |
| (55% to45%) | 98.89% |
| (65% to35%) | 99.43% |
| (70% to30%) | 99.67% |

Công việc liên quan trong nhận dạng khuôn mặt

|  |  |
| --- | --- |
| Related work | Accuracy |
| PCA + Euclidian Distance. (Riddhi A. Vyas 2016) | 93.33% |
| LBP + LTP. (Chi Kien Tran, Tsair Fwu Lee 2014) | 98.75% |
| DeepFace. [Taigman et al., 2014] | 97.53% |
| DeepID. [Yi Sun 2014] | 97.45% |
| DeepID2. [Yi Sun, Xiaoou Tang 2014] | 99.15% |
| DeepID3. [Yi Sun, Xiaoou Tang 2015] | 99.53% |
| Proposed method | 99.67% |

So sánh giữa phương pháp đề xuất và phương pháp thông thường trên cơ sở dữ liệu orl

|  |  |
| --- | --- |
| Proposed method | Conventional method |
| CNN Accuracy 97.89% | Fuzzy Hidden Markov Models (FHMM) classifier accuracy 95% |

**Bài báo 2**

Tiêu đề : Face Detection

Tác giả : Gary Chern, Paul Gurney, and Jared Starman

Bài báo trên web.stanford.edu

Hướng nghiên cứu :

Phát triển một thuật toán có khả năng định vị từng khuôn mặt trong ảnh màu của

lớp. Chúng tôi được cung cấp bảy hình ảnh đào tạo cùng với dữ liệu thực tế cơ bản tương ứng để phát triển và huấn luyện các thuật toán của chúng tôi. Kết quả cuối cùng của nhóm chúng tôi là một thuật toán có khả năng tìm thấy hơn 95% các khuôn mặt trong tất cả ngoại trừ một hình ảnh trong khoảng 30 giây. Ngoài ra, chúng tôi có thể xác định thành công một trong những con cái trong hai hình ảnh thử nghiệm.

**Kết quả đạt được**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Image # | #Faces Detected | #Faces in Image | Percentage Correct | # Repeated Faces and False Positives | Bonus |
| 1 | 20 | 21 | 95% | 0 | 1 |
| 2 | 23 | 24 | 96% | 0 | 1 |
| 3 | 25 | 25 | 100% | 0 | 0 |
| 4 | 23 | 24 | 96% | 0 | 0 |
| 5 | 21 | 24 | 88% | 0 | 0 |
| 6 | 23 | 24 | 96% | 0 | 0 |
| 7 | 22 | 22 | 100% | 0 | 0 |

Như có thể thấy từ bảng, thuật toán của chúng tôi hoạt động khá tốt. Hai hình ảnh có khả năng nhận diện khuôn mặt 100% tỷ lệ, trong khi tất cả trừ một hình ảnh có tỷ lệ nhận diện khuôn mặt trên 95%. Cũng cần lưu ý là chúng tôi không có khuôn mặt lặp lại hoặc kết quả dương tính giả trong bất kỳ hình ảnh huấn luyện nào. Hơn nữa, chúng tôi đã có thể điều chỉnh chính xác xác định một phụ nữ trong hai hình ảnh thử nghiệm.

# CHƯƠNG III : CƠ SỞ LÝ THUYẾT

## ****Deep Learning****

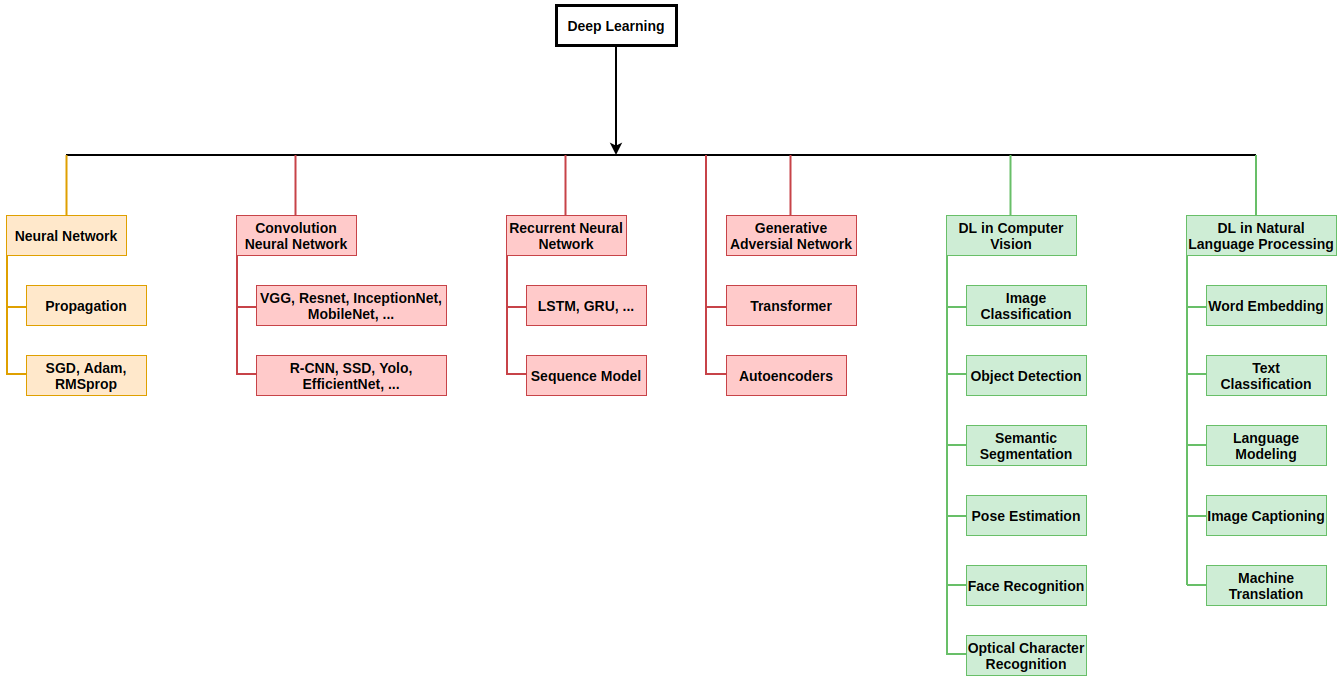
1. ****Deep Learning**** là gì?

“Deep learning (also known as deep structured learning or differential programming) là một phần của nhóm phương pháp học máy rộng hơn dựa trên mạng lưới thần kinh nhân tạo với học tập biểu diễn. Việc học có thể được giám sát, bán giám sát hoặc không giám sát”.

Tìm hiểu theo một cách khác thì DL là một tập hợp các thuật toán mô phỏng lại cách thức hoạt động của bộ não con người trong dữ liệu phân tích dữ liệu, nhằm tạo ra các mô hình được sử dụng cho công việc được quyết định dựa trên dữ liệu đầu vào. Về bản chất, DL bao gồm nhiều lớp, mỗi lớp có thể coi là một Mạng nơ-ron mạng (NN).

Giống như con người không có bộ nhớ, NN chứa các nơ-ron. Mỗi nơ-ron nhận các tín hiệu ở đầu vào, các nhân với các số quan trọng ( trọng số *)* , tổng hợp chúng lại và sau đó áp dụng các chức năng kích hoạt ( *thường là phi tuyến tính* ) lên chúng. Các nơ-ron được sắp xếp thành các lớp liên kết với nhau ( *stack* ).

1. ****Deep Learning thuật toán****



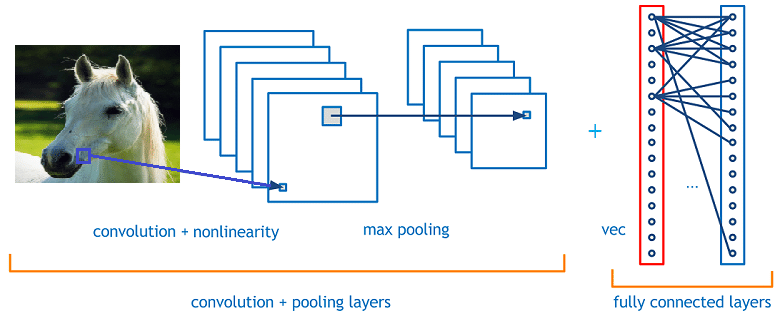
Hình : Các thuật toán của Deeplearning

***Convolution Neural Network (CNN)***

Mạng neural tích chập (CNN) sử dụng phép toán tích chập để tạo kết nối giữa các lớp trong mạng. Khá khác biệt so với mạng neural thông thường, mỗi nơ-ron trong CNN không kết nối với tất cả các nơ-ron khác, mà chỉ kết nối với một vài nơ-ron đại diện. Điều này cho phép CNN học được các mối quan hệ không gian của dữ liệu một cách hiệu quả hơn.

Điều này giải thích vì sao CNN được sử dụng rộng rãi trong các bài toán liên quan đến Thị giác máy tính, bao gồm phân loại hình ảnh, phát hiện đối tượng trong ảnh hoặc video, và nhiều ứng dụng khác.

Ngoài ra, có nhiều kiến trúc CNN kinh điển được phát triển như VGG, ResNet, MobiNet, InceptionNet, Yolo, SSD, mang lại nhiều cải tiến đáng kể trong việc xử lý hình ảnh và video.

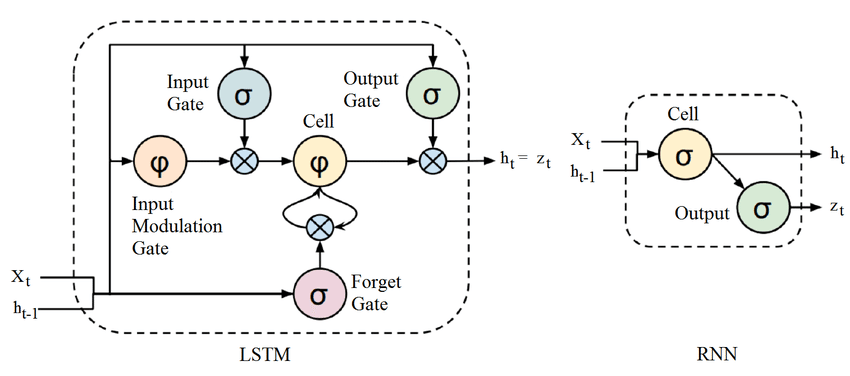


Hình : Mô tả mô hình CNN

***Recurrent Neural Network (RNN)***

Mạng neural hồi quy (RNN) được áp dụng phổ biến trong các loại dữ liệu liên quan đến thời gian, như dự báo chuỗi thời gian, và cũng được ưa chuộng trong xử lý ngôn ngữ tự nhiên (NLP). Điều đặc biệt về RNN đó là đầu ra của nó tại một thời điểm có thể trở thành đầu vào ở thời điểm tiếp theo, tạo nên một chuỗi liên kết. Nhờ điều này, RNN có khả năng ghi lại thông tin từ quá khứ và sử dụng thông tin đó để đưa ra kết quả dự đoán.

Một số kiến trúc mạng RNN kinh điển bao gồm GRU, LSTM và nhiều kiến trúc khác, mang lại nhiều cải tiến đáng kể trong việc xử lý dữ liệu liên quan đến thời gian và ngôn ngữ tự nhiên.



Hình : Mô tả mô hình RNN

***Generative Adversarial Networks (GAN)***

GAN cũng bao gồm 2 thành phần: Generator và Discriminator. Generator sinh ra dữ liệu giả và Discriminator sẽ cố gắng phân biệt dữ liệu mới sinh ra đó là giả hay thật. Hai thành phần này được huấn luyện cùng với nhau, chúng sẽ cạnh tranh nhau để cuối cùng tạo ra được model GAN tốt nhất.

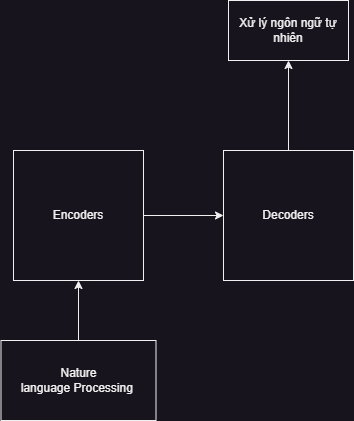
GAN được dùng để sinh ra dữ liệu mới từ dữ liệu ban đầu, ví dụ: sinh ra các mẫu thời trang mới, các mẫu nhân vật hoạt hình mới, …

**Transformer**

Mô hình kiến trúc Transformer là một trong những cải tiến quan trọng và thành công nhất trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) và máy học trong thập kỷ gần đây. Nó đã giúp cải thiện đáng kể hiệu suất của nhiều ứng dụng NLP, chẳng hạn như dịch máy, tổng hợp văn bản, và phân loại văn bản.

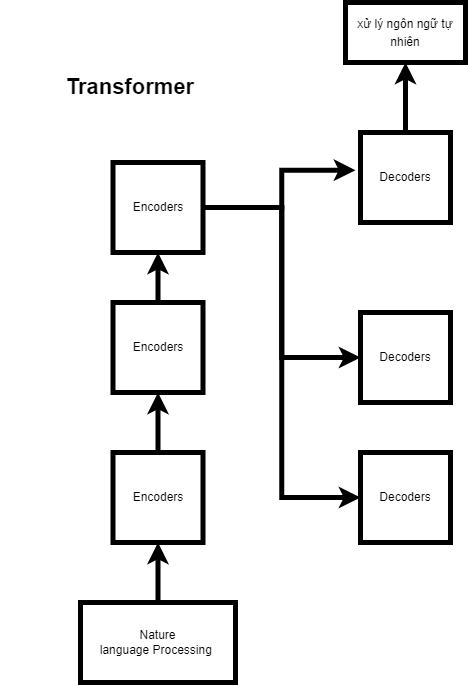
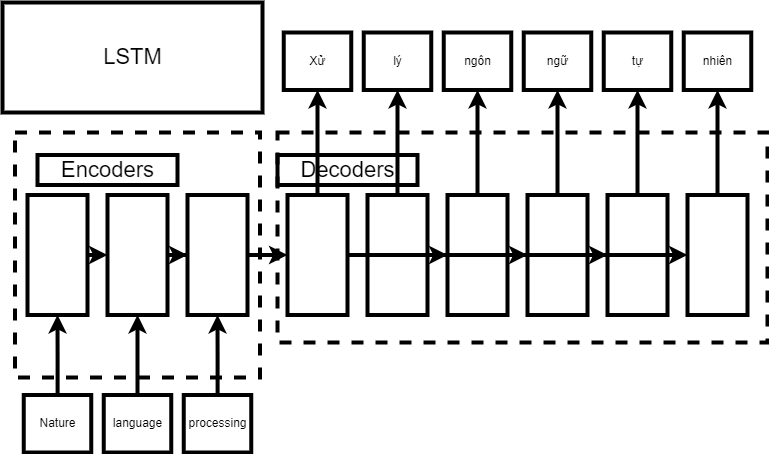
Giống như những mô hình dịch máy khác, kiến trúc tổng quan của mô hình transformer bao gồm 2 phần lớn là encoder và decoder. Encoder dùng để học vector biểu của câu với mong muốn rằng vector này mang thông tin hoàn hảo của câu đó. Decoder thực hiện chức năng chuyển vector biểu diễn kia thành ngôn ngữ đích.

Trong ví dụ ở dưới, encoder của mô hình transformer nhận một câu tiếng anh, và encode thành một vector biểu diễn ngữ nghĩa của câu Nature language Processing , sau đó mô hình decoder nhận vector biểu diễn này, và dịch nó thành câu tiếng việt xử lý ngôn ngữ tự nhiên.



Hình : Mô hình Transformer

Một ưu điểm nổi bật của mô hình Transformer là khả năng xử lý đồng thời các từ trong câu. Trong kiến trúc Transformer, Encoders được tổ chức thành nhiều layers, mỗi layer có khả năng xử lý các từ đồng thời. Điều này khác biệt với mô hình LSTM truyền thống, trong đó các từ phải được xử lý theo thứ tự tuần tự. Ngoài ra, mô hình Transformer cũng thực hiện xử lý theo hai hướng trong câu đầu vào mà không cần đến việc thêm một lớp LSTM hai chiều (Bidirectional LSTM) như trong các kiến trúc khác.



Hình : So sánh mô hình Transformer với LSTM

Hiện nay, BERT và GPT-3 là 2 pre-trained model của transformer được sử dụng rất nhiều trong lĩnh vực NLP.

**Graph Neural Network (GNN)**

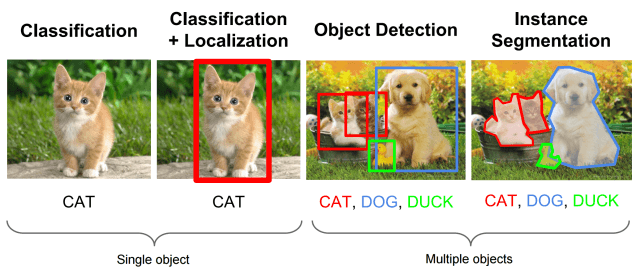
Mạng Nơ-ron Đồ Thị (GNN) là một loại mô hình học sâu được thiết kế đặc biệt để làm việc với dữ liệu có cấu trúc dưới dạng đồ thị. Trong hầu hết các trường hợp, dữ liệu thực tế thường không có cấu trúc rõ ràng và thường được biểu diễn dưới dạng đồ thị, như mạng xã hội hoặc cấu trúc phân tử trong hóa học.

GNN được thiết kế để phân tích các mối liên kết giữa các nút trong đồ thị và tạo ra các vectơ đặc trưng tương ứng, còn được gọi là "embedding". Kết quả này sau đó có thể được sử dụng như đầu vào cho các mạng nơ-ron khác để tiếp tục xử lý dữ liệu.

1. Các bài toán của DL trong Computer Vision

**Image Classification** là quá trình phân loại hình ảnh vào các danh mục, lớp, hoặc nhóm tương ứng dựa trên nội dung của hình ảnh. Điều này thường được thực hiện bằng cách sử dụng các mô hình học sâu, như các mạng nơ-ron sâu.

Các kiến trúc mạng kinh điển như VGG (VGG16, VGG19), ResNet, InceptionNet, MobiNet, AlexNet … giải quyết rất tốt bài toán này.

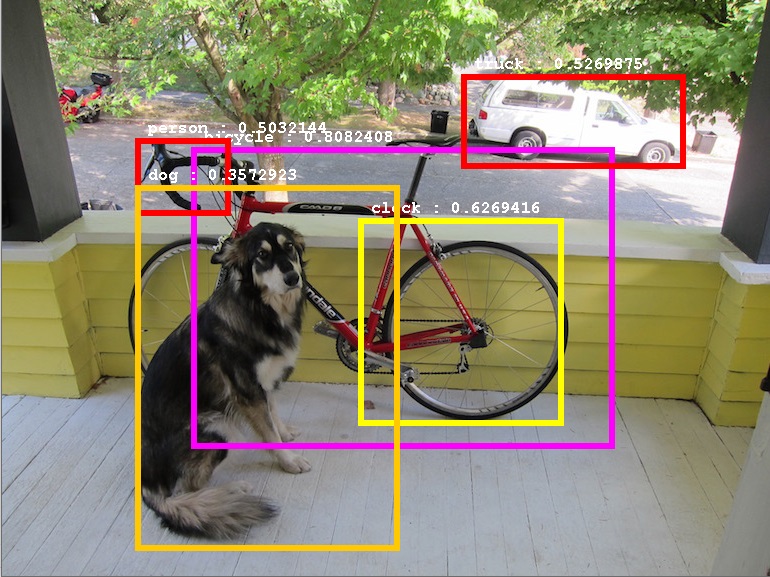


Hình : Mô tả phân loại hình ảnh

**Object Detection** là quá trình nhận diện và xác định vị trí của các đối tượng trong hình ảnh hoặc video. Điều này bao gồm việc tìm ra các bounding box (hình chữ nhật) xác định vị trí của các đối tượng cũng như xác định loại của từng đối tượng đó.

Object Detection cung cấp thông tin về không chỉ vị trí của các đối tượng mà còn loại của từng đối tượng đó. Nó có thể phục vụ nhiều mục đích khác nhau trong thị giác máy tính và phân tích hình ảnh, từ theo dõi vật thể trong video, nhận diện khuôn mặt, đếm đối tượng, đến các ứng dụng như xe tự lái, nhận diện biển báo giao thông, v.v.

Các phương pháp Object Detection bao gồm sử dụng các mô hình học sâu như YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), Faster R-CNN (Region-based Convolutional Neural Networks), và nhiều phương pháp khác.



Hình : Minh họa cho nhận diện đối tượng

Ngoài ra còn các dạng bài toán khác như Semantic Segmentation (Phân đoạn ngữ nghĩa), Instance Segmentation (Phân đoạn phiên bản), Face Recognition (Nhận diện khuôn mặt),… cũng là các bài toán nổi tiếng, phổ biến của Deep Learning trong Computer Vision.

## Computer Vision?

**Computer Vision là gì ?**

Chúng ta đang sống trong một thời đại Cách mạng Trí tuệ Nhân tạo, được đánh dấu bởi sự tiến bộ đáng kinh ngạc trong lĩnh vực học sâu. Chỉ trong vài tháng gần đây, chúng ta đã chứng kiến ứng dụng của trí tuệ nhân tạo khiến thế giới kinh ngạc bằng cách tạo ra những tác phẩm nghệ thuật chân thực, đỗ kỳ thi luật và viết mã Python để tạo ra các trang web.

Thị giác máy tính là một ứng dụng học sâu nằm ở trung tâm của cuộc cách mạng này. Nó cho phép máy tính rút ra thông tin từ đầu vào hình ảnh như hình ảnh và video. Các ví dụ về thị giác máy tính bao gồm nhận diện khuôn mặt, nhận diện khuôn mặt, ước lượng tư thế con người và nhận diện vật cản. Trong hướng dẫn này, chúng ta sẽ khám phá cách thực hiện nhận diện khuôn mặt với OpenCV, xem cả ảnh tĩnh và ảnh thời gian thực.

## OpenCV

Bây giờ chúng ta đã hiểu các ứng dụng thị giác máy tính hữu ích như thế nào, hãy xem xét một công cụ phổ biến được sử dụng để triển khai chúng. OpenCV là thư viện thị giác máy tính hỗ trợ các ngôn ngữ lập trình như Python, C++ và Java.

Gói này ban đầu được Intel tạo ra vào năm 1999 và sau đó được tạo thành nguồn mở và phát hành ra công chúng.

OpenCV cho phép các nhà phát triển và những người không phải là nhà toán học xây dựng các ứng dụng thị giác máy tính một cách dễ dàng mà không cần phải viết mã từ đầu. Thư viện có hơn 2.500 thuật toán cho phép người dùng thực hiện các tác vụ như nhận dạng khuôn mặt và phát hiện đối tượng.

Các nhà phát triển và người thực hiện dữ liệu tại các tổ chức có uy tín như Google, Microsoft, IBM và Intel sử dụng rộng rãi thư viện OpenCV, hiện miễn phí cho mục đích thương mại.

Trong bài viết này, chúng tôi sẽ sử dụng OpenCV để thực hiện nhận diện khuôn mặt trong Python.

Đến cuối hướng dẫn này, bạn sẽ biết cách:

* Phát hiện khuôn mặt người trong ảnh bằng OpenCV bằng Python
* Thực hiện nhận diện khuôn mặt theo thời gian thực trong buổi phát trực tiếp từ webcam
* Nhận biết và gắn nhãn khuôn mặt người nổi tiếng trong ảnh

## Face Detection ?

Face detection (Nhận diện khuôn mặt) là một trong những ứng dụng quan trọng của thị giác máy tính (Computer Vision). Nó có khả năng xác định và định vị vị trí của khuôn mặt trong hình ảnh hoặc video. Face detection được sử dụng rộng rãi để theo dõi và nhận diện người trong các hệ thống an ninh và giám sát.

Nhận diện và phân loại khuôn mặt trong ảnh và video: Đây là ứng dụng phổ biến trong các dịch vụ xã hội, ứng dụng mạng xã hội, và ảnh chụp kỷ niệm.

**Các bước tiến trình:**

Preprocessing: Hình ảnh đầu vào được tiền xử lý để tối ưu hóa việc nhận diện.

Feature extraction: Các đặc trưng quan trọng của khuôn mặt, chẳng hạn như mũi, mắt, miệng, được trích xuất.

Classification: Sử dụng mô hình học máy để xác định xem phần nào của hình ảnh chứa khuôn mặt.

**Các kỹ thuật và mô hình:**

* Viola-Jones Algorithm: Sử dụng AdaBoost để tìm các đặc trưng phổ biến giữa các khuôn mặt và các phần còn lại của hình ảnh.
* Deep Learning-based Models: Các mô hình như MTCNN, SSD, YOLO, và RetinaNet sử dụng deep learning để nhận diện khuôn mặt, cung cấp kết quả chính xác hơn đồng thời có thể xử lý hình ảnh thời gian thực.

**Ứng dụng thực tế:**

Ứng dụng di động: Face ID trên các điện thoại thông minh sử dụng face detection để mở khóa thiết bị,sử dụng trong các hệ thống an ninh tại sân bay, ngân hàng, sự kiện lớn, v.v. Facebook, Instagram, và nhiều ứng dụng xã hội khác sử dụng face detection để gợi ý các gương mặt trong ảnh.

Face detection đóng vai trò quan trọng trong nhiều lĩnh vực của cuộc sống hiện đại và công nghiệp, đồng thời mở ra nhiều tiềm năng ứng dụng trong tương lai.

# CHƯƠNG IV. XÂY DỰNG MÔ HÌNH

Quy trình thực hiện:

Webcam

Nhận dạng khuôn mặt trong luồng video

Vẽ khoanh vùng lại vùng nhận dạng khuôn mặt

Tạo vòng lặp để nhận diện khuôn mặt theo thời gian thực

Đánh giá mô hình

Hình : Quy trình thực hiện

**Web cam**

Chúng ta cần truy cập vào camera của thiết bị để đọc luồng dữ liệu video trực tiếp, lưu ý rằng chúng ta truyền tham số 0 cho hàm VideoCapture(). Điều này yêu cầu OpenCV sử dụng camera mặc định trên thiết bị của chúng ta. Nếu bạn có nhiều camera được gắn vào thiết bị của mình, bạn có thể thay đổi giá trị thông số này cho phù hợp.

Ngoài ra nhóm cũng có chuẩn bị 1 file .mp4 để có thể làm ví dụ cho mô hình nhận dạng khuôn mặt

**Nhận dạng khuôn mặt trong luồng video**

Mô hình mạng nơ-ron sâu sử dụng thư viện MediaPipe của Google sử dụng deep learning để nhận diện và vẽ bounding box (hộp giới hạn) xung quanh các khuôn mặt trong video. Quá trình này có thể được tối ưu hóa để hoạt động ở tốc độ thời gian thực trên video liên tục.

1. \_\_init\_\_: Phương thức khởi tạo, được gọi ngay khi một đối tượng của lớp được tạo. Nó nhận một tham số là minDetectionCon mặc định là 0.5. Trong phần này, các thuộc tính cơ bản được khởi tạo, bao gồm:
   * minDetectionCon: Ngưỡng nhận diện tối thiểu (mặc định là 0.5).
   * mpFaceDetection: Module mediapipe cho nhận diện khuôn mặt.
   * mpDraw: Module mediapipe cho các tiện ích vẽ.
   * faceDetection: Đối tượng FaceDetection được khởi tạo từ module mpFaceDetection với ngưỡng nhận diện được truyền vào.
2. findFaces: Phương thức này nhận một hình ảnh img và một cờ draw (mặc định là True). Nhiệm vụ của phương thức này là nhận diện các khuôn mặt trong hình ảnh. Các bước chính bao gồm:
   * Chuyển đổi không gian màu của ảnh thành RGB.
   * Tiến hành nhận diện khuôn mặt bằng cách gọi phương thức process của đối tượng faceDetection.
   * Nếu có kết quả từ quá trình nhận diện (self.results.detections), thì tiến hành duyệt qua từng kết quả và lấy ra thông tin về bounding box (bbox) và điểm độ tin cậy (confidence score).
   * Cập nhật danh sách bboxs với thông tin về từng khuôn mặt đã nhận diện.
   * Nếu cờ draw được thiết lập, thì tiến hành vẽ bounding box và hiển thị điểm độ tin cậy lên hình ảnh.
   * Trả về hình ảnh đã được xử lý và danh sách bboxs.

**Vẽ khoanh vùng lại vùng nhận dạng khuôn mặt**

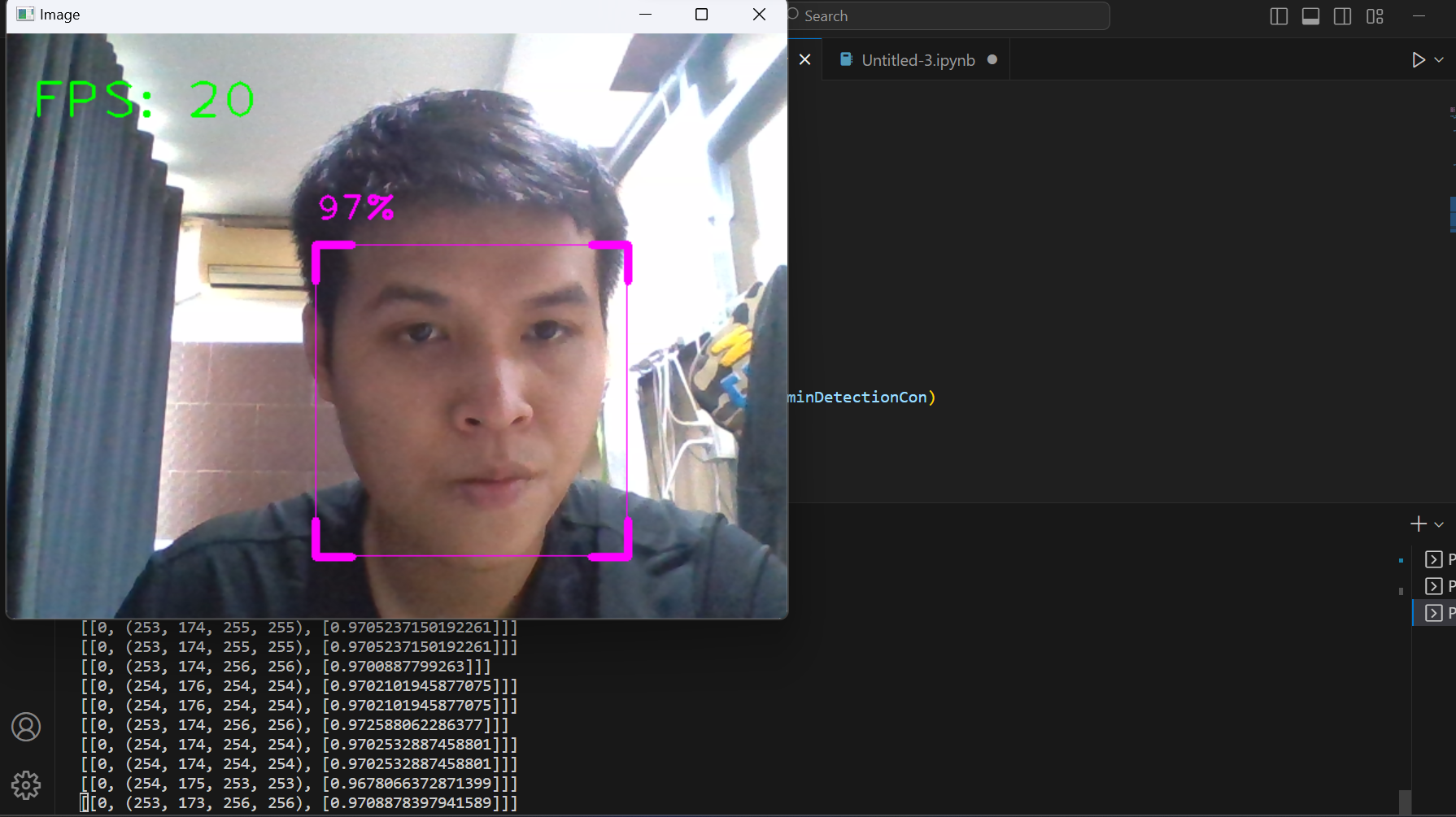
Phương thức fancyDraw được định nghĩa trong lớp FaceDetector và có chức năng vẽ các đường và hình chữ nhật xung quanh khuôn mặt đã nhận diện.

* **def fancyDraw(self, img, bbox, l=30, t=5, rt= 1):**: Khai báo phương thức **fancyDraw** nhận vào 4 tham số: **self** (chính là đối tượng lớp đang gọi phương thức), **img** (hình ảnh), **bbox** (bounding box - hộp giới hạn khuôn mặt được nhận diện), và 3 tham số tùy chọn với giá trị mặc định **l=30**, **t=5**, **rt=1**.
* **x, y, w, h = bbox**: Phân rã thông tin từ bounding box gồm tọa độ **(x, y)** của góc trên bên trái và kích thước **(w, h)**.
* **x1, y1 = x + w, y + h**: Tính toán tọa độ của góc dưới bên phải của bounding box.
* **cv2.rectangle(img, bbox, (255, 0, 255), rt)**: Vẽ hình chữ nhật xung quanh khuôn mặt, sử dụng màu hồng với độ dày được chỉ định bởi **rt**.
* Các dòng tiếp theo vẽ các đường:
  + **cv2.line(img, (x, y), (x + l, y), (255, 0, 255), t)**: Vẽ đường từ **(x, y)** đến **(x + l, y)** (đường ngang phía trên).
  + **cv2.line(img, (x, y), (x, y + l), (255, 0, 255), t)**: Vẽ đường từ **(x, y)** đến **(x, y + l)** (đường dọc bên trái).
  + Tương tự, các dòng sau vẽ các đường còn lại.
* **return img**: Trả về hình ảnh đã được vẽ các đường và hình chữ nhật.

**Tạo vòng lặp để nhận diện khuôn mặt theo thời gian thực**

Chương trình này sử dụng webcam hoặc camera trên máy tính để nhận diện khuôn mặt theo thời gian thực.

1. **pTime = 0**: Khởi tạo biến **pTime** để lưu thời điểm trước đó.
2. **detector = FaceDetector()**: Tạo một đối tượng **FaceDetector** để sử dụng các phương thức đã được định nghĩa trong lớp.
3. **while True:**: Bắt đầu một vòng lặp vô hạn, nghĩa là chương trình sẽ chạy liên tục cho đến khi gặp lỗi hoặc có lệnh thoát.
4. **success, img = cap.read()**: Đọc frame từ camera, **success** sẽ mang giá trị **True** nếu việc đọc frame thành công.
5. **img, bboxs = detector.findFaces(img)**: Sử dụng đối tượng **detector** để tìm khuôn mặt trong hình ảnh. Hàm **findFaces** trả về ảnh đã được vẽ bounding box và danh sách các bounding box đã tìm thấy.
6. **cTime = time.time()**: Ghi lại thời điểm hiện tại.
7. **fps = 1 / (cTime - pTime)**: Tính toán số frame mỗi giây (Frames Per Second).
8. **pTime = cTime**: Cập nhật thời điểm trước đó.
9. **cv2.putText(img, f'FPS: {int(fps)}', (20, 70), cv2.FONT\_HERSHEY\_PLAIN, 3, (0, 255, 0), 2)**: Hiển thị số FPS lên góc trái trên màn hình.
10. **cv2.imshow("Image", img)**: Hiển thị hình ảnh lên cửa sổ.
11. **key = cv2.waitKey(20)**: Chờ đợi một phím được nhấn trong 20 miligiây. Nếu có phím được nhấn, thì **key** sẽ mang giá trị Unicode của phím đó.
12. **if key == ord('q'): break**: Nếu phím được nhấn là 'q', thì thoát khỏi vòng lặp, kết thúc chương trình.



Hình : kết quả nhận diện khuôn mặt bằng webcam

Thuật toán sẽ theo dõi khuôn mặt của bạn và tạo một khung giới hạn màu tím xung quanh khuôn mặt đó bất kể bạn di chuyển đến đâu trong khung hình.

Góc trái trên là FPS ( Frame per second) góc trên của khung giới hạn là phần trăm thuật toán theo dõi và nhận dạng đó là khuôn mặt, phần trăm sẽ thay đổi khi môi trường của thuật toán gặp nhiều vật cản phần camera theo dõi, hoặc khi ta xoay nhiều góc khuất làm thuật toán không nhận diện được khuôn mặt.

Ta cũng có thể kiểm tra tính hiệu quả của mô hình này bằng cách giơ nhiều bức ảnh lên hoặc bằng cách yêu cầu những người khác nhau đứng ở các góc khác nhau phía sau máy ảnh. Mô hình phải có thể nhận dạng tất cả khuôn mặt người ở các nền hoặc cài đặt ánh sáng khác nhau.

****

Hình : kết quả nhận diện khuôn mặt qua file .mp4

Đây là kết quả khi cho mô hình chạy file .mp4, ta dễ dàng thấy do môi trường đối tượng di chuyển quay mặt liên tục, cũng như không phải nhìn vào máy ảnh 100% nên mô hình đã trả ra nhiều phần trăm khá thấp, tuy nhiên khi gặp môi trường thuận lợi thì mô hình vẫn trả ra kết quả đáng tin cậy 97%.

**Đánh giá mô hình**

1. **Độ chính xác (Precision)**: Độ chính xác của mô hình có thể được điều chỉnh thông qua biến minDetectionCon. Giá trị càng cao thì mô hình sẽ đưa ra kết quả với độ chính xác càng cao. Điều này có thể ảnh hưởng đến việc nhận diện khuôn mặt đúng và giảm số lượng false positives.
2. **Tốc độ xử lý:** Mô hình này có khả năng xử lý ảnh trong thời gian thực trên các video. Tuy nhiên, tốc độ xử lý có thể thay đổi tùy thuộc vào khả năng xử lý của thiết bị và độ phân giải của video đầu vào.
3. **Khả năng nhận diện khuôn mặt:** Mô hình này sử dụng relative bounding box để xác định vị trí của khuôn mặt trong khung hình. Có thể điều chỉnh các tham số như l, t, rt để tùy chỉnh vẽ bounding box và các đường nối đi kèm.

**Những điều cần lưu ý:**

* Độ chính xác của mô hình có thể bị ảnh hưởng bởi nhiều yếu tố như độ sáng, góc độ, màu da, v.v. Do đó, có thể cần điều chỉnh các tham số để đạt được kết quả tốt nhất cho tình huống cụ thể.
* Đối với video, việc nhận diện khuôn mặt có thể thay đổi tùy theo điều kiện ánh sáng và góc độ chụp.
* Kỹ thuật vẽ bounding box và các đường nối giúp hình ảnh trở nên trực quan và dễ nhìn hơn.

# CHƯƠNG V . KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Nhóm đã thực hiện và đạt được kết quả:

* Nghiên cứu lý thuyết về nhận diện khuôn mặt
* Nghiên cứu các phương pháp huấn luyện mô hình
* Kết quả đánh giá đạt hiệu suất cao

## Hướng phát triển

Hướng phát triển của nhóm :

Mở rộng xác định khuôn mặt con người trong các loại dữ liệu đầu vào khác nhau, chẳng hạn như tệp PDF hoặc hình ảnh giám sát. Thậm chí có thể thiết lập camera an ninh của riêng mình và thực hiện nhận diện khuôn mặt trên dữ liệu mà nó ghi lại trong thời gian thực.

# TÀI LIỆU THAM KHẢO

[1] Michael Elad, Yacov Hel-Or, Renato Keshet, “Rejection Based Classifier for Face Detection”, Hewlett Packard Laboratories, Israel, Pattern Recognition Letters, April 2001.

[2] Michael Elad, EE368 Lecture, 19 May 2003 and 21 May 2003.

[3] Face Recognition Data, University of Essex, UK, Face 94,

http://cswww.essex.ac.uk/mv/all faces/faces94.html.

**Link sản phẩm :**

**https://drive.google.com/drive/folders/1f267PQomWHSUyqLf6tiKue-\_WUMxPzQ3?usp=sharing**