

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

BỘ MÔN TRÍ TUỆ NHÂN TẠO



HCMUTE

BÁO CÁO CUỐI KỲ

**Phân biệt các loại mèo trên thế giới xử lý real – time
chạy trên webserver (app, ...)**

GVHD: Nguyễn Trường Thịnh

SVTH: Trần Ngọc Đoàn

MSSV: 19146175

Mã học phần: 212ARIN337629

Tp.Hồ Chí Minh, tháng 6 năm 2022

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	3
1.1. Mục đích đề tài	4
1.2. Tổng quan đề tài	4
1.3. Nhiệm vụ và giới hạn đề tài	4
1.4. Phương pháp nghiên cứu	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	6
2.1 Nghiên cứu về cơ sở lý thuyết chung và khái niệm	6
2.1.1 Khái niệm trí tuệ nhân tạo (Artificial Intelligence)	6
2.1.2 Khái niệm học máy (Machine Learning)	8
2.1.3 Khái niệm học sâu (Deep Learning)	8
2.1.4 Tổng quan về thư viện Tensorflow	8
2.2 Mạng nơ ron nhân tạo và thuật toán CNN	9
2.2.1 Khái niệm mạng nơ ron nhân tạo (Artificial Neural Networks)	9
2.2.2 Thuật toán CNN – Convolutional Neural Network	10
CHƯƠNG 3. TRÍCH XUẤT DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH	12
3.1 Giới thiệu về Google Colaboratory	12
3.2 Trích xuất và xử lý dữ liệu	12
3.2.1 Đánh giá sơ bộ và tiền xử lý dữ liệu	12
3.2.2 Truy xuất dữ liệu từ drive và chia thành các tập train, test	14
3.3 Xây dựng mô hình	15
CHƯƠNG 4. HUẤN LUYỆN, ĐÁNH GIÁ MÔ HÌNH VÀ KẾT QUẢ THU ĐƯỢC	16
4.1 Huấn luyện mô hình	16
4.2 Đánh giá mô hình	16
4.3 Thử nghiệm kết quả thu được	17
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	25
5.1 Kết luận	25
5.2 Hướng phát triển	25
PHỤ LỤC	26
TÀI LIỆU THAM KHẢO	27

DANH MỤC HÌNH ẢNH

Hình 1. 29 Classes từ tập dữ liệu

Hình 2. Kết quả thu được

Hình 3. Kết quả phân loại mèo

CHƯƠNG 1. TỔNG QUAN

Trí tuệ nhận tạo hiện đang phát triển với tốc độ “hàm mũ”, có nhiều đóng góp quan trọng vào sản xuất, dịch vụ và đời sống con người. Tuy nhiên, Trí tuệ nhân tạo là một lĩnh vực rất phức tạp và cũng tạo ra nhiều thách thức rất đáng lo ngại. Hiểu biết đúng về Trí tuệ nhận tạo để nắm bắt đúng và kịp thời các cơ hội và thách thức từ Trí tuệ nhân tạo là rất cần thiết đối với mỗi con người, tổ chức và mỗi quốc gia. Bấy ở chương này chúng ta hãy đi tìm hiểu tổng quan về Trí tuệ nhân tạo.

1.1. Mục đích đề tài

Hiện tại, ứng dụng của AI vào việc phân biệt các loài mèo trên thế giới chưa được phổ biến. Việc áp dụng trí thông minh nhân tạo vào đời sống thường nhật sẽ giúp chúng ta có thể nhận diện chính xác về các loài mèo lạ (hoặc lai) chưa được phổ biến ở Việt Nam.

Trong đề tài này sẽ tiếp bước các nghiên cứu về lĩnh vực trí tuệ nhân tạo ứng dụng trong việc phân biệt các loài mèo trên thế giới như đã nêu ở trên trong ngành khoa học máy tính để hoàn thiện và cải tiến, để người dùng có thể tiếp cận và sử dụng như một phần trong cuộc sống thường ngày. Vì vậy mục đích nghiên cứu là: nghiên cứu, ứng dụng và đưa ra những phương pháp của trí tuệ nhân tạo trong việc phân biệt các loài mèo sử dụng mạng nơ ron nhân tạo tích chập (CNN).

1.2. Tổng quan đề tài

Hệ thống phân biệt các loài mèo trên thế giới là sự kết hợp giữa trí tuệ nhân tạo và xử lý real - time trên webserver để nhận diện và phân loại đối tượng.

1.3. Nhiệm vụ và giới hạn đề tài

- Nhiệm vụ: Phân biệt các loài mèo trên thế giới xử lý real – time trên webserver
- Giới hạn:
 - + Đề tài chỉ dừng lại ở việc nghiên cứu và ứng dụng trí tuệ nhân tạo nên vẫn nhiều thiếu sót

- + Phân biệt các loài mèo trên thế giới khá phổ biến ở Việt Nam và các nước Phương Tây, chưa có khả năng phân biệt các loài mèo có đặc điểm ngoại hình, màu lông, ... tương đồng nhau.
- + Độ chính xác của mô hình không quá cao do chưa có được tập dữ liệu tối ưu và toàn diện (tự tạo tập dữ liệu).
- + Nhận dạng các loài mèo xử lý real – time bằng camera máy tính và google colaboratory.

1.4. Phương pháp nghiên cứu

Từ các cơ sở lý thuyết nền tảng đã học từ bộ môn Trí tuệ nhân tạo (Artificial Intelligence) và Học máy (Machine Learning), áp dụng vào việc xây dựng mô hình và xử lý dữ liệu đầu vào.

Nghiên cứu các tài liệu khoa học liên quan đến trí tuệ nhân tạo, các nghiên cứu đã thực hiện trước đây, các đề tài và nghiên cứu khoa học trong và ngoài nước, tham khảo các tài liệu trên Internet và các nguồn khác nhau.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Nghiên cứu về cơ sở lý thuyết chung và khái niệm

2.1.1 Khái niệm trí tuệ nhân tạo (Artificial Intelligence)

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.

AI có ba mức độ khác nhau:

- + Narrow AI: Trí tuệ nhân tạo được cho là hẹp khi máy có thể thực hiện một nhiệm vụ cụ thể tốt hơn so với con người. Nghiên cứu hiện tại về AI hiện đang ở cấp độ này.
- + General AI: Trí tuệ nhân tạo đạt đến trạng thái chung khi nó có thể thực hiện bất kỳ nhiệm vụ sử dụng trí tuệ nào có cùng độ chính xác như con người.
- + Strong AI: AI rất mạnh khi nó có thể đánh bại con người trong nhiều nhiệm vụ cụ thể.

Công nghệ AI được chia làm 4 loại chính:

Loại 1: Công nghệ AI phản ứng.

Công nghệ AI phản ứng có khả năng phân tích những động thái khả thi nhất của chính mình và của đối thủ, từ đó, đưa ra được giải pháp tối ưu nhất.

Một ví dụ điển hình của công nghệ AI phản ứng là Deep Blue. Đây là một chương trình chơi cờ vua tự động, được tạo ra bởi IBM, với khả năng xác định các nước cờ đồng thời dự đoán những bước đi tiếp theo của đối thủ. Thông qua đó, Deep Blue đưa ra những nước đi thích hợp nhất.

Loại 2: Công nghệ AI với bộ nhớ hạn chế

Đặc điểm của công nghệ AI với bộ nhớ hạn chế là khả năng sử dụng những kinh nghiệm trong quá khứ để đưa ra những quyết định trong tương lai. Công nghệ AI này thường kết hợp với cảm biến môi trường xung quanh nhằm mục đích dự đoán những trường hợp có thể xảy ra và đưa ra quyết định tốt nhất cho thiết bị.

Ví dụ như đối với xe không người lái, nhiều cảm biến được trang bị xung quanh xe và ở đầu xe để tính toán khoảng cách với các xe phía trước, công nghệ AI sẽ dự đoán khả năng xảy ra va chạm, từ đó điều chỉnh tốc độ xe phù hợp để giữ an toàn cho xe.

Loại 3: Lý thuyết trí tuệ nhân tạo

Công nghệ AI này có thể học hỏi cũng như tự suy nghĩ, sau đó áp dụng những gì học được để thực hiện một việc cụ thể. Hiện nay, công nghệ AI này vẫn chưa trở thành một phương án khả thi.

Loại 4: Tự nhận thức

Công nghệ AI này có khả năng tự nhận thức về bản thân, có ý thức và hành xử như con người. Thậm chí, chúng còn có thể bộc lộ cảm xúc cũng như hiểu được những cảm xúc của con người. Đây được xem là bước phát triển cao nhất của công nghệ AI và đến thời điểm hiện tại, công nghệ này vẫn chưa khả thi.

2.1.2 Khái niệm học máy (Machine Learning)

Machine Learning còn được gọi là học máy. Bạn có thể viết ứng dụng có AI mà không sử dụng học máy, nhưng bạn phải viết cả triệu triệu dòng code để xây dựng các trường hợp xảy ra.

Học máy là cách để có được AI, máy tự học mà không cần phải code nhiều như không có học máy. Nói cách khác, nếu AI là mục tiêu thì học máy là phương tiện để đạt được mục tiêu đó.

Máy sẽ được “học” bằng cách train nó với một lượng dữ liệu khổng lồ với một thuật toán, thuật toán có khả năng điều chỉnh và xây dựng model.

2.1.3 Khái niệm học sâu (Deep Learning)

Deep Learning được bắt nguồn từ thuật toán Neural network của AI, là một ngành nhỏ của Machine Learning.

Deep learning tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.

Trí tuệ nhân tạo có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, Machine learning nằm ở tầng tiếp theo và Deep learning nằm ở tầng trên cùng.

2.1.4 Tổng quan về thư viện Tensorflow

Với sự bùng nổ của lĩnh vực Trí Tuệ Nhân Tạo – A.I. trong thập kỷ vừa qua, machine learning và deep learning rõ ràng cũng phát triển theo cùng. Và ở thời điểm hiện tại, TensorFlow chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều. Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép bạn tính toán song song trên nhiều máy tính khác

nhau, thậm chí trên nhiều CPU, GPU trong cùng 1 máy hay tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model. Kiến trúc TensorFlow hoạt động được chia thành 3 phần:

- Tiền xử lý dữ liệu
- Dựng model
- Train và ước tính model

Tensor đại diện cho các loại dữ liệu được đưa vào trong Tensorflow. Mỗi thuộc tính trong tensor sẽ có những đặc điểm và tính năng khác nhau với những thuộc tính cơ bản sau đây:

- Rank: Trong các cấu trúc dữ liệu, thuộc tính bậc được hiểu một cách đơn giản chính là sự phân cấp bậc và là căn cứ cho việc phân loại các tensor. Mỗi tensor khi được phân bậc sẽ có tên gọi khác nhau, cụ thể bậc 0 là Scalar, bậc 1 là Vector bậc 2 là Matrix, các bậc cao hơn nữa sẽ được gọi là n-tensor.
- Shape: Đây là thuộc tính chiều của tensor các cấu trúc dữ liệu.
- Type: Kiểu dữ liệu của các element và là thuộc tính type duy nhất có trong tensor. Một tensor chỉ có một loại type duy nhất cho toàn bộ các element có trong tensor. Vì vậy mà cấu trúc dữ liệu có tính thống nhất.

2.2 Mạng nơ ron nhân tạo và thuật toán CNN

2.2.1 Khái niệm mạng nơ ron nhân tạo (Artificial Neural Networks)

Mạng nơ ron nhân tạo (Artificial Neural Networks) là một mô hình xử lý thông tin, cấu thành từ các lớp nơ ron, được ra đời trên cơ sở mô phỏng hoạt động não bộ của sinh vật. Mạng nơ ron nhân tạo gắn kết nhiều nơ ron theo một mô hình nhất định, được trải qua huấn luyện để rút ra được kinh nghiệm, và sử dụng các kinh nghiệm đã có để xử lý các thông tin mới. Mạng nơ ron nhân tạo thường áp dụng vào giải các bài toán nhận dạng mẫu, hoặc dự đoán.

Mạng nơ-ron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơ-ron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơ-ron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơ-ron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơ-ron.

2.2.2 Thuật toán CNN – Convolutional Neural Network

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.

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.

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 (training) 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.

Cách thức hoạt động của thuật toán Deep Learning diễn ra như sau: Các dòng thông tin sẽ được trải qua nhiều lớp cho đến lớp sau cùng. Lấy quy trình học của con người làm ví dụ cụ thể. Qua các lớp đầu tiên sẽ tập trung vào việc học các khái niệm cụ thể hơn trong khi các lớp sâu hơn sẽ sử dụng thông tin đã học để nghiên cứu và phân tích sâu hơn trong các khái niệm trừu tượng. Quy trình xây dựng biểu diễn dữ liệu này được gọi là trích xuất tính năng. Để có thể dễ hình dung về Deep Learning chúng ta sẽ tìm hiểu cách nó hoạt động thông qua một số ví dụ sau. Hãy bắt đầu với một ví dụ đơn giản về Deep Learning ở cấp độ khái niệm. Hãy cùng suy nghĩ làm thế nào để chúng ta có thể nhận biết được một hình nào đó là hình vuông.

Có thể đầu tiên bạn sẽ kiểm tra xem hình đó có 4 cạnh hay không, nếu nó đúng chúng ta sẽ kiểm tra tiếp 4 cạnh này có được kết nối với nhau thành 1 hình tứ giác hay không, nếu đúng chúng ta sẽ kiểm tra tiếp 4 cạnh này có vuông góc với nhau không và chúng có kích thước bằng nhau không. Nếu tất cả đều đúng thì kết quả nó là hình vuông. Nhìn chung thì cũng không có gì phức tạp cả nó chỉ là 1 hệ thống phân cấp các khái niệm. Chẳng hạn như ví dụ ở trên chúng ta đã chia nhiệm vụ xác định hình vuông thành những nhiệm vụ nhỏ và đơn giản hơn. Deep Learning cũng hoạt động tương tự như vậy nhưng ở quy mô lớn hơn.

CHƯƠNG 3. TRÍCH XUẤT DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

3.1 Giới thiệu về Google Colaboratory






























Google Colaboratory (gọi tắt là Google Colab hay Colab) là một sản phẩm của Google Research. Colab dựa trên Jupyter Notebook, người dùng có thể viết và thực thi đoạn mã python thông qua trình duyệt và đặc biệt rất phù hợp với data analysis, machine learning và giáo dục.

Google Colab là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp. Colab cung cấp nhiều loại GPU, thường là Nvidia K80s, T4s, P4s and P100s, tuy nhiên người dùng không thể chọn loại GPU trong Colab, GPU trong Colab thay đổi theo thời gian.

3.2 Trích xuất và xử lý dữ liệu

3.2.1 Đánh giá sơ bộ và tiền xử lý dữ liệu

Sau khi tải tập dữ liệu bao gồm 29 class và 2900 dữ liệu hình ảnh từ google ta có thể đánh giá sơ qua bộ dữ liệu.

Name	Date modified	Type
 Abyssinian	6/18/2022 5:43 PM	File folder
 Bengal	6/18/2022 5:43 PM	File folder
 Burmilla	6/18/2022 5:43 PM	File folder
 Garfield	6/18/2022 5:43 PM	File folder
 Gizmo	6/18/2022 5:43 PM	File folder
 Grumpy	6/18/2022 5:43 PM	File folder
 Havana Brown	6/18/2022 5:43 PM	File folder
 Luna	6/18/2022 5:43 PM	File folder
 Manie Coon	6/18/2022 5:43 PM	File folder
 Meo Anh Long Dai	6/18/2022 5:43 PM	File folder
 Meo Anh Long Ngan	6/18/2022 5:43 PM	File folder
 Meo Ba Tu	6/18/2022 5:43 PM	File folder
 Meo Rung Na-uy	6/18/2022 5:43 PM	File folder
 Meo Tam The	6/18/2022 5:43 PM	File folder
 Meo Xiem	6/18/2022 5:43 PM	File folder
 Mia	6/18/2022 5:43 PM	File folder
 Molly	6/18/2022 5:43 PM	File folder
 Munchkin	6/18/2022 5:43 PM	File folder
 Nala	6/18/2022 5:43 PM	File folder
 Ocicat	6/18/2022 5:43 PM	File folder
 Oriental	6/18/2022 5:43 PM	File folder
 Princess	6/18/2022 5:43 PM	File folder
 Ragdoll	6/18/2022 5:43 PM	File folder
 Scottist Fold	6/18/2022 5:43 PM	File folder
 Simba	6/18/2022 5:43 PM	File folder
 Singapura	6/18/2022 5:43 PM	File folder
 Somali	6/18/2022 5:43 PM	File folder
 Sophie	6/18/2022 5:43 PM	File folder
 Sphynx	6/18/2022 5:43 PM	File folder

Hình 1. 29 classes từ tập dữ liệu

3.2.2 Truy xuất dữ liệu từ drive và chia thành các tập train, test

Khai báo một số thư viện cần thiết:

```
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Dropout, Conv2D, MaxPooling2D, Flatten
```

Truy xuất tập dữ liệu đã tải từ google drive:

```
from google.colab import drive
drive.mount('/content/drive/')
```

Phân chia tập dữ liệu đã tải thành 2 tập train và test

```
import os
train_image_files_path = "/content/drive/MyDrive/AI/Project/training set"
valid_image_files_path = "/content/drive/MyDrive/AI/Project/test set"
```

Xử lý dữ liệu đầu vào của dữ liệu với kích thước mỗi ảnh là 200x200, class model là categorical

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_data_gen = ImageDataGenerator(rescale=1/255)
validation_data_gen = ImageDataGenerator(rescale=1/255)
train_generator = train_data_gen.flow_from_directory(train_image_files_path, target_size=(200, 200), class_mode='categorical')
validation_generator = validation_data_gen.flow_from_directory(valid_image_files_path, target_size=(200, 200), class_mode='categorical')
```

Kết quả thu được khi chạy đoạn chương trình:

Found 2900 images belonging to 29 classes.

Found 580 images belonging to 29 classes.

Có thể thấy được sau khi tách dữ liệu ta được 2900 dữ liệu thuộc 29 classes ở tập dữ liệu train và 580 dữ liệu thuộc 29 classes ở tập dữ liệu test.

3.3 Xây dựng mô hình

Xây dựng mô hình CNN bằng Sequential từ thư viện keras.models với 1 lớp input với dữ liệu đầu vào là dữ liệu với kích thước 200x200x3, 3 lớp ẩn và lớp output với 29 classes đầu ra:

```
model = Sequential ()
model.add(Conv2D(32,(3,3), activation='relu', input_shape = (200,200,3)))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(64,(3,3), activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Conv2D(128,(3,3), activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Flatten())
model.add(Dense(512, activation=tf.nn.relu))
model.add(Dense(29, activation=tf.nn.softmax))
from tensorflow.keras.optimizers import Adam
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy', metrics=['acc'])
```

CHƯƠNG 4. HUẤN LUYỆN, ĐÁNH GIÁ MÔ HÌNH VÀ KẾT QUẢ THU ĐƯỢC

4.1 Huấn luyện mô hình

Huấn luyện mô hình đã tạo với tập huấn luyện là `train_generator`, tập thẩm định là `validation_generator`, số lần học là 100 lần:

```
history = model.fit(train_generator, steps_per_epoch=3, epochs=100,  
                    verbose=1, validation_data = validation_generator,  
                    validation_steps=3)
```

Kết quả thu được ở 5 lần huấn luyện thu được ta có thể thấy độ chính xác đối với tập train là 87% và độ chính xác của mô hình sau khi chạy với tập thẩm định là 85%

```
Epoch 95/100  
3/3 [=====] - 1s 553ms/step - loss: 0.5647 -  
acc: 0.8646 - val_loss: 0.4810 - val_acc: 0.8438  
Epoch 96/100  
3/3 [=====] - 2s 811ms/step - loss: 0.6287 -  
acc: 0.8646 - val_loss: 0.4399 - val_acc: 0.8958  
Epoch 97/100  
3/3 [=====] - 2s 818ms/step - loss: 0.3517 -  
acc: 0.9167 - val_loss: 0.4709 - val_acc: 0.9062  
Epoch 98/100  
3/3 [=====] - 1s 294ms/step - loss: 0.4469 -  
acc: 0.8750 - val_loss: 0.3332 - val_acc: 0.9062  
Epoch 99/100  
3/3 [=====] - 2s 566ms/step - loss: 0.4459 -  
acc: 0.9062 - val_loss: 0.6501 - val_acc: 0.8021  
Epoch 100/100  
3/3 [=====] - 1s 304ms/step - loss: 0.5602 - acc:  
0.8646 - val_loss: 0.6071 - val_acc: 0.8125
```

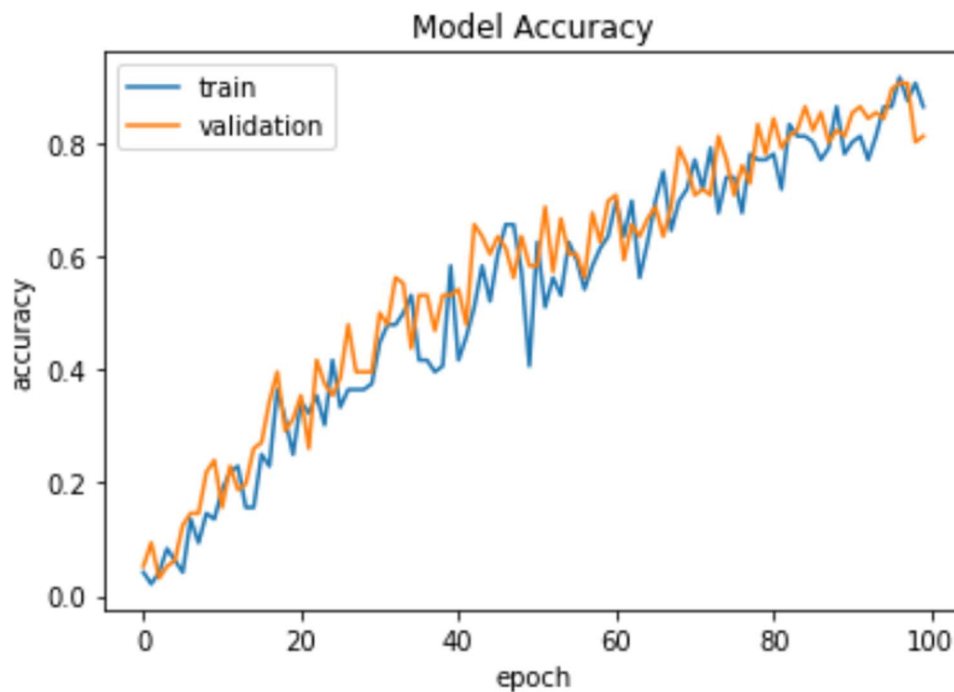
4.2 Đánh giá mô hình

Từ kết quả mô hình sau khi huấn luyện như trên ta có thể đánh giá sơ qua kết quả của mô hình như sau: khi huấn luyện độ chính xác của mô hình khá cao đạt gần 87% ở kết quả thu được nhưng khi thẩm định thì kết quả thu được độ chính xác chỉ dao động ở 85% cho thấy được chất lượng mô hình và data chưa thực sự tối ưu nhưng vẫn trong khoảng chấp nhận được.

Tiến hành vẽ biểu đồ thể hiện đánh giá thu được sau mỗi lần huấn luyện với thư viện matplotlib:

```
plt.plot(history.history['acc'])  
plt.plot(history.history['val_acc'])  
plt.title('Model Accuracy')  
plt.ylabel('accuracy')  
plt.xlabel('epoch')  
plt.legend(['train', 'validation'], loc='upper-left')
```

Kết quả thu được:



Hình 2. Kết quả thu được

4.3 Thử nghiệm kết quả thu được

Ta tiến hành thử nghiệm kết quả thu được của mô hình đã huấn luyện bằng cách tải mô hình đã huấn luyện bằng load_model từ thư viện keras

```
model.save('meo.h5')
```

```
from tensorflow.keras.models import load_model
```

```
model=load_model('meo.h5')
```

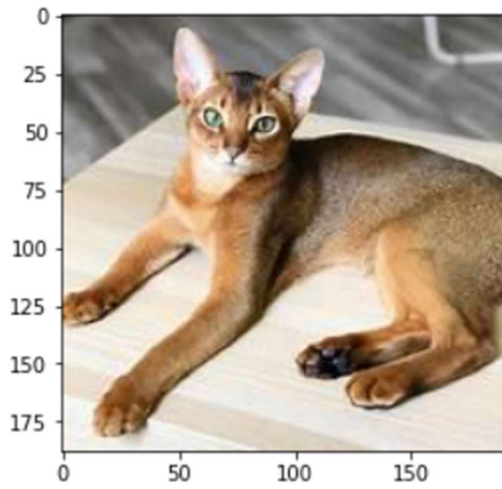
Ta tiến hành đặt tên cho 29 classes ứng với 29 loại mèo:

```
label = ['Abyssinian', 'Bengal', 'Burmilla', 'Garfield', 'Gizmo', 'Grumpy',  
'Havana Brown', 'Luna', 'Manie Coon', 'Meo Anh Long Dai',  
'Meo Anh Long Ngan', 'Meo Ba Tu', 'Meo Rung Na-uy', 'Meo Tam The',  
'Meo Xiem', 'Mia', 'Molly', 'Munchkin', 'Nala', 'Ocicat', 'Oriental', 'Princess',  
'Ragdoll', 'Scottist Fold', 'Simba', 'Singapura', 'Somali', 'Sophie', 'Sphynx']
```

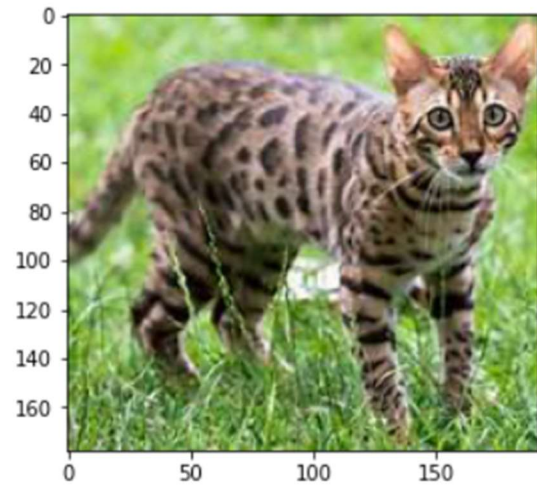
Sử dụng image từ thư viện keras.preprocessing và mpimg từ thư viện matplotlib.image để thử nghiệm với các kết quả bên ngoài

```
uploaded=files.upload()  
for fn in uploaded.keys():  
    #predicting images  
    path='/content/' + fn  
    #In ảnh đọc được  
    plt.imshow(mpimg.imread(path))  
    img=image.load_img(path,target_size=(200,200))  
    x=image.img_to_array(img)  
    x=np.expand_dims(x,axis=0)  
    images=np.vstack([x])  
    y_predict = model.predict(images,batch_size=10)  
    print (y_predict)  
    print ('Giá trị dự đoán: ', label[np.argmax(y_predict)])  
    plt.show()
```

Giá trị dự đoán: Abyssinian



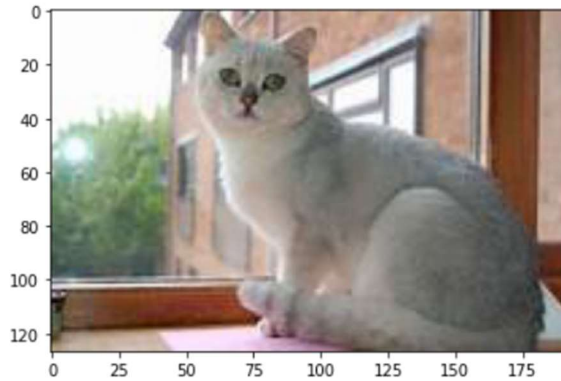
Giá trị dự đoán: Bengal



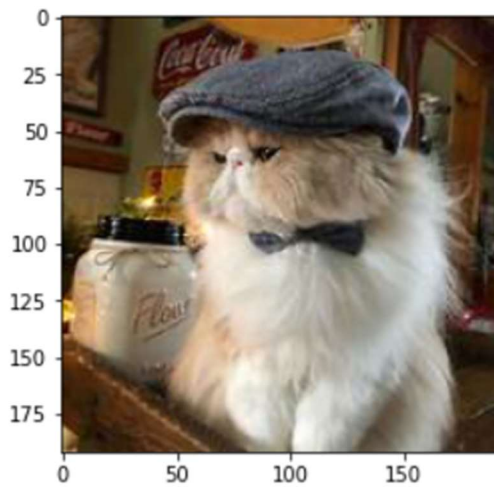
Giá trị dự đoán: Garfield



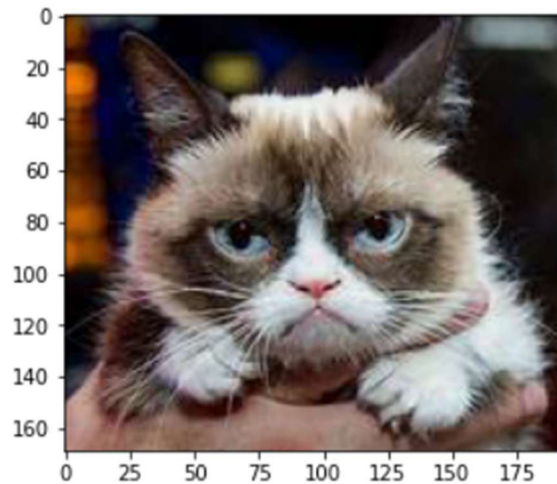
Giá trị dự đoán: Burmilla



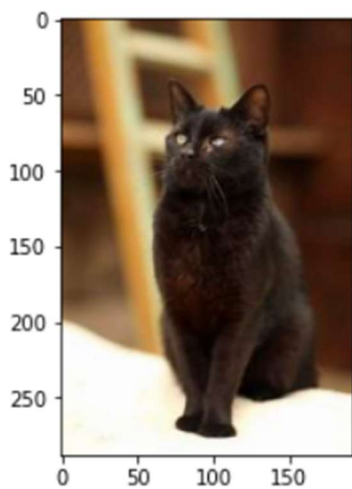
Giá trị dự đoán: Gizmo



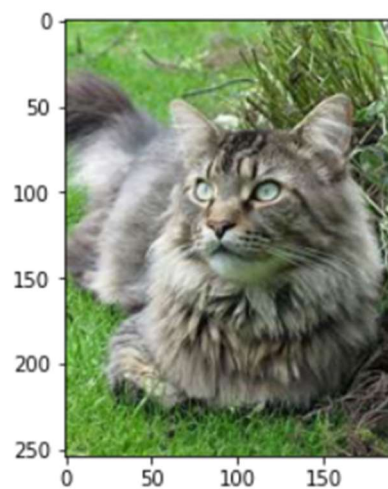
Giá trị dự đoán: Grumpy



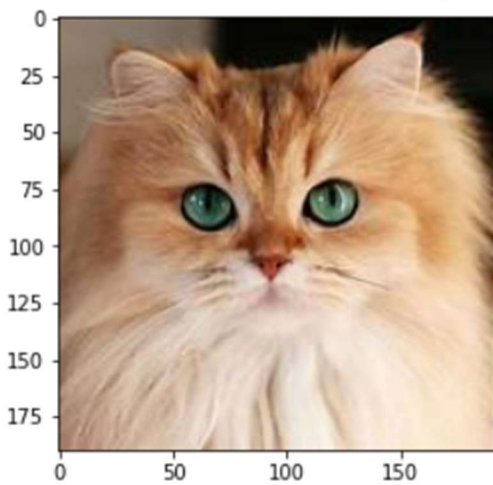
Giá trị dự đoán: Havana Brown



Giá trị dự đoán: Manie Coon



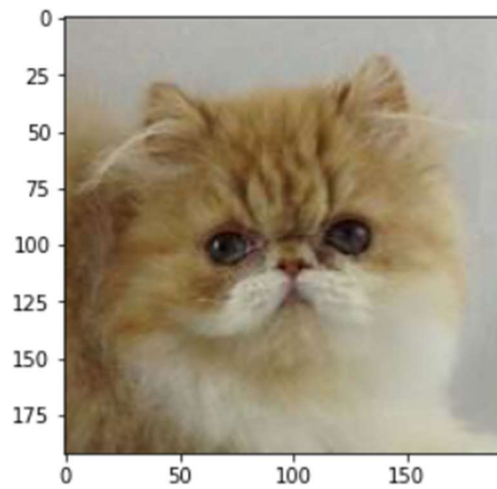
Giá trị dự đoán: Meo Anh Long Dai



Giá trị dự đoán: Meo Anh Long Ngan



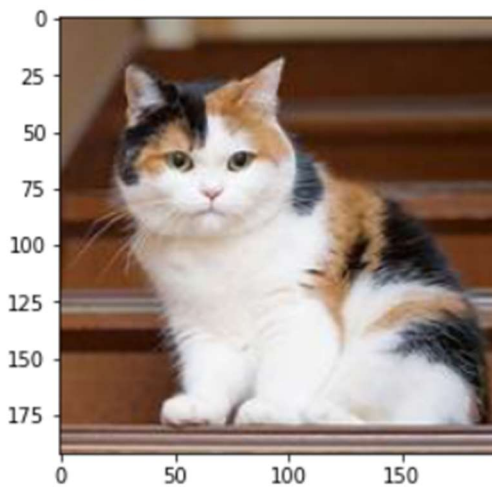
Giá trị dự đoán: Meo Ba Tu



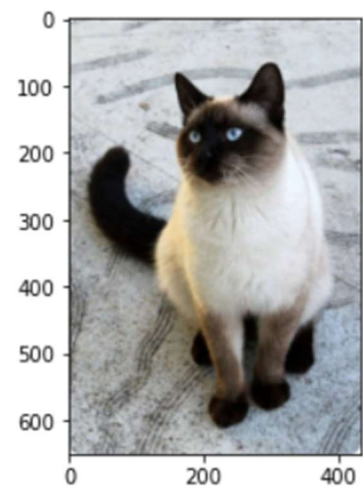
Giá trị dự đoán: Meo Rung Na-uy



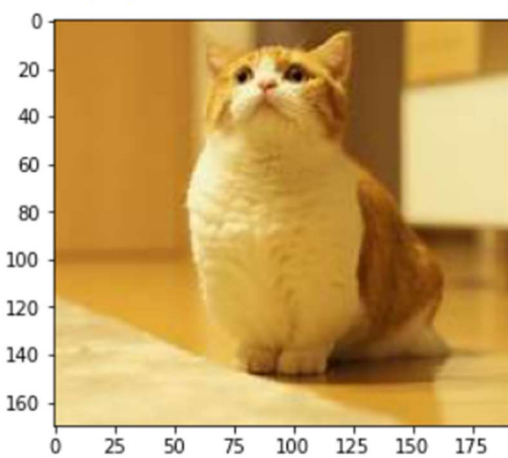
Giá trị dự đoán: Meo Tam The



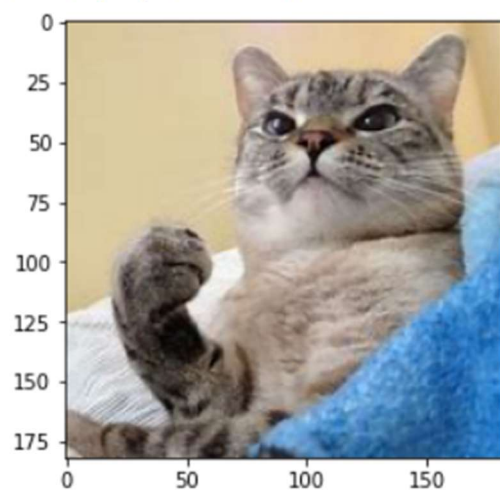
Giá trị dự đoán: Meo Xiem



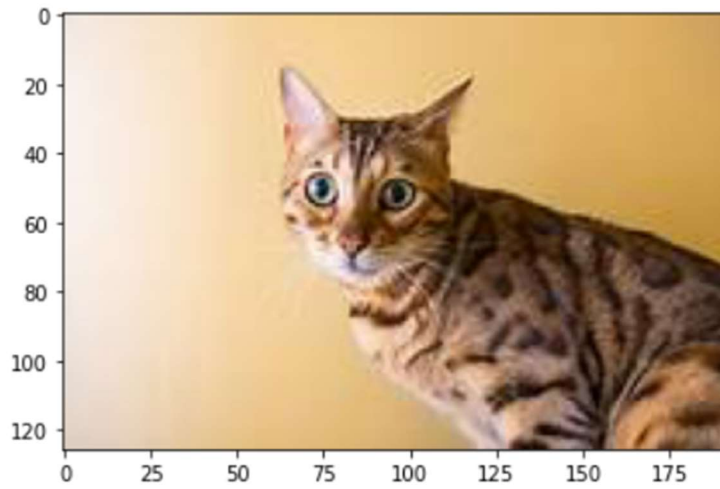
Giá trị dự đoán: Munchkin



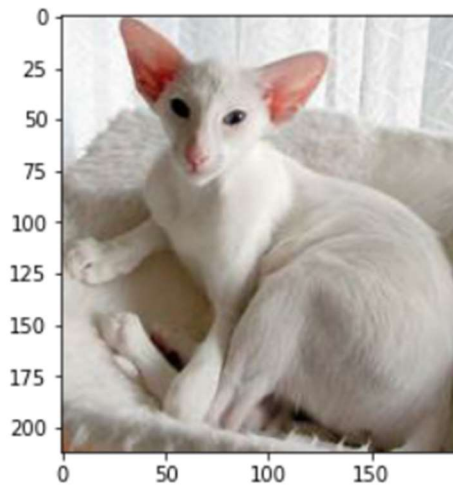
Giá trị dự đoán: Nala



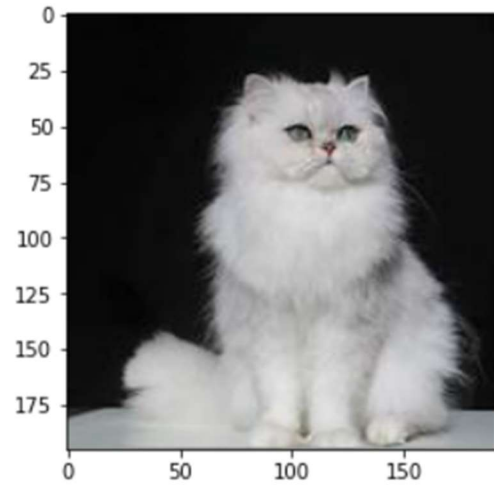
Giá trị dự đoán: Ocicat



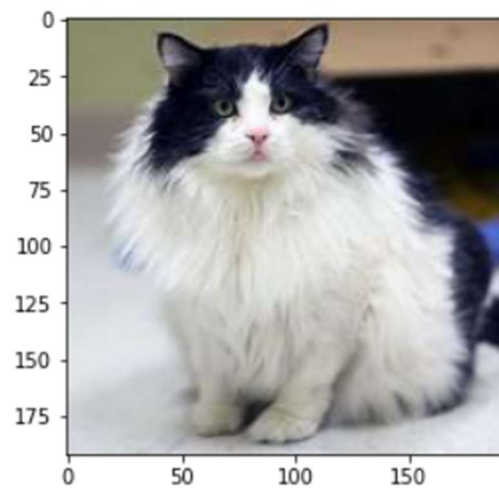
Giá trị dự đoán: Oriental



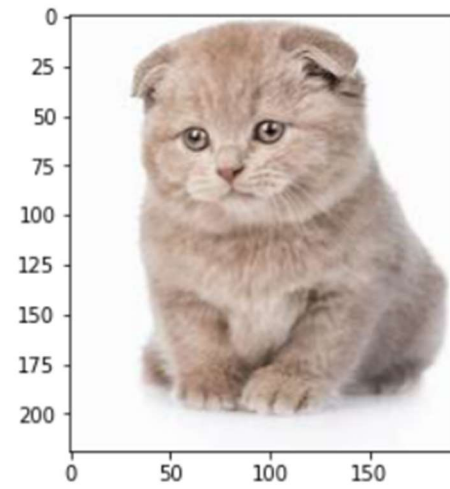
Giá trị dự đoán: Princess



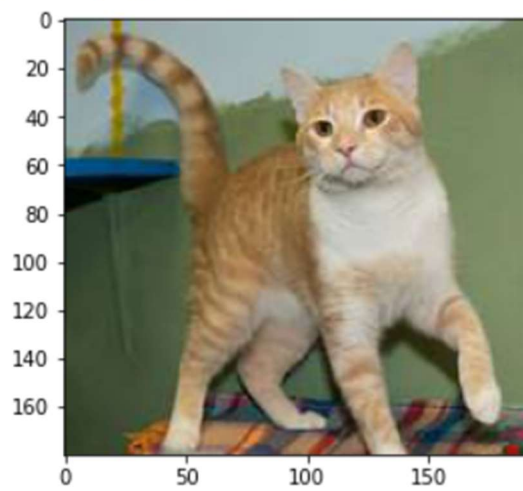
Giá trị dự đoán: Ragdoll



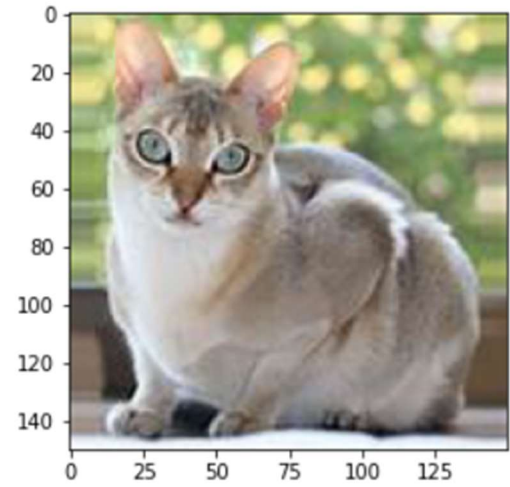
Giá trị dự đoán: Scottist Fold



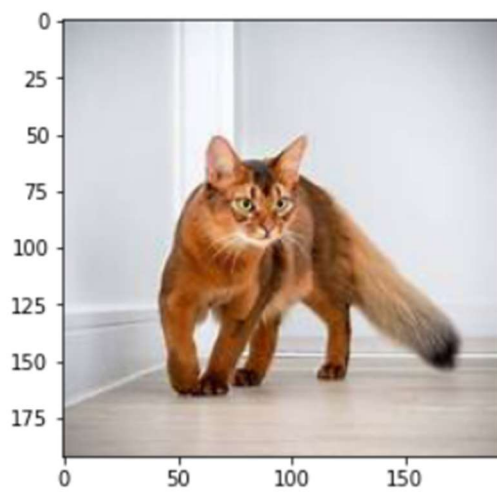
Giá trị dự đoán: Simba



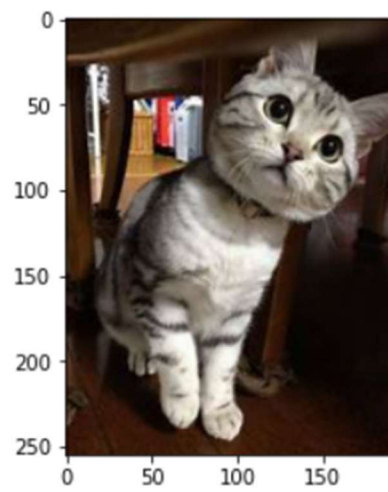
Giá trị dự đoán: Singapura



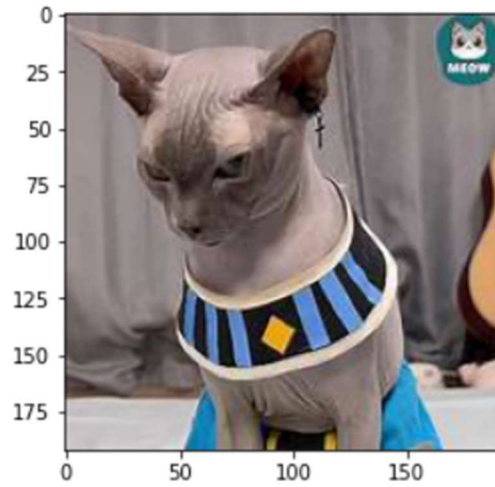
Giá trị dự đoán: Somali



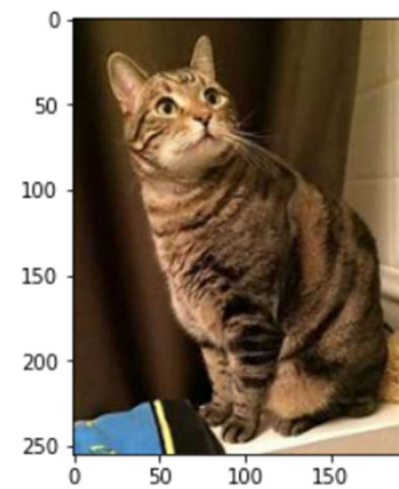
Giá trị dự đoán: Sophie



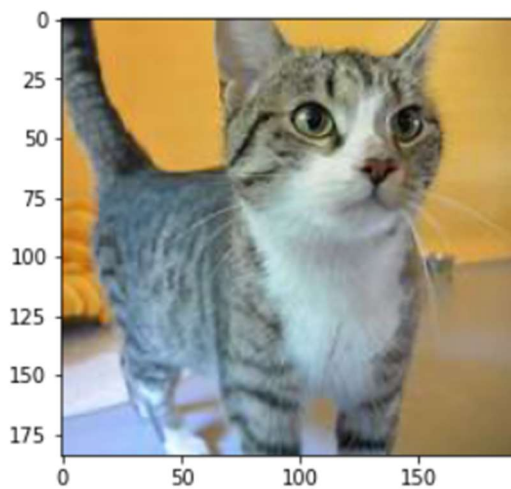
Giá trị dự đoán: Sphynx



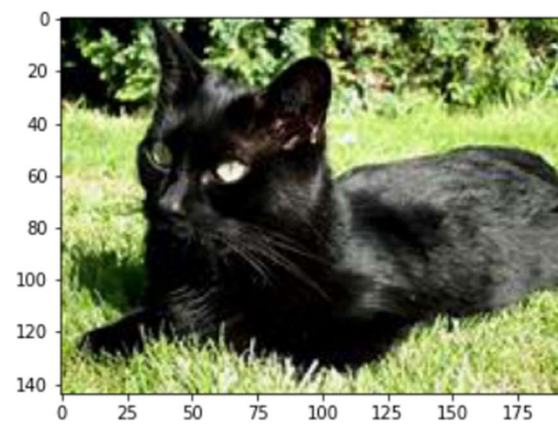
Giá trị dự đoán: Bengal



Giá trị dự đoán: Munchkin



Giá trị dự đoán: Gizmo



Hình 3. Kết quả phân loại mèo

Có thể thấy được từ 29 giống mèo trên, mô hình đem lại độ chính xác nằm trong khoảng có thể chấp nhận được.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình thực hiện đồ án nghiên cứu và ứng dụng trí tuệ nhân tạo (Artificial Intelligence) phân biệt các loài mèo trên thế giới đã hoàn được một số công việc cơ bản như sau:

- Hoàn thành các công việc đúng thời hạn
- Nghiên cứu được các cơ sở lý thuyết và chọn lựa được các phương pháp, xây dựng mô hình huấn luyện.
- Hoàn thành việc xây dựng và viết code chạy mô hình trên google colab.
- Vì thời gian thực hiện đồ án nghiên cứu và ứng dụng trí tuệ nhân tạo (Artificial Intelligence) phân biệt các loài mèo trên thế giới là có giới hạn và còn nhiều thiếu sót về kiến thức dẫn đến đồ án còn nhiều sai sót và nhiều lỗi cũng như độ chính xác và hàm lượng khoa học chưa cao.

5.2 Hướng phát triển

Do phụ thuộc vào nhiều yếu tố khách quan lẫn chủ quan, trình độ có giới hạn nên hiện tại đề tài vẫn còn nhiều vấn đề chưa thể thực hiện, cần tiếp tục nghiên cứu và thực nghiệm để hoàn thiện mô hình phân biệt các loài mèo trên thế giới. Vì vậy hướng phát triển tiếp theo của đề tài cần tập trung vào những cần đề sau để hoàn thiện mô hình đưa vào thực tế:

- + Hoàn thiện mô hình và kiểm nghiệm những tính toán.
- + Thực hiện khảo sát thực tế để biết được nhiều yếu tố khách quan hơn.
- + Thử nghiệm trên các mô hình khác nhau để đưa ra kết quả tốt hơn.
- + Bổ sung tập dữ liệu để dữ liệu phong phú hơn cũng như tăng thêm độ chính xác cho mô hình.

PHỤ LỤC

Link Github: <https://github.com/DoanAI/PROJECT.git>

TÀI LIỆU THAM KHẢO

- [1] Huỳnh Thanh Việt (2020), “*Tìm hiểu về mạng nơ-ron tích chập (convolution neural networks)*”, thanhviet.com, ngày truy cập 20/06/2022, <https://thanhvie.com/tim-hieu-ve-mang-no-ron-tich-chap-convolutional-neural-networks/>
- [2] Trần Đức Trung (2020), “*Tìm hiểu về Convolutional Neural Network và làm một ví dụ nhỏ về phân loại ảnh*”, VIBLO.asia, ngày truy cập 20/06/2022, <https://viblo.asia/p/tim-hieu-ve-convolutional-neural-network-va-lam-mot-vi-du-nho-ve-phan-loai-anh-aWj53WXo56m>
- [3] Lê Thị Thu Hằng, “*Nghiên cứu về mạng neural tích chập và ứng dụng cho bài toán nhận dạng biển số xe*”, trường đại học Công Nghệ, 2016.