**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 15: Xây dựng hệ thống phát triển mô hình phân loại ảnh để nhận diện các loài động vật**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210221 | Lê Văn Chiến | DC.CNTT12.10.1 |
| **2** | 20210332 | Hoàng Hữu Dũng | DC.CNTT12.10.1 |
| **3** | 20210144 | Nguyễn Đức Minh Hiếu | DC.CNTT12.10.1 |
| **4** | 20210346 | Nguyễn Đình Vỹ | DC.CNTT12.10.1 |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 15: Xây dựng hệ thống phát triển mô hình phân loại ảnh để nhận diện các loài động vật**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20210221 | Lê Văn Chiến | DC.CNTT12.10.1 |
| **2** | 20210332 | Hoàng Hữu Dũng | DC.CNTT12.10.1 |
| **3** | 20210144 | Nguyễn Đức Minh Hiếu | DC.CNTT12.10.1 |
| **4** | 20210346 | Nguyễn Đình Vỹ | DC.CNTT12.10.1 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 15 : Xây dựng hệ thống phát triển mô hình phân loại ảnh để nhận diện các loài động vật**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín chỉ:**  **XATGMT.03.K12.01.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  *Lương Thị Hồng Lan* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **LÊ VĂN CHIẾN** | **HOÀNG HỮU DŨNG** | **NGUYỄN ĐỨC MINH HIẾU** | **NGUYỄN ĐÌNH VỸ** |
| --- | --- | --- | --- | --- | --- | --- |
| 20210221 | 20210332 | 20210144 | 20210346 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

LỜI CẢM ƠN

Tri thức là một biển cả mênh mông, mỗi con người đều cần nỗ lực học tập để trang bị cho mình hành trang bước vào cuộc sống. Chúng em sinh ra và lớn lên trong sự nuôi dưỡng của cha mẹ và may mắn được các thầy cô "gieo mầm tri thức", để từ đó có nền tảng trở thành công dân tốt, đóng góp vào sự phát triển của xã hội.

Để có thể hoàn thành đề tài "Xây dựng hệ thống phát triển mô hình phân loại ảnh để nhận diện các loài động vật", chúng em xin gửi lời cảm ơn chân thành đến Ban Giám Hiệu Trường Đại Học Công Nghệ Đông Á cùng tập thể quý thầy cô khoa Công Nghệ Thông Tin đã tận tình giảng dạy, trang bị cho chúng em kiến thức và kỹ năng cần thiết trong suốt quá trình học tập tại trường.

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến cô Lương Thị Hồng Lan, người đã trực tiếp hướng dẫn, chỉ bảo tận tình và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài này. Những kiến thức và kinh nghiệm quý báu mà cô chia sẻ không chỉ giúp chúng em hoàn thành tốt báo cáo mà còn là nền tảng cho những chặng đường học tập và làm việc trong tương lai.

Mặc dù đã cố gắng hoàn thành báo cáo với sự nỗ lực cao nhất, nhưng do thời gian và năng lực còn hạn chế, chắc chắn không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý, nhận xét từ quý thầy cô để có thể cải thiện và hoàn thiện bản thân tốt hơn nữa.

Cuối cùng, chúng em xin kính chúc Ban Giám Hiệu nhà trường, cô Lương Thị Hồng Lan và toàn thể quý thầy cô giáo sức khỏe dồi dào, thành công trong sự nghiệp và hạnh phúc trong cuộc sống.

Trân trọng!

MỤC LỤC

[LỜI CẢM ƠN 6](#_Toc184513552)

[MỤC LỤC 7](#_Toc184513553)

[DANH MỤC HÌNH ẢNH 9](#_Toc184513554)

[CHƯƠNG 1. TỔNG QUAN 10](#_Toc184513555)

[1.1. Tổng quan về thị giác máy tính 10](#_Toc184513556)

[1.1.1. Thị giác máy tính là gì ? 10](#_Toc184513557)

[1.1.2. Cách thị giác máy tính hoạt động 10](#_Toc184513558)

[1.1.3. Lịch sử phát triển của thị giác máy tính 11](#_Toc184513559)

[1.2. Tổng quan về nhận dạng đối tượng 12](#_Toc184513560)

[1.2.1. Nhận dạng đối tượng là gì ? 12](#_Toc184513561)

[1.2.2. Sự khác biệt giữa nhận dạng đối tượng và phát hiện đối tượng 13](#_Toc184513562)

[1.2.3. Cách nhận dạng đối tượng hoạt động 14](#_Toc184513563)

[1.3. Tổng quan về học máy 14](#_Toc184513564)

[1.3.1. Khái niệm học máy 14](#_Toc184513565)

[1.3.2. Cách học máy hoạt động 14](#_Toc184513566)

[1.3.3. Phương pháp học máy 15](#_Toc184513567)

[1.3.4. Các thuật toán học máy phổ biến 15](#_Toc184513568)

[1.3.5. Các trường hợp sử dụng máy học trong thế giới thực 16](#_Toc184513569)

[1.4. Tổng quan về Deep Learning 17](#_Toc184513570)

[1.4.1. Khái niệm Deep Learning 17](#_Toc184513571)

[1.4.2. Cách Deep Learning hoạt động 18](#_Toc184513572)

[1.4.3. Các loại mô hình trong Deep Learning 18](#_Toc184513573)

[1.5. Các kỹ thuật nhận diện đối tượng bằng cách sử dụng hình ảnh 19](#_Toc184513574)

[1.5.1. Kỹ thuật xử lý ảnh truyền thống 19](#_Toc184513575)

[1.5.2. Kỹ thuật dựa trên mạng neuron 21](#_Toc184513576)

[1.6. Thư viện Framework TensorFlow/Keras 25](#_Toc184513577)

[CHƯƠNG 2. XÂY DỰNG HỆ THỐNG 30](#_Toc184513578)

[2.1. Yêu cầu bài toán 30](#_Toc184513579)

[2.2. Phân tích bài toán 31](#_Toc184513580)

[2.3. Xây dựng hệ thống với mô hình VGG-16 31](#_Toc184513581)

[2.3.1. Tổng quát 31](#_Toc184513582)

[2.3.2. Mục tiêu của mô hình VGG-16 32](#_Toc184513583)

[2.3.3. Cấu tạo của VGG-16 35](#_Toc184513584)

[2.3.4. Quy trình hoạt động của VGG-16 36](#_Toc184513585)

[2.3.5. Vai trò của VGG-16 39](#_Toc184513586)

[2.4. Xây dựng giao diện chương trình 40](#_Toc184513587)

[CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM 42](#_Toc184513588)

[3.1. Dữ liệu đầu vào 42](#_Toc184513589)

[3.2. Độ đo đánh giá 42](#_Toc184513590)

[3.2.1. Công thức tính độ đo 42](#_Toc184513591)

[3.2.2. Kết quả đo 43](#_Toc184513592)

[3.3. Giao diện chương trình 48](#_Toc184513593)

[3.4. Kết quả thực nghiệm 49](#_Toc184513594)

[3.5. Kết quả đạt được 51](#_Toc184513595)

[3.6. Hạn chế 51](#_Toc184513596)

[DANH MỤC TÀI LIỆU THAM KHẢO 53](#_Toc184513597)

DANH MỤC HÌNH ẢNH

|  |  |  |
| --- | --- | --- |
| Số hiệu | Tên | Trang |
| Hình 1.1.2 | Cách Thị giác máy tính hoạt động | 12 |
| Hình 1.2.1 | Nhận dạng các loại đối tượng khác nhau | 13 |
| Hình 1.2.2 | Nhận dạng đối tượng (trái) và phát hiện đối tượng (phải). | 14 |
| Hình 1.2.3 | Kỹ thuật học máy và học sâu để nhận dạng đối tượng | 14 |
| Hình 1.4.1 | Minh họa cách hoạt động của thuật toán HOG | 18 |
| Hình 1.4.2.1 | Cấu tạo đơn giản của CNN | 21 |
| Hình 1.4.2.2 | Cách hoạt động của CNN | 21 |
| Hình 1.4.2.3 | Cách hoạt động của R-CNN | 22 |
| Hình 1.4.2.4 | Cách hoạt động của YOLO | 24 |
| Hình 2.3.1.1 | Mô hình xử lý VGG-16 | 27 |
| Hình 2.3.1.2 | Cấu tạo của VGG-16 | 30 |
| Hình 3.1 | Giao diện chính chương trình | 38 |
| Hình 3.2.1. | Kết quả phân loại động vật | 39 |
| Hình 3.2.2 | Kết quả phân loại động vật | 39 |
| Hình 3.2.3 | Kết quả sau khi train | 40 |

1. TỔNG QUAN
   1. Tổng quan về thị giác máy tính
      1. Thị giác máy tính là gì ?

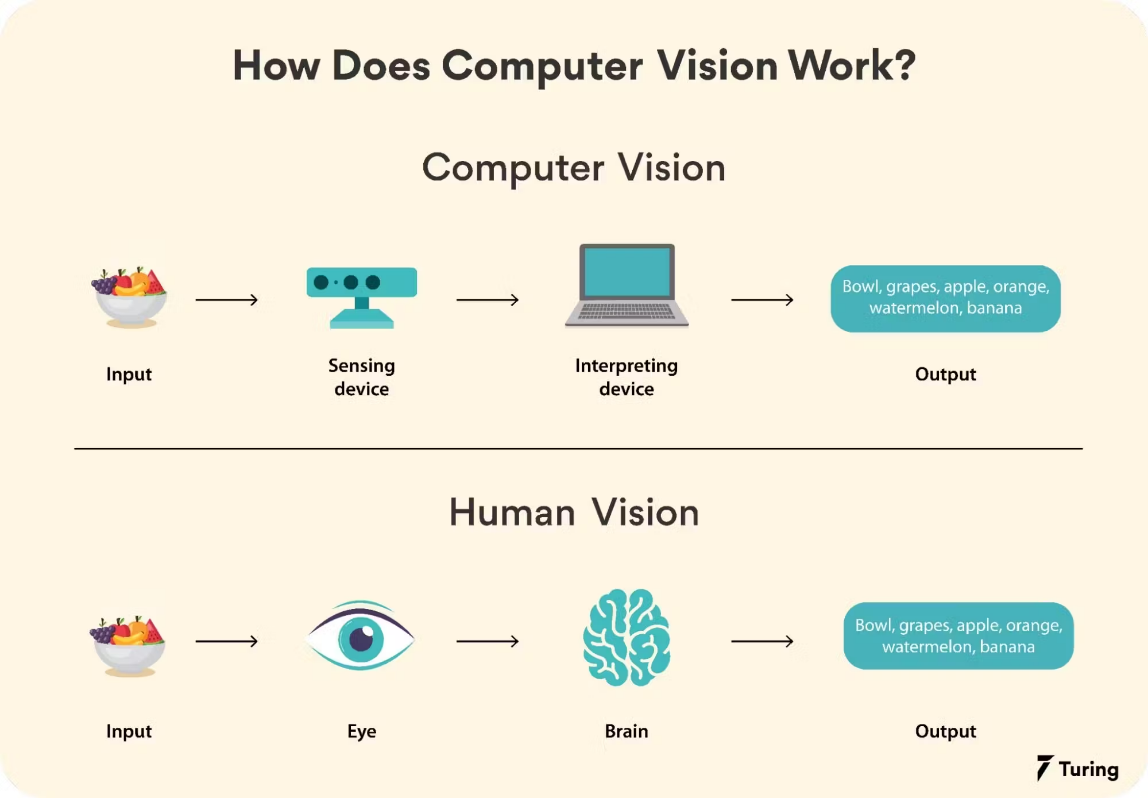
Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo (AI) sử dụng machine learning và mạng nơ-ron để dạy máy tính và hệ thống cách trích xuất thông tin có ý nghĩa từ hình ảnh kỹ thuật số, video và các đầu vào trực quan khác, cũng như đưa ra khuyến nghị hoặc thực hiện hành động khi chúng nhìn thấy lỗi hoặc sự cố.

Thị giác máy tính hoạt động giống như thị giác của con người, ngoại trừ việc con người có lợi thế hơn. Thị giác của con người có lợi thế là có nhiều bối cảnh trong suốt cuộc đời để đào tạo cách phân biệt các vật thể, chúng cách xa bao nhiêu, chúng có đang di chuyển hay có gì đó không ổn với hình ảnh không.

Thị giác máy tính đào tạo máy móc thực hiện các chức năng này, nhưng phải thực hiện trong thời gian ngắn hơn nhiều bằng máy ảnh, dữ liệu và thuật toán thay vì võng mạc, dây thần kinh thị giác và vỏ não thị giác. Bởi vì một hệ thống được đào tạo để kiểm tra sản phẩm hoặc theo dõi tài sản sản xuất có thể phân tích hàng nghìn sản phẩm hoặc quy trình mỗi phút, nhận thấy những khiếm khuyết hoặc vấn đề không thể nhận thấy, nên nó có thể nhanh chóng vượt qua khả năng của con người. [1]

* + 1. Cách thị giác máy tính hoạt động

Thay vì nhìn vào toàn bộ hình ảnh như chúng ta vẫn làm, máy tính sẽ chia hình ảnh thành các pixel và sử dụng các giá trị RGB của từng pixel để hiểu xem hình ảnh có chứa các đặc điểm quan trọng hay không. Các thuật toán thị giác máy tính tập trung vào một đốm pixel tại một thời điểm và sử dụng một hạt nhân hoặc bộ lọc chứa các giá trị nhân pixel để phát hiện cạnh của các đối tượng. Máy tính nhận dạng và phân biệt hình ảnh bằng cách quan sát mọi khía cạnh của hình ảnh bao gồm màu sắc, bóng đổ và các bản vẽ đường nét.



*Hình 1.1.2. Cách Thị giác máy tính hoạt động*

Ngày nay, chúng ta sử dụng mạng nơ-ron tích chập (CNN) để lập mô hình và đào tạo. CNN là mạng nơ-ron đặc biệt, được thiết kế riêng để xử lý dữ liệu pixel, được sử dụng để nhận dạng và xử lý hình ảnh. Lớp tích chập chứa nhiều nơ-ron và tenxơ. Chúng xử lý các tập dữ liệu lớn bằng cách học cách điều chỉnh các giá trị của chúng để phù hợp với các đặc điểm quan trọng để phân biệt các lớp khác nhau. Điều này được thực hiện bằng cách đào tạo mô hình một cách chuyên sâu.

Một cách giúp máy tính học cách nhận dạng mẫu là cung cấp cho chúng nhiều hình ảnh được gắn nhãn để chúng có thể tìm kiếm các mẫu trong tất cả các phần tử. Ví dụ, nếu bạn đưa một triệu bức ảnh về một con 'sư tử' vào máy tính, chúng sẽ trải qua các thuật toán phân tích màu sắc, hình dạng, khoảng cách giữa các hình dạng, ranh giới giữa các vật thể, v.v., để chúng trở thành các mặt cắt. Sau đó, máy tính có thể sử dụng trải nghiệm khi đưa vào các hình ảnh khác không có nhãn để biết liệu hình ảnh được hiển thị có phải là hình ảnh của một con sư tử hay không. [1]

* + 1. Lịch sử phát triển của thị giác máy tính

Các nhà khoa học và kỹ sư đã cố gắng phát triển các phương pháp để máy móc có thể nhìn và hiểu dữ liệu hình ảnh trong khoảng 60 năm. Thí nghiệm bắt đầu vào năm 1959 khi các nhà sinh lý học thần kinh cho một con mèo xem một loạt hình ảnh, cố gắng liên hệ phản ứng trong não của nó. Họ phát hiện ra rằng nó phản ứng đầu tiên với các cạnh cứng hoặc đường thẳng và về mặt khoa học, điều này có nghĩa là quá trình xử lý hình ảnh bắt đầu bằng các hình dạng đơn giản như các cạnh thẳng.

Vào khoảng thời gian đó, công nghệ quét hình ảnh máy tính đầu tiên đã được phát triển, cho phép máy tính số hóa và thu thập hình ảnh. Một cột mốc khác đã đạt được vào năm 1963 khi máy tính có thể chuyển đổi hình ảnh hai chiều thành dạng ba chiều. Vào những năm 1960, AI nổi lên như một lĩnh vực nghiên cứu hàn lâm và nó cũng đánh dấu sự khởi đầu của nhiệm vụ AI nhằm giải quyết vấn đề thị giác của con người.

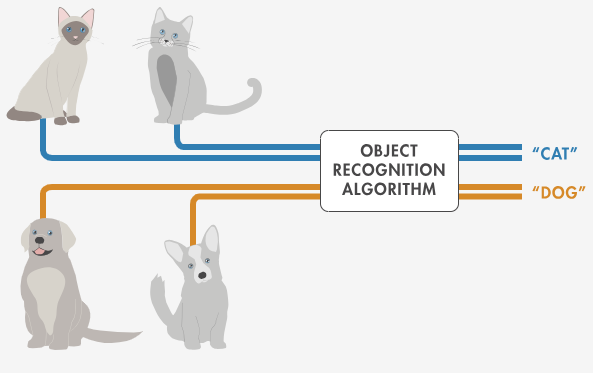
Năm 1974 chứng kiến ​​sự ra đời của công nghệ nhận dạng ký tự quang học (OCR), có thể nhận dạng văn bản được in bằng bất kỳ phông chữ hoặc kiểu chữ nào. Tương tự như vậy, nhận dạng ký tự thông minh (ICR) có thể giải mã văn bản viết tay bằng mạng nơ-ron. Kể từ đó, OCR và ICR đã tìm được đường vào xử lý tài liệu và hóa đơn, nhận dạng biển số xe, thanh toán di động, chuyển đổi máy và các ứng dụng phổ biến khác.

Năm 1982, nhà khoa học thần kinh David Marr đã xác định rằng thị giác hoạt động theo thứ bậc và giới thiệu các thuật toán để máy móc phát hiện các cạnh, góc, đường cong và các hình dạng cơ bản tương tự. Đồng thời, nhà khoa học máy tính Kunihiko Fukushima đã phát triển một mạng lưới các tế bào có thể nhận dạng các mẫu. Mạng lưới này, được gọi là Neocognitron, bao gồm các lớp tích chập trong mạng nơ-ron.

Đến năm 2000, trọng tâm nghiên cứu là nhận dạng đối tượng; và đến năm 2001, các ứng dụng nhận dạng khuôn mặt thời gian thực đầu tiên đã xuất hiện. Việc chuẩn hóa cách gắn thẻ và chú thích các tập dữ liệu trực quan đã xuất hiện trong những năm 2000. Năm 2010, tập dữ liệu ImageNet đã có sẵn. Nó chứa hàng triệu hình ảnh được gắn thẻ trên một nghìn lớp đối tượng và cung cấp nền tảng cho CNN và các mô hình học sâu được sử dụng ngày nay. Năm 2012, một nhóm từ Đại học Toronto đã đưa CNN vào cuộc thi nhận dạng hình ảnh. Mô hình có tên AlexNet đã giảm đáng kể tỷ lệ lỗi khi nhận dạng hình ảnh. Sau bước đột phá này, tỷ lệ lỗi đã giảm xuống chỉ còn vài phần trăm. [1]

* 1. Tổng quan về nhận dạng đối tượng
     1. Nhận dạng đối tượng là gì ?

Nhận dạng đối tượng là một kỹ thuật thị giác máy tính để xác định các đối tượng trong hình ảnh hoặc video. Nhận dạng đối tượng là đầu ra chính của thuật toán học sâu và học máy. Khi con người nhìn vào một bức ảnh hoặc xem video, chúng ta có thể dễ dàng phát hiện ra người, vật thể, cảnh và chi tiết trực quan. Mục tiêu là dạy máy tính làm những gì con người có thể làm một cách tự nhiên: đạt được mức độ hiểu biết về nội dung của hình ảnh. [2]

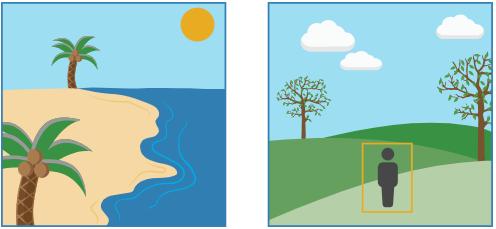


*Hình 1.2.1. Sử dụng nhận dạng đối tượng để xác định các loại đối tượng khác nhau.*

Nhận dạng đối tượng là công nghệ then chốt đằng sau xe không người lái, cho phép chúng nhận ra biển báo dừng hoặc phân biệt người đi bộ với cột đèn. Nó cũng hữu ích trong nhiều ứng dụng khác nhau như xác định bệnh trong hình ảnh sinh học, kiểm tra công nghiệp và thị giác robot. [2]

* + 1. Sự khác biệt giữa nhận dạng đối tượng và phát hiện đối tượng

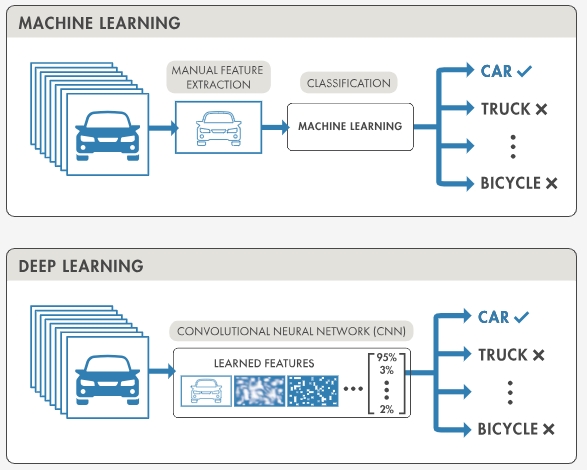
Phát hiện đối tượng và nhận dạng đối tượng là những kỹ thuật tương tự để xác định đối tượng, nhưng chúng khác nhau về cách thực hiện. Phát hiện đối tượng là quá trình tìm kiếm các trường hợp đối tượng trong hình ảnh. Trong trường hợp học sâu, phát hiện đối tượng là một tập hợp con của nhận dạng đối tượng, trong đó đối tượng không chỉ được xác định mà còn được định vị trong một hình ảnh. Điều này cho phép nhiều đối tượng được xác định và định vị trong cùng một hình ảnh. [2]



*Hình 1.2.2. Nhận dạng đối tượng (trái) và phát hiện đối tượng (phải).*

* + 1. Cách nhận dạng đối tượng hoạt động

Bạn có thể sử dụng nhiều phương pháp khác nhau để nhận dạng đối tượng. Gần đây, các kỹ thuật trong học máy và học sâu đã trở thành những phương pháp phổ biến để giải quyết các vấn đề nhận dạng đối tượng. Cả hai kỹ thuật đều học cách nhận dạng đối tượng trong hình ảnh, nhưng chúng khác nhau về cách thực hiện.



*Hình 1.2.3: Kỹ thuật học máy và học sâu để nhận dạng đối tượng.*

* 1. Tổng quan về học máy
     1. Khái niệm học máy

Nghiên cứu về các thuật toán và mô hình toán học (thường mô hình thống kê) để các chương trình máy tính tự động cải thiện khả năng giải quyết bài toán [3]

* + 1. Cách học máy hoạt động

Quy trình ra quyết định: Nhìn chung, các thuật toán học máy được sử dụng để đưa ra dự đoán hoặc phân loại. Dựa trên một số dữ liệu đầu vào, có thể được gắn nhãn hoặc không gắn nhãn, thuật toán của bạn sẽ đưa ra ước tính về một mẫu trong dữ liệu.

Hàm lỗi: Hàm lỗi đánh giá dự đoán của mô hình. Nếu có các ví dụ đã biết, hàm lỗi có thể thực hiện so sánh để đánh giá độ chính xác của mô hình.

Quy trình tối ưu hóa mô hình: Nếu mô hình có thể phù hợp hơn với các điểm dữ liệu trong tập huấn luyện, thì trọng số sẽ được điều chỉnh để giảm sự khác biệt giữa ví dụ đã biết và ước tính của mô hình. Thuật toán sẽ lặp lại quy trình "đánh giá và tối ưu hóa" lặp đi lặp lại này, tự động cập nhật trọng số cho đến khi đạt đến ngưỡng chính xác. [3]

* + 1. Phương pháp học máy

Các mô hình học máy được chia thành ba loại chính.

* Học máy có giám sát

Học máy có giám sát, còn được gọi là học máy có giám sát, được định nghĩa bằng cách sử dụng các tập dữ liệu được gắn nhãn để đào tạo các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác. Khi dữ liệu đầu vào được đưa vào mô hình, mô hình sẽ điều chỉnh trọng số của nó cho đến khi nó được điều chỉnh phù hợp. Điều này xảy ra như một phần của quy trình xác thực chéo để đảm bảo rằng mô hình tránh tình trạng quá khớp hoặc không khớp. Học có giám sát giúp các tổ chức giải quyết nhiều vấn đề thực tế ở quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt với hộp thư đến của bạn. Một số phương pháp được sử dụng trong học có giám sát bao gồm mạng nơ-ron, bayes ngây thơ, hồi quy tuyến tính, hồi quy logistic, rừng ngẫu nhiên và máy vectơ hỗ trợ (SVM). [4]

* Học máy không giám sát

Học không giám sát, còn được gọi là học máy không giám sát, sử dụng các thuật toán học máy để phân tích và nhóm các tập dữ liệu không có nhãn (các tập hợp con được gọi là cụm). Các thuật toán này khám phá các mẫu ẩn hoặc nhóm dữ liệu mà không cần sự can thiệp của con người. Khả năng khám phá điểm tương đồng và khác biệt trong thông tin của phương pháp này khiến nó trở nên lý tưởng cho phân tích dữ liệu thăm dò, chiến lược bán chéo, phân khúc khách hàng và nhận dạng hình ảnh và mẫu. Nó cũng được sử dụng để giảm số lượng các tính năng trong một mô hình thông qua quá trình giảm chiều. Phân tích thành phần chính (PCA) và phân tích giá trị kỳ dị (SVD) là hai phương pháp phổ biến cho mục đích này. Các thuật toán khác được sử dụng trong học không giám sát bao gồm mạng nơ-ron, cụm k-means và phương pháp cụm xác suất.

* Học bán giám sát

Học bán giám sát cung cấp phương pháp trung gian giữa học có giám sát và học không giám sát. Trong quá trình đào tạo, nó sử dụng một tập dữ liệu có nhãn nhỏ hơn để hướng dẫn phân loại và trích xuất tính năng từ một tập dữ liệu lớn hơn, không có nhãn. Học bán giám sát có thể giải quyết vấn đề không có đủ dữ liệu có nhãn cho thuật toán học có giám sát. Nó cũng hữu ích nếu việc dán nhãn đủ dữ liệu quá tốn kém. [4]

* + 1. Các thuật toán học máy phổ biến

Một số thuật toán học máy thường được sử dụng. Bao gồm:

* Mạng nơ-ron: Mạng nơ-ron mô phỏng cách não người hoạt động, với số lượng lớn các nút xử lý được liên kết. Mạng nơ-ron rất giỏi trong việc nhận dạng các mẫu và đóng vai trò quan trọng trong các ứng dụng bao gồm dịch ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói và tạo hình ảnh.
* Hồi quy tuyến tính: Thuật toán này được sử dụng để dự đoán các giá trị số, dựa trên mối quan hệ tuyến tính giữa các giá trị khác nhau. Ví dụ, kỹ thuật này có thể được sử dụng để dự đoán giá nhà dựa trên dữ liệu lịch sử của khu vực.
* Hồi quy logistic: Thuật toán học có giám sát này đưa ra dự đoán cho các biến phản hồi theo danh mục, chẳng hạn như câu trả lời "có/không" cho các câu hỏi. Thuật toán này có thể được sử dụng cho các ứng dụng như phân loại thư rác và kiểm soát chất lượng trên dây chuyền sản xuất.
* Phân cụm: Sử dụng học không giám sát, các thuật toán phân cụm có thể xác định các mẫu trong dữ liệu để có thể nhóm lại. Máy tính có thể giúp các nhà khoa học dữ liệu bằng cách xác định sự khác biệt giữa các mục dữ liệu mà con người đã bỏ qua.
* Cây quyết định: Cây quyết định có thể được sử dụng để dự đoán các giá trị số (hồi quy) và phân loại dữ liệu thành các danh mục. Cây quyết định sử dụng chuỗi phân nhánh các quyết định được liên kết có thể được biểu diễn bằng sơ đồ cây. Một trong những ưu điểm của cây quyết định là chúng dễ xác thực và kiểm tra, không giống như hộp đen của mạng nơ-ron.
* Random forests: Trong Random forests, thuật toán học máy dự đoán giá trị hoặc danh mục bằng cách kết hợp các kết quả từ một số cây quyết định. [4]
  + 1. Các trường hợp sử dụng máy học trong thế giới thực

Sau đây chỉ là một vài ví dụ về máy học mà bạn có thể gặp hàng ngày:

* Nhận dạng giọng nói: Còn được gọi là nhận dạng giọng nói tự động (ASR), nhận dạng giọng nói trên máy tính hoặc chuyển giọng nói thành văn bản và đây là khả năng sử dụng xử lý ngôn ngữ tự nhiên (NLP) để dịch giọng nói của con người thành định dạng văn bản. Nhiều thiết bị di động tích hợp nhận dạng giọng nói vào hệ thống của họ để thực hiện tìm kiếm bằng giọng nói—ví dụ: Siri—hoặc cải thiện khả năng truy cập để nhắn tin.
* Dịch vụ khách hàng: Các chatbot trực tuyến đang thay thế các tác nhân con người trong suốt hành trình của khách hàng, thay đổi cách chúng ta nghĩ về sự tương tác của khách hàng trên các trang web và nền tảng truyền thông xã hội. Các chatbot trả lời các câu hỏi thường gặp (FAQ) về các chủ đề như vận chuyển hoặc cung cấp lời khuyên được cá nhân hóa, bán chéo sản phẩm hoặc gợi ý kích thước cho người dùng. Ví dụ bao gồm các tác nhân ảo trên các trang web thương mại điện tử; bot nhắn tin, sử dụng Slack và Facebook Messenger; và các tác vụ thường do trợ lý ảo và trợ lý giọng nói thực hiện.
* Tầm nhìn máy tính: Công nghệ AI này cho phép máy tính thu thập thông tin có ý nghĩa từ hình ảnh kỹ thuật số, video và các đầu vào trực quan khác, sau đó thực hiện hành động thích hợp. Được hỗ trợ bởi mạng nơ-ron tích chập, thị giác máy tính có ứng dụng trong gắn thẻ ảnh trên phương tiện truyền thông xã hội, chụp X-quang trong chăm sóc sức khỏe và xe tự lái trong ngành công nghiệp ô tô.
* Công cụ đề xuất: Sử dụng dữ liệu về hành vi tiêu dùng trong quá khứ, thuật toán AI có thể giúp khám phá xu hướng dữ liệu có thể được sử dụng để phát triển các chiến lược bán chéo hiệu quả hơn. Các công cụ đề xuất được các nhà bán lẻ trực tuyến sử dụng để đưa ra các đề xuất sản phẩm có liên quan cho khách hàng trong quá trình thanh toán.
* Tự động hóa quy trình bằng rô-bốt (RPA): Còn được gọi là rô-bốt phần mềm, RPA sử dụng các công nghệ tự động hóa thông minh để thực hiện các tác vụ thủ công lặp đi lặp lại.
* Giao dịch chứng khoán tự động: Được thiết kế để tối ưu hóa danh mục đầu tư chứng khoán, các nền tảng giao dịch tần suất cao do AI điều khiển thực hiện hàng nghìn hoặc thậm chí hàng triệu giao dịch mỗi ngày mà không cần sự can thiệp của con người.
* Phát hiện gian lận: Các ngân hàng và các tổ chức tài chính khác có thể sử dụng máy học để phát hiện các giao dịch đáng ngờ. Học có giám sát có thể đào tạo một mô hình bằng cách sử dụng thông tin về các giao dịch gian lận đã biết. Phát hiện bất thường có thể xác định các giao dịch có vẻ bất thường và cần được điều tra thêm. [4]
  1. Tổng quan về Deep Learning
     1. Khái niệm Deep Learning

Học sâu là một tập hợp con của học máy sử dụng mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu, để mô phỏng khả năng ra quyết định phức tạp của não người. Một số dạng học sâu hỗ trợ hầu hết các ứng dụng trí tuệ nhân tạo (AI) trong cuộc sống của chúng ta ngày nay.

Sự khác biệt chính giữa học sâu và học máy là cấu trúc của kiến ​​trúc mạng nơ-ron cơ bản. Các mô hình học máy truyền thống "không sâu" sử dụng mạng nơ-ron đơn giản với một hoặc hai lớp tính toán. Các mô hình học sâu sử dụng ba lớp trở lên—nhưng thường là hàng trăm hoặc hàng nghìn lớp—để đào tạo các mô hình.

Trong khi các mô hình học có giám sát yêu cầu dữ liệu đầu vào có cấu trúc, được gắn nhãn để tạo ra đầu ra chính xác, các mô hình học sâu có thể sử dụng học không giám sát. Với học không giám sát, các mô hình học sâu có thể trích xuất các đặc điểm, tính năng và mối quan hệ mà chúng cần để tạo ra đầu ra chính xác từ dữ liệu thô, không có cấu trúc. Ngoài ra, các mô hình này thậm chí có thể đánh giá và tinh chỉnh đầu ra của chúng để tăng độ chính xác.

Học sâu là một khía cạnh của khoa học dữ liệu thúc đẩy nhiều ứng dụng và dịch vụ cải thiện tự động hóa, thực hiện các tác vụ phân tích và vật lý mà không cần sự can thiệp của con người. Điều này cho phép nhiều sản phẩm và dịch vụ hàng ngày như trợ lý kỹ thuật số, điều khiển TV bằng giọng nói, phát hiện gian lận thẻ tín dụng, xe tự lái và AI tạo ra.

* + 1. Cách Deep Learning hoạt động

Mạng nơ-ron nhân tạo, hay mạng nơ-ron nhân tạo, cố gắng mô phỏng bộ não con người thông qua sự kết hợp giữa dữ liệu đầu vào, trọng số và độ lệch - tất cả đều hoạt động như nơ-ron silicon. Các yếu tố này hoạt động cùng nhau để nhận dạng, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

Mạng nơ-ron sâu bao gồm nhiều lớp nút được kết nối với nhau, mỗi lớp xây dựng trên lớp trước đó để tinh chỉnh và tối ưu hóa dự đoán hoặc phân loại. Tiến trình tính toán này thông qua mạng được gọi là truyền tiến. Các lớp đầu vào và đầu ra của mạng nơ-ron sâu được gọi là các lớp hiển thị. Lớp đầu vào là nơi mô hình học sâu thu thập dữ liệu để xử lý và lớp đầu ra là nơi đưa ra dự đoán hoặc phân loại cuối cùng.

Một quy trình khác được gọi là truyền ngược sử dụng các thuật toán, chẳng hạn như giảm dần độ dốc, để tính toán lỗi trong các dự đoán, sau đó điều chỉnh trọng số và độ lệch của hàm bằng cách di chuyển ngược qua các lớp để đào tạo mô hình. Cùng nhau, truyền tiến và truyền ngược cho phép mạng nơ-ron đưa ra dự đoán và sửa mọi lỗi. Theo thời gian, thuật toán dần trở nên chính xác hơn.

Học sâu đòi hỏi một lượng lớn sức mạnh tính toán. Các đơn vị xử lý đồ họa (GPU) hiệu suất cao là lý tưởng vì chúng có thể xử lý khối lượng tính toán lớn trong nhiều lõi với bộ nhớ dồi dào. Điện toán đám mây phân tán cũng có thể hỗ trợ. Mức công suất tính toán này là cần thiết để đào tạo các thuật toán sâu thông qua học sâu. Tuy nhiên, việc quản lý nhiều GPU tại cơ sở có thể tạo ra nhu cầu lớn về tài nguyên nội bộ và cực kỳ tốn kém để mở rộng quy mô. Đối với các yêu cầu về phần mềm, hầu hết các ứng dụng học sâu đều được mã hóa bằng một trong ba khuôn khổ học tập sau: JAX, PyTorch hoặc TensorFlow. [4]

* + 1. Các loại mô hình trong Deep Learning

Thuật toán học sâu cực kỳ phức tạp và có nhiều loại mạng nơ-ron khác nhau để giải quyết các vấn đề hoặc tập dữ liệu cụ thể. Sau đây là sáu loại. Mỗi loại có những ưu điểm riêng và chúng được trình bày ở đây theo thứ tự phát triển của chúng, với mỗi mô hình kế tiếp điều chỉnh để khắc phục điểm yếu trong mô hình trước đó.

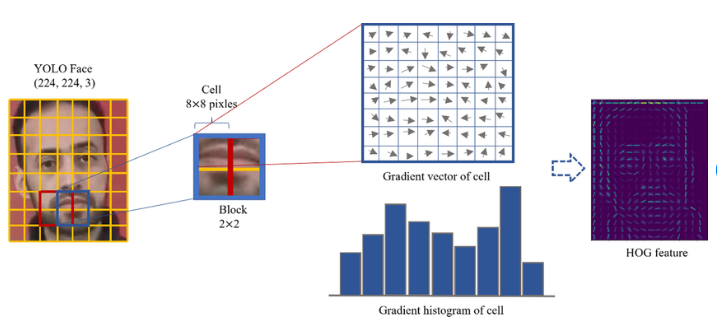
Một điểm yếu tiềm ẩn của tất cả chúng là các mô hình học sâu thường là "hộp đen", khiến việc hiểu hoạt động bên trong của chúng trở nên khó khăn và đặt ra những thách thức về khả năng diễn giải. Nhưng điều này có thể được cân bằng với những lợi ích chung của độ chính xác cao và khả năng mở rộng. [4]

Tựu chung thì Deep Learning có các mô hình sau:

* CNN
* RNN
* Autoencoders and variational autoencoders
* GAN
* Diffusion models
  1. Các kỹ thuật nhận diện đối tượng bằng cách sử dụng hình ảnh
     1. Kỹ thuật xử lý ảnh truyền thống

Các kỹ thuật xử lý hình ảnh truyền thống để phát hiện đối tượng thường bao gồm trích xuất đặc điểm sau đó là phân loại. Một số phương pháp đáng chú ý bao gồm:

* + - 1. Histogram of Oriented Gradients (HOG) :
* Nhằm tạo ra các bộ mô tả đặc trưng (feature descriptor) của đối tượng
* HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.
* Bản chất của HOG là sử dụng thông tin về sự phân bốcủa các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh.Cách hoạt động: HOG (Histogram of Oriented Gradients) hoạt động bằng cách chia hình ảnh thành các ô nhỏ (cells), tính gradient tại mỗi pixel để xác định hướng và độ lớn, sau đó nhóm lại thành histogram theo các hướng cụ thể. Nhiều ô được gom thành khối (blocks) để chuẩn hóa, và các histogram trên toàn hình ảnh được ghép lại tạo thành vector đặc trưng, giúp nhận diện hình dạng hoặc đối tượng trong ảnh. [5]



*Hình 1.4.1. Minh họa cách hoạt động của thuật toán HOG để trích xuất đặc trưng từ một hình ảnh.*

* Ưu điểm của HOG:
* Độ bền trước những thay đổi về ánh sáng : Các tính năng HOG không thay đổi theo những thay đổi về điều kiện ánh sáng, khiến chúng phù hợp để phát hiện đối tượng trong các tình huống thực tế.
* Khả năng chống che khuất : Các tính năng HOG có thể xử lý được tình trạng che khuất một phần, cho phép phát hiện đối tượng chính xác ngay cả khi đối tượng bị ẩn một phần.
* Hiệu quả tính toán : Các tính năng HOG có thể được tính toán hiệu quả, phù hợp với các ứng dụng phát hiện đối tượng theo thời gian thực.
* Tính linh hoạt : Các tính năng HOG có thể được sử dụng với nhiều thuật toán phân loại khác nhau, chẳng hạn như Máy vectơ hỗ trợ (SVM) và Rừng ngẫu nhiên, v.v.
* Nhược điểm của HOG
* Không xử lý tốt biến dạng và góc nhìn: HOG hoạt động tốt với các đối tượng có cấu trúc cố định (ví dụ: khuôn mặt, xe cộ), nhưng kém hiệu quả khi đối tượng có biến dạng hoặc ở góc nhìn khác nhau.
* Hiệu năng giảm trên hình ảnh phức tạp: Với các cảnh có nhiều nhiễu hoặc phông nền phức tạp, HOG có thể khó phân biệt đối tượng chính với các vùng không liên quan.
* Yêu cầu căn chỉnh đối tượng: HOG yêu cầu đối tượng phải được căn chỉnh (aligned) để tối ưu hóa việc phát hiện, như khuôn mặt phải hướng chính diện.
* Tốn kém tài nguyên tính toán: Mặc dù nhanh hơn một số phương pháp khác, việc tính toán gradient và histogram trên các ô và khối vẫn tốn tài nguyên khi áp dụng trên hình ảnh lớn hoặc nhiều hình ảnh liên tiếp.
* Không phát hiện tốt các chi tiết nhỏ: HOG tập trung vào các mẫu gradient cục bộ, dẫn đến khó phát hiện các chi tiết nhỏ hoặc các đặc trưng không rõ ràng.
* Không nhạy với thay đổi ánh sáng: Dù được chuẩn hóa, HOG vẫn có thể bị ảnh hưởng bởi điều kiện ánh sáng không đồng đều trong ảnh.
  + - 1. Thuật toán Viola-Jones
* Tổng quan: Được sử dụng rộng rãi để phát hiện khuôn mặt, thuật toán này sử dụng các đặc điểm giống Haar và một loạt các bộ phân loại tăng cường để phát hiện các đối tượng theo thời gian thực.
* Ưu điểm của Viola-Jones:
* Phát hiện theo thời gian thực: Thuật toán có tốc độ nhanh trong việc phát hiện khuôn mặt, phù hợp với các ứng dụng thời gian thực như camera giám sát hoặc hệ thống tương tác người-máy.
* Đơn giản và hiệu quả: Sử dụng đặc trưng Haar-like và hình ảnh tích phân (Integral Image), giúp giảm độ phức tạp trong tính toán.
* Tỷ lệ phát hiện cao: Thuật toán đạt hiệu quả cao trong việc phát hiện khuôn mặt chính diện, ngay cả khi có nhiều khuôn mặt trong một bức ảnh.
* Khả năng loại bỏ vùng không cần thiết: Cascade classifier loại bỏ nhanh các vùng không chứa đối tượng, giảm tải tính toán cho các bước tiếp theo.
* Không yêu cầu ảnh màu: Chỉ cần ảnh grayscale, giúp giảm kích thước và tài nguyên cần thiết khi xử lý.
* Nhược điểm của Viola-Jones:
* Nhạy với góc nhìn và tư thế: Hoạt động tốt nhất với khuôn mặt chính diện và có thể không phát hiện chính xác khi đối tượng quay đầu hoặc nghiêng.
* Không phù hợp với ánh sáng thay đổi: Khó hoạt động hiệu quả trong các điều kiện ánh sáng không đồng đều.
* Quá trình huấn luyện chậm: Mặc dù phát hiện nhanh, việc huấn luyện cascade classifier yêu cầu nhiều thời gian và tài nguyên.
* Hạn chế trong việc phát hiện đối tượng khác: Được tối ưu chủ yếu cho phát hiện khuôn mặt, khó mở rộng hiệu quả cho các đối tượng khác.
* Dễ bỏ sót chi tiết nhỏ: Với các khuôn mặt nhỏ hoặc chi tiết tinh tế, thuật toán có thể không phát hiện chính xác.
* Hiệu suất giảm với nền phức tạp: Khi có nhiều nhiễu hoặc đối tượng trong nền, độ chính xác của thuật toán có thể bị ảnh hưởng.
  + 1. Kỹ thuật dựa trên mạng neuron

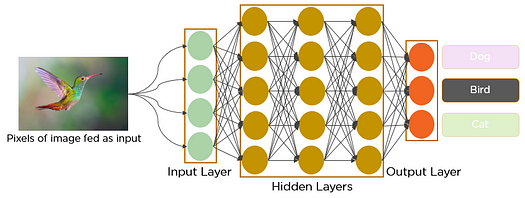
Với sự ra đời của học sâu, các kỹ thuật dựa trên mạng nơ-ron đã trở thành tiêu chuẩn để phát hiện đối tượng. Các phương pháp này bao gồm:

* + - 1. CNN – Mạng neuron tích chấp
* Tổng quan: CNN là phiên bản mở rộng của Artificial Neural Network (ANN) chủ yếu được sử dụng để trích xuất tính năng từ tập dữ liệu ma trận dạng lưới. Ví dụ, tập dữ liệu trực quan như hình ảnh hoặc video, trong đó các mẫu dữ liệu đóng vai trò rộng lớn. CNN được sử dụng rộng rãi để phát hiện đối tượng do khả năng tự động học các tính năng từ dữ liệu. Chúng là xương sống của nhiều mô hình phát hiện đối tượng hiện đại.
* Cấu tao của CNN: Mạng nơ-ron tích chập bao gồm nhiều lớp như lớp đầu vào (input layer), lớp tích chập (convolutional layer), lớp gộp (max pooling layer) và các lớp kết nối đầy đủ (dense layer). [6]



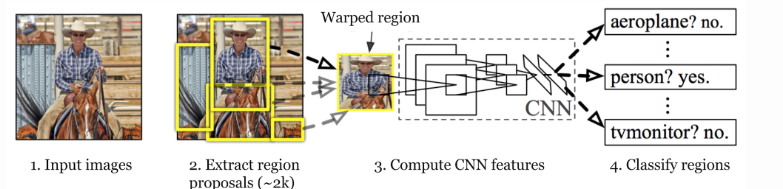
*Hình 1.4.2.1. Cấu tạo đơn giản của CNN*

* Cách hoạt động của CNN: CNN hoạt động bằng cách áp dụng một loạt các lớp tích chập và gộp vào hình ảnh hoặc video đầu vào. Các lớp tích chập trích xuất các đặc điểm từ đầu vào bằng cách trượt một bộ lọc nhỏ hoặc hạt nhân trên hình ảnh hoặc video và tính toán tích vô hướng giữa bộ lọc và đầu vào. Các lớp gộp sau đó sẽ giảm mẫu đầu ra của các lớp tích chập để giảm chiều của dữ liệu và làm cho dữ liệu hiệu quả hơn về mặt tính toán. [6]



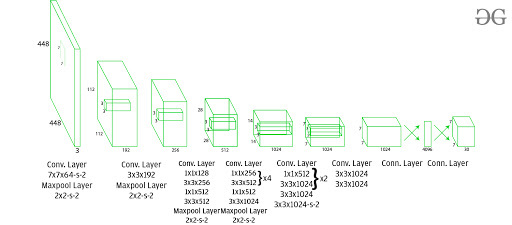
*Hình 1.4.2.2. Cách hoạt động của CNN*

* Ưu điểm của CNN
* Học tính năng tự động: CNN có thể tự động học các tính năng có liên quan từ dữ liệu đầu vào thô, giúp loại bỏ nhu cầu thiết kế tính năng thủ công.
* Độ chính xác cao: CNN có thể đạt được hiệu suất tiên tiến trong nhiều tác vụ nhận dạng hình ảnh và video.
* Độ bền với nhiễu: CNN có khả năng chống nhiễu và biến dạng trong dữ liệu đầu vào, giúp chúng cực kỳ hiệu quả trong các ứng dụng thực tế.
* Nhược điểm
* Yêu cầu tính toán cao: CNN yêu cầu tài nguyên tính toán đáng kể, bao gồm cả GPU cao cấp, để đào tạo và triển khai.
* Lượng dữ liệu lớn: CNN yêu cầu lượng lớn dữ liệu đào tạo được gắn nhãn để đạt được độ chính xác cao, điều này có thể khó đạt được trong một số miền.
* Quá khớp: CNN dễ bị quá khớp khi dữ liệu đào tạo bị hạn chế hoặc nhiễu, điều này có thể dẫn đến hiệu suất tổng quát kém trên dữ liệu mới. [6]
  + - 1. RCNN – Mạng neuron tích chập dựa theo dùng
* Tổng quan: Để giải quyết những thách thức của việc phát hiện đối tượng, Ross Girshick đã giới thiệu R-CNN. Phương pháp này sử dụng thuật toán tìm kiếm có chọn lọc để tạo ra khoảng 2.000 đề xuất vùng, sau đó được xử lý thông qua Mạng nơ-ron tích chập (CNN) để trích xuất các đặc điểm. Các đặc điểm này được phân loại bằng Máy vectơ hỗ trợ (SVM) , trong khi bộ hồi quy hộp giới hạn được sử dụng để cải thiện độ chính xác định vị. [9]
* Cách hoạt động: R-CNN xác định và định vị các đối tượng trong hình ảnh bằng cách đề xuất Vùng quan tâm (RoI) và phân loại chúng thông qua CNN . Khung phát hiện đối tượng bắt đầu bằng một hình ảnh đầu vào chứa các đối tượng tiềm năng và sử dụng Mạng đề xuất vùng (RPN) , giống như Tìm kiếm có chọn lọc , để tạo các hộp giới hạn có khả năng chứa các đối tượng. [9]



*Hình 1.4.2.3. Cách hoạt động của R-CNN*

* Ưu điểm của R-CNN
* Phát hiện đối tượng chính xác: R-CNN cung cấp khả năng phát hiện đối tượng chính xác bằng cách tận dụng các tính năng tích chập dựa trên vùng. Nó vượt trội trong các tình huống mà việc định vị và nhận dạng đối tượng chính xác là rất quan trọng.
* Độ mạnh mẽ đối với các biến thể đối tượng: Các mô hình R-CNN có thể xử lý các đối tượng có kích thước, hướng và tỷ lệ khác nhau, khiến chúng phù hợp với các tình huống thực tế với nhiều đối tượng và bối cảnh phức tạp.
* Tính linh hoạt: R-CNN là một khuôn khổ linh hoạt có thể được điều chỉnh cho nhiều tác vụ phát hiện đối tượng khác nhau, bao gồm phân đoạn thể hiện và theo dõi đối tượng. Bằng cách sửa đổi các lớp cuối cùng của mạng, bạn có thể tùy chỉnh R-CNN để phù hợp với nhu cầu cụ thể của mình.
* Nhược điểm của R-CNN
* Độ phức tạp về mặt tính toán: R-CNN đòi hỏi nhiều tính toán. Nó bao gồm việc trích xuất các đề xuất vùng, áp dụng CNN cho từng đề xuất, sau đó chạy các tính năng đã trích xuất thông qua một trình phân loại. Quy trình nhiều giai đoạn này có thể chậm và đòi hỏi nhiều tài nguyên.
* Suy luận chậm: Do xử lý tuần tự các đề xuất vùng, R-CNN tương đối chậm trong quá trình suy luận. Các ứng dụng thời gian thực có thể thấy độ trễ này là không thể chấp nhận được.
* Đề xuất vùng chồng chéo: R-CNN có thể tạo ra nhiều đề xuất vùng chồng chéo đáng kể, dẫn đến tính toán dư thừa và có khả năng ảnh hưởng đến hiệu suất phát hiện.
* R-CNN không phải là End-to-End: Không giống như các kiến ​​trúc phát hiện đối tượng hiện đại hơn như Faster R-CNN, R-CNN không phải là mô hình end-to-end. Nó bao gồm các mô-đun riêng biệt cho đề xuất vùng và phân loại, điều này có thể dẫn đến hiệu suất không tối ưu so với các mô hình tối ưu hóa cả hai tác vụ cùng nhau. [9]
  + - 1. Nhìn 1 lần – YOLO
* Tổng quan: YOLO được Joseph Redmond và cộng sự đề xuất vào năm 2015. Nó được đề xuất để giải quyết các vấn đề mà các mô hình nhận dạng đối tượng gặp phải vào thời điểm đó, Fast R-CNN là một trong những mô hình tiên tiến nhất vào thời điểm đó nhưng nó có những thách thức riêng như mạng này không thể sử dụng theo thời gian thực, vì phải mất 2-3 giây để dự đoán một hình ảnh và do đó không thể sử dụng theo thời gian thực. Trong khi đó, trong YOLO, chúng ta chỉ phải xem một lần trong mạng, tức là chỉ cần một lần duyệt qua mạng để đưa ra dự đoán cuối cùng. [8]
* Cấu tạo của YOLO: YOLO lấy một hình ảnh làm đầu vào và thay đổi kích thước thành 448\*448 bằng cách giữ nguyên tỷ lệ khung hình và thực hiện đệm. Sau đó, hình ảnh này được truyền vào mạng CNN. Mô hình này có 24 lớp tích chập, 4 lớp max-pooling theo sau là 2 lớp được kết nối đầy đủ . Để giảm số lớp (Kênh), chúng tôi sử dụng tích chập 1\*1 theo sau là tích chập 3\*3 . Lưu ý rằng lớp cuối cùng của YOLOv1 dự đoán đầu ra hình khối. Điều này được thực hiện bằng cách tạo (1, 1470) từ lớp được kết nối đầy đủ cuối cùng và định hình lại thành kích thước (7, 7, 30). [8]



*Hình 1.4.2.4. Cách hoạt động của YOLO*

* Ưu điểm của YOLO:
* Xử lý khung hình ở tốc độ 45 fps (mạng lớn hơn) đến 150 fps (mạng nhỏ hơn) tốt hơn thời gian thực.
* Mạng có khả năng khái quát hóa hình ảnh tốt hơn.
* Nhược điểm của YOLO:
* Khả năng thu hồi tương đối thấp và lỗi định vị nhiều hơn so với Faster R\_CNN.
* Khó phát hiện các vật thể gần vì mỗi lưới chỉ có thể đề xuất 2 hộp giới hạn.
* Khó phát hiện các vật thể nhỏ.
  1. Thư viện Framework TensorFlow/Keras
     + 1. TensorFlow

TensorFlow là một nền tảng mã nguồn mở dành cho học máy và là thư viện toán học biểu tượng được sử dụng cho các ứng dụng học máy

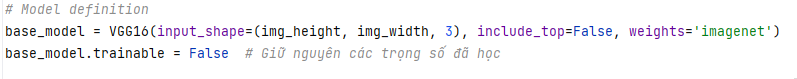
* Ưu điểm của TensorFlow:
* Tensor Flow có biểu diễn đồ thị tốt hơn cho một dữ liệu nhất định so với bất kỳ nền tảng hàng đầu nào khác.
* Tensor Flow có lợi thế là hỗ trợ và sử dụng nhiều phần mềm phụ trợ như GUI và ASIC.
* Khi nói đến hỗ trợ cộng đồng, Tensor Flow là tốt nhất.
* Tensor Flow cũng giúp gỡ lỗi phần phụ của đồ thị.
* Tensor Flow đã cho thấy hiệu suất tốt hơn khi so sánh với các nền tảng khác.
* Dễ dàng mở rộng vì nó cho phép tự do thêm các khối tùy chỉnh để xây dựng trên các ý tưởng mới..
* Nhược điểm của TensorFlow
* Tensor flow không được thiết kế riêng cho hệ điều hành Windows nhưng được thiết kế cho các hệ điều hành khác như Linux nhưng tensor flow có thể được cài đặt trong windows với sự trợ giúp của trình cài đặt gói python (pip).
* Tốc độ của tensor flow chậm hơn khi so sánh với các nền tảng khác cùng loại.
* Để hiểu rõ hơn về tensor flow, người dùng phải có kiến ​​thức cơ bản về phép tính.
* Tensor flow không hỗ trợ OpenCL.
* Vai trò:

Trong hệ thống phân loại động vật này, TensorFlow đóng vai trò quan trọng trong việc xây dựng, huấn luyện và triển khai mô hình học sâu. Cụ thể, TensorFlow là nền tảng hỗ trợ toàn bộ quy trình xử lý mô hình mạng nơ-ron nhân tạo, từ việc tạo mô hình cho đến tối ưu hóa và dự đoán kết quả.

TensorFlow được sử dụng thông qua Keras, một API cao cấp của TensorFlow, để tạo và huấn luyện mô hình học sâu. Trong phần mềm của chúng tôi, TensorFlow thực hiện các chức năng quan trọng như:

* Tạo và cấu hình mô hình:

Mô hình VGG-16 được sử dụng thông qua API của TensorFlow/Keras. VGG-16 là một mô hình học sâu đã được huấn luyện sẵn trên bộ dữ liệu ImageNet, và TensorFlow cung cấp khả năng tải mô hình này vào để sử dụng. Cụ thể, TensorFlow cho phép chúng ta dễ dàng sử dụng mô hình VGG-16 thông qua dòng lệnh:

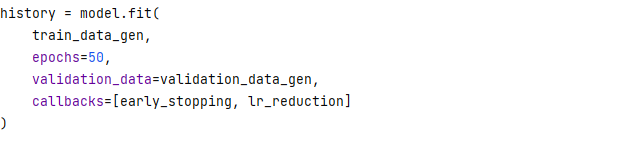


Sau đó, chúng ta tạo mô hình mới bằng cách kết hợp các lớp Flatten, Dense và Dropout để hoàn thiện mô hình phân loại động vật.

* Huấn luyện mô hình:

Sau khi mô hình được xây dựng, TensorFlow sử dụng bộ tối ưu hóa và hàm mất mát để huấn luyện mô hình trên dữ liệu huấn luyện. Các thuật toán học sâu của TensorFlow sử dụng các phép toán số học phức tạp để điều chỉnh các trọng số trong mạng nơ-ron nhằm giảm thiểu hàm mất mát. Cụ thể:

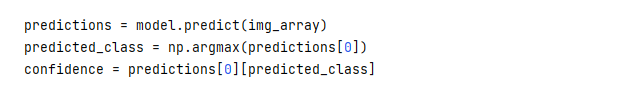




Dòng mã trên cho thấy việc sử dụng bộ tối ưu hóa adam và hàm mất mát categorical\_crossentropy,early\_stopping và ReduceLROnPlateau để giảm tốc độ học khi cần thiết.

* Dự đoán và đánh giá mô hình:

Sau khi mô hình đã được huấn luyện, TensorFlow cho phép chúng ta sử dụng mô hình đã học để dự đoán các lớp động vật từ các ảnh đầu vào mới. Cụ thể, khi người dùng chọn một ảnh từ giao diện người dùng, ảnh đó sẽ được xử lý và đưa qua mô hình để phân loại:



TensorFlow sử dụng mô hình đã huấn luyện để tính toán xác suất của từng lớp động vật, giúp đưa ra dự đoán về loài động vật trong ảnh.

* Tăng gốc GPU:

TensorFlow hỗ trợ tăng tốc phần cứng thông qua GPU, giúp giảm thời gian huấn luyện và tăng hiệu quả tính toán khi làm việc với bộ dữ liệu lớn. Điều này rất quan trọng trong các mô hình học sâu, khi mà thời gian huấn luyện có thể kéo dài nếu chỉ sử dụng CPU.

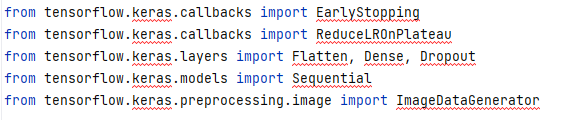
* + - 1. Keras

Đây là thư viện mạng nơ-ron nguồn mở chạy trên Theano hoặc Tensorflow. Nó được thiết kế để người dùng có thể sử dụng nhanh chóng và dễ dàng. Đây là thư viện hữu ích để xây dựng bất kỳ thuật toán học sâu nào tùy theo lựa chọn của chúng ta.

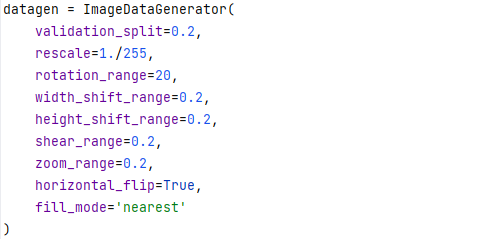
* Ưu điểm:
* Keras là nền tảng tốt nhất hiện có để làm việc trên các mô hình mạng nơ-ron.
* API mà Keras có thân thiện với người dùng, nơi người mới bắt đầu có thể dễ dàng hiểu được.
* Keras có lợi thế là có thể chọn bất kỳ thư viện nào hỗ trợ nó để hỗ trợ phần phụ trợ.
* Keras cung cấp nhiều mô hình được đào tạo trước giúp người dùng cải thiện thêm các mô hình mà người dùng đang thiết kế.
* Khi nói đến hỗ trợ cộng đồng, Keras có hỗ trợ tốt nhất như stack overflow.
* Nhược điểm:
* Nhược điểm chính của Keras là nó là một giao diện lập trình ứng dụng cấp thấp.
* Một số mô hình được đào tạo trước mà Keras không hỗ trợ nhiều khi thiết kế một số mô hình.
* Các lỗi do thư viện Keras đưa ra không hữu ích cho người dùng.
* Vai trò

Keras là API cấp cao của TensorFlow, giúp việc xây dựng mô hình học sâu trở nên dễ dàng và trực quan hơn. Trong hệ thống này, Keras cung cấp các công cụ để tạo mô hình học sâu, chuẩn bị dữ liệu, và huấn luyện mô hình. Với Keras, quá trình xây dựng mô hình học sâu trở nên đơn giản nhờ vào các lớp như Sequential, Dense, Flatten, và các phương thức như fit() để huấn luyện mô hình.

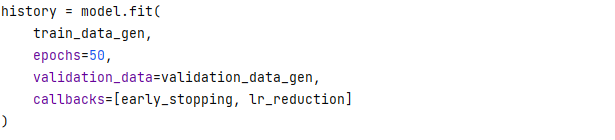
Mô hình phân loại động vật được xây dựng theo kiểu Sequential, tức là các lớp được xếp theo thứ tự liên tiếp. Mô hình này bắt đầu với phần mạng VGG16 đã được huấn luyện sẵn, sau đó là các lớp mới như Flatten để chuyển đổi dữ liệu đầu ra của mạng VGG16 thành một vector một chiều, Dense để thêm các lớp fully connected, và Dropout để giảm thiểu overfitting trong quá trình huấn luyện.



Keras cũng hỗ trợ việc tăng cường dữ liệu thông qua lớp ImageDataGenerator, giúp mô hình học được nhiều đặc trưng hơn từ dữ liệu huấn luyện bằng cách tạo ra các biến thể của ảnh gốc. Điều này đặc biệt quan trọng khi làm việc với các bộ dữ liệu hạn chế.



Trong lúc huần huyện, các callback EarlyStopping và ReduceLROnPlateau được sử dụng để điều chỉnh quá trình huấn luyện, dừng lại nếu mô hình không cải thiện sau một số epoch nhất định và giảm tốc độ học khi mô hình không cải thiện.



1. XÂY DỰNG HỆ THỐNG
   1. Yêu cầu bài toán

* Mục tiêu chính:
* Phát triển một hệ thống phân loại hình ảnh, giúp nhận diện và phân loại chính xác loài động vật xuất hiện trong ảnh đầu vào.
* Hệ thống phải có khả năng đưa ra tên loài động vật và độ tin cậy (confidence score) cho dự đoán.
* Phạm vi:
* Hệ thống làm việc với một tập hợp các loài động vật cố định (15 loài, bao gồm: Bear, Bird, Cat, Dog,...).
* Tập dữ liệu huấn luyện được tổ chức trong các thư mục tương ứng với từng loài.
* Chức năng chính:
* Huấn luyện mô hình: Sử dụng dữ liệu động vật đã phân loại để huấn luyện một mô hình phân loại đa lớp (multi-class classification).
* Giao diện người dùng (GUI): Cung cấp một giao diện để người dùng tải ảnh lên, nhận diện loài động vật trong ảnh, và hiển thị kết quả (gồm tên loài và độ tin cậy).
* Lưu trữ mô hình: Lưu trữ mô hình đã huấn luyện dưới định dạng .keras để tái sử dụng.
* Công nghệ sử dụng:
* Deep Learning: Sử dụng mạng nơ-ron tích chập (CNN) với kiến trúc VGG16 làm backbone, kết hợp các tầng tùy chỉnh cho bài toán hiện tại.
* TensorFlow/Keras: Thư viện chính để xây dựng, huấn luyện và triển khai mô hình.
* Tkinter: Thư viện GUI để tạo giao diện người dùng.
* Đầu vào/Đầu ra:
* Đầu vào:
* Tập dữ liệu huấn luyện: Các hình ảnh đã được phân loại theo loài.
* Hình ảnh từ người dùng (thông qua giao diện).
* Đầu ra:
* Tên loài động vật (dựa trên kết quả phân loại).
* Độ tin cậy của dự đoán.
* Hiển thị hình ảnh đầu vào trên giao diện.
* Yêu cầu chất lượng:
* Mô hình phải đạt được độ chính xác phân loại cao trên tập dữ liệu kiểm tra.
* Giao diện đơn giản, dễ sử dụng, phản hồi nhanh.
  1. Phân tích bài toán
* Tính thách thức:
* Đảm bảo chất lượng tập dữ liệu: Dữ liệu phải đa dạng (góc chụp, điều kiện ánh sáng...) để mô hình tổng quát tốt.
* Giảm hiện tượng overfitting: Do số lượng lớp khá lớn (15 loài động vật), cần sử dụng các kỹ thuật như augment dữ liệu và Dropout.
* Tối ưu hóa giao diện để người dùng không cần hiểu sâu về kỹ thuật nhưng vẫn có trải nghiệm tốt.
* Ứng dụng thực tế:
* Phân loại động vật trong các công viên hoang dã, sở thú, hoặc các nghiên cứu sinh thái học.
* Hỗ trợ giáo dục, như nhận diện loài động vật qua hình ảnh để cung cấp thông tin.
  1. Xây dựng hệ thống với mô hình VGG-16
     1. Tổng quát

Mô hình VGG-16 là kiến ​​trúc mạng nơ-ron tích chập (CNN) được đề xuất bởi Visual Geometry Group (VGG) tại Đại học Oxford. Nó được đặc trưng bởi độ sâu của nó, bao gồm 16 lớp, bao gồm 13 lớp tích chập và 3 lớp được kết nối đầy đủ. VGG-16 nổi tiếng vì tính đơn giản và hiệu quả, cũng như khả năng đạt được hiệu suất mạnh mẽ trên nhiều tác vụ thị giác máy tính khác nhau, bao gồm phân loại hình ảnh và nhận dạng đối tượng. Kiến trúc của mô hình có một chồng các lớp tích chập theo sau là các lớp gộp tối đa, với độ sâu tăng dần. Thiết kế này cho phép mô hình học các biểu diễn phân cấp phức tạp của các tính năng trực quan, dẫn đến các dự đoán mạnh mẽ và chính xác. Mặc dù đơn giản hơn so với các kiến ​​trúc gần đây hơn, VGG-16 vẫn là lựa chọn phổ biến cho nhiều ứng dụng học sâu do tính linh hoạt và hiệu suất tuyệt vời của nó.

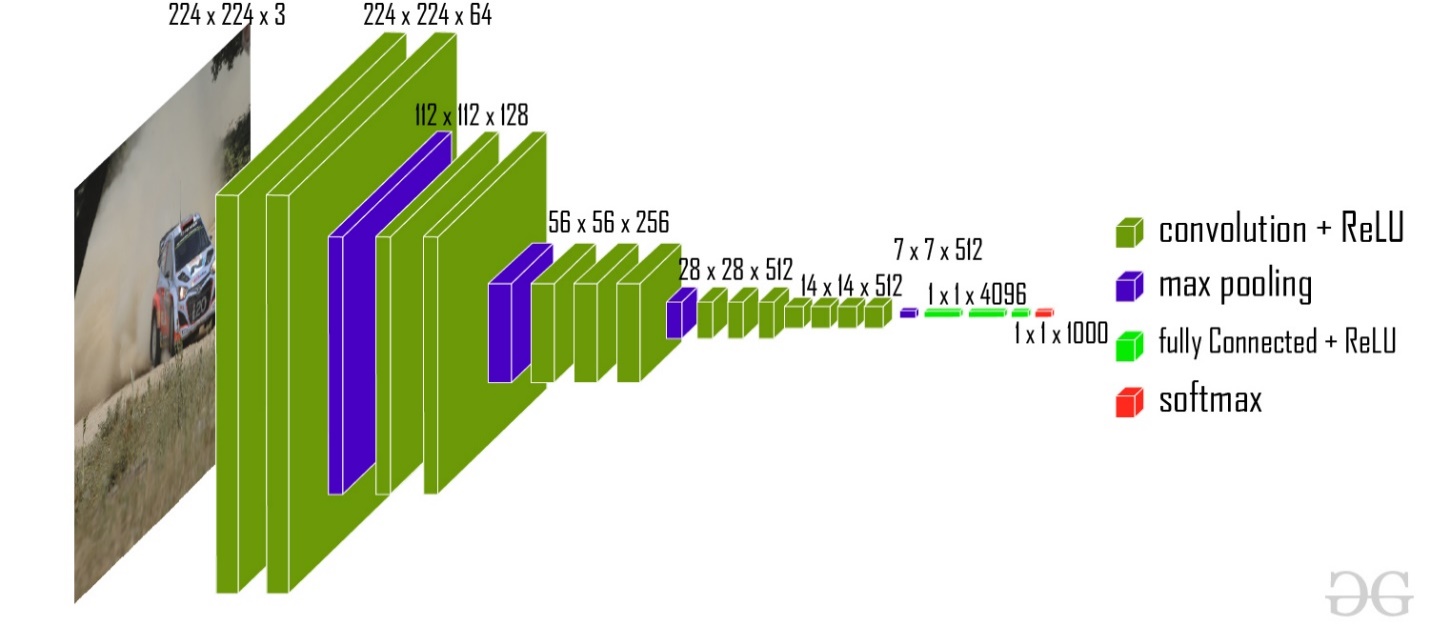
16 trong VGG16 đề cập đến 16 lớp có trọng số. Trong VGG16 có mười ba lớp tích chập, năm lớp Max Pooling và ba lớp Dense tổng cộng là 21 lớp nhưng chỉ có mười sáu lớp trọng số, tức là lớp tham số có thể học được.

VGG16 lấy kích thước tenxơ đầu vào là 224x224 với 3 kênh RGB

Điều độc đáo nhất về VGG16 là thay vì có một số lượng lớn các siêu tham số, họ tập trung vào việc có các lớp tích chập của bộ lọc 3x3 với bước tiến 1 và luôn sử dụng cùng một lớp đệm và lớp maxpool của bộ lọc 2x2 của bước tiến 2.

Các lớp tích chập và lớp maxpool được sắp xếp liên tục trong toàn bộ kiến ​​trúc

Thử thách Nhận dạng Hình ảnh Quy mô Lớn của ImageNet (ILSVRC) là cuộc thi thường niên về thị giác máy tính, trong đó các đội giải quyết các nhiệm vụ bao gồm định vị đối tượng và phân loại hình ảnh. VGG16, do Karen Simonyan và Andrew Zisserman đề xuất vào năm 2014, đã đạt thứ hạng cao nhất trong cả hai nhiệm vụ, phát hiện đối tượng từ 200 lớp và phân loại hình ảnh thành 1000 danh mục. [7]

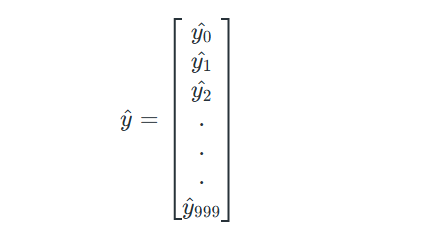


*Hình 2.3.1.1. Mô hình xử lý VGG-16*

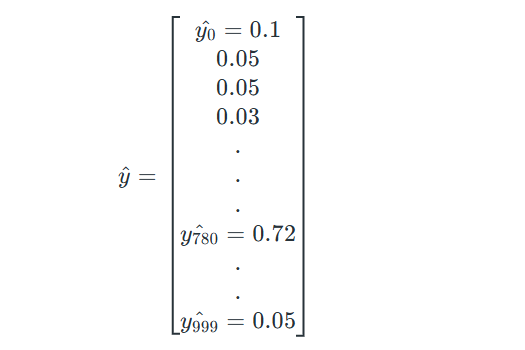
Mô hình này đạt độ chính xác kiểm tra top-5 là 92,7% trên tập dữ liệu ImageNet chứa 14 triệu hình ảnh thuộc 1000 lớp.

* + 1. Mục tiêu của mô hình VGG-16

Bộ dữ liệu ImageNet chứa các hình ảnh có kích thước cố định là 224x224 và có các kênh RGB. Vì vậy, chúng ta có một tenxơ (224, 224, 3) làm đầu vào. Mô hình này xử lý hình ảnh đầu vào và đưa ra một vectơ gồm 1000 giá trị:

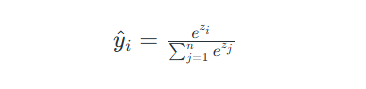


Vectơ này biểu diễn xác suất phân loại cho lớp tương ứng. Giả sử chúng ta có một mô hình dự đoán rằng hình ảnh thuộc về lớp 0 với xác suất 1, lớp 1 với xác suất 0,05, lớp 2 với xác suất 0,05, lớp 3 với xác suất 0,03, lớp 780 với xác suất 0,72, lớp 999 với xác suất 0,05 và tất cả các lớp khác với 0. Vậy, vectơ phân loại cho điều này sẽ là:

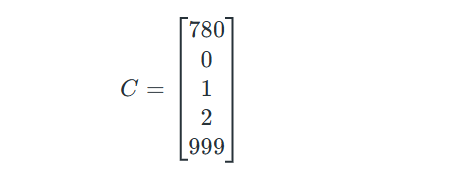


Để đảm bảo các xác suất này cộng lại bằng 1, chúng ta sử dụng hàm softmax.

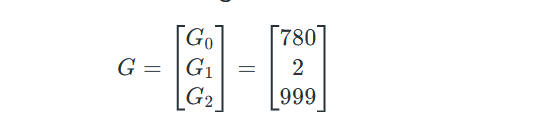
Hàm softmax này được định nghĩa như sau:



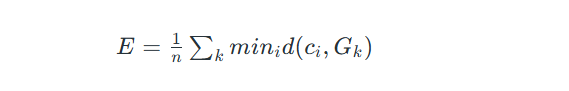
Sau đó, chúng ta đưa 5 ứng viên có khả năng nhất vào vectơ.



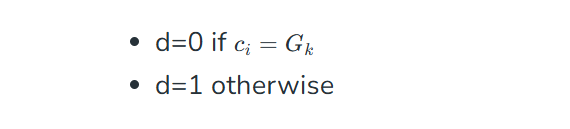
Và vectơ ground truth cơ bản của chúng tôi được định nghĩa như sau:



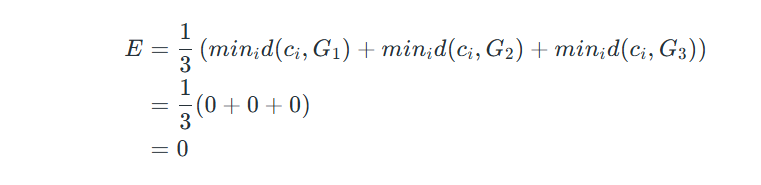
Sau đó chúng ta định nghĩa hàm Error như sau:



Nó tính toán khoảng cách tối thiểu giữa mỗi lớp thực tế và các ứng viên được dự đoán, trong đó hàm khoảng cách d được định nghĩa như sau:



Vì vậy, hàm mất mát cho ví dụ này là:

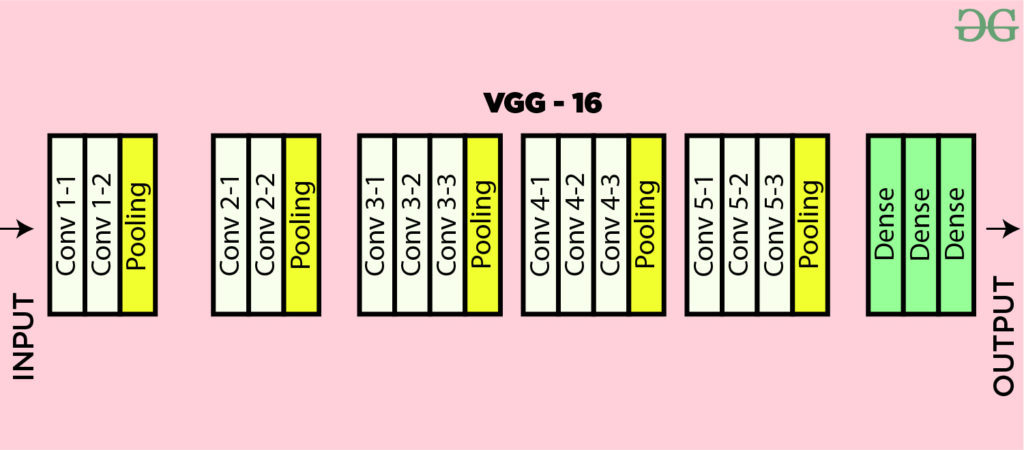


Vì tất cả các danh mục trong dữ liệu thực tế đều nằm trong ma trận dự đoán top-5 nên tổn thất sẽ bằng 0. [7]

* + 1. Cấu tạo của VGG-16

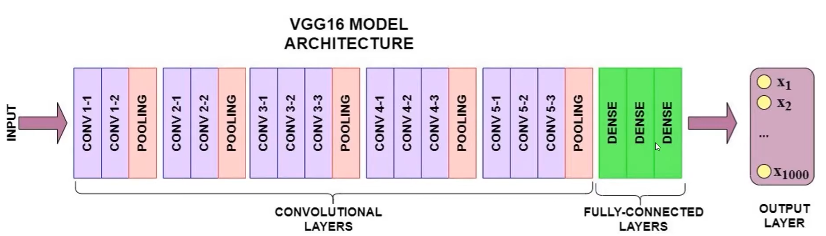
Kiến trúc VGG-16 là một mạng nơ-ron tích chập sâu (CNN) được thiết kế cho các tác vụ phân loại hình ảnh. Nó được giới thiệu bởi Visual Geometry Group tại Đại học Oxford. VGG-16 được đặc trưng bởi tính đơn giản và kiến ​​trúc đồng nhất, giúp dễ hiểu và triển khai. [7]

Cấu hình VGG-16 thường bao gồm 16 lớp, bao gồm 13 lớp tích chập và 3 lớp được kết nối đầy đủ. Các lớp này được tổ chức thành các khối, với mỗi khối chứa nhiều lớp tích chập theo sau là một lớp gộp tối đa để hạ mẫu. [7]

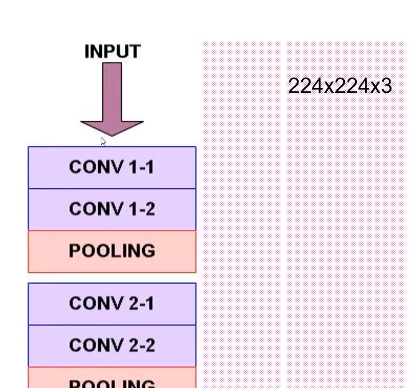


*Hình 2.3.1.2. Cấu tạo của VGG-16*

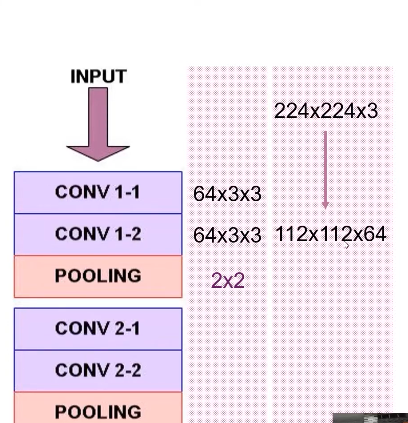
* Sau đây là phân tích về kiến ​​trúc VGG-16:
* Lớp đầu vào: Kích thước đầu vào: (224, 224, 3)
* Lớp tích chập (64 bộ lọc, bộ lọc 3×3, cùng một lớp đệm):
* Hai lớp tích chập liên tiếp, mỗi lớp có 64 bộ lọc và kích thước bộ lọc là 3×3.
* Cùng một lớp đệm được áp dụng để duy trì kích thước không gian.
* Lớp gộp tối đa (2×2, bước 2): Lớp gộp tối đa có kích thước nhóm là 2×2 và bước 2.
* Lớp tích chập (128 bộ lọc, bộ lọc 3×3, cùng một lớp đệm): Hai lớp tích chập liên tiếp, mỗi lớp có 128 bộ lọc và kích thước bộ lọc là 3×3.
* Lớp gộp tối đa (2×2, bước 2): Lớp gộp tối đa có kích thước nhóm là 2×2 và bước 2.
* Lớp tích chập (256 bộ lọc, bộ lọc 3×3, cùng một lớp đệm): Hai lớp tích chập liên tiếp, mỗi lớp có 256 bộ lọc và kích thước bộ lọc là 3×3.
* Lớp tích chập (512 bộ lọc, bộ lọc 3×3, cùng một lớp đệm): Hai bộ gồm ba lớp tích chập liên tiếp, mỗi lớp có 512 bộ lọc và kích thước bộ lọc là 3×3.
* Lớp gộp tối đa (2×2, bước 2): Lớp gộp tối đa có kích thước nhóm là 2×2 và bước 2.
* Ngăn xếp các lớp tích chập và gộp tối đa:
* Hai lớp tích chập bổ sung sau ngăn xếp trước đó.
* Kích thước bộ lọc: 3×3.
* Làm phẳng: Làm phẳng bản đồ đặc trưng đầu ra (7x7x512) thành một vectơ có kích thước 25088.
* Các lớp được kết nối đầy đủ:
* Ba lớp được kết nối đầy đủ với kích hoạt ReLU.
* Lớp đầu tiên có kích thước đầu vào là 25088 và kích thước đầu ra là 4096.
* Lớp thứ hai có kích thước đầu vào là 4096 và kích thước đầu ra là 4096.
* Lớp thứ ba có kích thước đầu vào là 4096 và kích thước đầu ra là 1000, tương ứng với 1000 lớp trong thử thách ILSVRC.
* Kích hoạt Softmax được áp dụng cho đầu ra của lớp được kết nối đầy đủ thứ ba để phân loại.
* Kiến trúc này tuân theo các thông số kỹ thuật được cung cấp, bao gồm việc sử dụng hàm kích hoạt ReLU và lớp kết nối đầy đủ cuối cùng đưa ra xác suất cho 1000 lớp sử dụng kích hoạt softmax.
  + 1. Quy trình hoạt động của VGG-16
* VGG16 1 ảnh đưa vào thì nó sẽ qua nhiều lớp tích chập liên tiếp rồi sau đó nó mới qua max polling lí do là để lấy nhiều đặc trưng của ảnh hơn và độ chính xác cũng sẽ cao hơn
* VGG16 sẽ gồm 16 lớp trong đó có 13 lớp tích chập và 3 lớp kết nối
* Trong 13 lớp tích chập sẽ có 2 loại, lại thứ nhât đó là một cái ảnh đưa vào nó sẽ phải qua 2 lớp tích chập với mới qua lớp max polling
* Loại thứ 2 đó chính là sau mỗi ảnh đưa ra từ 2 lớp tích chập conv conv thì nó sẽ qua 3 lớp tích chập liên tiếp, tiếp đến mới qua lớp max polling



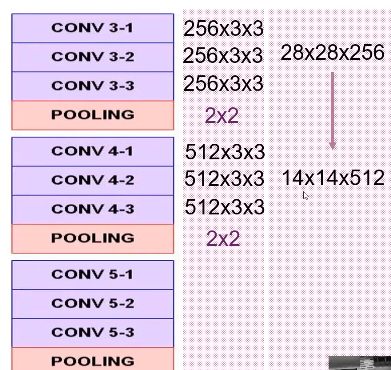
* Bắt buộc: Đưa vào ảnh 224x224x3 và sau đó nó sẽ qua 2 lớp conv



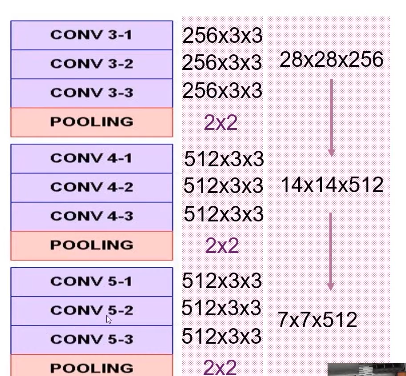
* 2 lớp conv đầu tiên sẽ có 64 kernel kích thước 3x3. Sau đó qua lớp max pooling có kích thước là 2x2. Nó sẽ cho hình ảnh là 112x112x64



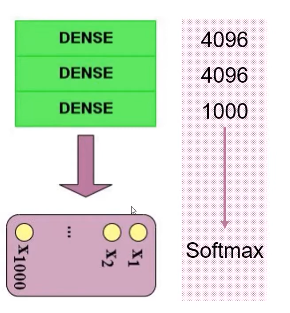
* Từ ảnh 112x112x64, qua liên tiếp 3 lớp conv có 512 kernel với kích thước 3x3 sau đó qua lớp maxpooling 2x2. Sau đó cho ảnh 14x14x512



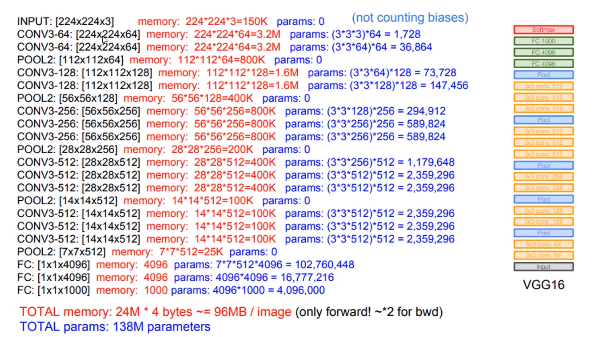
* Tiếp đến từ ảnh đó qua liên tiếp 3 conv có 512kernel 3x3, qua pooling 2x2 , cho ảnh cuối cùng 7x7x512



* Lớp dense – kết nối đầy đủ. Từ ảnh 7x7x512 , nó sẽ được duỗi ra 1 vector 1 chiều gồm 4096 nút và nó sẽ được duỗi tiếp 4096 nút nữa và nó được duỗi ra 1000 nút
* Cuối cùng nó qua một hàm softmax để cho ra 1000 nút, cho ra xác suất để phân loại



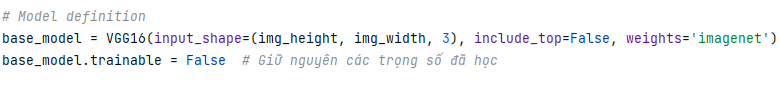
* Trong quá trình huấn luyện của VGG16 thì trọng số sẽ được biểu diễn như sau:
  + Ảnh đầu tiên vào là có kích thước 224x224x3 thì trọng số được huấn luyện
  + Ta có 64 kernel 3x3 cho nên tổng trọng số được huấn luyện đó là 3x3x3x64 =1728 trọng số được huấn luyện
  + Sau khi qua đó đưuọc một ảnh là 224x224x64 thì nó tiếp tục qua lớp conv là 64kerrnel 3x3 thì tổng trọng số là 64x64x3, và tương tự qua mối lớp maxpooling thì không có trọng số để huấn luyện cho nên tổng trọng số được huấn luyện là khoảng 138tr trọng số được huấn luyện, do số lớn nên VGG-16 tốc độ khá chậm, VGG016 sẽ được ứng dựng trong nhiều chỗ như phân loại chó mèo vì VGG16 được huấn luyện lâu nên áp dụng trực tiếp các tọng số đã được huấn luyện trong thưu viện keras đã đươc huấn luyện



* + 1. Vai trò của VGG-16

VGG16 là một trong những mô hình học sâu đã được huấn luyện sẵn trên tập dữ liệu ImageNet, một tập dữ liệu lớn bao gồm hàng triệu hình ảnh thuộc nhiều lớp khác nhau. VGG16 có kiến trúc mạng nơ-ron sâu, bao gồm nhiều lớp tích chập (convolutional layers) và các lớp kết nối đầy đủ (fully connected layers). Mô hình này đặc biệt hữu ích trong các bài toán phân loại hình ảnh, vì đã được huấn luyện sẵn để nhận diện các đặc trưng từ các hình ảnh đa dạng.

Trong hệ thống này, VGG16 được sử dụng như một mô hình cơ sở (base model) trong khuôn khổ của transfer learning. Cụ thể, các lớp tích chập của VGG16 được giữ nguyên và không được huấn luyện lại, giúp tận dụng các đặc trưng mà mô hình đã học được từ ImageNet. Các lớp cuối cùng của mô hình (lớp phân loại) được thay thế bởi các lớp mới, được thiết kế để phân loại các loài động vật cụ thể mà hệ thống hướng đến. [7]

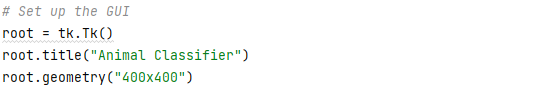


Ở đây, tham số include\_top=False cho phép loại bỏ các lớp phân loại cuối cùng của VGG16 và chỉ giữ lại phần mạng tích chập. Điều này giúp giảm thiểu độ phức tạp của mô hình và tập trung vào các lớp mới để phân loại các loài động vật. [7]

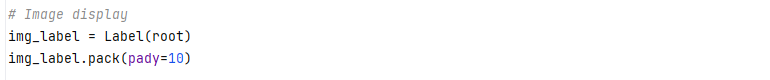
* 1. Xây dựng giao diện chương trình

Giao diện người dùng (GUI) được xây dựng với các thành phần chính như sau:

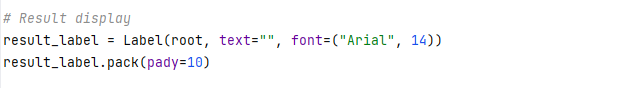
* Khung chương trình chính được tạo bằng thư viện tkinter:



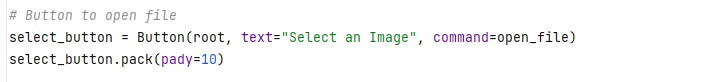
* root.title: Đặt tiêu đề cửa sổ là "Animal Classifier".
* root.geometry: Đặt kích thước cửa sổ giao diện là 400x400 pixel.
* Hiển thị ảnh đã chọn: Giao diện cho phép hiển thị ảnh được chọn từ máy tính của người dùng.



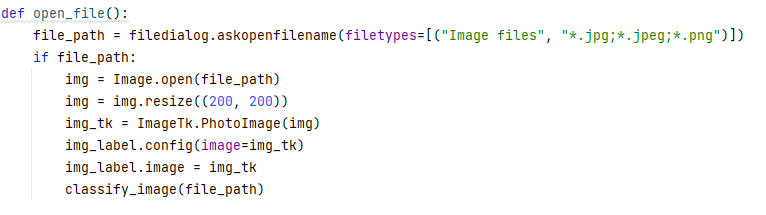
* Label: Dùng để tạo khung hiển thị ảnh.
* pack(pady=10): Sắp xếp vị trí khung ảnh với khoảng cách dọc 10 pixel.
* Hiển thị kết quả phân loại: Một nhãn (label) được dùng để hiển thị kết quả phân loại loài động vật từ ảnh đã chọn.



* text: Ban đầu để trống, sẽ được cập nhật sau khi ảnh được phân loại.
* font: Sử dụng font chữ Arial với kích cỡ 14.
* Nút chọn ảnh từ thư mục: Người dùng có thể nhấn nút để mở cửa sổ chọn file ảnh từ máy tính.



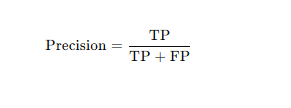
* text: Văn bản hiển thị trên nút ("Select an Image").
* command: Liên kết với hàm open\_file để xử lý sự kiện khi nút được nhấn.
* Hàm mở file ảnh (open\_file): Hàm này cho phép người dùng chọn ảnh từ máy tính, hiển thị ảnh lên giao diện, và gửi ảnh đến mô hình để phân loại.



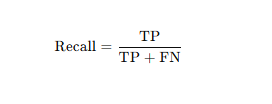
* filedialog.askopenfilename: Mở hộp thoại chọn file, chỉ chấp nhận các định dạng ảnh như .jpg, .jpeg, .png.
* Image.open: Mở ảnh đã chọn.
* ImageTk.PhotoImage: Chuyển đổi ảnh thành định dạng hiển thị được trên Tkinter.
* classify\_image: Hàm phân loại ảnh, được gọi sau khi ảnh được tải thành công

1. KẾT QUẢ THỰC NGHIỆM
   1. Dữ liệu đầu vào

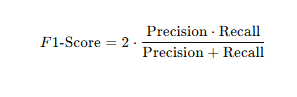
* Dữ liệu đầu vào được lấy từ kaggle.com với bộ ảnh động vật
* Dữ liệu được xử lý thủ công bằng lật ngang, lật dọc, tăng cường độ sáng, giảm cường độ sáng
* Chia train-test lần lượt là
* Train 40 – Test 60
* Train 60 – Test 40
* Train 30 – Test 70
* Train 70 – Test 30
* Train 20 – Test 80
* Train 80 – Test 20
  1. Độ đo đánh giá
     1. Công thức tính độ đo
        1. Precision (Độ chính xác)
* Precision đo lường độ chính xác của các dự đoán dương tính (Positive Predictions). Nó trả lời câu hỏi: "Trong tất cả các mẫu được dự đoán là dương tính, có bao nhiêu mẫu là đúng?"
* Precision cao đồng nghĩa với việc mô hình ít đưa ra các dự đoán sai dương tính (False Positives).
* Công thức



* Trong đó:
* TP (True Positive): Số mẫu được dự đoán đúng là dương tính.
* FP (False Positive): Số mẫu được dự đoán nhầm là dương tính.
  + - 1. Recall (Độ nhạy hoặc Độ bao phủ)
* Recall đo lường khả năng phát hiện các trường hợp dương tính thực tế. Nó trả lời câu hỏi: "Trong tất cả các mẫu thực sự dương tính, mô hình đã dự đoán đúng bao nhiêu?"
* Recall cao cho thấy mô hình ít bỏ sót các trường hợp dương tính (False Negatives).
* Công thức:



* Trong đó:
* TP (True Positive): Số mẫu được dự đoán đúng là dương tính.
* FN (False Negative): Số mẫu thực tế là dương tính nhưng bị dự đoán nhầm là âm tính.
  + - 1. F1-Score
* F1-Score là trung bình điều hòa (harmonic mean) giữa Precision và Recall. Nó cung cấp một thước đo cân bằng, kết hợp cả hai độ đo này để đánh giá hiệu suất của mô hình, đặc biệt hữu ích trong các tình huống dữ liệu bị mất cân đối (imbalanced dataset).
* Công thức:



* + 1. Kết quả đo
* Tỉ lệ train 40, test 60
* Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Bear | 0.62 | 0.37 | 0.47 | 75 |
| Bird | 0.72 | 0.67 | 0.7 | 82 |
| Cat | 0.37 | 0.52 | 0.43 | 73 |
| Cow | 0.55 | 0.46 | 0.5 | 78 |
| Deer | 0.43 | 0.49 | 0.46 | 76 |
| Dog | 0.61 | 0.47 | 0.53 | 73 |
| Dolphin | 0.67 | 0.81 | 0.73 | 77 |
| Elephant | 0.53 | 0.46 | 0.49 | 79 |
| Giraffe | 0.76 | 0.66 | 0.71 | 77 |
| Horse | 0.36 | 0.5 | 0.42 | 78 |
| Kangaroo | 0.41 | 0.37 | 0.39 | 75 |
| Lion | 0.48 | 0.53 | 0.5 | 78 |
| Panda | 0.54 | 0.62 | 0.58 | 81 |
| Tiger | 0.8 | 0.69 | 0.74 | 77 |
| Zebra | 0.87 | 0.89 | 0.88 | 82 |
| **Accuracy** |  |  | **0.57** | 1161 |
| **Macro Avg** | 0.58 | 0.57 | 0.57 | 1161 |
| **Weighted Avg** | 0.58 | 0.57 | 0.57 | 1161 |

* Confusion Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bear** | **Bird** | **Cat** | **Cow** | **Deer** | **Dog** | **Dolphin** | **Elephant** | **Giraffe** | **Horse** | **Kangaroo** | **Lion** | **Panda** | **Tiger** | **Zebra** |
| **Bear** | 28 | 0 | 4 | 3 | 2 | 1 | 10 | 7 | 0 | 3 | 1 | 4 | 11 | 1 | 0 |
| **Bird** | 0 | 55 | 5 | 3 | 1 | 0 | 2 | 0 | 3 | 5 | 5 | 3 | 0 | 0 | 0 |
| **Cat** | 2 | 0 | 38 | 1 | 5 | 2 | 4 | 1 | 4 | 1 | 0 | 8 | 4 | 2 | 1 |
| **Cow** | 1 | 2 | 4 | 36 | 3 | 0 | 2 | 4 | 0 | 8 | 5 | 3 | 6 | 3 | 1 |
| **Deer** | 4 | 2 | 1 | 2 | 37 | 1 | 0 | 2 | 1 | 14 | 5 | 4 | 3 | 0 | 0 |
| **Dog** | 4 | 2 | 13 | 4 | 0 | 34 | 0 | 2 | 1 | 2 | 1 | 6 | 3 | 0 | 1 |
| **Dolphin** | 1 | 6 | 2 | 0 | 0 | 1 | 62 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 2 |
| **Elephant** | 1 | 1 | 2 | 5 | 0 | 0 | 7 | 36 | 1 | 10 | 8 | 0 | 7 | 1 | 0 |
| **Giraffe** | 0 | 4 | 4 | 0 | 4 | 1 | 1 | 0 | 51 | 6 | 0 | 3 | 0 | 2 | 1 |
| **Horse** | 1 | 1 | 2 | 9 | 5 | 2 | 3 | 4 | 1 | 39 | 10 | 1 | 0 | 0 | 0 |
| **Kangaroo** | 0 | 0 | 6 | 0 | 12 | 0 | 0 | 8 | 5 | 9 | 28 | 7 | 0 | 0 | 0 |
| **Lion** | 0 | 3 | 7 | 0 | 9 | 4 | 0 | 2 | 0 | 4 | 3 | 41 | 3 | 1 | 1 |
| **Panda** | 3 | 0 | 9 | 1 | 1 | 10 | 1 | 0 | 0 | 0 | 2 | 4 | 50 | 0 | 0 |
| **Tiger** | 0 | 0 | 5 | 1 | 6 | 0 | 0 | 0 | 0 | 2 | 1 | 2 | 3 | 53 | 4 |
| **Zebra** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 3 | 0 | 0 | 0 | 3 | 73 |

* Tỉ lệ train 60, test 40
* Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Bear | 0.47 | 0.58 | 0.52 | 50 |
| Bird | 0.84 | 0.80 | 0.82 | 54 |
| Cat | 0.63 | 0.53 | 0.58 | 49 |
| Cow | 0.48 | 0.46 | 0.47 | 52 |
| Deer | 0.53 | 0.60 | 0.56 | 50 |
| Dog | 0.51 | 0.44 | 0.47 | 48 |
| Dolphin | 0.71 | 0.86 | 0.78 | 51 |
| Elephant | 0.52 | 0.64 | 0.58 | 53 |
| Giraffe | 0.80 | 0.84 | 0.82 | 51 |
| Horse | 0.52 | 0.63 | 0.57 | 52 |
| Kangaroo | 0.67 | 0.40 | 0.50 | 50 |
| Lion | 0.67 | 0.56 | 0.61 | 52 |
| Panda | 0.75 | 0.80 | 0.77 | 54 |
| Tiger | 0.85 | 0.65 | 0.73 | 51 |
| Zebra | 0.89 | 0.91 | 0.90 | 54 |
| **Accuracy** |  |  | **0.65** | 771 |
| **Macro Avg** | 0.66 | 0.65 | 0.65 | 771 |
| **Weighted Avg** | 0.66 | 0.65 | 0.65 | 771 |

* Confusion Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bear** | **Bird** | **Cat** | **Cow** | **Deer** | **Dog** | **Dolphin** | **Elephant** | **Giraffe** | **Horse** | **Kangaroo** | **Lion** | **Panda** | **Tiger** | **Zebra** |
| **Bear** | 29 | 0 | 1 | 3 | 1 | 1 | 5 | 8 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| **Bird** | 0 | 43 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 5 | 0 | 0 | 0 | 0 | 0 |
| **Cat** | 1 | 0 | 26 | 2 | 0 | 5 | 2 | 0 | 0 | 0 | 0 | 5 | 4 | 2 | 2 |
| **Cow** | 5 | 2 | 0 | 24 | 7 | 1 | 1 | 2 | 1 | 3 | 1 | 2 | 2 | 1 | 0 |
| **Deer** | 2 | 0 | 0 | 4 | 30 | 0 | 0 | 1 | 2 | 7 | 2 | 1 | 1 | 0 | 0 |
| **Dog** | 8 | 0 | 4 | 7 | 2 | 21 | 0 | 2 | 1 | 0 | 0 | 0 | 3 | 0 | 0 |
| **Dolphin** | 0 | 4 | 0 | 0 | 0 | 0 | 44 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Elephant** | 3 | 0 | 0 | 1 | 2 | 1 | 3 | 34 | 1 | 5 | 1 | 0 | 2 | 0 | 0 |
| **Giraffe** | 1 | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 43 | 3 | 0 | 0 | 0 | 0 | 0 |
| **Horse** | 1 | 1 | 0 | 3 | 2 | 0 | 0 | 7 | 2 | 33 | 3 | 0 | 0 | 0 | 0 |
| **Kangaroo** | 7 | 0 | 1 | 0 | 5 | 2 | 2 | 2 | 2 | 6 | 20 | 3 | 0 | 0 | 0 |
| **Lion** | 5 | 1 | 0 | 1 | 3 | 2 | 1 | 3 | 0 | 2 | 3 | 29 | 0 | 1 | 1 |
| **Panda** | 0 | 0 | 2 | 1 | 1 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 43 | 0 | 0 |
| **Tiger** | 0 | 0 | 4 | 1 | 2 | 2 | 3 | 0 | 0 | 0 | 0 | 2 | 1 | 33 | 3 |
| **Zebra** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 49 |

* Tỉ lệ train 70, test 30
* Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Bear | 0.56 | 0.62 | 0.59 | 37 |
| Bird | 0.74 | 0.71 | 0.72 | 41 |
| Cat | 0.62 | 0.44 | 0.52 | 36 |
| Cow | 0.42 | 0.44 | 0.43 | 39 |
| Deer | 0.68 | 0.45 | 0.54 | 38 |
| Dog | 0.62 | 0.56 | 0.59 | 36 |
| Dolphin | 0.62 | 0.95 | 0.75 | 38 |
| Elephant | 0.65 | 0.56 | 0.60 | 39 |
| Giraffe | 0.82 | 0.87 | 0.85 | 38 |
| Horse | 0.46 | 0.72 | 0.56 | 39 |
| Kangaroo | 0.67 | 0.38 | 0.48 | 37 |
| Lion | 0.66 | 0.74 | 0.70 | 39 |
| Panda | 0.78 | 0.90 | 0.84 | 40 |
| Tiger | 0.84 | 0.71 | 0.77 | 38 |
| Zebra | 0.95 | 0.85 | 0.90 | 41 |
| **Accuracy** |  |  | **0.66** | 576 |
| **Macro Avg** | 0.67 | 0.66 | 0.66 | 576 |
| **Weighted Avg** | 0.66 | 0.65 | 0.65 | 771 |

* Confusion Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bear** | **Bird** | **Cat** | **Cow** | **Deer** | **Dog** | **Dolphin** | **Elephant** | **Giraffe** | **Horse** | **Kangaroo** | **Lion** | **Panda** | **Tiger** | **Zebra** |
| **Bear** | 23 | 0 | 0 | 1 | 0 | 0 | 7 | 2 | 0 | 2 | 0 | 0 | 1 | 1 | 0 |
| **Bird** | 0 | 29 | 0 | 3 | 0 | 0 | 4 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 |
| **Cat** | 1 | 0 | 16 | 1 | 0 | 3 | 2 | 0 | 0 | 1 | 0 | 6 | 5 | 0 | 1 |
| **Cow** | 1 | 2 | 0 | 17 | 4 | 0 | 1 | 1 | 1 | 7 | 0 | 2 | 2 | 1 | 0 |
| **Deer** | 1 | 0 | 1 | 4 | 17 | 2 | 0 | 0 | 1 | 8 | 4 | 0 | 0 | 0 | 0 |
| **Dog** | 5 | 0 | 1 | 8 | 0 | 20 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **Dolphin** | 0 | 1 | 0 | 0 | 0 | 0 | 36 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **Elephant** | 2 | 0 | 0 | 1 | 1 | 3 | 4 | 22 | 0 | 4 | 0 | 1 | 0 | 1 | 0 |
| **Giraffe** | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 0 | 33 | 1 | 0 | 0 | 0 | 0 | 0 |
| **Horse** | 1 | 2 | 0 | 0 | 0 | 0 | 2 | 4 | 1 | 28 | 1 | 0 | 0 | 0 | 0 |
| **Kangaroo** | 2 | 1 | 3 | 3 | 2 | 2 | 1 | 1 | 1 | 4 | 14 | 3 | 0 | 0 | 0 |
| **Lion** | 3 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 29 | 0 | 0 | 0 |
| **Panda** | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 36 | 0 | 0 |
| **Tiger** | 2 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 1 | 27 | 1 |
| **Zebra** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 2 | 35 |

* Tỉ lệ train 80, test 20
* Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** |
| Bear | 0.67 | 0.72 | 0.69 | 25 |
| Bird | 0.81 | 0.93 | 0.86 | 27 |
| Cat | 0.50 | 0.58 | 0.54 | 24 |
| Cow | 0.50 | 0.58 | 0.54 | 26 |
| Deer | 0.50 | 0.64 | 0.56 | 25 |
| Dog | 0.79 | 0.62 | 0.70 | 24 |
| Dolphin | 0.63 | 0.68 | 0.65 | 25 |
| Elephant | 0.71 | 0.38 | 0.50 | 26 |
| Giraffe | 0.61 | 1.00 | 0.76 | 25 |
| Horse | 0.52 | 0.65 | 0.58 | 26 |
| Kangaroo | 0.67 | 0.32 | 0.43 | 25 |
| Lion | 0.69 | 0.69 | 0.69 | 26 |
| Panda | 0.88 | 0.81 | 0.85 | 27 |
| Tiger | 0.92 | 0.44 | 0.59 | 25 |
| Zebra | 0.92 | 0.89 | 0.91 | 27 |
| **Accuracy** |  |  | **0.67** | 383 |
| **Macro Avg** | 0.69 | 0.66 | 0.66 | 383 |
| **Weighted Avg** | 0.69 | 0.67 | 0.66 | 383 |

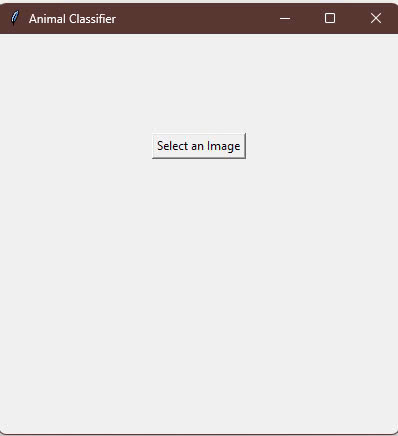
* Confusion Matrix

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Bear** | **Bird** | **Cat** | **Cow** | **Deer** | **Dog** | **Dolphin** | **Elephant** | **Giraffe** | **Horse** | **Kangaroo** | **Lion** | **Panda** | **Tiger** | **Zebra** |
| **Bear** | 18 | 0 | 0 | 2 | 0 | 0 | 2 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| **Bird** | 0 | 25 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Cat** | 1 | 0 | 14 | 0 | 1 | 3 | 0 | 0 | 0 | 1 | 0 | 4 | 0 | 0 | 0 |
| **Cow** | 1 | 0 | 0 | 15 | 3 | 0 | 1 | 0 | 3 | 2 | 1 | 0 | 0 | 0 | 0 |
| **Deer** | 0 | 0 | 1 | 1 | 16 | 0 | 0 | 1 | 1 | 5 | 0 | 0 | 0 | 0 | 0 |
| **Dog** | 2 | 1 | 0 | 5 | 0 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| **Dolphin** | 0 | 3 | 1 | 0 | 2 | 0 | 17 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Elephant** | 1 | 0 | 1 | 2 | 4 | 0 | 4 | 10 | 2 | 1 | 0 | 0 | 0 | 1 | 0 |
| **Giraffe** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Horse** | 3 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 17 | 1 | 0 | 1 | 0 | 0 |
| **Kangaroo** | 1 | 0 | 3 | 1 | 3 | 0 | 1 | 0 | 2 | 5 | 8 | 1 | 0 | 0 | 0 |
| **Lion** | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 2 | 18 | 0 | 0 | 0 |
| **Panda** | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 22 | 0 | 0 |
| **Tiger** | 0 | 0 | 5 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 0 | 2 | 1 | 11 | 2 |
| **Zebra** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 24 |

* 1. Giao diện chương trình

Người dùng khởi động chương trình bằng cách chạy file thực thi hoặc file Python của ứng dụng.

Cửa sổ giao diện chính xuất hiện, với một nút bấm duy nhất có nhãn "Select an Image" ở giữa màn hình.

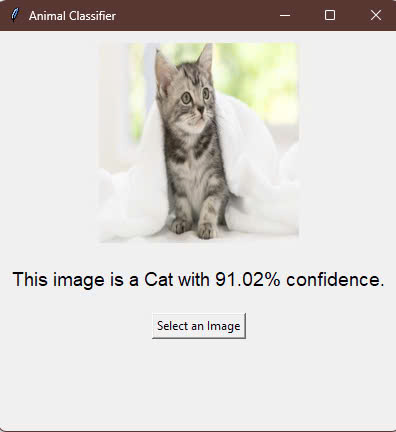


*Hình 3.3 Giao diện chính chương trình*

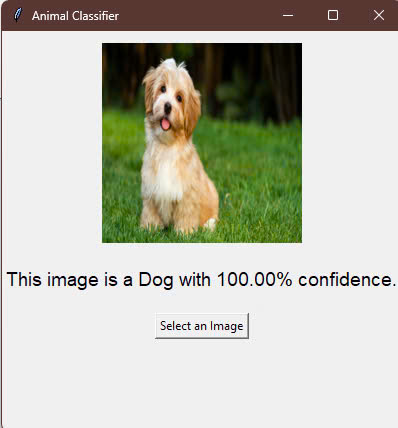
* 1. Kết quả thực nghiệm

Sau khi quá trình phân loại hoàn tất kết quả được hiển thị ngay bên dưới ảnh, bao gồm:

* Loại động vật (ví dụ: Dog, Cat, Lion).
* Độ chính xác của dự đoán (ví dụ: 91.02%).
* Người dùng có thể xem kết quả ngay lập tức mà không cần thao tác thêm.

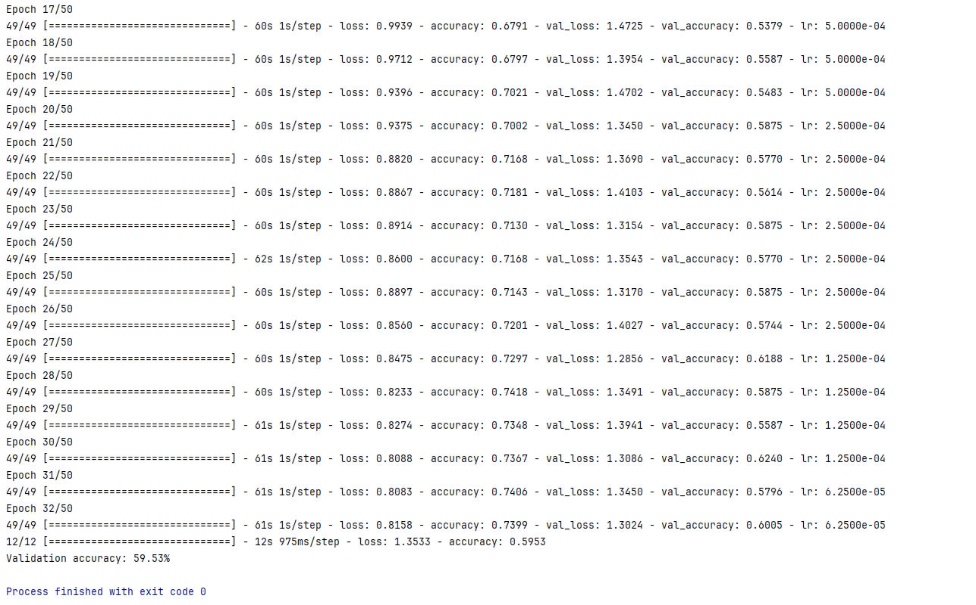


*Hình 3.4.1. Kết quả phân loại động vật*



*Hình 3.4.2. Kết quả phân loại động vật*

* Kết quả được hiển thị ở Terminal và được lưu vào file animal-classifier- model.keras



*Hình 3.4.3. Kết quả sau khi train*

* 1. Kết quả đạt được
* Khả năng phân loại chính xác: Chương trình đạt độ chính xác cao khi phân loại các loài động vật. Với sự hỗ trợ của mô hình VGG16, hệ thống có thể nhận diện nhiều loài động vật khác nhau (như chó, mèo, sư tử...) với mức độ tự tin rõ ràng (confidence score).
* Giao diện thân thiện: Giao diện đơn giản, dễ sử dụng, chỉ với một nút bấm để chọn ảnh, giúp người dùng không cần nhiều kiến thức kỹ thuật cũng có thể thao tác.
* Tích hợp hiệu quả mô hình học sâu: Hệ thống sử dụng mô hình VGG16 đã được huấn luyện trước, giúp giảm thời gian huấn luyện và tối ưu hóa hiệu năng.
* Xử lý ảnh nhanh chóng:Kết quả được trả về nhanh chóng sau khi người dùng chọn ảnh, phù hợp cho các tình huống sử dụng thực tế.
  1. Hạn chế
* Phụ thuộc vào mô hình VGG16: Mô hình VGG16 tuy mạnh nhưng không phải là tối ưu nhất hiện nay. Hiệu suất có thể bị giới hạn khi so với các mô hình hiện đại hơn như EfficientNet hay ResNet.
* Giới hạn loại động vật: Chương trình chỉ phân loại các loài động vật mà mô hình được huấn luyện (dựa trên tập dữ liệu ImageNet hoặc tương tự). Những loài động vật hiếm gặp hoặc không có trong tập dữ liệu có thể bị nhận diện sai.
* Giao diện đơn giản: Giao diện hiện tại chỉ có một nút bấm và hiển thị kết quả cơ bản. Không có tính năng mở rộng như hiển thị lịch sử các dự đoán, hoặc lưu kết quả.
* Yêu cầu phần cứng: Quá trình phân loại đòi hỏi sức mạnh tính toán từ GPU hoặc CPU mạnh, khiến hệ thống khó chạy mượt mà trên các thiết bị cấu hình thấp.
* Không hỗ trợ phân tích đa ảnh: Hệ thống chỉ hỗ trợ phân loại từng ảnh đơn lẻ. Việc phân tích nhiều ảnh cùng lúc chưa được triển khai.

DANH MỤC TÀI LIỆU THAM KHẢO

[1]. What is computer vision ?, Ambika 19/12/2023, www.medium.com

[2]. Object Recognition 3 things you need to know, [www.mathworks.com](http://www.mathworks.com)

[3]. XLA\_C2\_3.pdf, Trường Đại Học Công Nghệ Đông Á

[4]. What is Machine Learning, [www.ibm.com](http://www.ibm.com)

[5]. Phân lớp DL.pdf, Trường Đại Học Công Nghệ Đông Á

[6]. Introduction to Convolution Neural Network, [www.geeksforgeeks.org](http://www.geeksforgeeks.org)

[7]. VGG-16 | CNN model, [www.geeksforgeeks.org](http://www.geeksforgeeks.org)

[8]. YOLO : You Only Look Once – Real Time Object Detection, geeksforgeeks

[9]. R-CNN – Region-Based Convolutional Neural Networks, geeksforgeeks