**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**KHOA ĐÀO TẠO KIẾN THỨC CHUNG**

****

**TIỂU LUẬN KẾT THÚC HỌC PHẦN**

**THIẾT KẾ WEBSITE**

**ĐỀ TÀI: THIẾT KẾ WEBSITE LAPTOP và PHỤ KIỆN**

**Giảng viên hướng dẫn:** ThS.Trần Bá Minh Sơn

Nguyễn Hữu Nghĩa MSSV: 2124802050013

Lớp: D21TTNT01

***Bình Dương, tháng 03 năm 2022***

MỤC LỤC

[MỤC LỤC v](#_Toc100200382)

[DANH MỤC HÌNH vii](#_Toc100200383)

[A. PHẦN MỞ ĐẦU 1](#_Toc100200384)

[1. Lí do chọn đề tài tiểu luận: 1](#_Toc100200385)

[2. Mục tiêu nghiên cứu: 1](#_Toc100200386)

[3. Đối tượng nghiên cứu: 1](#_Toc100200387)

[4. Phạm vi nghiên cứu: 1](#_Toc100200388)

[5. Phương pháp nghiên cứu 1](#_Toc100200389)

[6. Ý nghĩa đề tài: 1](#_Toc100200390)

[7. Kết cấu tiểu luận 1](#_Toc100200391)

[B. PHẦN NỘI DUNG 2](#_Toc100200392)

[CHƯƠNG 1. CƠ SỞ LÍ THUYẾT 2](#_Toc100200393)

[1.1. Tổng quan tình hình nghiên cứu: 2](#_Toc100200394)

[1.2. Một số khái niệm 2](#_Toc100200395)

[1.2.1. Khái niệm giấc ngủ 2](#_Toc100200396)

[1.2.2. Khái niệm xử lí ảnh 6](#_Toc100200397)

[1.2.3. Machine learning 11](#_Toc100200398)

[1.2.4. Deep learning 15](#_Toc100200399)

[CHƯƠNG 2. MÔ HÌNH BÀI TOÁN 19](#_Toc100200400)

[2.1. Quy trình thực hiện 19](#_Toc100200401)

[2.2. Mô hình bài toán 19](#_Toc100200402)

[2.2.1. Train model 19](#_Toc100200403)

[2.2.2. Thuật toán được sử dụng 19](#_Toc100200404)

[2.2.3. Test model 25](#_Toc100200405)

[2.3. Quy trình thực hiện 25](#_Toc100200406)

[2.4. Mô hình bài toán 25](#_Toc100200407)

[CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ 28](#_Toc100200408)

[3.1. Chuẩn bị 28](#_Toc100200409)

[3.1.1. Cấu hình máy 28](#_Toc100200410)

[3.1.2. Phần mềm 28](#_Toc100200411)

[3.1.3. Ngôn ngữ lập trình 28](#_Toc100200412)

[3.1.4. Các thư viện được sử dụng 30](#_Toc100200413)

[3.1.5. Mô tả bộ dữ liệu 30](#_Toc100200414)

[3.2. Thực hành 33](#_Toc100200415)

[3.2.1. Train model 33](#_Toc100200416)

[3.2.2. Test model 33](#_Toc100200417)

[3.3. Kết quả 33](#_Toc100200418)

[3.4. Đánh giá 33](#_Toc100200419)

[C. PHẦN KẾT LUẬN 34](#_Toc100200420)

[1. Kết quả đạt được 34](#_Toc100200421)

[2. Khuyến nghị 34](#_Toc100200422)

[TÀI LIỆU THAM KHẢO 35](#_Toc100200423)

DANH MỤC HÌNH

[**Hình 1.1:** Các giai đoạn chính trong xử lý ảnh. 7](#_Toc100214304)

[**Hình 1.2:** Quan hệ trong vùng lân cận giữa các điểm ảnh. 10](#_Toc100214305)

[**Hình 1.3:** Thuật toán trong machine learning 12](#_Toc100214306)

[**Hình 1.4:** AlphaGo chơi cờ vây với Lee Sedol. 14](#_Toc100214307)

[**Hình 1.5:** Cách thức hoạt động của deeplearning. 16](#_Toc100214308)

[**Hình 1.6:** Hình mô phỏng sử dụng deeplearning dự đoán giá vé máy bay. 17](#_Toc100214309)

[**Hình 2.1:** Quy trình thực hiện 20](#_Toc100214310)

[**Hình 2.2:** Mô hình bài toán 21](#_Toc100214311)

[**Hình 2.3:** Ma trận mô tả Convolutional 22](#_Toc100214312)

[**Hình 2.4:** Convolution của ảnh. 23](#_Toc100214313)

[**Hình 2.5:** Cấu trúc mạng CNN. 24](#_Toc100214314)

[**Hình 2.6:** Trường tiếp nhận cục bộ (local receptive field). 25](#_Toc100214315)

[**Hình 3.1:** OpenCV. 31](#_Toc100214316)

[**Hình 3.2:** Keras. 31](#_Toc100214317)

[**Hình 3.3:** TensorFlow. 32](#_Toc100214318)

[**Hình 3.4:** Numpy. 33](#_Toc100214319)

[**Hình 3.5:** Matplotlib. 34](#_Toc100214320)

[**Hình 3.6:** Folder Open trong dataset. 35](#_Toc100214321)

[**Hình 3.7:** Folder Closed trong dataset. 35](#_Toc100214322)

[**Hình 3.8:** Folder Open trong dataset2. 36](#_Toc100214323)

[**Hình 3.9:** Folder Closed trong dataset2. 36](#_Toc100214324)

[**Hình 3.10:** Folder Open tự thu thập. 37](#_Toc100214325)

[**Hình 3.11:** Folder Closed tự thu thập. 37](#_Toc100214326)

A. PHẦN MỞ ĐẦU

1. Lí do chọn đề tài:

Với tình hình hiện nay, tai nạn giao thông do tài xế ngủ gật đang được đưa vào tình trạng đáng báo động. Theo thống kê của Ủy ban An toàn giao thông quốc gia, mỗi năm có hơn 6000 người chết từ những vụ tai nạn giao thông liên quan tới tài xế ngủ gật. Nguyên nhân do thiếu ngủ chiếm tới 30% tổng các vụ giao thông trong một năm.

Có thể thấy tỉ lệ người tử vong do tai nạn giao thông trong năm 2020 (từ ngày 15.12.2019 đến 14.12.2020) toàn quốc đã xảy ra 14.510 vụ tai nạn giao thông, làm chết 6.700 người, bị thương 10.804 ngươi và vấn đề tài xế xe ngủ không đủ giấc gây nên tai nạn giao thông cũng chiếm tỉ lệ không hề nhỏ trong việc làm tăng số vụ tai nạn giao thông cũng như là số ca tử vọng và bị thương lên đến một con số vô cùng lớn. Vì vậy chung tôi đã quyết định cùng nhau tìm tòi tận dụng những tiền đề của những người nghiên cứu đi trước nhắm phát huy cái mới của họ đồng thời đưa thêm những ý tưởng sáng tạo của những người mới như chúng tôi để có thể phát triển đưa ra một phần mềm ứng dụng Trí Tuệ Nhân Tạo (AI) vào cuộc sống thực tế nhằm giải quyết vấn đề cấp thiết hiện nay đồng thời giảm thiểu đi nhưng vụ tai nạn không đáng có. Chúng tôi đã chọn khu vực địa bàn tỉnh bình dương để nghiên cứu đề tài này vì chúng tôi muốn thử sức mình ở khu vực hiện tai mình đang sinh sống nếu thành công chúng tôi cũng muốn phát triển rộng và tiến xa hơn nữa và đây cũng chính là lý do mà chúng tôi muốn cùng nhau làm việc để tìm ra một phần mềm tối ưu nhất có thể, để giúp cho mọi người giảm bớt đi nhưng lo lắng khi tham gia giao thông.

1. Mục tiêu nghiên cứu:

Xây dựng một hệ thống giúp theo dõi tài xế xem họ có dấu hiệu buồn ngủ hay không.

1. Đối tượng nghiên cứu:

Tài xế khi tham gia giao thông.

1. Phạm vi nghiên cứu:

Địa bàn tỉnh Bình Dương.

1. Phương pháp nghiên cứu

**Phương pháp nghiên cứu lý thuyết:**

Thị giác máy tính, xử lý hình ảnh, xử lý video

**Các phương pháp học máy:**

Ngôn ngữ và kỹ thuật lập trình

**Phương pháp thực nghiệm:**

Lập trình xây dựng ứng dụng học sâu.

1. Ý nghĩa đề tài:

Nhắc nhở các tài xế hoàn thành tốt công việc của mình một cách hiệu quả và giảm thiểu tai nạn giao thông vào những giờ cao điểm.

1. Kết cấu tiểu luận

Chương 1: Cơ sở lý thuyết và mô hình bài toán

Chương 2: Tài nguyên

Chương 3: Đánh giá kết quả thực nghiệm

B. PHẦN NỘI DUNG

1. CƠ SỞ LÍ THUYẾT
   1. Tổng quan tình hình nghiên cứu:

**Trong nước:**

Nghiên cứu xây dựng hệ thống nhúng phát hiện và cảnh báo lái xe ngủ gật dựa trên kỹ thuật xử lý ảnh (bắt đầu 1/2014 dự kiến hoàn thành 12/2016)

Nghiên cứu: ứng dụng sóng não phát hiện dấu hiệu buồn ngủ và đưa ra tín hiệu cảnh báo đối người lái

Năm 2015, học sinh sáng chế thiết bị chống ngủ gật cho tài xế ô tô

Năm 2012, Nhóm sinh viên Đại học Bách khoa TP.HCM đã nghiên cứu và cho ra đời phần mềm nhận diện chớp mắt và cảnh báo buồn ngủ khi lái xe trên nền Windows và Android

**Ngoài nước:**

Tại Mỹ:Máy chống ngủ gật đeo tai của những nhãn hiệu như Nap Zapper1, No Nap, Doze Alert.

Tại Đức:Sản phẩm có tên là Stopsleep, là thiết bị đeo trên ngón tay dựa vào độ dẫn điện của da, phản ánh hoạt động của não để có thể phát hiện tình trạng buồn ngủ của người lái xe.

Tại Hong Kong: Công ty Holux3 đã cho ra thiết bị có thể đặt trên dây thắt an toàn và phát hiện tình trạng mệt mỏi buồn ngủ dựa trên phân tích nhịp tim.

Một số các hang xe ô tô lớn cũng đã tích hợp các thiết bị chống ngủ gật an toàn trên các loại xe của họ.

* 1. Một số khái niệm
     1. Khái niệm giấc ngủ

Ngủ là một hoạt động tự nhiên theo định kỳ mà những cảm giác và vận động tạm thời bị hoãn lại một cách tương đối, với đặc điểm dễ nhận thấy là cơ thể bất tỉnh hoàn toàn hoặc một phần và sự bất động gần như hầu các cơ bắp. Nó được phân biệt với sự tỉnh táo bằng khả năng giảm các phản ứng với sự kích thích, và nó dễ dàng bị chấm dứt hơn so với ngủ đông hoặc hôn mê. Giấc ngủ là một trạng thái đồng bộ cao, tăng cường sự tăng trưởng và trẻ hóa của hệ thống miễn dịch, thần kinh, xương và hệ thống cơ bắp. Nó được quan sát thấy ở tất cả động vật có vú, tất cả các loài chim, và nhiều loài bò sát, động vật lưỡng cư, cá. Ở con người, các động vật có vú khác, và đa số phân loại động vật khác đã được nghiên (như một số loài cá, chim, kiến, ruồi quả), giấc ngủ thường xuyên rất cần thiết cho sự sống.

Hầu hết mọi người đều cho rằng giấc ngủ là một hoạt động thụ động mà khi đó cơ thể và não bộ không hoạt động. Nhưng thực chất cho thấy não bộ đang tham gia vào một số hoạt động cần thiết của chính cuộc sống của con người trong suốt một chu trình giấc ngủ có liên quan rất chặt chẽ đến chất lượng cuộc sống của con người chúng ta. Ngủ là một trạng thái mà nhận thức đối với các kích thích của môi trường bị giảm sút. Giấc ngủ khác với trạng thái hôn mê, ngủ đông và chết bởi thực tế là nó có thể được đảo ngược nhanh chóng.

Các hóa chất truyền tín hiệu thần kinh được gọi là chất dẫn truyền thần kinh kiểm soát việc chúng ta đang ngủ hay đang thức bằng cách tác động lên các nhóm tế bào thần kinh hoặc tế bào thần kinh khác nhau trong não bộ con người. Các tế bào thần kinh trong thân não, kết nối não với tủy sống, sản xuất chất dẫn truyền thần kinh như serotonin và norepinephrine giúp giữ cho một số bộ phận của não hoạt động trong khi chúng ta thức. Các tế bào thần kinh khác ở đáy não bắt đầu phát tín hiệu khi chúng ta chìm vào giấc ngủ. Những tế bào thần kinh này dường như "tắt" các tín hiệu giúp chúng ta tỉnh táo. Nghiên cứu cũng cho thấy rằng một chất hóa học có tên là adenosine tích tụ trong máu khi chúng ta thức và gây buồn ngủ. Hóa chất này dần dần bị phá vỡ trong khi chúng ta ngủ.

Trong chu kỳ của một giấc ngủ, chúng ta thường trải qua 5 giai đoạn: giai đoạn 1, 2, 3, 4 và giấc ngủ REM (chuyển động mắt nhanh). Các giai đoạn này của giấc ngủ tiến triển theo chu kỳ từ giai đoạn 1 đến giai đoạn REM, sau đó chu kỳ bắt đầu lại với giai đoạn 1. Trẻ em và người lớn dành gần 50% tổng thời gian ngủ của họ ở giai đoạn 2, khoảng 20% ​​trong giai đoạn REM, và 30 phần trăm còn lại trong các giai đoạn khác. Ngược lại, trẻ sơ sinh dành khoảng một nửa thời gian ngủ trong giấc ngủ REM.

Trong giai đoạn 1, tức là ngủ nhẹ, chúng ta chìm vào giấc ngủ và có thể dễ dàng bị đánh thức. Đôi mắt của chúng ta di chuyển rất chậm và hoạt động của cơ cũng chậm lại. Những người bị đánh thức từ giấc ngủ giai đoạn 1 thường ghi nhớ những hình ảnh trực quan bị phân mảnh. Nhiều người cũng trải qua các cơn co thắt cơ đột ngột được gọi là giảm trương lực cơ hoặc giật cơ giảm trương lực, thường xảy ra trước cảm giác bắt đầu ngã. Những chuyển động đột ngột này tương tự như "bước nhảy" mà chúng ta thực hiện khi bị giật mình. Một số người gặp phải chứng rối loạn giấc ngủ được gọi là Chuyển động chân tay định kỳ của giấc ngủ ( PLMS), nơi họ trải qua các chuyển động chân lặp đi lặp lại.

Khi chúng ta bước vào giai đoạn 2 của giấc ngủ, chuyển động mắt của chúng ta dừng lại và sóng não (dao động của hoạt động điện có thể được đo bằng điện cực) trở nên chậm hơn, với những đợt bùng phát sóng nhanh không thường xuyên được gọi là trục quay khi ngủ .

Trong giai đoạn 3, các sóng não cực kỳ chậm được gọi là sóng delta bắt đầu xuất hiện, xen kẽ với các sóng nhỏ hơn, nhanh hơn. Đến giai đoạn 4, não hầu như chỉ tạo ra sóng delta. Rất khó đánh thức ai đó trong giai đoạn 3 và 4, gọi chung là giấc ngủ sâu. Trong giai đoạn này, không có hoạt động cơ bắp hoặc không có cử động mắt. Những người bị đánh thức trong giấc ngủ sâu không điều chỉnh ngay lập tức và thường cảm thấy chệnh choạng và mất phương hướng trong vài phút sau khi thức dậy. Một số trẻ em bị đái dầm , kinh hãi ban đêm hoặc mộng du khi ngủ sâu.

Khi chúng ta chuyển sang giấc ngủ REM , hơi thở của chúng ta trở nên nhanh hơn, không đều và nông hơn, mắt chúng ta giật nhanh theo nhiều hướng khác nhau và các cơ tay chân của chúng ta tạm thời bị tê liệt trong khi ngủ . Nhịp tim của chúng ta tăng lên, huyết áp của chúng ta tăng lên và nam giới sẽ cương cứng dương vật. Có một khoảng trong giấc ngủ REM, vài người thường mô tả những câu chuyện kỳ ​​quái và phi logic còn được gọi là những giấc mơ.

Giai đoạn giấc ngủ REM đầu tiên thường xảy ra khoảng 70 đến 90 phút sau khi chúng ta chìm vào giấc ngủ. ​​90 đến 110 phút là khoảng thời gian trung bình cho một chu kỳ ngủ hoàn chỉnh của con người. Các chu kỳ ngủ đầu tiên mỗi đêm chứa các giai đoạn REM tương đối ngắn và thời gian ngủ sâu kéo dài. Càng về đêm, thời gian ngủ REM càng dài ra tương ứng là giấc ngủ sâu giảm đi. Vào buổi sáng, mọi người dành gần như tất cả thời gian để ngủ trong các giai đoạn 1, 2 và REM.

Vì giấc ngủ và sự tỉnh táo bị ảnh hưởng bởi các tín hiệu dẫn truyền thần kinh khác nhau trong não, các loại thực phẩm và thuốc làm thay đổi sự cân bằng của các tín hiệu này sẽ ảnh hưởng đến việc chúng ta cảm thấy tỉnh táo hay buồn ngủ và chúng ta ngủ ngon như thế nào. Đồ uống chứa cafein như cà phê, và các loại thuốc như thuốc chế độ ăn uống và thuốc thông mũi kích thích một số bộ phận của não và có thể gây mất ngủ , hoặc không có khả năng ngủ. Nhiều loại thuốc chống trầm cảm ngăn chặn giấc ngủ REM. Những người nghiện thuốc lá nặng thường ngủ rất nhẹ và giảm số lượng giấc ngủ REM. Họ cũng có xu hướng thức dậy sau 3 hoặc 4 giờ ngủ do cai nicotine. Nhiều người bị mất ngủ cố gắng giải quyết vấn đề bằng rượu - cái được gọi là mũ ban đêm. Trong khi rượu giúp con người chìm vào giấc ngủ nhẹ, nó cũng cướp đi giấc ngủ say của họ và giấc ngủ sâu hơn, phục hồi hơn các giai đoạn của giấc ngủ. Thay vào đó, nó giữ họ ở giai đoạn nhẹ hơn của giấc ngủ , từ đó họ có thể dễ dàng bị đánh thức.

Chúng ta mất một số khả năng điều chỉnh nhiệt độ cơ thể trong giai đoạn REM, vì vậy nhiệt độ nóng hoặc lạnh bất thường trong môi trường có thể làm gián đoạn giai đoạn này của giấc ngủ. Nếu giấc ngủ REM của chúng ta bị gián đoạn vào một đêm, cơ thể chúng ta sẽ không tuân theo tiến trình chu kỳ giấc ngủ bình thường vào lần ngủ gật tiếp theo. Thay vào đó, chúng ta thường trực tiếp đi vào giấc ngủ REM và trải qua giai đoạn REM kéo dài cho đến khi chúng ta “bắt kịp” giai đoạn này của giấc ngủ.

Nếu chúng ta bị đánh thức sau khi ngủ hơn vài phút, chúng ta thường không thể nhớ lại vài phút cuối cùng trước khi chìm vào giấc ngủ. Dạng mất trí nhớ liên quan đến giấc ngủ này là lý do khiến mọi người thường quên các cuộc điện thoại hoặc cuộc trò chuyện mà họ đã có vào lúc nửa đêm. Nó cũng giải thích tại sao chúng ta thường không nhớ chuông báo thức của mình đổ chuông vào buổi sáng nếu chúng ta đi ngủ ngay sau khi tắt chúng đi.

Những người đang được gây mê hoặc hôn mê thường được cho là đang ngủ. Tuy nhiên, những người trong tình trạng này không thể bị đánh thức và không tạo ra các mô hình sóng não hoạt động, phức tạp được thấy trong giấc ngủ bình thường. Thay vào đó, sóng não của họ rất chậm và yếu, đôi khi không thể phát hiện được.

Giấc ngủ là cần thiết để có một hệ thống thần kinh hoạt động bình thường. Do đó, nếu chúng ta ngủ quá ít thì cơ thể sẽ uể oải và khó để tập trung vào ngày hôm sau. Ngoài ra, nó cũng dẫn đến suy giảm trí nhớ và hoạt động thể chất, đồng thời giảm khả năng thực hiện các phép tính toán học. Ảo giác và thay đổi tâm trạng có thể phát triển nếu tình trạng thiếu ngủ kéo dài. Thậm chí, các tế bào thần kinh có thể trở nên cạn kiệt năng lượng hoặc bị ô nhiễm bởi các sản phẩm phụ của hoạt động của các tế bào bình thường nếu chúng ta không ngủ. Ngoài ra, giấc ngủ cũng tạo cơ hội để kết nối tế bào thần kinh quan trọng vốn có thể xấu đi do thiếu hoạt động.

Ở trẻ em và thanh thiếu niên, sự phát triển thường xảy ra trong khi ngủ do giấc ngủ sâu đồng thời với việc tiết ra hormone tăng trưởng. Trong giai đoạn ngủ sâu, nhiều tế bào của cơ thể cũng cho thấy sự gia tăng sản xuất và giảm sự phân hủy protein. Vì protein là các khối xây dựng cần thiết cho sự phát triển của tế bào và để sửa chữa các tổn thương do các yếu tố như căng thẳng và phơi nhiễm tia cực tím, nên giấc ngủ sâu có thể thực sự là “giấc ngủ đẹp”.

Một nghiên cứu thực hiện trên chuột chỉ ra rằng một số mô hình tín hiệu thần kinh mà chuột tạo ra trong ngày được lặp lại khi ngủ sâu. Sự lặp lại này có thể giúp mã hóa ký ức và cải thiện việc học.

Có một loại giấc ngủ có tên gọi là “Microsleep” là thuật ngữ nói về giấc ngủ không giống như giấc ngủ thông thường, “Microsleep” không rơi trực tiếp vào một trong những loại giấc ngủ, vì nó là một khoảng thời gian thoáng qua và không thể kiểm soát mà lại không kéo dài đủ lâu để các đặc tính của trạng thái ngủ xuất hiện. “Microsleep” là sự kiện một người mất ý thức hoặc sự chú ý do cảm thấy mệt mỏi hoặc buồn ngủ. Các sự kiện “Microsleep” thường chỉ kéo dài khoảng một giây đến hai phút, nhưng thời gian có thể lâu hơn. Microsleep thường xảy ra khi một cá nhân bị thiếu ngủ. Ban đêm là khi phần lớn các sự cố microsleep xảy ra, trong điều kiện lái xe. Lái xe vào ban đêm thường trở nên mệt mỏi.  Các triệu chứng liên quan đến tình trạng này bao gồm:

+ không phản hồi thông tin

+ một cái nhìn trống rỗng

+ gục đầu xuống

+ bị giật cơ thể đột ngột

+ không thể nhớ một hoặc hai phút cuối cùng

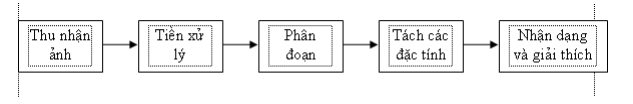
+ chớp mắt chậm

Một người trãi qua “Microsleep” không nhận ra rằng mình đang ngủ hoặc sắp bước vào trạng thái ngủ, tình trạng này cũng có thể xảy ra khi mắt mở và trống rỗng. “Microsleep” cũng được đặc trưng bởi các chuyển động của đầu như gật đầu và chớp mắt quá thường xuyên và không thể nhớ những điều đã xãy ra trước đó một phút. Sau khi chìm vào giấc ngủ, một người trãi qua giấc ngủ nhỏ thường thức dậy với cảm giác sảng khoái hơn trong một thời gian ngắn.

* + 1. Khái niệm xử lí ảnh

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh. Cũng như xử lý dữ liệu bằng đồ hoạ, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hoá các ảnh tự nhiên. Mục đích của xử lý ảnh gồm: Biến đổi ảnh làm tăng chất lượng ảnh. Tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh. Nhận biết và đánh giá các nội dung của ảnh là sự phân tích một hình ảnh thành những phần có ý nghĩa để phân biệt đối tượng này với đối tượng khác, dựa vào đó ta có thể mô tả cấu trúc của hình ảnh ban đầu. Có thể liệt kê một số phương pháp nhận dạng cơ bản như nhận dạng ảnh của các đối tượng trên ảnh, tách cạnh, phân đoạn hình ảnh,… Kỹ thuật này được dùng nhiều trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng chữ trong văn bản.

* Các quá trình xử lý ảnh:



* + - 1. Các giai đoạn chính trong xử lý ảnh.

Thu nhận ảnh: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình XLA. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

Tiền xử lý: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,v.v… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình XLA. Quá trình này thường được thực hiện bởi các bộ lọc.

Phân đoạn ảnh: phân đoạn ảnh là bước then chốt trong XLA. Giai đoạn này phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng mức xám v.v… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều phần tử khác nhau cấu tạo lên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn, trong khi đa số các ứng dụng chúng ta chỉ cần trích một vài đặc trưng nào đó, do vậy cần có một quá trình để giảm lượng thông tin khổng lồ đó. Quá trình này bao gồm phân vùng ảnh và trích chọn đặc tính chủ yếu.

Tách các đặc tính: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng ảnh dưới dạng biên hay dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu trúc xương của nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

Nhận dạng và giải thích: Đây là bước cuối cùng trong quá trình XLA. Nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn gán nghĩa cho một tập các đối tượng đã được nhận biết.

Chúng ta cũng có thể thấy rằng, không phải bất kỳ một ứng dụng XLA nào cũng bắt buộc phải tuân theo tất cả các bước xử lý đã nêu ở trên, ví dụ như các ứng dụng chỉnh sửa ảnh nghệ thuật chỉ dừng lại ở bước tiền xử lý. Một cách tổng quát thì những chức năng xử lý bao gồm cả nhận dạng và giải thích thường chỉ có mặt trong hệ thống phân tích ảnh tự động hoặc bán tự động, được dùng để rút trích ra những thông tin quan trọng từ ảnh, ví dụ như các ứng dụng nhận dạng ký tự quang học, nhận dạng chữ viết tay v.v…

* Ảnh và biểu diễn ảnh:

Ảnh trong thực tế là một ảnh liên tục cả về không gian và giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính thì cần thiết phải tiến hành số hóa ảnh. Quá trình số hóa biến đổi các tín hiệu liên tục sang tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng tử hóa các thành phần giá trị mà về nguyên tắc bằng mắt thường không thể phân biệt được hai điểm liền kề nhau. Các điểm như vậy được gọi là các pixel (Picture Element) hay các phần tử ảnh hoặc điểm ảnh. Ở đây cần phân biệt khái niệm pixel hay đề cập đến trong các hệ thống đồ họa máy tính. Để tránh nhầm lẫn ta gọi khái niệm pixel này là pixel thiết bị. Khái niệm pixel thiết bị có thể xem xét như sau: khi ta quan sát màn hình (trong chế độ đồ họa), màn hình không liên tục mà gồm các điểm nhỏ, gọi là pixel. Mỗi pixel gồm một tập tọa độ (x, y) và màu.

Như vậy mỗi ảnh là tập hợp các điểm ảnh. Khi được số hóa nó thường được biểu diễn bởi mảng 2 chiều I(n,p): n là dòng và p là cột.

Về mặt toán học có thể xem ảnh là một hàm hai biến f(x,y) với x, y là các biến tọa độ. Giá trị số ở điểm (x,y) tương ứng với giá trị xám hoặc độ sáng của ảnh (x là các cột còn y là các hàng). Giá trị của hàm ảnh f(x,y) được hạn chế trong phạm vi của các số nguyên dương.

0 ≤ f(x,y) ≤ fmax

Với ảnh đen trắng mức xám của ảnh có thể được biểu diễn bởi một số như sau:

Trong đó là đặc tính phổ của cảm biến được sử dụng và k là hệ số tỷ lệ xích. Vì sự cảm nhận độ sáng có tầm quan trọng hàng đầu đối với ảnh đen trắng nên được chọn giống như là hiệu suất sáng tương đối. Vì biểu diễn công suất trên đơn vị diện tích, nên nó bao giờ cũng không âm và hữu hạn.

0≤ f ≤ fmax

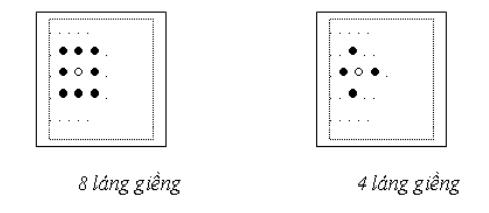
Trong đó là giá trị lớn nhất mà đạt được. Trong xử lý ảnh, được chia thang sao cho nó nằm trong một phạm vi thuận lợi nào đó.

Thông thường đối với ảnh xám, giá trị là bởi vì mỗi phần tử ảnh được mã hóa bởi một byte. Khi quan tâm đến ảnh màu ta có thể mô tả màu qua ba hàm số: thành phần màu đỏ qua R(x,y), thành phần màu lục qua G(x,y) và thành phần màu lam qua B(x,y). Bộ ba giá trị R, G, và B nhận được từ:

Ở đó theo thứ tự là những đặc tính phổ của các cảm biến (bộ lọc) đỏ, lục và lam. cũng không âm và hữu hạn.

Ảnh có thể được biểu diễn theo một trong hai mô hình: mô hình Vector hoặc mô hình Raster

Mô hình Vector: Ngoài mục đích tiết kiệm không gian lưu trữ, dễ dàng hiển thị và in ấn, các ảnh biểu diễn theo mô hình vector còn có ưu điểm cho phép dễ dàng lựa chọn, sao chép, di chuyển, tìm kiếm…Theo những yêu cầu này thì kỹ thuật biểu diễn vector tỏ ra ưu việt hơn. Trong mô hình này, người ta sử dụng hướng vector của các điểm ảnh lân cận để mã hóa và tái tạo lại hình ảnh ban đầu. Các ảnh vector được thu nhận trực tiếp từ các thiết bị số hóa như Digitalize hoặc được chuyển đổi từ các ảnh Raster thông qua các chương trình vector hóa. Mô hình Raster: là mô hình biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận các điểm ảnh. Tùy theo nhu cầu thực tế mà mỗi điểm ảnh có thể được biểu diễn bởi một hay nhiều bit. Mô hình Raster thuận lợi cho việc thu nhận, hiển thị và in ấn. Các ảnh được sử dụng trong phạm vi của đề tài này cũng là các ảnh được biểu diễn theo mô hình Raster. Khi xử lý các ảnh Raster chúng ta có thể quan tâm đến mối quan hệ trong vùng lân cận của các điểm ảnh. Các điểm ảnh có thể xếp hàng trên một lưới (raster) hình vuông, lưới hình lục giác hoặc theo một cách hoàn toàn ngẫu nhiên với nhau.



* + - 1. Quan hệ trong vùng lân cận giữa các điểm ảnh.

Cách sắp xếp theo hình vuông là được quan tâm đến nhiều nhất và có hai loại: điểm 4 láng giềng (4 liền kề) hoặc 8 láng giềng (8 liền kề). Với điểm 4 láng giềng, một điểm ảnh I(i, j) sẽ có điểm kế cận theo 2 hướng i và j; trong khi đó với điểm 8 láng giềng, điểm ảnh I(i, j) sẽ có 4 điểm kế cận theo 2 hướng i, j và 4 điểm kế cận theo hướng chéo .

* Phạm vi ứng dụng của xử lý ảnh:

Xử lý ảnh đã đem lại nhiều ứng dụng trong nhiều lĩnh vực khác nhau: y học, khoa học hình hình sự, khí tƣợng thuỷ văn, quản lý, ... Quản lý là là một trong những ứng dụng quan trọng của xử lý ảnh. Cùng với sự bùng nổ của kinh tế thị trƣờng. Khối lƣợng quản lý càng lớn, nhƣ quản lý hồ sơ, quản lý phiếu điều tra trong công tác thống kê, các câu hỏi trắc nghiệm. Để thực hiện các công việc trên một cách chính xác, nhanh chóng và hiệu quả. Xử lý ảnh và nhận dạng đã nghiên cứu và phát triển mạnh mẽ bài toán nhập liệu tự động

* + 1. Machine learning

Một trong những khác biệt chính giữa con người và máy tính là con người học hỏi từ những kinh nghiệm trong quá khứ, nhưng với máy tính hoặc máy móc cần được phải được thực hiện theo một quy trình có sẵn. Máy tính là những máy logic nghiêm ngặt với ý nghĩa thông thường. Điều đó có nghĩa là nếu chúng ta muốn máy làm điều gì đó, chúng ta phải cung cấp cho nó những quy trình và các hướng dẫn chi tiết, từng bước về chính xác những việc cần làm. Vì vậy, con người đã viết nên các kịch bản và lập trình để máy tính làm theo các hướng dẫn và có khả năng tự học hỏi. Đó là cái cách mà Machine Learning ra đời. Khái niệm máy học chính xác là việc máy tính học hỏi từ dữ liệu trong quá khứ và rút kinh nghiệm qua thời gian.

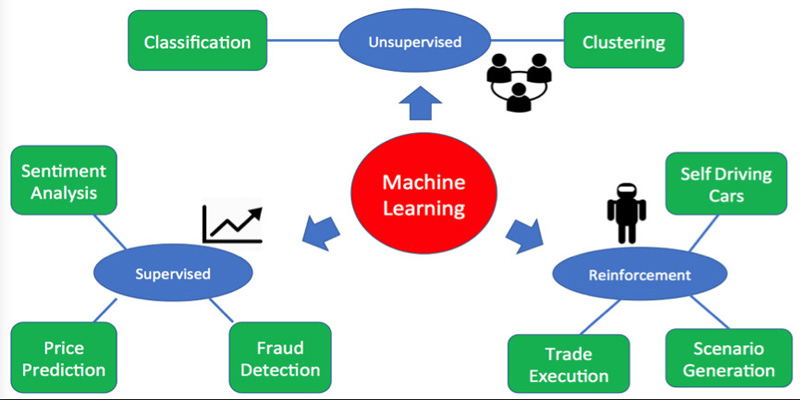
Học máy là một ứng dụng của trí tuệ nhân tạo (AI) cung cấp cho các hệ thống khả năng tự động học hỏi và cải thiện từ kinh nghiệm mà không cần lập trình rõ ràng. Học máy tập trung vào việc phát triển các chương trình máy tính có thể truy cập dữ liệu và sử dụng nó để tự học. Quá trình học bắt đầu bằng các quan sát hoặc dữ liệu. Ví dụ, để tìm kiếm các mẫu trong dữ liệu và đưa ra quyết định tốt hơn trong tương lai dựa trên các ví dụ mà chúng tôi cung cấp. Mục đích chính là cho phép các máy tính tự động học mà không cần sự can thiệp hay trợ giúp của con người và điều chỉnh các hành động tương ứng.

Phương pháp tiếp cận:

Machine learning có liên quan chặt chẽ với thống kê tính toán, do đó, có kiến thức nền tảng về thống kê rất hữu ích để hiểu và tận dụng các thuật toán machine learning.

Đối với những người có thể chưa nghiên cứu thống kê, xác định tương quan và hồi quy có thể hữu ích đầu tiên, vì chúng là các kỹ thuật thường được sử dụng để nghiên cứu mối quan hệ giữa các biến định lượng. **Correlation** là thước đo liên kết giữa hai biến không được chỉ định là phụ thuộc hay độc lập. **Regression** ở mức cơ bản được sử dụng để kiểm tra mối quan hệ giữa một biến phụ thuộc và một biến độc lập. Bởi vì thống kê hồi quy có thể được sử dụng để dự đoán biến phụ thuộc khi biết biến độc lập, hồi quy cho phép khả năng dự đoán.

Các thuật toán trong machine learning thường được chia làm 3 loại:



* + - 1. Thuật toán trong machine learning
* Supervised Learning (Học có giám sát):

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào và một tập hợp nhãn tương ứng , trong đó là các vector. Các cặp dữ liệu biết trước được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập *training data* này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập sang một phần tử (xấp xỉ) tương ứng của tập .

Mục đích là tìm ra hàm thật tốt để khi có một dữ liệu mới, chúng ta có thể tính được nhãn tương ứng của nó với công thức:

* Unsupervised Learning (Học không giám sát):

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào  mà không biết nhãn tương ứng.

Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm không giám sát được đặt tên theo nghĩa này.

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

+ Clustering (phân nhóm)

Một bài toán phân nhóm toàn bộ dữ liệu XX thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

+ Association

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

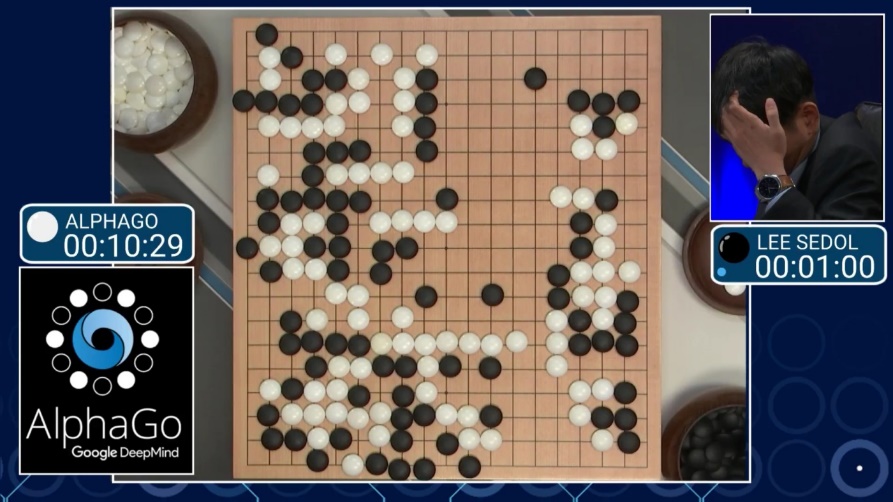
* Semi-Supervised Learning (Học bán giám sát):

Các bài toán khi chúng ta có một lượng lớn dữ liệu  nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.

* Reinforcement Learning (Học củng cố):

Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.



* + - 1. AlphaGo chơi cờ vây với Lee Sedol.

Quy trình xây dựng mô hình machine learning:

* **Thu thập dữ liệu:** Thu thập dữ liệu để mô hình học.
* **Chuẩn bị dữ liệu:** Xử lý và đưa dữ liệu về định dạng tối ưu, trích chọn [đặc trưng](https://trituenhantao.io/tu-dien-thuat-ngu/dac-trung/" \t "_blank) hoặc giảm chiều dữ liệu.
* **Huấn luyện:** Tại pha này, thuật toán [machine learning](https://trituenhantao.io/tu-dien-thuat-ngu/machine-learning/) thực hiện việc học thông qua các ví dụ đã được thu thập và chuẩn bị từ hai bước trên.
* **Đánh giá:** Kiểm thử mô hình để đánh giá xem chất lượng của mô hình tốt đến đâu.
* **Tinh chỉnh: Tinh chỉnh mô hình để tối ưu hiệu quả.**

Một số ứng dụng của machine learning hiện nay:

* Tự động phân loại:

Phân loại tin tức là một ứng dụng điểm chuẩn khác của phương pháp học máy.Vận dụng như thế nào? Như một vấn đề thực tế là bây giờ khối lượng thông tin đã tăng lên rất nhiều trên web. Tuy nhiên, mỗi người có sở thích hoặc lựa chọn cá nhân của mình. Vì vậy, để chọn hoặc thu thập một phần thông tin phù hợp trở thành một thách thức đối với người dùng từ vô số nội dung trên trang web.

Phân loại các danh mục một cách rõ ràng, dễ điều hướng giúp cho các khách hàng mục tiêu chắc chắn sẽ tăng khả năng truy cập các trang tin tức. Hơn nữa, độc giả hoặc người dùng có thể tìm kiếm tin tức cụ thể một cách hiệu quả và nhanh chóng.

Có một số phương pháp học máy trong mục đích này, tức là, máy vectơ hỗ trợ, naive Bayes, k-nearest neighbor, v.v.

* Ứng dụng trong các mạng xã hội:

Học máy đang được sử dụng trong một loạt các ứng dụng ngày nay. Một trong những ví dụ nổi tiếng nhất là Facebook News Feed. Nguồn cấp tin tức sử dụng học máy để cá nhân hóa từng nguồn cấp dữ liệu thành viên.

 Nếu một thành viên thường xuyên dừng lại để đọc hoặc thích một bài đăng của một người bạn cụ thể, News Feed sẽ bắt đầu hiển thị nhiều hơn về hoạt động của người bạn đó trước đó trong nguồn cấp dữ liệu.

Đằng sau hệ thống ấy, phần mềm sử dụng phân tích thống kê và phân tích dự đoán để xác định các mẫu trong dữ liệu người dùng và sử dụng các mẫu đó để điền vào News Feed. Nếu thành viên không còn dừng lại để đọc, thích hoặc bình luận trên các bài đăng của bạn bè, dữ liệu mới đó sẽ được bao gồm trong tập dữ liệu và News Feed sẽ điều chỉnh tương ứng.

Không chỉ riêng facebook, ta có thể bắt gặp những tính năng tương tự đó qua các mạng xã hội khác như google, instagram,....

* Nhận diện hình ảnh:

Nhận dạng hình ảnh là một trong những ví dụ về máy học và trí tuệ nhân tạo phổ biến nhất. Về cơ bản, nó là một cách tiếp cận để xác định và phát hiện các đặc trưng của một đối tượng trong hình ảnh kỹ thuật số. Hơn nữa, kỹ thuật này có thể được sử dụng để phân tích sâu hơn, chẳng hạn như nhận dạng mẫu, nhận diện hình khuôn, nhận dạng khuôn mặt, nhận dạng ký tự quang học và nhiều hơn nữa,...

* + 1. Deep learning

Deep Learning là tập hợp con của Machine Learning và nó có tác dụng hỗ trợ cho máy tính tự huấn luyện chính nó để có thể thực hiện mọi tác vụ tương tự như con người. Điều này chính là giúp máy tính bắt chước con người cách học hỏi và suy nghĩ. Các hệ thống của Deep Learning có khả năng cải thiện được những hiệu suất  của chúng với quyền truy cập vào dữ liệu sẽ được nhiều hơn. Thông thường, phiên bản máy sẽ có nhiều kinh nghiệm hơn; những máy móc đã có đủ kinh nghiệm thì sẽ được mang đi phục vụ cho những công việc như: lái xe, phát hiện cỏ dại,... Deep Learning có hỗ trợ cho việc dịch ngôn ngữ, phân loại các hình ảnh, nhận dạng giọng nói. Chính vì thế, nó có thể được ứng dụng để giải quyết mọi nhu cầu cần nhận dạng mẫu mà không cần đến sự can thiệp của con người.

Bộ phận mạg thần kinh nhân tạo này giống với não người bao gồm: Các node (nó là đơn vị thần kinh trong mạng thần kinh nhân tạo) là những neuron thần kinh. Hầu hết các node bản thân thường chỉ có khả năng trả lời cho những câu hỏi  đơn giản và cơ bản nhất; với những tác vụ khó thì chúng sẽ tiến hành liên kết với nhau để trả lời.

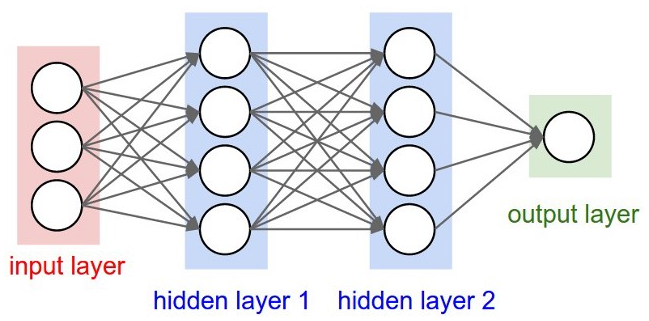
Bạn có thể dạy hoặc có thể huấn luyện chúng bằng những  thuật toán cụ thể. Những node giải đáp các câu hỏi phức tạp thì sẽ được gọi là mạng lưới thần kinh sâu (DNN), nó được định nghĩa như sau: Mạng lưới thần kinh sâu có khả năng thực hiện các hoạt động phức tạp như biểu diễn, trừu tượng có mang các ý nghĩa về âm thanh, văn bản và hình ảnh. Chúng được đánh giá là lĩnh vực phát triển nhất trong Machine Learning.

Cách thức hoạt động của deeplearning:

Để hiểu cách hoạt động của Deep Learning, chúng ta sẽ build giả thiết là dịch vụ ước tính giá vé máy bay, train nó bằng method supervised learning (phương thức học có giám sát).

Chúng ta muốn có 1 công cụ để dự đoán giá vé bằng cách sử dụng các input sau (để đơn giản, chúng ta sẽ loại bỏ các vé khứ hồi): Sân bay khởi hành, sân bay đến, ngày khởi hành, tên hãng máy bay.

Cũng giống như động vật, bộ não của AI dự đoán này có các nơ-ron. Các nơ ron này được thể hiện bằng các vòng tròn và những vòng tròn này lại kết nối với nhau.



* + - 1. Cách thức hoạt động của deeplearning.

Những nơ-ron này được nhóm lại thành 3 loại layer khác nhau: Layer Input, Layer(s) Hidden, Layer Output

**Layer input**nhận các input data. Trong trường hợp này, chúng ta có 4 nơ-ron trong layer inout: Sân bay khởi hành, Sân bay đến, Ngày khởi hành và Hãng máy bay. Layer input sẽ đưa các inputs này đến layer hidden đầu tiên.

Các **hidden layers** thực hiện các tính toán toán học dựa trên outputs của chúng ta. Một trong những thách thức khi tạo các neural networks chính là việc quyết định số lượng các hidden layers và số lượng các nơ-ron cho mỗi layer.

**Layer output**trả về data output mà trường hợp này là dự đoán về giá vé máy bay.



* + - 1. Hình mô phỏng sử dụng deeplearning dự đoán giá vé máy bay.

Mỗi kết nối giữa các nơ-ron được liên kết với 1 **weight**. Weight này thể hiện tầm quan trọng của giá trị input. Các weight ban đầu được set ngẫu nhiên.

Khi dự đoán giá của 1 tấm vé máy bay, ngày khởi hành là 1 trong những yếu tố có sức “nặng” hơn. Vì vậy, các liên kết nơ-ron của ngày khởi hành sẽ có weight (trọng lượng) lớn.

Mỗi nơ-ron có 1 [Activation Function](https://en.wikipedia.org/wiki/Activation_function), để đơn giản, 1 trong những mục đích chính của nó là “standardize” – “tiêu chuẩn hóa” output từ nơ-ron. Một khi 1 set của data input đã vượt qua tất cacr các layers của neural network, nó sẽ trả lại data output qua layer output.

Cách huấn luyện Neural Network:

Bạn cần **1 set dữ liệu đủ lớn** và lượng lớn **computational power**. Đối việc dự đoán giá vé máy bay, chúng ta cần có data lịch sử của các giá vé máy bay. Và lượng lớn các mối liên hệ tiềm năng giữa sân bay và ngày khởi hành, chúng ta cần 1 danh sách rất lớn các giá vé.

Để train AI, chúng ta gần đưa cho nó các inputs từ data set, và so sánh các outputs của nó với các output từ data set. Vì AI vẫn chưa được train, output của nó sẽ bị sai.

Một khi đã xem xét toàn bộ data set, chúng ta có thể tạo hàm để hiển thị các output của AI đã sai khác như thế nào so với các output thực sự. Hàm này được gọi là [Cost Function](https://en.wikipedia.org/wiki/Loss_function).

Lý tưởng nhất, bạn sẽ muốn hàm cost này bằng 0. Đó là khi các output của AI hoàn toàn giống với các output của data set. Để giảm hàm cost này, chúng ta sẽ sử dụng kĩ thuật Gradient Descent. Là 1 kĩ thuật cho phép chúng ta tìm thấy phần tối thiểu của 1 hàm. Trong trường hợp này là phần tối thiểu (minimum) của cost function. Gradient Descent **sẽ thay đổi các weights**theo cấp độ tăng nhỏ **sau khi mỗi lần lặp đi lặp lại data set.**Bằng cách tính toán phái sinh (hoặc độ dốc) của cost function ở 1 set weight nào đó, chúng ta có thể nhận thấy được khuynh hướng của minimum.

Bạn cần phải lặp đi lặp lại các data set nhiều lần. Đó là lý do tại sao bạn cần 1 lượng computational power, cập nhật các weights bằng gradient descent sẽ được thực hiện **1 cách tự động.**Đây chính là điều kì diệu của Deep Learning. Một khi đã train cho công cụ AI đo lường giá vé máy bay, chúng ta có thể sử dụng nó để dự đoán giá vé trong tương lai.

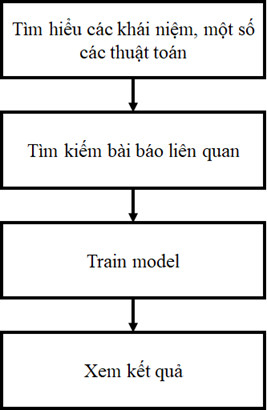
1. MÔ HÌNH BÀI TOÁN
   1. Quy trình thực hiện

Bước 1: Nắm kiến thức Python cơ bản, tìm hiểu các thư viện Keras, OpenCV, Tensorflow,..và các thuật toán liên quan đến marchine learning, deep learning.

Bước 2: Tìm kiếm các bài báo và các khái niệm liên quan đến đề tài.

Bước 3: Train model với bộ dữ liệu có sẵn và bộ dữ liệu tự thu thập.

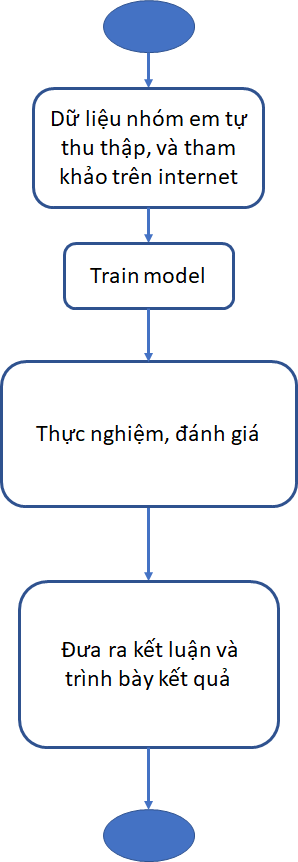
Bước 4: Chạy chương trình và xem kết quả.



* + - 1. Quy trình thực hiện
  1. Mô hình bài toán

Nhóm em tham khảo dữ liệu trên internet và tự thu thập dữ liệu. Sau đó sử dụng bộ dữ liệu có được để train ra 2 file model. Tiếp theo, nhóm em sử dụng 2 file model đó vào chương trình chạy đã chuẩn bị trước để xem và đánh giá model được train từ bộ dữ liệu như thế nào sẽ tốt hơn từ đó đưa ra kết luận và trình bày.

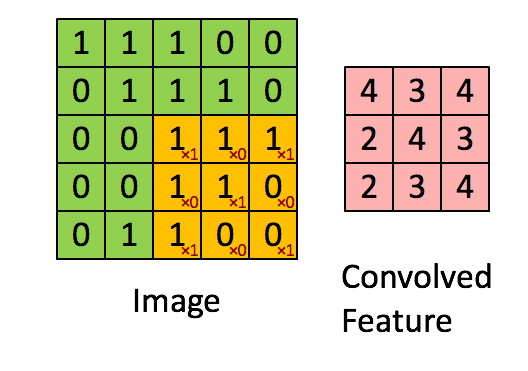
Mô hình bài toán xử lý được biểu diễn như sau:



* + - 1. Mô hình bài toán
    1. Train model
* Xây dựng trình phân tích cú pháp
* Xây dựng trình tạo hình ảnh đào tạo
* Xây dựng phần đầu của mô hình, phần này sẽ được đặt trên phần mô hình cơ sở.
* Vòng lặp các phần trong mô hình cơ sở, và bảo đảm chúng không được cập nhật trong lần train đầu tiên
* Biên dịch mô hình
* Đưa ra dự đoán về bộ thử nghiệm
* Lưu model với định dạng là h5.
  + 1. Thuật toán được sử dụng

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh.

Convolutional là một cửa sổ trượt (Sliding Windows) trên một ma trận như mô tả hình dưới:



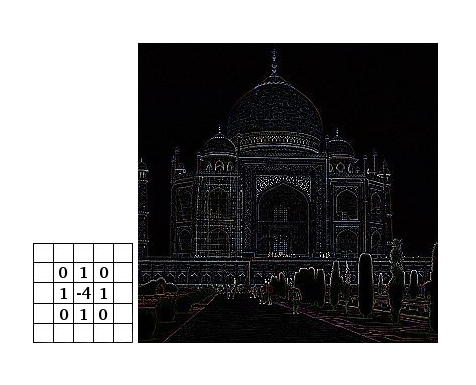
* + - 1. Ma trận mô tả Convolutional

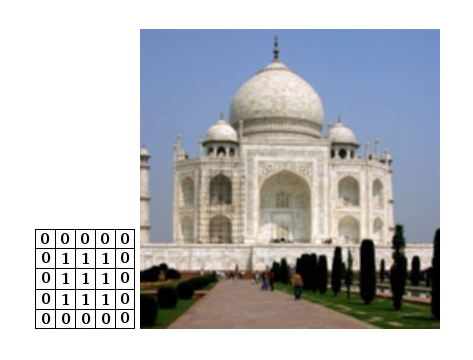
Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Trong hình ảnh ví dụ trên, ma trận bên trái là một hình ảnh trắng đen được số hóa. Ma trận có kích thước 5×5 và mỗi điểm ảnh có giá trị 1 hoặc 0 là giao điểm của dòng và cột.

Convolution hay tích chập là nhân từng phần tử trong ma trận 3. Sliding Window hay còn gọi là kernel, filter hoặc feature detect là một ma trận có kích thước nhỏ như trong ví dụ trên là 3×3.

Convolution hay tích chập là nhân từng phần tử bên trong ma trận 3×3 với ma trận bên trái. Kết quả được một ma trận gọi là Convoled feature được sinh ra từ việc nhận ma trận Filter với ma trận ảnh 5×5 bên trái.





* + - 1. Convolution của ảnh.

Cấu trúc mạng CNN:

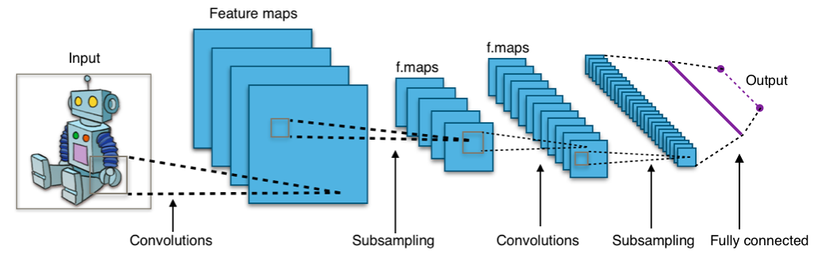
Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (traning) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

* + - 1. Cấu trúc mạng CNN.

Trong mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các gốc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

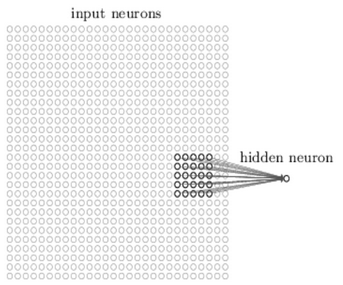
Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling). Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter.

Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên.

Mạng CNN sử dụng 3 ý tưởng cơ bản: các trường tiếp nhận cục bộ (local receptive field), trọng số chia sẻ (shared weights), tổng hợp (pooling).

Trường tiếp nhận cục bộ(local receptive field): Đầu vào của mạng CNN là một ảnh. Ví dụ như ảnh có kích thước 28×28 thì tương ứng đầu vào là một ma trận có 28×28 và giá trị mỗi điểm ảnh là một ô trong ma trận. Trong mô hình mạng ANN truyền thống thì chúng ta sẽ kết nối các neuron đầu vào vào tầng ảnh.

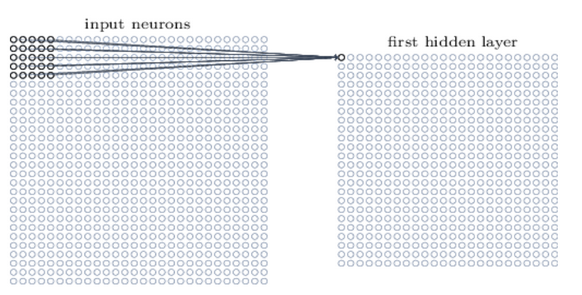
Tuy nhiên trong CNN chúng ta không làm như vậy mà chúng ta chỉ kết nối trong một vùng nhỏ của các neuron đầu vào như một filter có kích thước 5×5 tương ứng (28- 5 + 1) 24 điểm ảnh đầu vào. Mỗi một kết nối sẽ học một trọng số và mỗi neuron ẩn sẽ học một bias. Mỗi một vùng 5×5 đấy gọi là một trường tiếp nhận cục bộ.



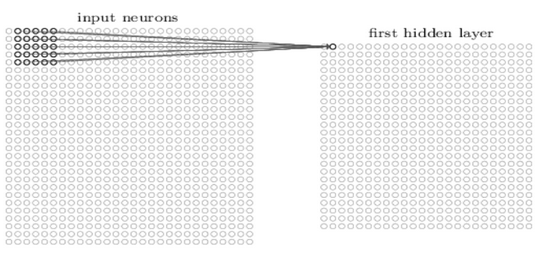
* + - 1. Trường tiếp nhận cục bộ (local receptive field).

Một cách tổng quan, ta có thể tóm tắt các bước tạo ra 1 hidden layer bằng các cách sau:

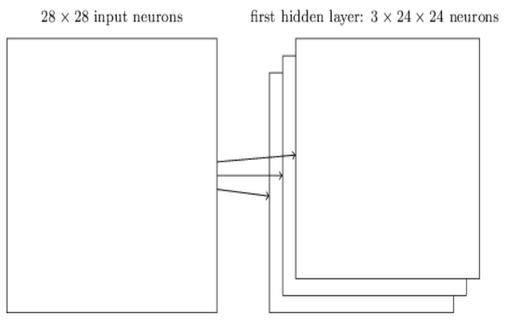
* **Tạo ra neuron ẩn đầu tiên trong lớp ẩn 1**



* **Dịch filter qua bên phải một cột sẽ tạo được neuron ẩn thứ 2.**



Với bài toán nhận dạng ảnh người ta thường gọi ma trận lớp đầu vào là feature map, trọng số xác định các đặc trương là shared weight và độ lệch xác định một feature map là shared bias. Như vậy đơn giản nhất là qua các bước trên chúng ta chỉ có 1 feature map. Tuy nhiên trong nhận dạng ảnh chúng ta cần nhiều hơn một feature map.



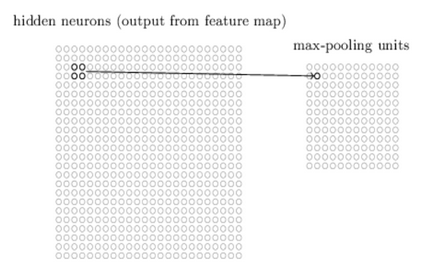
Như vậy, local receptive field thích hợp cho việc phân tách dữ liệu ảnh, giúp chọn ra những vùng ảnh có giá trị nhất cho việc đánh giá phân lớp.

Trọng số chia sẻ (Shared weight and bias): Đầu tiên, các trọng số cho mỗi filter (kernel) phải giống nhau. Tất cả các nơ-ron trong lớp ẩn đầu sẽ phát hiện chính xác feature tương tự chỉ ở các vị trí khác nhau trong hình ảnh đầu vào. Chúng ta gọi việc map từ input layer sang hidden layer là một feature map.

Một convolutional layer bao gồm các feature map khác nhau. Mỗi một feature map giúp detect một vài feature trong bức ảnh. Lợi ích lớn nhất của trọng số chia sẻ là giảm tối đa số lượng tham số trong mạng CNN.

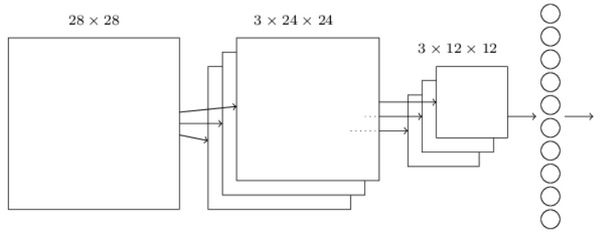
Lớp tổng hợp (pooling layer): Lớp pooling thường được sử dụng ngay sau lớp convulational để đơn giản hóa thông tin đầu ra để giảm bớt số lượng neuron.

Thủ tục pooling phổ biến là max-pooling, thủ tục này chọn giá trị lớn nhất trong vùng đầu vào 2×2.



Như vậy qua lớp Max Pooling thì số lượng neuron giảm đi phân nửa. Trong một mạng CNN có nhiều Feature Map nên mỗi Feature Map chúng ta sẽ cho mỗi Max Pooling khác nhau. Chúng ta có thể thấy rằng Max Pooling là cách hỏi xem trong các đặc trưng này thì đặc trưng nào là đặc trưng nhất. Ngoài Max Pooling còn có L2 Pooling.

Cuối cùng ta đặt tất cả các lớp lại với nhau thành một CNN với đầu ra gồm các neuron với số lượng tùy bài toán.



2 lớp cuối cùng của các kết nối trong mạng là một lớp đầy đủ kết nối (fully connected layer) . Lớp này nối mọi nơron từ lớp max pooled tới mọi nơron của tầng ra.

Cách chọn tham số cho CNN:

* Số các convolution layer: càng nhiều các convolution layer thì performance càng được cải thiện. Sau khoảng 3 hoặc 4 layer, các tác động được giảm một cách đáng kể
* Filter size: thường filter theo size 5×5 hoặc 3×3
* Pooling size: thường là 2×2 hoặc 4×4 cho ảnh đầu vào lớn
* Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.
  + 1. Test model
* Tải model được lưu sau khi train xong
* Lọc ra các phát hiện và đảm bảo độ tin cậy là lớn hơn độ tin cậy tối thiểu, sau đó tính toạ độ (x, y) cho đối tượng và đảm bảo các hộp giới hạn nằm trong kích thước của khung. Tiếp theo, trích xuất khuôn mặt để đổi màu, và thay đổi kích thước và xử lý trước.
* Đưa khuôn mặt qua mô hình để xác định xem là đang nhắm mắt hay đang mở mắt.
* Xác định nhãn và màu sắc dùng để vẽ hộp giới hạn và văn bản, và xác suất
* Hiển thị nhãn và hình chữ nhật ra output
* Kiểm tra nếu mắt nhắm trong khoảng thời gian nhất định thì sẽ phát còi báo hiệu.

1. THỰC NGHIỆM VÀ ĐÁNH GIÁ
   1. Chuẩn bị
      1. Cấu hình máy

HĐH: windows 10 Home Single Language 64-bit (10.0, Build 19044).

CPU: Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz, ~ 2.5GHz

RAM: 8GB

SSD: 512GB

GPU: GTX 1650 4GB

* + 1. Phần mềm

Visual Studio Code

NVIDIA Graphics Driver 469.49

Python 3.6.8(64-bit)

Python Launcher

* + 1. Ngôn ngữ lập trình

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990).

Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o). Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc. Python hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl). Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.[[28]](https://vi.wikipedia.org/wiki/Python_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)#cite_note-28) Ban đầu, Python được phát triển để chạy trên nền [Unix](https://vi.wikipedia.org/wiki/Unix). Nhưng rồi theo thời gian, Python dần mở rộng sang mọi [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) từ [MS-DOS](https://vi.wikipedia.org/wiki/MS-DOS) đến [Mac OS](https://vi.wikipedia.org/wiki/Mac_OS), [OS/2](https://vi.wikipedia.org/wiki/OS/2), [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [các hệ điều hành khác thuộc họ Unix](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%E1%BB%B1_Unix). Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python. Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

Ưu điểm của Python:

Có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linix và các OS khác thuộc họ Unix.

Tương thích mạnh mẽ với Unix, hardware, thirt-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng)

[Python](http://itplus-academy.edu.vn/Khoa-hoc-lap-trinh-Python-PYTHON-FOR-EVERYBODY.html) với tốc độ xử lý cực nhanh, python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Biender 3D

* Dễ đọc và dễ học:

Python là một ngôn ngữ có hình thức sáng sủa, cấu trúc rõ rang, cú pháp ngắn gọn. Nó không có cú pháp phức tạp như các ngôn ngữ khác như C hoặc C ++. Nhờ ít phức tạp hơn, Python cho phép bạn suy nghĩ rõ ràng hơn và tập trung vào việc xây dựng logic.

* Khả năng ứng dụng rộng rãi:

Một tính năng thiết yếu khác của ngôn ngữ này là nó có thể áp dụng rộng rãi. Các kỹ sư, nhà khoa học và nhà toán học sử dụng rộng rãi nó.

* Quản lý bộ nhớ:

Python có một thư viện rộng lớn với khả năng quản lý bộ nhớ, điều này làm cho nó nổi bật so với các ngôn ngữ lập trình khác. Nó bao gồm một heap riêng chứa tất cả các đối tượng và cấu trúc dữ liệu Python, một trình quản lý bộ nhớ tích hợp để duy trì heap riêng tư này.

* Đơn giản và nhanh chóng:

Cộng đồng Python cung cấp hỗ trợ nhanh chóng và thiết thực cho người dùng cũng như khả năng thích ứng nhanh của mã. Một số chuyên gia thích đặt biệt danh cho Python là “ngôn ngữ sẵn sàng để chạy” vì nó chỉ yêu cầu mã đơn giản để được thực thi. Nâng cao và kiểm tra mã thoải mái hơn nhiều với Python.

* Mã hoá không đồng bộ:

Mã hóa không đồng bộ sử dụng một vòng lặp sự kiện duy nhất để hoàn thành công việc trong những khoảng thời gian nhỏ. Python rất hữu ích để viết mã không đồng bộ vì nó dễ viết và dễ bảo trì. Nó không yêu cầu bất kỳ nội dung nghiên cứu phức tạp, bế tắc hoặc bất kỳ sự phức tạp nào khác.

* Tích hợp với các ngôn ngữ khác:

Python có các thư viện như Cython và Jython, cho phép tích hợp với các ngôn ngữ khác như C, C ++ và Java để phát triển đa nền tảng. Đây là một trong những đặc quyền chính của Python vì không có ngôn ngữ nào là hoàn hảo và đôi khi sự phát triển đòi hỏi các chức năng ngôn ngữ đa dạng.

Nhược điểm của Python:

Python không có các thuộc tính như :protected,private hay public, không có vòng lặp do…while và switch….case.

* Khó kiểm tra:

Vì nó là một ngôn ngữ dựa trên trình thông dịch, rất khó để chạy các bài kiểm tra trên mã được viết bằng Python. Tất cả các lỗi chỉ xuất hiện trong thời gian chạy, điều này khiến việc kiểm tra các đoạn mã được viết bằng Python rất khó khăn.

* Tốc độ thực thi chậm:

Python là một ngôn ngữ thông dịch, có nghĩa là nó hoạt động với trình thông dịch, không phải với trình biên dịch. Do đó, nó thực thi tương đối chậm hơn C, C ++, Java và nhiều ngôn ngữ khác.

* Tiêu thụ bộ nhớ lớn:

Các cấu trúc của Python đòi hỏi nhiều không gian bộ nhớ hơn. Ngôn ngữ này không thích hợp để sử dụng cho sự phát triển trong điều kiện bộ nhớ hạn chế.

* Phát hiện lỗi trong mã:

Vì Python được thực thi thông qua trình thông dịch thay vì trình biên dịch, nên không thể phát hiện lỗi trong quá trình biên dịch và điều đó không tốt cho các nhà phát triển.

* + 1. Các thư viện được sử dụng

Các Thư viện mã nguồn mở OpenCV, Keras, Tensorflow, Numpy, Matplotlib.

OpenCV là dự án bắt đầu tại hãng Intel vào năm 1999 bởi Gary Bradsky và ra mắt lần đầu tiên vào năm 2000. Sau đó Vadim Pisarevsky gia nhập và quản lý nhóm. Vào năm 2005, OpenCV được sử dụng trên xe tự lái Stanley và chiếc này đã vô địch giải đấu 2005 DARPA Grand. Tiếp theo nó tiếp tục được cải tiến và phát triển dưới sự hỗ trợ của Willow Garage bên cạnh với Gary Bradsky và Vadim Pisarevsky. Hiện tại OpenCV hỗ trợ vô số các thuật toán liên quan để lĩnh vực thị giác máy tính (computer vision) và lĩnh vực học máy (machine learning).



* + - 1. OpenCV.

OpenCV được phát hành theo giấy phép BSD (\*), do đó nó miễn phí cho cả học tập và sử dụng với mục đích thương mại. Nó có trên các giao diện C++, C, Python và Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để hỗ trợ hiệu quả về tính toán và chuyên dùng cho các ứng dụng real-time (thời gian thực). Nếu được viết trên C/C++ tối ưu, thư viện này có thể tận dụng được bộ xử lý đa lõi (multi-core processing).

OpenCV có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ. OpenCV hiện được ứng dụng rộng rãi toàn cầu, với cộng đồng hơn 47.000 người, với nhiều mục đích và tính năng khác nhau từ interactive art, đến khai thác mỏ, khai thác web map hoặc qua robotic cao cấp.



* + - 1. Keras.

Keras chạy trên các thư viện máy mã nguồn mở như TensorFlow, Theano hoặc Bộ công cụ nhận thức (CNTK). Theano là một thư viện python được sử dụng cho các tác vụ tính toán số nhanh. TensorFlow là thư viện toán học biểu tượng nổi tiếng nhất được sử dụng để tạo mạng nơ-ron và mô hình học sâu. TensorFlow rất linh hoạt và lợi ích chính là tính toán phân tán. CNTK là khung học sâu được phát triển bởi Microsoft. Nó sử dụng các thư viện như Python, C #, C ++ hoặc các bộ công cụ học máy độc lập. Theano và TensorFlow là những thư viện rất mạnh nhưng khó hiểu để tạo mạng nơ-ron.

Keras dựa trên cấu trúc tối thiểu, cung cấp một cách dễ dàng và dễ dàng để tạo các mô hình học sâu dựa trên TensorFlow hoặc Theano. Keras được thiết kế để xác định nhanh các mô hình học sâu. Chà, Keras là một lựa chọn tối ưu cho các ứng dụng học sâu.



* + - 1. TensorFlow.

Vài năm trước, khi phải xử lý lượng dữ liệu khổng lồ, deep learning bắt đầu cho thấy hiệu năng vượt trội so với tất cả các thuật toán machine learning khác. Google sớm nhận ra tiềm năng này và nghĩ rằng họ nên sử dụng deep neural network để cải thiện các dịch vụ của mình, trong đó có:

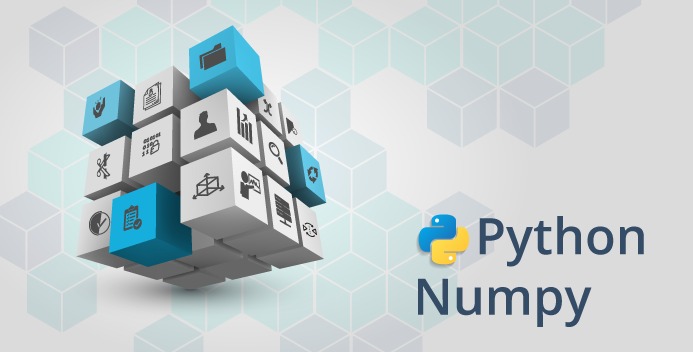
– Gmail.

– Hình ảnh

– Google search engine

Thế là họ dựng 1 framework có tên là TensorFlow để các nhà nghiên cứu cũng như lập trình viên có thể làm việc cùng nhau trên model A.I. 1 khi đã được phát triển và scale hoàn chỉnh, rất nhiều người đã có thể sử dụng được nó.

Ra mắt lần đầu vào cuối năm 2015, phiên bản TensorFlow ổn định cuối cùng cũng xuất hiện vào năm 2017. Là mã nguồn mở dưới sự cho phép của Apache Open Source, giờ đây bạn có thể sử dụng, điều chỉnh và tái đóng góp phiên bản được điều chỉnh đó, đổi lại không cần phải trả bất cứ đồng nào cho Google.



* + - 1. Numpy.

NumPy hay Numeric Python là thư viện lõi phục vụ cho khoa học máy tính của Python. Nó cung cấp một đối tượng mảng đa chiều hiệu suất cao và các công cụ để làm việc với các mảng này.

NumPy chứa các tính năng khác nhau bao gồm những tính năng quan trọng sau:

* Đối tượng mảng N-chiều mạnh mẽ
* Các chức năng broadcasting
* Phép biến đổi Fourier, khả năng số ngẫu nhiên
* Các công cụ để tích hợp mã C / C ++ và Fortran.

Bên cạnh những công dụng khoa học rõ ràng, NumPy cũng có thể được sử dụng như một nơi chứa dữ liệu chung đa chiều hiệu quả. Các kiểu dữ liệu tùy ý có thể được xác định bằng cách sử dụng NumPy, cho phép NumPy tích hợp liền mạch và nhanh chóng với nhiều loại cơ sở dữ liệu.



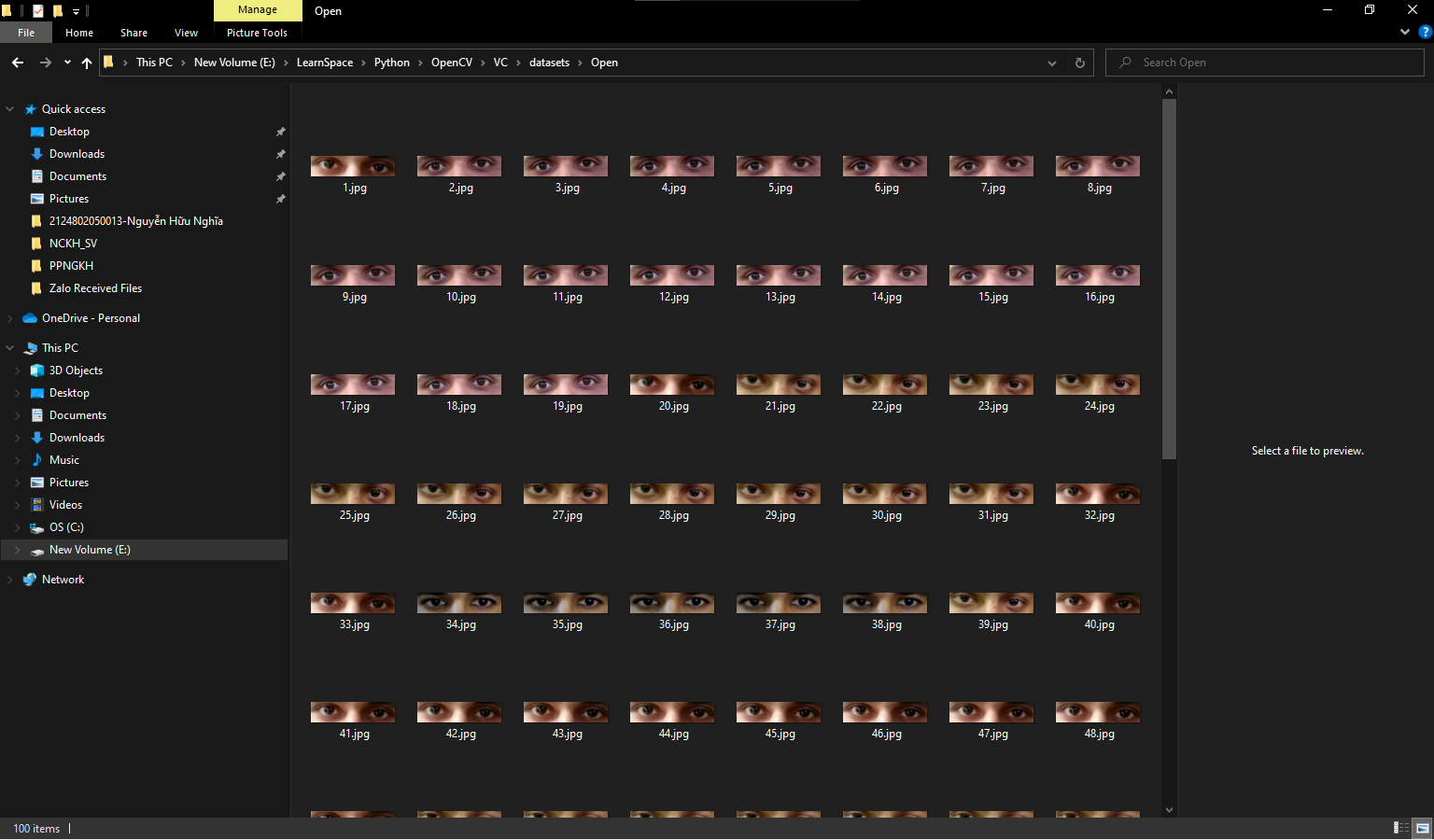
* + - 1. Matplotlib.

Như chúng ta đã biết Python được sử dụng nhiều nhất trong lĩnh vực phân tích dữ liệu, mà trong khoa học dữ liệu, việc trực quan hóa thông qua các đồ thị, biểu đồ giúp cho chúng ta hiểu được các mối quan hệ trong dữ liệu dễ dàng hơn rất nhiều. Matplotlib là một thư viện sử dụng để vẽ các đồ thị trong Python, chính vì vậy nó là thư viện cực phổ biến của Python. Bạn thử tưởng tượng một file dữ liệu khoảng 20MB, khi vẽ ra đồ thị từ dữ liệu này bạn sướng như phát điên vì có thể hiểu được ngay 20MB đó nói lên cái gì? Ngày nay, khi Big data đang trở thành thực tế, hàng ngày có quá nhiều dữ liệu, việc trực quan hóa dữ liệu càng trở nên cấp thiết và càng thúc đẩy những thư viện như Matplotlib phát triển hơn.

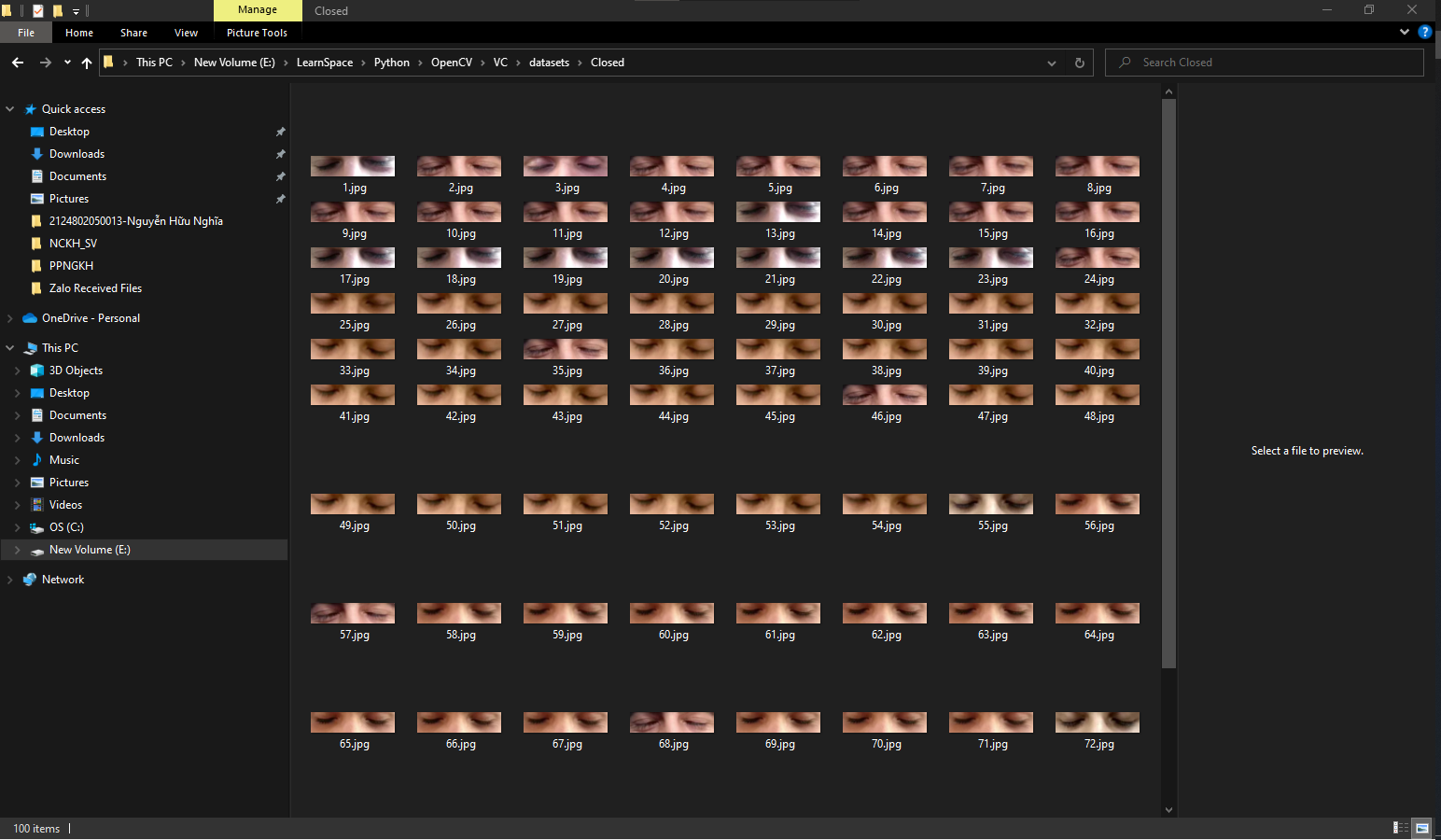
* + 1. Mô tả bộ dữ liệu

Bộ dữ liệu chúng em tham khảo được từ trang web stackoverflow bao gồm:

Một folder dataset chứa 100 tấm hình màu đôi mắt khi đang nhắm và 100 tấm hình màu đôi mắt khi đang mở được chia ra thành 2 folder con là Open và Closed.



* + - 1. Folder Open trong dataset.

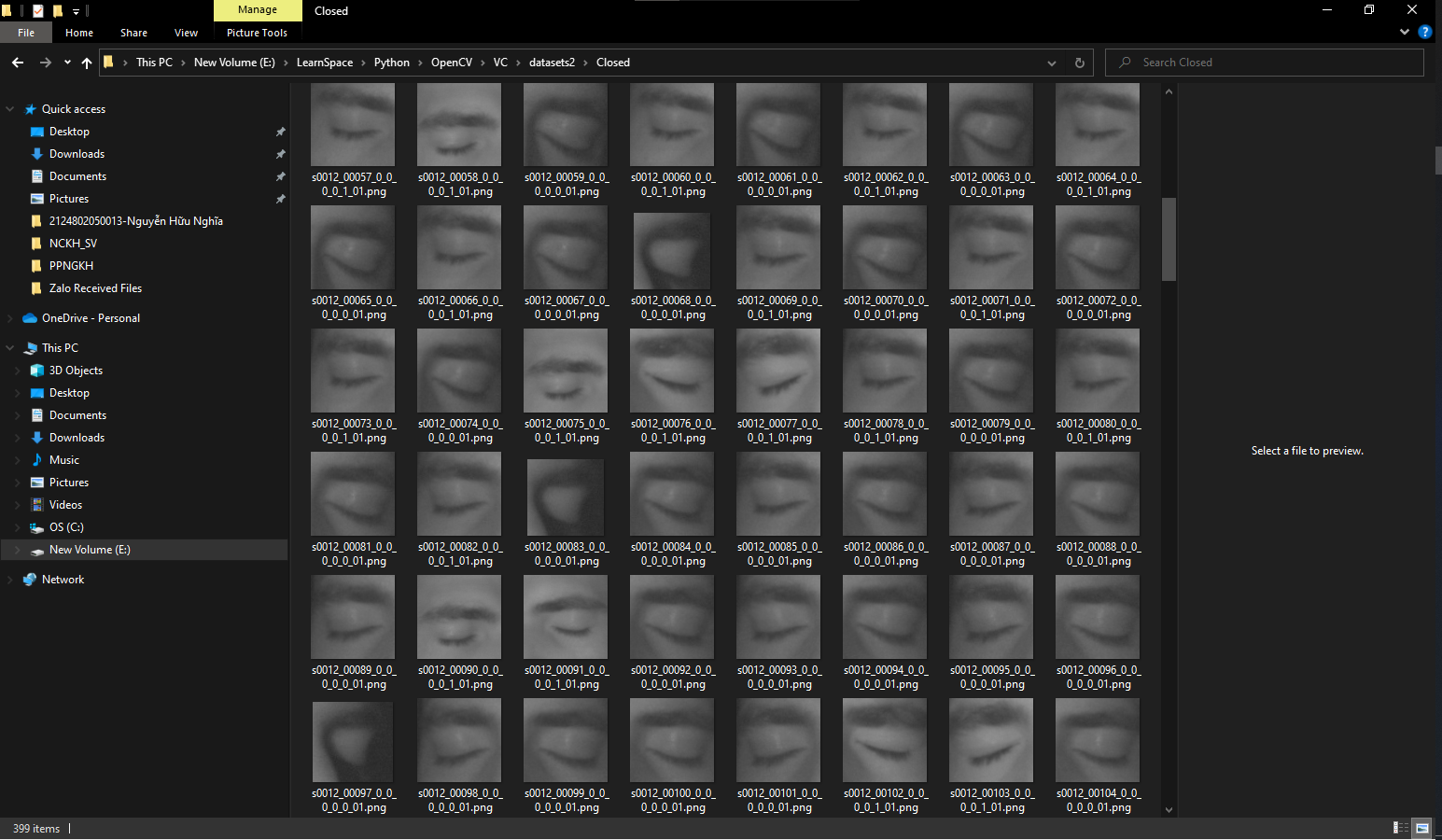


* + - 1. Folder Closed trong dataset.

Một folder dataset2 chứa 399 tấm hình trắng đen từng con mắt nhắm, và 399 tấm hình trắng đen từng con mắt mở ở những góc độ khác nhau được chia thành 2 folder con là Open và Closed.

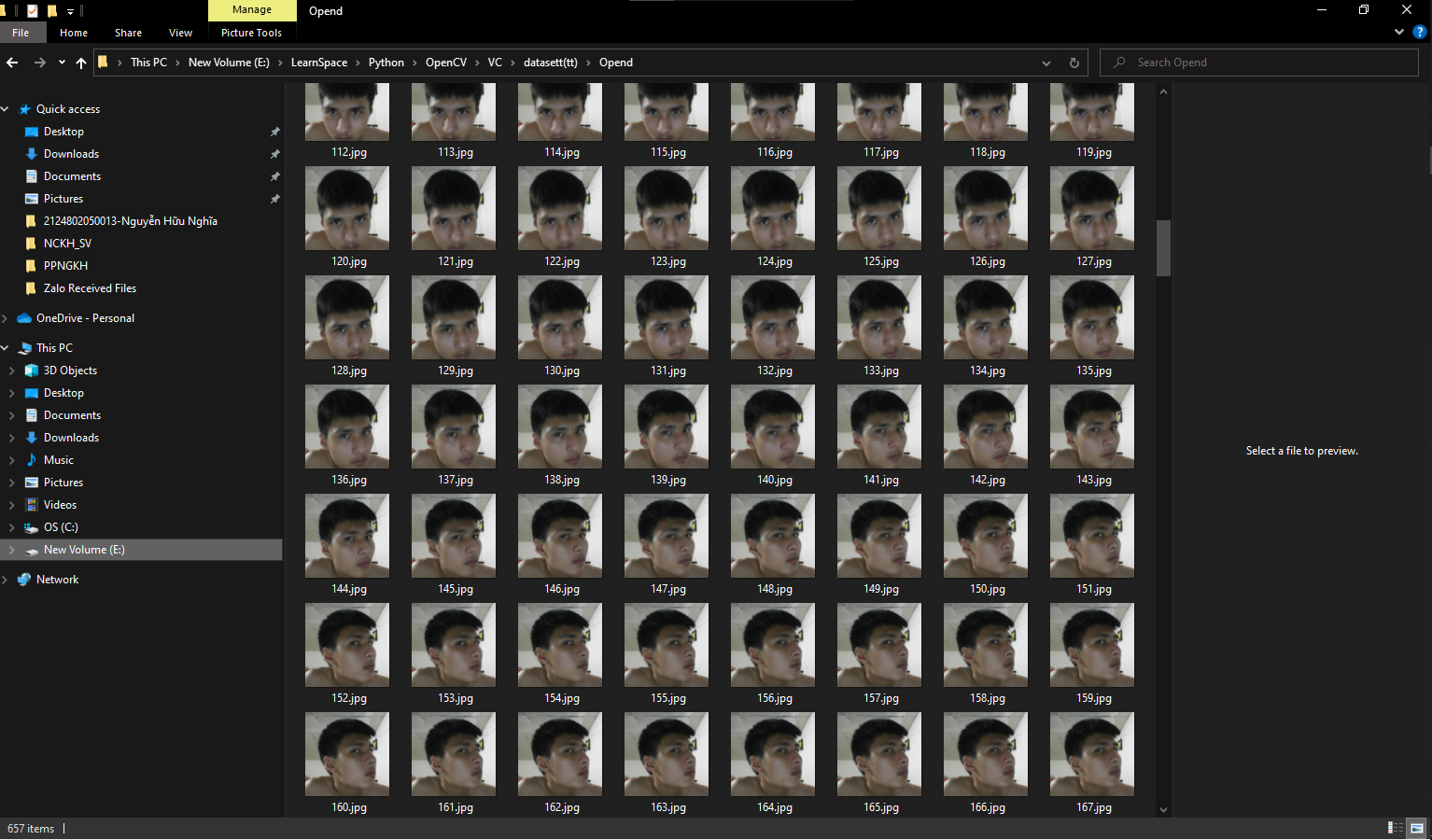


* + - 1. Folder Open trong dataset2.

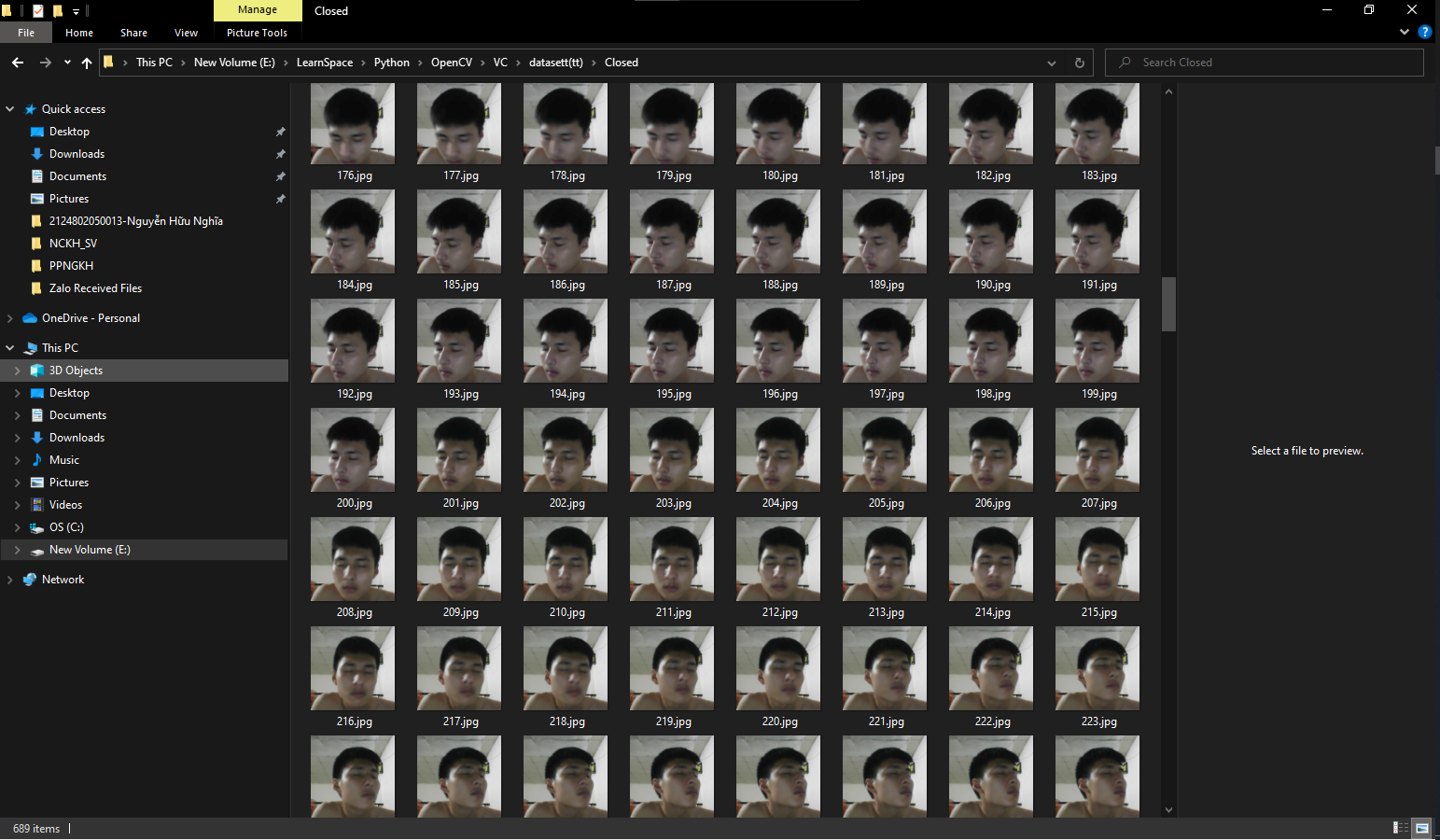


* + - 1. Folder Closed trong dataset2.

Bộ dữ liệu chúng em tự thu thập được gồm 1 folder data trong đó có 1 folder Closed chứa 689 tấm hình màu khuôn mặt nhắm mắt và 1 folder Opend chứa 657 tấm hình màu khuôn mặt mở mắt.



* + - 1. Folder Open tự thu thập.



* + - 1. Folder Closed tự thu thập.
  1. Thực hành
     1. Train model

ap = argparse.ArgumentParser()

ap.add\_argument("-d", "--datasets2", default="datasets2",

help="path to input dataset")

ap.add\_argument("-p", "--plot", type=str, default="plot.png",

help="path to output loss/accuracy plot")

ap.add\_argument("-m", "--model", type=str,

default="eye1\_detector.model",

help="path to output face mask detector model")

args = vars(ap.parse\_args())

aug = ImageDataGenerator(

rotation\_range=20,

zoom\_range=0.15,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.15,

horizontal\_flip=True,

fill\_mode="nearest")

headModel = baseModel.output

headModel = AveragePooling2D(pool\_size=(7, 7))(headModel)

headModel = Flatten(name="flatten")(headModel)

headModel = Dense(128, activation="relu")(headModel)

headModel = Dropout(0.5)(headModel)

headModel = Dense(2, activation="softmax")(headModel)

for layer in baseModel.layers:

layer.trainable = False

opt = Adam(lr=INIT\_LR, decay=INIT\_LR / EPOCHS)

model.compile(loss="binary\_crossentropy", optimizer=opt,

metrics=["accuracy"])

predIdxs = model.predict(testX, batch\_size=BS)

model.save(args["model"], save\_format="h5")

* + 1. Test model

prototxtPath = os.path.join(os.getcwd(), 'face\_detector', 'deploy.prototxt')

weightsPath = os.path.join(os.getcwd(), 'face\_detector', 'res10\_300x300\_ssd\_iter\_140000.caffemodel')

net = cv2.dnn.readNet(prototxtPath, weightsPath)

model = load\_model(args['model'])

if confidence > args["confidence"]:

box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])

(startX, startY, endX, endY) = box.astype("int")

(startX, startY) = (max(0, startX), max(0, startY))

(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

face = frame[startY:endY, startX:endX]

face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)

face = cv2.resize(face, (224, 224))

face = img\_to\_array(face)

face = preprocess\_input(face)

face = np.expand\_dims(face, axis=0)

(Closed, Open) = model.predict(face)[0]

label = "Tinh Tao" if Open > Closed else "Buon Ngu"

color = (0, 255, 0) if label == "Tinh Tao" else (0, 0, 255)

label = "{}: {:.2f}%".format(label, max(Open, Closed) \* 100)

cv2.putText(frame, label, (startX, startY - 10),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)

cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

if Open > Closed:

if keyboard.is\_pressed('v'):

pygame.mixer.music.pause()

count = 0

else:

ount += 1

if count == 15:

pygame.mixer.music.play()

* 1. Kết quả

Thông mô hình huấn luyện triển khai viết chương trình sau khi chạy thu được 2 tập tin “eye\_1.model”, “eye\_2.model” chứa đặc trưng của khuôn mặt khi nhắm hoặc mở mắt.

Sử dụng 2 tập tin trên chạy mẫu chương trình thực tế thấy được chương trình đều có thể nhận dạng được khuôn mặt người đang nhắm mắt hay mở mắt và phát còi tính hiệu khi khuôn mặt đang nhắm mắt trong một khoảng thời gian.

* 1. Đánh giá

Từ kết quả thực nghiệm cho thấy bộ dữ liệu tự thu thập chưa được qua xử lý trước, nên khi đưa vào train sẽ cho ra mô hình có độ chính xác thấp hơn bộ dữ liệu đã được xử lý trước khi đưa vào train.

C. PHẦN KẾT LUẬN

1. Kết quả đạt được

Đã đưa ra được kết luận là bộ dữ liệu trên internet có độ chính xác cao hơn bộ dữ liệu tự thu thập được thông qua chương trình chạy mẫu dựa trên mô hình nhận diện khuôn mặt đang nhắm mắt hay mở mắt sử dụng thuật toán CNN. Đây là kiến trúc mô hình CNN ứng dụng vào việc phát hiện đóng mở mắt được đề xuất, tập ảnh 224x224 điểm ảnh đa cấp xám được sử dụng trong thực nghiệm đánh giá mô hình.

Nhận diện được khuôn mặt đang đóng hay mở mắt nhưng vẫn còn chưa nhạy cho lắm. Tuy nhiên lợi thế là mô hình không quá lớn(dưới 4 MB) nên có thể tiếp tục tối ưu để triển khai trên các thiết bị, mô hình cũng có thể cài đặt trên các thiết bị nhúng vốn yêu cầu chặt chẽ về bộ nhớ.

1. Khuyến nghị

Đây là báo cáo nghiên cứu khoa học cấp đơn vị. Do hạn chế về mặt thời gian nên nhóm mới chỉ nghiên cứu về “nhận diện đóng mở mắt”, thêm vào tiếng chuông báo động Những vấn đề nhóm chưa làm được như:

* Thu thập thêm nhiều dữ liệu, ứng dụng thêm nhiều thuật toán và nhiều công nghệ tiên tiến để tăng độ chính xác của mô hình.

Và còn rất nhiều vấn đề chưa nghiên cứu tới.

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Thái Thị Hoà Vân, nghiên cứu tình trạng buồn ngủ của người lái xe dựa trên nhận dạng cử chỉ khuôn mặt
2. Nguyễn Minh Sơn, Nguyễn Văn Bình, Nguyễn Ngọc Lâm, Nghiên cứu thiết kế hệ thống giám sát trạng thái buồn ngủ của lái xe
3. ThS.Nguyễn Duy Linh, ThS. Trần Công Trung, Tìm hiểu một số hệ thống cảnh báo lái xe ngủ gật (Drowsy Driver Alert System) và đề xuất phương pháp xây dựng ứng dụng với thư viện mã nguồn mở OpenCV

**Website**

1. [tensorflow - Detect open and closed eyes with Python DeepLearning - Stack Overflow](https://stackoverflow.com/questions/65542271/detect-open-and-closed-eyes-with-python-deeplearning)
2. [python - How to train a tensorflow model to accurately classify open eyes and closed eyes? - Stack Overflow](https://stackoverflow.com/questions/63014116/how-to-train-a-tensorflow-model-to-accurately-classify-open-eyes-and-closed-eyes)
3. [Cảnh báo ngủ gật cho tài xế bằng Dlib và Resnet - Mì AI (miai.vn)](https://miai.vn/2019/09/05/cv-thu-lam-model-canh-bao-ngu-gat-cho-tai-xe-oto-bang-dlib-va-resnet/)