

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



HCMUTE

BÁO CÁO

**ĐỀ TÀI: NHẬN DIỆN MANG GIÀY HAY KHÔNG
MANG GIÀY VÀ LOẠI GIÀY (3 LOẠI)**

GVHD: PGS. TS. Nguyễn Trường Thịnh

Lớp học phần: 212ARIN337629_03CLC

Môn học: Trí tuệ nhân tạo

Sinh viên thực hiện:

Phạm Trần Anh Tuấn - 19146298

TP.HCM, Tháng 06/2022

LỜI NÓI ĐẦU

Trí tuệ nhân tạo ngày nay không còn là một thuật ngữ mới mẻ hay xa lạ nữa, mà ngày nay việc máy móc hiện hữu bên cạnh cũng như gần gũi hơn với con người. Trí tuệ nhân tạo có thể biến những thứ không thể thành có thể, biến thứ không tưởng thành hiện thực. Nó phát triển đáng kinh ngạc và dần dần thay thế con người những công việc nặng nhọc cũng như các công việc khó khăn. Những căn nhà thông minh, những robot chẩn đoán bệnh hay hoạt động trong lĩnh vực quân sự, những chiếc xe không người lái,.. Tương lai sẽ còn hơn thế nữa nó có thể phát triển đến chóng mặt hơn cả những gì ta nghĩ.

Nhận thấy sự cần thiết của AI trong tương lai cũng như sự cần thiết của nó trong công việc sau này của sinh viên ngành Cơ điện tử nên trường Đại học Sư phạm Kỹ thuật TP.HCM đã thêm môn học “Trí tuệ nhân tạo” trong chương trình đào tạo 18 của trường. Môn học “Trí tuệ nhân tạo” hướng dẫn một cách tổng quát, giúp sinh viên hiểu khái quát nhất về AI, tạo nền móng cho sinh viên học tập, nghiên cứu sâu thêm sau này.

Do đó giảng viên hướng dẫn của môn học là PSG.TS Nguyễn Trường Thịnh đã cho các bạn sinh viên đang học môn này làm các dự án về AI để báo cáo môn học cuối kì để có thể nắm rõ hơn môn học và ứng dụng môn học này vào thực tiễn. Chúng em mong rằng sau khi học môn học này thì tất cả các bạn sinh viên đã và đang học môn “Trí tuệ nhân tạo” sẽ có thêm nhiều kiến thức để tạo nền cho những bước ban đầu trong con đường trở thành một kỹ sư giỏi phụng sự cho Tổ quốc.

Với kiến thức còn hạn hẹp thì chúng em rất mong sự đóng góp, sửa chữa và góp ý của thầy để chúng em rút kinh nghiệm cũng như bổ sung những thứ còn thiếu cho mình. Cuối cùng, chúng em xin được chân thành cảm ơn sự hướng dẫn của thầy Nguyễn Trường Thịnh và các anh chị trợ giảng đã giúp tụi em hoàn thành dự án này.

MỤC LỤC

LỜI NÓI ĐẦU	2
CHƯƠNG 1: TỔNG QUAN	4
1.1 Lí do chọn đề tài.....	4
1.2 Tổng quan về trí tuệ nhân tạo.....	4
1.2.1 Trí tuệ nhân tạo là gì ?	4
1.2.2 Lịch sử hình thành trí tuệ nhân tạo	5
1.3 Tổng quan về ngôn ngữ lập trình python	8
1.3.1 Lịch sử ra đời của Python	8
1.3.2 Các tính năng của Python	9
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	10
2.1 Học máy	10
2.2 Các thư viện sử dụng.....	10
2.2.1 Thư viện OpenCV.....	10
2.2.2 Thư viện Numpy	11
2.2.3 Thư viện Matplotlib.....	12
2.3 Các module.....	12
CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH.....	13
3.1 Nhập các thư viện cần thiết.....	13
3.2 Tải tập dữ liệu từ các file trong máy	14
3.3 Đặt tên và đưa đường dẫn file vào các biến mình cần nhận diện	14
3.4 Tạo model.....	15
3.5 Train model	17
3.6 Biểu đồ học của máy	17
3.7 Khởi tạo các thư viện cần thiết khi xử lí realtime	18
3.8 Khởi tạo camera và tạo frame	19
3.9 Tạo nhãn đầu ra cho output và hiển thị realtime.....	19
3.10 Tạo hiển thị trên khung ảnh frame	20
CHƯƠNG 4: KẾT QUẢ	22
4.1 Kết quả thực nghiệm	22
4.2 Các hạn chế của đề tài.....	24
4.3 Hướng phát triển	25
KẾT LUẬN	26

CHƯƠNG 1: TỔNG QUAN

1.1 Lí do chọn đề tài

Việc cấp sách đến trường luôn là việc mà chúng ta đã từng trải qua suốt hành trình ngồi trên ghế nhà trường. Thế nhưng liệu chúng ta có nhớ khi đi học từ cấp 1 đến cấp 3 và cụ thể hiện nay chúng ta đang đi học đại học thì chúng ta mang giày hay mang dép đi học không? Chắc hẳn sẽ hơi khó để nhớ hồi cấp 2 và cấp 3 vì lúc ấy là giai đoạn dậy thì, giai đoạn nông nổi thích thể hiện thì thật khó để chúng ta mang một đôi giày hay scandal đi học để lịch sự hướng chi là mang giày. Thế nhưng ngày nay với việc đất nước chúng ta ngày càng phát triển hội nhập văn hóa các nước phương tây thì giày xuất hiện càng nhiều, và đó cũng coi là một thời trang và được mang vào môi trường học tập. Khi mang giày đi học chúng ta thấy được sự lịch thiệp và đẹp, thời trang. Vậy có thể những học sinh khác sẽ cảm thấy bó buộc khi mang giày nên sẽ tìm cách lách không mang giày thay vào đó mang dép hoặc scandal. Do đó em chọn đề tài này nhằm mục đích phát hiện được các bạn học sinh đang mang giày hay không và xử lý theo quy định mà nhà trường đã đặt ra nhằm thống nhất giữa các học sinh.

Cũng có thể áp dụng vào các buổi tiệc, hội nghị yêu cầu đảm bảo trang phục chỉnh tề đẹp đẽ cao. Chính vì thế em chọn đề tài nhận diện mang giày hay không mang giày.

1.2 Tổng quan về trí tuệ nhân tạo

1.2.1 Trí tuệ nhân tạo là gì ?

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi,...

Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong các tác phẩm khoa học viễn tưởng, nó là một trong những ngành trọng yếu của tin học. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc.

1.2.2 Lịch sử hình thành trí tuệ nhân tạo

Nghiên cứu lịch sử trí tuệ nhân tạo AI sớm vào những năm thập niên 60 đã khám phá các vấn đề mà công nghệ này có thể giải quyết. Vào những năm 1960, Bộ Quốc phòng Hoa Kỳ đã quan tâm đến loại công việc này và bắt đầu đào tạo máy tính để bắt chước lý luận cơ bản của con người. Ví dụ, Cơ quan Dự án Nghiên cứu Quốc phòng Tiên tiến (DARPA) đã hoàn thành các dự án lập bản đồ đường phố vào những năm 1970. Và DARPA đã sản xuất trợ lý cá nhân thông minh vào năm 2003...

Công việc ban đầu này đã mở đường cho tự động hóa và lý luận chính thức mà chúng ta thấy trong các máy tính ngày nay, bao gồm các hệ thống hỗ trợ quyết định và hệ thống tìm kiếm thông minh có thể được thiết kế để bổ sung và tăng cường khả năng của con người.

- Năm 1943 Warren McCullough và Walter Pitts xuất bản cuốn “A Logical Calculus of Ideas Immanent in Nervous Activity”, dịch ra là “Một tính toán logic của những ý tưởng tiềm ẩn trong hoạt động thần kinh”. Bài viết đề xuất mô hình toán học đầu tiên để xây dựng một mạng lưới thần kinh.
- Năm 1949 Trong cuốn sách “The Organization of Behavior: A Neuropsychological Theory”- *Tổ chức hành vi: Một lý thuyết thần kinh học*, Donald Hebb đề xuất lý thuyết về các hệ thống con đường thần kinh được tạo ra từ các kết nối giữa các tế bào thần kinh trở nên mạnh mẽ hơn.

- Năm 1950 Alan Turing xuất bản “Computing Machinery and Intelligence” – Máy tính và trí thông minh, đề xuất Thử nghiệm Turing, một phương pháp để xác định xem một máy tính có thông minh hay không.
- Năm 1952 Arthur Samuel phát triển một chương trình tự học để chơi cờ.
- Năm 1954 Thí nghiệm dịch máy Georgetown-IBM tự động dịch 60 câu tiếng Nga được chọn cẩn thận sang tiếng Anh.
- Năm 1956 Cụm từ trí tuệ nhân tạo lần đầu tiên được nói đến tại “Dự án nghiên cứu mùa hè về trí tuệ nhân tạo”. Với sự dẫn đầu bởi John McCarthy, hội nghị, trong đó xác định phạm vi và mục tiêu của AI, được coi là sự ra đời của trí tuệ nhân tạo như chúng ta biết ngày nay.
- Năm 1958 John McCarthy phát triển ngôn ngữ lập trình AI Lisp và xuất bản bài báo “Programs with Common Sense”. Bài viết đã đề xuất nhà tư vấn giả thuyết, một hệ thống AI hoàn chỉnh với khả năng học hỏi kinh nghiệm hiệu quả như con người.
- Năm 1959 Allen Newell, Herbert Simon và JC Shaw giải quyết vấn đề chung (GPS), một chương trình được thiết kế để bắt chước giải quyết vấn đề của con người. Herbert Gelernter phát triển chương trình Định lý hình học. Arthur Samuel đồng xu với thuật ngữ học máy khi còn ở IBM. John McCarthy và Marvin Minsky đã tìm thấy Dự án Trí tuệ nhân tạo MIT.
- Năm 1963 John McCarthy bắt đầu Phòng thí nghiệm AI tại Stanford.
- Năm 1966 Báo cáo của Ủy ban Tư vấn xử lý ngôn ngữ tự động (ALPAC) của chính phủ Hoa Kỳ nêu chi tiết về sự thiếu tiến bộ trong nghiên cứu dịch máy, một sáng kiến lớn của chiến tranh lạnh với lời hứa dịch tự động tiếng Nga. Báo cáo ALPAC dẫn đến việc hủy bỏ tất cả các dự án MT do chính phủ tài trợ.

- Năm 1969 Các hệ thống chuyên gia thành công đầu tiên được phát triển trong DENDRAL, một chương trình XX và MYCIN, được thiết kế để chẩn đoán nhiễm trùng máu, được tạo ra tại Stanford.
- Năm 1972 Ngôn ngữ lập trình logic PRITAL được tạo ra.
- Năm 1973 “Báo cáo Lighthill”, nêu chi tiết về sự thất bại trong nghiên cứu AI, được chính phủ Anh công bố, từ đây dẫn đến việc cắt giảm nghiêm trọng tài trợ cho các dự án trí tuệ nhân tạo.
- Năm 1974-1980 Liên tiếp là sự thất vọng với sự phát triển của AI dẫn đến sự cắt giảm DARPA lớn trong các khoản trợ cấp học thuật. Kết hợp với báo cáo ALPAC trước đó và “Báo cáo Lighthill” năm trước, tài trợ trí tuệ nhân tạo làm khô và các quỹ nghiên cứu. Thời kỳ này được gọi là “Mùa đông AI đầu tiên.”
- Năm 1980 Tập đoàn thiết bị kỹ thuật số phát triển R1 (còn được gọi là XCON), hệ thống chuyên gia thương mại thành công đầu tiên. Được thiết kế để định cấu hình các đơn đặt hàng cho các hệ thống máy tính mới, R1 khởi đầu sự bùng nổ đầu tư vào các hệ thống chuyên gia sẽ tồn tại trong phần lớn thập kỷ, kết thúc hiệu quả “Mùa đông AI” đầu tiên.
- Năm 1982 Bộ Thương mại Quốc tế và Công nghiệp Nhật Bản khởi động dự án Hệ thống máy tính thế hệ thứ năm đầy tham vọng. Mục tiêu của FGCS là phát triển hiệu năng giống như siêu máy tính và một nền tảng để phát triển trí tuệ nhân tạo AI.
- Năm 1985 Các công ty đang chi hơn một tỷ đô la một năm cho các hệ thống chuyên gia và toàn bộ ngành công nghiệp được gọi là thị trường máy Lisp mọc lên để hỗ trợ họ. Các công ty như Symbolics và Lisp Machines Inc. xây dựng các máy tính chuyên dụng để chạy trên ngôn ngữ lập trình AI Lisp.
- Năm 1987-1993 Khi công nghệ điện toán đám mây được cải thiện, có nhiều lựa chọn thay thế rẻ hơn xuất hiện và thị trường máy Lisp sụp đổ vào năm 1987, mở ra “Mùa đông AI thứ hai”. Các chuyên gia AI rất chật vật và không được sự ủng

hộ trong giai đoạn này. DARPA kết thúc Sáng kiến Điện toán Chiến lược vào năm 1993 sau khi chi gần 1 tỷ đô la và không đạt được kỳ vọng như đã tính toán

- Năm 1991 Lực lượng Hoa Kỳ triển khai DART, một công cụ lập kế hoạch và lập kế hoạch hậu cần tự động, trong Chiến tranh vùng Vịnh
- Năm 2005 STANLEY, một chiếc xe tự lái, chiến thắng DARPA Grand Challenge. Quân đội Hoa Kỳ bắt đầu đầu tư vào các robot tự hành như “Big Dog” của Boston Dynamic và “PackBot” của iRobot.
- Năm 2008 Google tạo ra những bước đột phá trong nhận dạng giọng nói và giới thiệu tính năng này trong ứng dụng iPhone.
- Năm 2011 Watson của IBM tuyên bố cạnh tranh về *Jeopardy!*
- Năm 2012 Andrew Ng, người sáng lập dự án Google Brain Deep Learning, cung cấp một mạng lưới thần kinh bằng cách sử dụng thuật toán 10 triệu video YouTube dưới dạng tập huấn luyện. Mạng lưới thần kinh đã học cách nhận ra một con mèo mà không được cho biết con mèo là gì
- Năm 2014 Google tạo ra chiếc xe tự lái đầu tiên để vượt qua bài kiểm tra lái xe của nhà nước.
- Năm 2016 AlphaGo của Google DeepMind đánh bại nhà vô địch thế giới cờ vây Lee Sedol. Sự phức tạp của trò chơi Trung Quốc cổ đại được coi là một trở ngại lớn để giải tỏa trong AI.

1.3 Tổng quan về ngôn ngữ lập trình python

1.3.1 Lịch sử ra đời của Python

Python đã được hình thành vào cuối những năm 1980 và được bắt đầu thực hiện vào tháng 12/1989 bởi Guido van Rossum tại CWI tại Hà Lan như là người kế thừa của ngôn ngữ ABC (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba. Van Rossum là tác giả chính của Python, và vai trò trung tâm của ông tiếp tục trong việc quyết định hướng phát triển của Python được

phản ánh trong tiêu đề mà cộng đồng Python dành cho ông “Độc tài nhân từ cho cuộc sống” (benevolent dictator for life)(BDFL).

Python 2.0 được phát hành vào ngày 16/10/2000, với nhiều tính năng chính mới bao gồm một bộ dọn rác đầy đủ và hỗ trợ Unicode. Với phiên bản này, quá trình phát triển đã được thay đổi và trở thành minh bạch hơn và được cộng đồng ủng hộ.

Python 3.0 (còn được gọi là Python 3000 hoặc Py3k), một bản phát hành lớn, không tương thích ngược, được phát hành vào ngày 03/12/2008 sau một thời gian dài thử nghiệm. Nhiều trong số các tính năng chính của nó đã được điều chỉnh để tương thích ngược với Python 2.6 và 2.7.

1.3.2 Các tính năng của Python

Các tính năng của python bao gồm:

- Dễ học: Python có ít từ khóa, cấu trúc đơn giản và cú pháp được định nghĩa rõ ràng. Dễ đọc Mã Python được định nghĩa rõ ràng hơn và có thể nhìn thấy bằng mắt.
- Dễ bảo trì, thư viện tiêu chuẩn rộng: Phần lớn thư viện Python rất dễ đính kèm và đa nền tảng.
- Chế độ tương tác: Python có hỗ trợ cho chế độ tương tác cho phép kiểm tra tương tác và debug.
- Portable: Python có thể chạy trên nhiều nền tảng phần cứng khác nhau và có cùng giao diện trên tất cả các nền tảng.
- Có thể mở rộng: Bạn có thể thêm các module cấp thấp vào trình thông dịch Python. Các module này cho phép các lập trình viên thêm hoặc tùy chỉnh các công cụ của mình để hiệu quả hơn.
- Cơ sở dữ liệu: Python cung cấp phương thức giao tiếp cho tất cả các cơ sở dữ liệu.
- Khả năng mở rộng: Python cung cấp cấu trúc và hỗ trợ tốt hơn cho các chương trình lớn hơn so với kịch bản lệnh shell.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Học máy

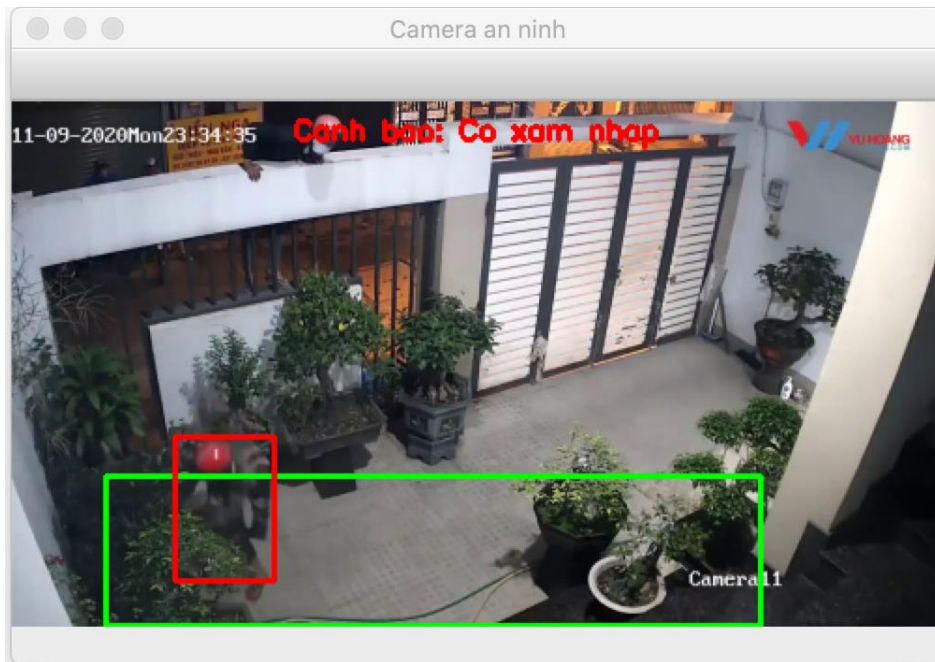
Học máy hay máy học trong tiếng Anh là Machine learning, viết tắt: ML. Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo. Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian. ML vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kỹ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả. Các mô hình ML yêu cầu lượng dữ liệu đủ lớn để "huấn luyện" và đánh giá mô hình. Trước đây, các thuật toán ML thiếu quyền truy cập vào một lượng lớn dữ liệu cần thiết để mô hình hóa các mối quan hệ giữa các dữ liệu. Sự tăng trưởng trong dữ liệu lớn (big data) đã cung cấp các thuật toán ML với đủ dữ liệu để cải thiện độ chính xác của mô hình và dự đoán.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia, Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

2.2 Các thư viện sử dụng

2.2.1 Thư viện OpenCV

OpenCV là viết tắt của Open Source Computer Vision, tạm dịch là Thị giác máy tính mã nguồn mở, đúng như tên gọi mục đích chính của nó được tạo ra là nhằm tạo ra các chức năng thị giác máy tính, giám sát các chức năng tổng thể tập trung vào thị giác máy tính tức thì. Vì OpenCV không có tài liệu thích hợp, cho nên theo nhiều nhà phát triển, nó là một trong những thư viện khó học nhất. Tuy nhiên, nó vẫn cung cấp đầy đủ khả năng để giúp bạn học thị giác máy tính dễ hơn.



Hình 2.1. Ví dụ về OpenCV.

Những khả năng của OpenCV:

- OpenCV là một gói mô-đun hình ảnh lý tưởng cho phép bạn đọc và ghi, thay đổi dữ liệu nhiều hình ảnh cùng một lúc.
- Tạo ra thị giác máy tính cho phép bạn xây dựng lại, gián đoạn và thông hiểu môi trường 3D từ môi trường 2D tương ứng của nó.
- OpenCV được sử dụng nhiều trong nhận diện vật thể và hình ảnh được thiết lập trước, chẳng hạn như khuôn mặt, động vật, cây cối, các vật thể di chuyển, etc.
- Bạn cũng có thể lưu và chụp bất kỳ khoảnh khắc nào của video và cũng có thể phân tích các thuộc tính khác nhau của nó như chuyển động, nền, etc.

2.2.2 Thư viện Numpy

Numpy (Numeric Python): là một thư viện toán học rất phổ biến và mạnh mẽ của Python. NumPy được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng Python đơn thuần.

NumPy làm phong phú ngôn ngữ lập trình Python với các cấu trúc dữ liệu mạnh mẽ để tính toán hiệu quả các mảng và ma trận đa chiều. Numpy không chỉ là một gói

mô-đun để xử lý mảng mà nó còn cung cấp khả năng quản lý mảng cực kỳ vượt trội. Nhanh chóng, vượt trội, hiệu quả là những gì tôi được trải nghiệm với Numpy.

2.2.3 Thư viện Matplotlib

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotlib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở. Matplotlib cho phép các lập trình viên trực quan hóa các tập dữ liệu với nhiều loại biểu đồ khác nhau. Nó được sử dụng phù hợp nhất cho việc khám phá và trình bày thông tin chi tiết về dữ liệu đã xử lý. Cho dù đó là biểu đồ tĩnh, động hay biểu đồ tương tác, Matplotlib đều giúp dữ liệu của bạn trở nên sống động. Điều này rất quan trọng với lập trình viên khi họ phải giao tiếp với các đối tượng không chuyên về kỹ thuật.

2.3 Các module

Module OS: Module là một module tích hợp sẵn với Python, module này cho phép chúng ta thao tác với tệp và thư mục.

CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH

3.1 Nhập các thư viện cần thiết

```
# Import Libraries
import tensorflow as tf
import matplotlib.pyplot as plt
import cv2 as cv
import os
import numpy as np
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras import layers
import numpy as np
import pandas as pd # Xu lý bảng
import seaborn as sns # Vẽ biểu đồ thị của dữ liệu
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler # Xử lý chuẩn hóa dữ liệu
from sklearn.model_selection import train_test_split # Chia dữ liệu ra làm 2 phần
from keras.layers import Dense, Activation, Dropout, BatchNormalization, LSTM # LSTM biên dạng ANN, BatchNormalization: cho nhỏ lại
from keras.models import Sequential
from tensorflow.keras.utils import to_categorical # Sử dụng để làm nổi đối tượng cần phân loại
from keras import callbacks
from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_report, accuracy_score, f1_score # Để đo lường
from keras.utils import np_utils
from keras.callbacks import EarlyStopping
from keras.utils.np_utils import to_categorical
from keras.layers import Dropout, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
import cv2
from sklearn.model_selection import train_test_split
```

```
import random
from tensorflow.keras.optimizers import RMSprop # tối ưu
from sklearn import preprocessing
import keras
from keras.datasets import boston_housing
from keras.callbacks import EarlyStopping # dừng lại ngay lập tức
from sklearn.preprocessing import scale # xử lý dữ liệu
from sklearn.preprocessing import StandardScaler # xử lý dữ liệu
```

Hình 3.1. Các thư viện cần thiết.

CV2: Nhập để sử dụng thư viện OpenCV để xử lý hình ảnh.

Matplotlib: Thư viện này được sử dụng để trực quan dữ liệu và vẽ biểu đồ. Do đó, nó được nhập để tạo thành hình ảnh.

OS: Để tương tác với hệ điều hành. Tại đây, để đọc đường dẫn và lưu hình ảnh vào đường dẫn đó.

3.2 Tải tập dữ liệu từ các file trong máy

```
image_generator = ImageDataGenerator(rescale=1/255, validation_split=0.2)

train_dataset = image_generator.flow_from_directory(batch_size=32,
                                                    directory=r"C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train",
                                                    shuffle=True,
                                                    target_size=(150,150),
                                                    subset="training",
                                                    class_mode='categorical')

validation_dataset = image_generator.flow_from_directory(batch_size=32,
                                                         directory=r"C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train",
                                                         shuffle=True,
                                                         target_size=(150,150),
                                                         subset="validation",
                                                         class_mode='categorical')
```

Hình 3.2. Code tải dữ liệu.

Đoạn mã này gán các biến từ tập Train và tập Validation vào, ở đây ta không cần phải chia sẵn 2 tập train và validation thay vào đó ta cứ lấy tất cả ảnh trong tập train, lúc này máy sẽ tự động chia dữ liệu cho chúng ta thành train và validation.

Trước đó ta có bước rescale kích thước ảnh lại.

3.3 Đặt tên và đưa đường dẫn file vào các biến mình cần nhận diện

```
import glob
from keras.utils import load_img
from PIL import Image
adidas = glob.glob(r'C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train\adidas/*.')
nike = glob.glob(r'C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train\nike/*.')
converse = glob.glob(r'C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train\converse/*.')
no_shoe = glob.glob(r'C:\Users\TUAN\Desktop\AI_cuoiky\lan_cuoik8\train\no-shoe/*.')
✓ 0.4s
```

Hình 3.3 Đặt tên biến và đưa địa chỉ của file từ máy.

```
train_dataset.class_indices
✓ 0.4s
{'adidas': 0, 'converse': 1, 'nike': 2, 'no-shoe': 3}
```

Hình 3.4. Các biến lần lượt được gán nhãn đầu ra

Ở đây ta nhận nhiệm mang giày hay không mang giày và loại giày (2 loại). Lần lượt là Adidas, Nike và No_shoe. Các biến trên lần lượt được đưa vào các nhãn đầu ra 0, 1, 2 và 3.

3.4 Tạo model

```
from keras.layers import Conv2D, MaxPooling2D
model = Sequential()
model.add(Conv2D(32,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same',input_shape=(150,150,3)))
model.add(Conv2D(32,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))

model.add(Conv2D(64,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same')) # 64 lần tích chập
model.add(Conv2D(64,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))

model.add(Conv2D(128,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same')) # 128 lần tích chập
model.add(Conv2D(128,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))

model.add(Conv2D(256,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same')) # 256 lần tích chập
model.add(Conv2D(256,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))

model.add(Conv2D(512,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same')) # 512 lần tích chập
model.add(Conv2D(512,(3,3), activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.2))
```

Hình 3.5. Code tạo Model.

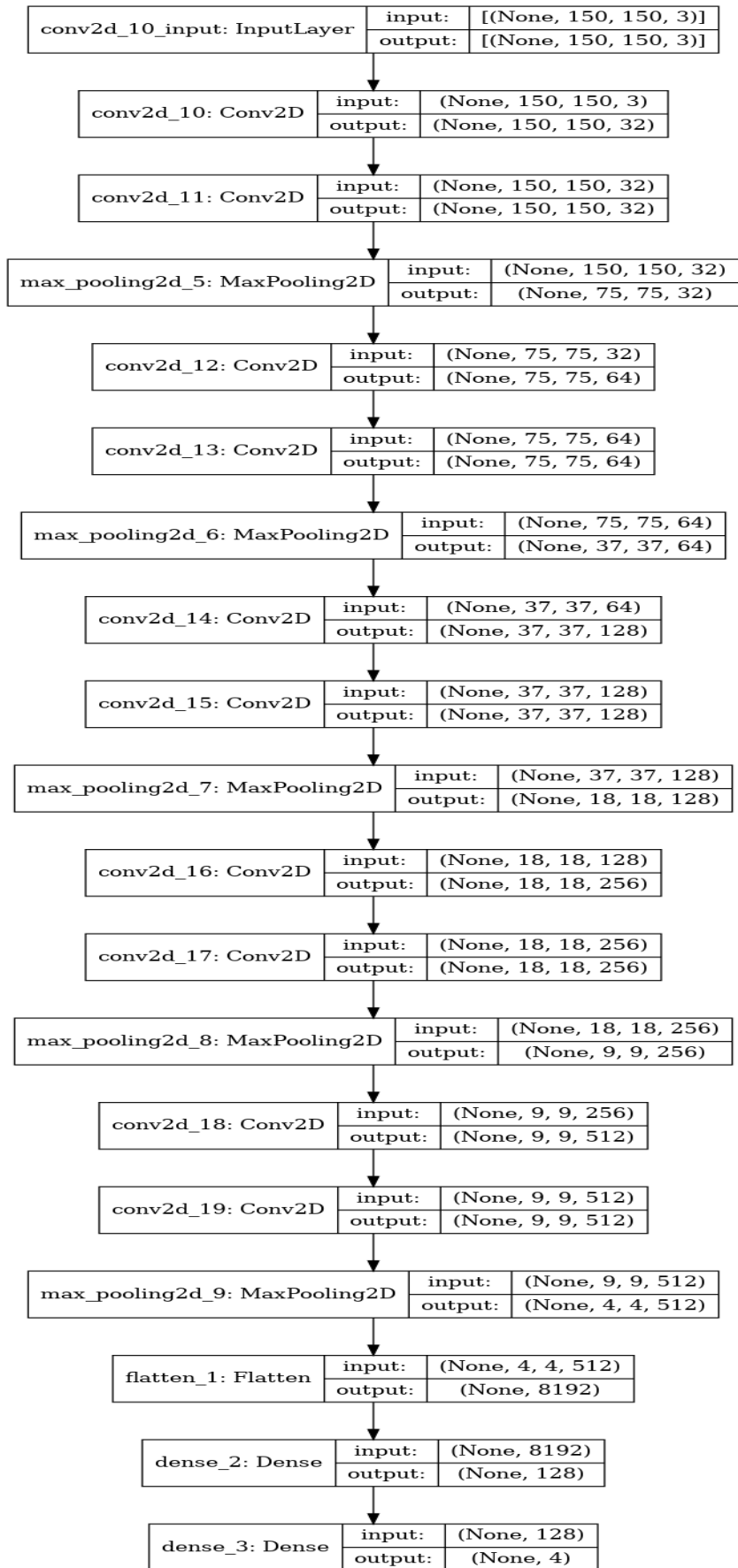
Model có input_size là 150x150x3, nghĩa là các ảnh đều resize xuống 64x64. Model có 4 output.

```
=====
Total params: 5,761,444
Trainable params: 5,761,444
Non-trainable params: 0
```

Hình 3.6. Tổng tham số là.

Sử dụng CNN để tạo model với các lần tích chập. Đầu tiên là 32 lần tích chập sau đó đến 64, 128, 256, 512 lần tích chập. Với lần tích chập đầu tiên đầu vào là kích thước ảnh 150.

Ba dòng tiếp là tạo lớp CNN với 32 lần tích chập với hàm phi tuyến Relu và lớp gộp MaxPooling. Tương tự cho các lớp CNN với 64, 128, 256, 512 lần tích chập. Sử dụng câu lệnh dropout mục đích là để tránh hiện overfitting.



Hình 3.7. Ví dụ một trong các lớp ẩn sau khi tích chập.

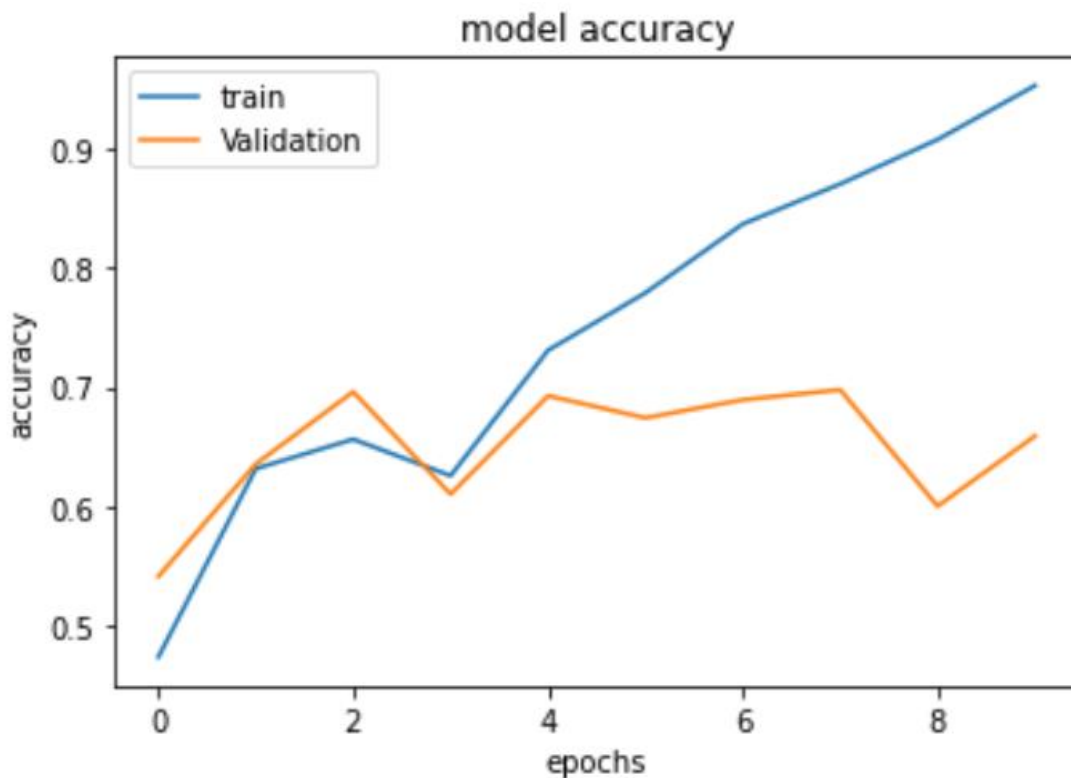
3.5 Train model

```
# Train model
model.compile(loss='mse',optimizer=RMSprop(),metrics=['accuracy'])
es = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=50)
history=model.fit(train_dataset,batch_size=128,epochs=150,validation_data=validation_dataset,callbacks=es)
```

Hình 3.8. Code train mdel.

Ta thực hiện train model 150 lần với hàm es là EarlyStop mục đích nhằm để những cái nào khi máy học không tăng thì sẽ bỏ qua cái phần đó giúp giảm thời gian train đáng kể.

3.6 Biểu đồ học của máy



Hình 3.9. Biểu đồ học của máy.

Ta thấy rằng lúc đầu máy học khá là tốt bám theo đường train nhưng về sau thì máy bị học overftting. Nguyên do chính là do data của chúng ta soạn về sau chưa được tốt khiến máy học không có được ổn cho lắm và nguyên nhân thứ hai là do model của chúng ta chưa được tối ưu nên sự học của máy nó không được hiệu quả khiến cho độ chính xác không được cao chỉ trung bình khá.

3.7 Khởi tạo các thư viện cần thiết khi xử lý realtime

```
# Camera
import cv2
import numpy as np
from imutils.video import VideoStream
from tensorflow.keras.models import load_model
import os
from tensorflow.keras.utils import load_img, img_to_array
```

Hình 3.10. Các thư viện.

```
model = load_model('Final.h5')
# Create a VideoCapture object and read from input file
cap = cv2.VideoCapture('test2.mp4')
try:
    # creating a folder named data
    if not os.path.exists('data'):
        os.makedirs('data')
# if not created then raise error
except OSError:
    print ('Error: Creating directory of data')
```

Hình 3.11. Load model đã lưu từ train.

Tiếp đến ta lấy dữ liệu từ tập tập ta đã train, sau đó khởi tạo camera ở đây là 1 video quay sẵn. Sau đó tạo một folder có tên là data lưu các ảnh ta cắt từ video lưu vào data. Sau đó lấy các ảnh từ tệp data ta nhận diện mà hiển thị lên video.

Khi mà chúng ta test trên camera máy tính thì chỉ việc đổi dòng lệnh `cap = cv2.VideoCapture(0)` là lúc đó ta có thể sử dụng cam máy tính cho việc thời gian thực.

3.8 Khởi tạo camera và tạo frame

```
# frame
currentframe = 0
# Check if camera opened successfully
if (cap.isOpened()== False):
    print("Error opening video file")
# Read until video is completed
while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret == True:
        # Display the resulting frame
        # if video is still left continue creating images
        name = './data/frame' + str(currentframe) + '.jpg'
        print ('Creating...' + name)
```

Hình 3.12. Code khởi tạo camera.

Đầu tiên ta tạo frame, sau đó ta tạo đoạn code xem camera hoặc video của mình có được mở thành công hay chưa.

Tiếp đến nếu camera đã được đọc và khi video vẫn đang tiếp tục chạy thì tạo một folder có tên là data và lưu các ảnh cắt từ video lần lượt là frame1, frame2,... tới khi nào video kết thúc là được lưu dưới định dạng .jpg.

3.9 Tạo nhãn đầu ra cho output và hiển thị realtime

Mục đích của việc này là để làm cho công việc nhận diện của máy sau khi hoàn thành và hiển thị cho ta biết được rằng là máy có nhận diện chính xác hay là không.

```

# writing the extracted images
cv2.imwrite(name, frame)
filename = name
predict = ['adidas', 'nike', 'no_shoe', 'converse']
predict = np.array(predict)
img = load_img(filename, target_size=(150,150))
img = load_img(filename, target_size=(150,150))
img = img_to_array(img)
img = img.reshape(1,150,150,3)
img = img.astype('float32')
img = img/255
result = np.argmax(model.predict(img), axis=-1)
predict[result]
if(result==0):
    a = "Brand: ADIDAS"
if(result==2):
    a = "Brand: NIKE"
if(result==3):
    a = "Khong mang giay"
if(result==1):
    a = "converse"

```

Hình 3.13. Tạo predict và các nhãn đầu ra.

3.10 Tạo hiển thị trên khung ảnh frame

```

# org
org = (20, 185)
# fontScale
fontScale = 1.5
# Blue color in BGR
color = (255, 0, 0)
font = cv2.FONT_HERSHEY_SIMPLEX
# Line thickness of 2 px
thickness = 2
cv2.putText(frame, a, org, font, fontScale, color, 1)
cv2.imshow('Frame', frame)
# increasing counter so that it will
# show how many frames are created
currentframe += 1

# Press Q on keyboard to exit
if cv2.waitKey(25) & 0xFF == ord('q'):
    break

```

Hình 3.14. Setup video hiển thị.

Với các giá trị lần lượt là:

- Org: là vị trí mà chữ mình hiển thị.
- FontScale: là kích cỡ chữ hiển thị.
- Color và font : lần lượt là màu và phong chữ.

Và cuối là dòng lệnh ấn ‘q’ để thoát.

CHƯƠNG 4: KẾT QUẢ

4.1 Kết quả thực nghiệm

Kết quả khi thực hiện thử đưa những tấm ảnh không có trong tập train thì máy nhận khá là chính xác sai số khoảng 2 tấm trên 10 tấm ảnh:



Hình 4.1. Ảnh nhận diện khi test.

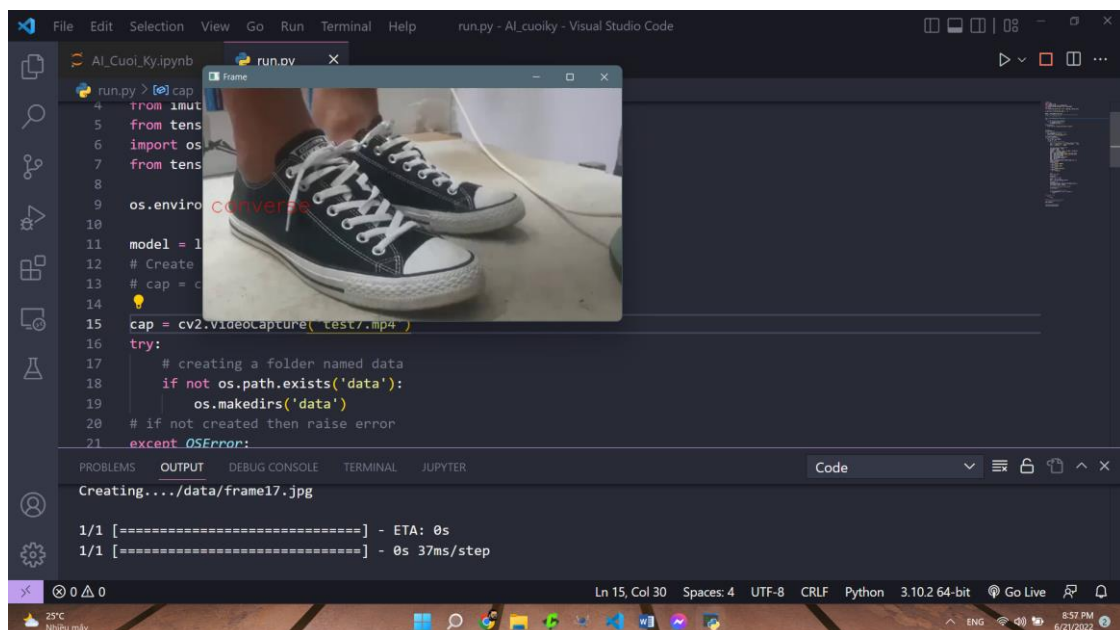


Hình 4.2. Ảnh nhận diện Nike.

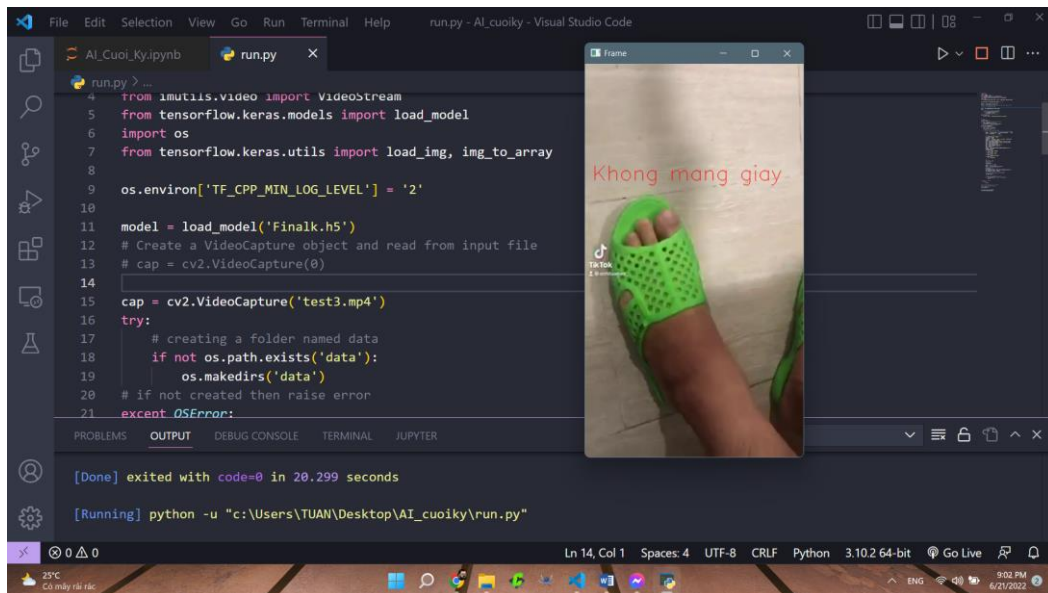


Hình 4.3. Ảnh nhận diện Adidas.

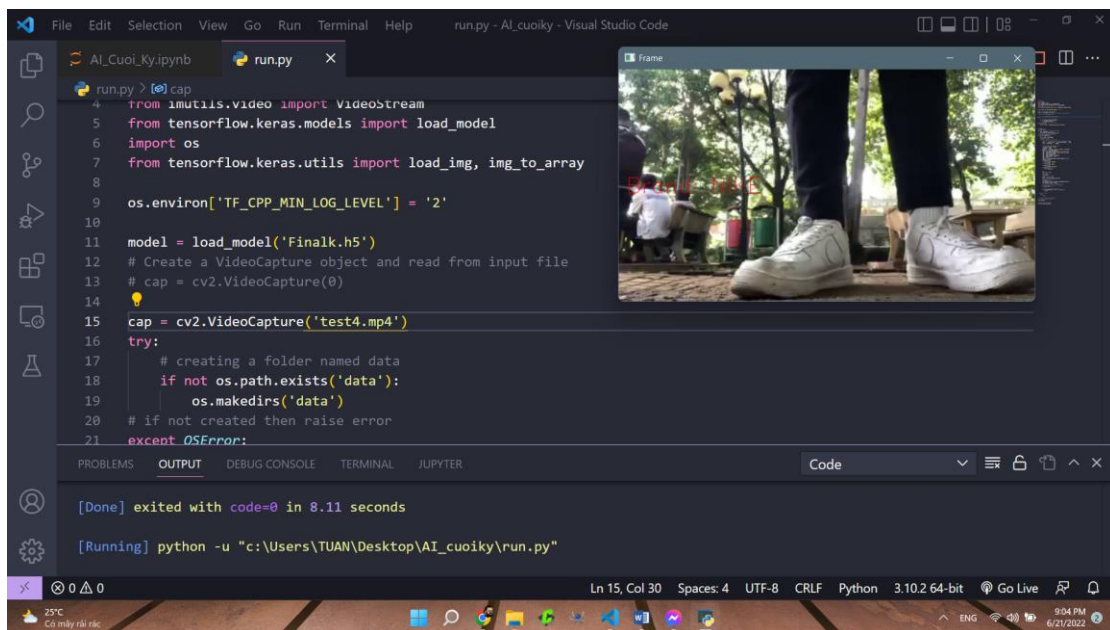
Kết quả thực nghiệm cho ta thấy được nhận diện ổn độ chính xác trung bình khá. Và đây là các sản phẩm khi thực hiện thời gian thực và video:



Hình 4.4. Nhận diện được đôi giày loại Converse.



Hình 4.5. Nhận diện không mang giày.



Hình 4.6. Nhận diện được giày loại Nike.

4.2 Các hạn chế của đề tài

Sau khi làm thì em rút ra được khá là thú vị, mô hình chạy khá ổn nhưng bên cạnh đó cũng có nhiều hạn chế khiến cho mô hình chạy không được tối ưu, độ chính xác chưa cao cho lắm:

- Hạn chế thứ nhất: Giới hạn về không được sử dụng các model train sẵn như Yolo, VGG,... điều này làm cho việc nhận diện thời gian thực giảm độ chính

xác xuống. Thay vì sẽ có khung xanh để bắt đôi giày của chúng ta thì phải để camera dưới đất và khá sát giày thì độ chính xác mới cao hơn.

- Hạn chế thứ hai: Tập dữ liệu train chưa thực sự hoàn hảo, cụ thể như ở trên biểu đồ học của máy ta thấy rằng lúc đầu máy bám khá sát đường train nhưng càng về sau thì không còn được bám sát như lúc đầu nữa(overfitting). Việc chưa có kinh nghiệm soạn tập dữ liệu cũng là một hạn chế.
- Hạn chế thứ ba: Chưa có tối ưu hóa hết được thuật toán trong nhận diện thời gian thực khiến cho việc nhận diện đôi lúc không chính xác.

4.3 Hướng phát triển

Có thể sử dụng thêm các model được huấn luyện sẵn để tối ưu hóa được nhận diện thời gian thực. Bên cạnh đó cải thiện thêm phần dữ liệu huấn luyện được tối ưu đầy đủ và chi tiết hơn các đặc trưng của giày để máy có thể học hiệu quả và được tối ưu hơn.

KẾT LUẬN

Sau thời gian nghiên cứu tìm hiểu đề tài và tiến hành thi công đề đồ án tuy gặp không ít khó khăn và trở ngại khi thực hiện đề tài nhưng em cũng đã đạt được kết quả khả quan và đó cũng là động lực để em hoàn thiện đề tài tốt hơn.

Qua đây em xin cảm ơn các anh chị trợ giảng đã góp ý và luôn nhẫn nại với các câu hỏi từ tụi em thăm mắc. Cũng cảm ơn nhà trường đã tạo điều kiện cho chúng em học môn trí tuệ nhân tạo này rất bổ ích và tiếp thu được nhiều kiến thức mới.

Trong quá trình làm đề tài cũng có nhiều vấn đề mà trong khả năng của em có thể chưa giải quyết được, em rất mong được sự hướng dẫn thêm nhiều điều từ thầy Nguyễn Trường Thịnh để tụi em có thêm nhiều kiến thức ngoài những kiến thức thầy đã dạy trong môn học “Trí tuệ nhân tạo”.

TÀI LIỆU THAM KHẢO

1. <https://vdodata.vn/tri-tue-nhan-cao-la-gi-lich-su-phat-trien-tri-tue-nhan-cao-ai/>
2. Các slide powerpoint của Thầy PGS. TS Nguyễn Trường Thịnh.