**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** | 20211610 | Nguyễn Thị Băng | DCCNTT12.10.6 |
| **2** | 20211423 | Nguyễn Thị Duyên | DCCNTT12.10.6 |
| **3** | 20211660 | Nguyễn Thu Hà | DCCNTT12.10.6 |
| **4** | 20211663 | Nguyễn Hồng Phượng | DCCNTT12.10.6 |
| **5** | 20211729 | Đào Thị Tâm | DCCNTT12.10.6 |

**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** | 20211610 | Nguyễn Thị Băng | DCCNTT12.10.6 |
| **2** | 20211423 | Nguyễn Thị Duyên | DCCNTT12.10.6 |
| **3** | 20211660 | Nguyễn Thu Hà | DCCNTT12.10.6 |
| **4** | 20211663 | Nguyễn Hồng Phượng | DCCNTT12.10.6 |
| **5** | 20211729 | Đào Thị Tâm | DCCNTT12.10.6 |

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

**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**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp tín chỉ: DCCNTT12.10.6** | |
| **Cán bộ chấm thi 1**  *Lương Thị Hồng Lan* | **Cán bộ chấm thi 2** |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Thị**  **Băng** | **Nguyễn Thị Duyên** | **Nguyễn Thu**  **Hà** | **Nguyễn Hồng Phượng** | **Đào Thị Tâm** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20211610 | 20211423 | 20211660 | 20212663 | 20211729 |
| **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* |  |  |  |  |  |

**MỤC LỤC**

[DANH MỤC BẢNG BIỂU, SƠ ĐỒ 8](#_Toc184683834)

[LỜI MỞ ĐẦU 9](#_Toc184683835)

[CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT 10](#_Toc184683836)

[I. Tổng quan về thị giác máy tính 10](#_Toc184683837)

[**1. Khái niệm** 10](#_Toc184683838)

[**1.2. Ứng dụng của thị giác máy tính** 11](#_Toc184683839)

[II. Tổng quan về học máy 12](#_Toc184683840)

[**1. Học có giám sát** 12](#_Toc184683841)

[**1.1. KNN** 13](#_Toc184683842)

[**1.2. Mô hình mạng nơ-ron** 16](#_Toc184683843)

[**2. Học không giám sát** 18](#_Toc184683844)

[**2.1. K – Means** 19](#_Toc184683845)

[**2.2. Fuzzy C – Means** 21](#_Toc184683846)

[**3. Học bán giám sát** 22](#_Toc184683847)

[CHƯƠNG 2 - XÂY DỰNG HỆ THỐNG PHÁT TRIỂN MÔ HÌNH PHÂN LOẠI NHẬN DIỆN CÁC LOÀI ĐỘNG VẬT 26](#_Toc184683848)

[1 Mô tả bài toán 26](#_Toc184683849)

[**1.1. Bối cảnh** 26](#_Toc184683850)

[**1.2. Mục tiêu** 26](#_Toc184683851)

[**1.3. Đặc điểm bài toán** 26](#_Toc184683852)

[2. Xây dựng hệ thống phát triển mô hình phân loại nhận diện các loài động vật 26](#_Toc184683853)

[**2.1 Mô hình sử dụng** 26](#_Toc184683854)

[**2.2 Ngôn ngữ:** 28](#_Toc184683855)

[**2.3.Các bước thực hiện** 28](#_Toc184683856)

[**2.4. Một số mã nguồn quan trọng** 31](#_Toc184683857)

[CHƯƠNG 3 – THỰC NGHIỆM 32](#_Toc184683858)

[I. Dữ liệu 32](#_Toc184683859)

[**1. Dữ liệu huấn luyện (Training Data)** 32](#_Toc184683860)

[**2. Dữ liệu dự đoán (Prediction Data)** 33](#_Toc184683861)

[II. Độ đo đánh giá 34](#_Toc184683862)

[**1.Độ chính xác (Accuracy)** 34](#_Toc184683863)

[**2. Độ nhạy (Recall)** 34](#_Toc184683864)

[**3. Độ chính xác trung bình (Precision)** 34](#_Toc184683865)

[**4. Điểm F1 (F1 Score)** 34](#_Toc184683866)

[III. Kết quả thực nghiệm 37](#_Toc184683868)

[**1. Hình ảnh minh họa** 37](#_Toc184683869)

[**2. Thời gian chạy** 39](#_Toc184683870)

[**3. Nhận xét chung** 39](#_Toc184683871)

[KẾT LUẬN 40](#_Toc184683872)

[TÀI LIỆU THAM KHẢO 41](#_Toc184683873)

# DANH MỤC BẢNG BIỂU, SƠ ĐỒ

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng biểu và sơ đồ** | **Trang** |
| Bảng 1 | Bảng đo độ đánh giá của Model | 51 |
| Hình 1 | Giao diện file tệp trong Visual Studio Code | 58 |
| Hình 2 | Giao diện chạy bài | 59 |
| Hình 3 | Giao diện chọn dữ liệu | 60 |
| Hình 4 | Giao diện nhận dạng con chó | 61 |
| Hình 5 | Giao diện nhận dạng con mèo | 61 |
| Hình 6 | Giao diện nhận dạng con gà | 62 |
| Hình 7 | Giao diện nhận dạng con capybara | 62 |
| Hình 8 | Giao diện nhận dạng con chim cánh cụt | 63 |

# LỜI MỞ ĐẦU

Xử lý ảnh số có nhiều ứng dụng trong thực tế. Một trong những ứng dụng sớm nhất là xử lý ảnh từ nhiệm vụ Ranger 7 tại phòng thí nghiệm Jet Propulsion vào những năm đầu của thập kỷ 60. Hệ thống chụp hình gắn trên tàu vũ trụ có một số hạn chế về kích thước và trọng lượng, do đó ảnh nhận được bị giảm chất lượng như mờ, méo hình học và nhiễu nền. Các ảnh đó được xử lý thành công nhờ máy tính số. Hình ảnh của mặt trăng và sao hỏa mà chúng ta thấy trong các tạp chí đều được xử lý bằng máy tính số.

 Bên cạnh ngôn ngữ giao tiếp, các thông tin dưới dạng hình ảnh đóng một vai trò rất quan trọng trong việc trao đổi thông tin. Trong công nghệ thông tin, xử lý ảnh và đồồ họa đã chiếm một vị trí rất quan trọng bởi vì các đặc tính đầy hấp dẫn đã tạo nên một sự phân biệt với các lĩnh vực khác. Chúng giới thiệu các phương pháp và kỹ thuật để tạo ra các ảnh và xử lý các ảnh này. Ta biết rằng phần lớn các thông tin mà con người thu thập được qua thị giác đều bắt nguồn từ các ảnh. Do đó việc xử lý ảnh và đồ họa là một bộ phận quan trọng trong việc trao đổi thông tin giữa người và máy.

Trong cuộc sống hiện đại ngày nay, người máy càng đóng vai trò quan trọng trong công nghiệp và gia đình. Chúng sẽ thực hiện những công việc rất nhàm chán hoặc nguy hiểm, và những công việc mà tốc độ và độ chính xác vượt quá khả năng của con người. Khi người máy trở nên tinh vi hơn, thị giác máy tính sẽ đóng vai trò ngày càng quan trọng. Người ta sẽ đòi hỏi người máy không những phát hiện và nhận dạng các bộ phận công nghiệp, mà còn hiểu được những gì chúng thấy và đưa ra những hành động phù hợp. Xử lý ảnh sẽ tác động lớn đến thị giác máy tính. Những ứng dụng khác của xử lý ảnh là vô hạn. Ngoài những ứng dụng đã thảo luận ở trên, còn bao gồm cả các lĩnh vực khác như điện tử gia đình, thiên văn học, sinh vật học, vật lý, nông nghiệp, địa lý, nhân chủng học,..v.v..

Xử lý ảnh ngoài được sử dụng để nhận diện gương mặt con người còn được áp dụng vô nhận diện các loài động vật. Để có thể nghiên cứu và tìm hiểu rõ hơn nhóm chúng em đã chọn ra đề tài “*Xây dựng hệ thống phát triển mô hình phân loại ảnh để nhận diện các loài động vật*” dưới sự hướng dẫn của giảng viên *Lương Thị Hồng Lan*.

# CHƯƠNG 1 – CƠ SỞ LÝ THUYẾT

## I. Tổng quan về thị giác máy tính

### **1. Khái niệm**

**- Định nghĩa:**

Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc giúp máy tính thu nhận, xử lý và diễn giải dữ liệu từ hình ảnh hoặc video. Với mục tiêu cải thiện chất lượng và phân tích nội dung ảnh, các kỹ thuật hiện đại như mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đã trở thành nền tảng quan trọng trong việc xây dựng hệ thống cải thiện độ phân giải ảnh.

**-Mục tiêu chính:**

* Tăng cường độ phân giải và chi tiết của hình ảnh thông qua các thuật toán học sâu, đặc biệt là CNN.
* Tự động hóa việc khôi phục và nâng cao chất lượng hình ảnh để phục vụ các ứng dụng trong thực tế.

**-Các ứng dụng phổ biến của việc cải thiện độ phân giải ảnh:**

* Nâng cao chất lượng video: Dùng trong truyền phát video hoặc chỉnh sửa hậu kỳ.
* Hình ảnh y tế: Cải thiện độ rõ nét trong hình ảnh CT, MRI để hỗ trợ chẩn đoán chính xác hơn.
* Thực tế ảo và tăng cường (VR/AR): Đảm bảo chất lượng hình ảnh mượt mà và chi tiết trong môi trường ảo.
* Phân tích ảnh vệ tinh: Nâng cao độ chi tiết của hình ảnh phục vụ trong nông nghiệp, quân sự, và môi trường.
* Phục chế ảnh cũ: Khôi phục và làm sắc nét những hình ảnh lịch sử, bị mờ hoặc tổn hại theo thời gian.

**-Các bài toán chính liên quan đến cải thiện độ phân giải ảnh:**

* Super-Resolution (Siêu phân giải): Dự đoán và tái tạo các chi tiết bị mất trong hình ảnh có độ phân giải thấp.
* Noise Reduction (Giảm nhiễu): Loại bỏ nhiễu trong hình ảnh mà vẫn giữ được chi tiết.
* Image Denoising (Khử nhiễu ảnh): Kết hợp với các kỹ thuật tăng cường độ phân giải để cải thiện chất lượng tổng thể của hình ảnh.

**-Kỹ thuật CNN áp dụng trong cải thiện độ phân giải ảnh:**

* Super-Resolution CNN (SRCNN): Mô hình CNN cơ bản đầu tiên ứng dụng trong việc cải thiện độ phân giải ảnh.
* Enhanced Deep Super-Resolution (EDSR): Mô hình nâng cao loại bỏ các phần dư thừa trong mạng để tập trung vào chất lượng đầu ra.
* Generative Adversarial Networks (GANs): Ứng dụng GAN để tạo ra các chi tiết chân thực hơn khi tăng độ phân giải ảnh.
* Residual Networks (ResNet): Tăng hiệu quả của CNN bằng cách xử lý thông tin trong các tầng sâu hơn mà không làm mất mát dữ liệu.

-**Lợi ích của hệ thống cải thiện độ phân giải ảnh bằng CNN:**

* Tăng cường chất lượng hình ảnh: Phục hồi chi tiết bị mất một cách chính xác và tự nhiên.
* Tự động hóa: Giảm thiểu sự can thiệp thủ công trong các công đoạn chỉnh sửa ảnh.
* Ứng dụng rộng rãi: Từ công nghiệp, y tế, giải trí đến nghiên cứu khoa học.

**-Công cụ và thư viện phổ biến:**

* OpenCV: Hỗ trợ các bước xử lý tiền đề trước khi áp dụng mạng CNN.
* TensorFlow/Keras: Phát triển và huấn luyện các mô hình CNN cải thiện độ phân giải ảnh.
* PyTorch: Một framework mạnh mẽ khác để triển khai các mô hình học sâu.
* ESRGAN: Một mô hình GAN tiên tiến trong việc cải thiện độ phân giải ảnh.

### **1.2. Ứng dụng của thị giác máy tính**

* **Nhận diện khuôn mặt**
* **Xe tự lái**
* **Chẩn đoán y khoa:**
* **Phân tích hành vi trong bán lẻ**
* **Giám sát giao thông**
* **Nông nghiệp thông minh**
* **Hỗ trợ người khuyết tật**
* **Công nghiệp sản xuất**

## II. Tổng quan về học máy

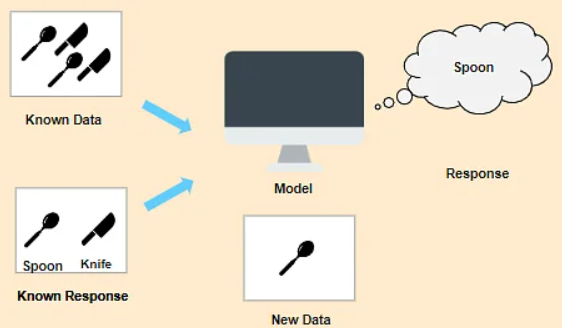
### **1. Học có giám sát**

Trong kỹ thuật học có giám sát, các hệ thống máy tính sẽ đào tạo mô hình, học hỏi từ dữ liệu dưới sự giám sát từ bên ngoài.

Chúng sẽ chứa một mô hình có thể dự đoán, với sự tập hợp của các dữ liệu đã được gắn nhãn trước đó. Dữ liệu được gắn nhãn là những dữ liệu đã biết câu trả lời đích.

Các loại học máy có giám sát:

* Hồi quy: Dự đoán một giá trị đầu ra duy nhất bằng cách sử dụng dữ liệu huấn luyện.
* Hồi quy logistic: Được sử dụng để ước tính các giá trị rời rạc dựa trên một tập hợp các biến độc lập đã cho. Nó giúp bạn dự đoán xác suất xảy ra sự kiện bằng cách khớp dữ liệu với hàm logit. Vì vậy, nó còn được gọi là hồi quy logistic. Vì nó dự đoán xác suất nên giá trị đầu ra của nó nằm trong khoảng từ 0 đến 1.
* Phân loại: Là nhóm đầu ra bên trong một lớp. Nếu thuật toán cố gắng gắn nhãn đầu vào thành hai lớp riêng biệt thì nó được gọi là phân loại nhị phân. Việc lựa chọn giữa nhiều hơn hai lớp được gọi là phân loại nhiều lớp.



* Điểm mạnh: Cây phân loại hoạt động rất tốt trong thực tế.
* Điểm yếu: Không bị ràng buộc, các cây riêng lẻ có xu hướng bị trang bị quá mức.
* Phân loại Naïve Bayes: Mô hình Naive Bayesian (NBN) rất dễ xây dựng và rất hữu ích cho các tập dữ liệu lớn. Phương pháp này bao gồm các biểu đồ chu kỳ trực tiếp với một cha và một số con. Nó giả định sự độc lập giữa các nút con tách biệt khỏi nút cha của chúng.
* Cây quyết định: Phân loại cá thể bằng cách sắp xếp chúng dựa trên giá trị đặc trưng. Trong phương pháp này, mỗi chế độ là một tính năng của một thể hiện. Nó phải được phân loại và mỗi nhánh đại diện cho một giá trị mà nút có thể đảm nhận. Đây là một kỹ thuật được sử dụng rộng rãi để phân loại. Trong phương pháp này, phân loại là một cây được gọi là cây quyết định. Nó giúp bạn ước tính giá trị thực.
* Máy hỗ trợ vector: Là một loại thuật toán học được phát triển vào năm 1990. Phương pháp này dựa trên kết quả của lý thuyết học thống kê do Vap Nik giới thiệu.
* Các máy SVM cũng được kết nối chặt chẽ với các hàm kernel, đây là khái niệm trung tâm của hầu hết các nhiệm vụ học tập. Kernel framework và SVM được sử dụng trong nhiều lĩnh vực khác nhau. Nó bao gồm việc truy xuất thông tin đa phương tiện, tin sinh học và nhận dạng mẫu.

### **1.1. KNN**

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.



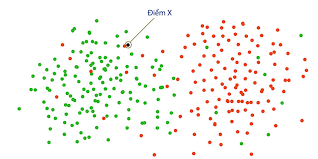
Thuật toán KNN dựa trên nguyên tắc các đối tượng hoặc điểm dữ liệu gần nhau trong không gian đặc trưng có khả năng thuộc cùng một lớp hoặc có kết quả đầu ra tương tự nhau.

Thuật toán KNN gán một điểm dữ liệu mới cho lớp phổ biến nhất trong k hàng xóm gần nhất của nó. Giá trị của k là một siêu tham số cần được chỉ định trước khi đào tạo mô hình.

* KNN hoạt động theo quy trình gồm 4 bước chính:
* Xác định tham số K (số láng giềng gần nhất).
* Tính khoảng cách từ điểm đang xét đến tất cả các điểm trong tập dữ liệu cho trước.
* Sắp xếp các khoảng cách đó theo thứ tự tăng dần.
* Xét trong tập K điểm gần nhất với điểm đang xét, nếu số lượng điểm của loại nào cao hơn thì coi như điểm đang xét thuộc loại đó.

KNN giả định rằng những dữ liệu tương tự nhau sẽ nằm gần nhau trong một không gian đặc trưng, và nhiệm vụ của thuật toán là tìm ra k điểm dữ liệu gần nhất với điểm cần phân loại. Khoảng cách giữa các điểm thường được tính bằng các phương pháp như khoảng cách Euclid, khoảng cách Manhattan, hoặc các công thức tính khoảng cách.

* Ưu điểm:
* Đơn giản và dễ hiểu: Thuật toán KNN đơn giản và dễ hiểu. Nó không yêu cầu bất kỳ giả định nào về phân phối dữ liệu cơ bản và có thể dễ dàng triển khai bằng bất kỳ ngôn ngữ lập trình nào.
* Không tham số: Thuật toán KNN không tham số, có nghĩa là nó không đưa ra bất kỳ giả định nào về phân phối dữ liệu cơ bản. Điều này làm cho nó phù hợp cho cả phân phối dữ liệu tuyến tính và phi tuyến tính.
* Không có giai đoạn đào tạo: Thuật toán KNN không yêu cầu giai đoạn đào tạo. Mô hình được lưu trữ đơn giản trong bộ nhớ và các điểm dữ liệu mới có thể được phân loại hoặc dự đoán trong thời gian thực.
* Linh hoạt: Thuật toán KNN có thể được sử dụng cho cả nhiệm vụ phân loại và hồi quy. Nó cũng có thể được điều chỉnh để phân loại nhiều lớp bằng cách sử dụng một sửa đổi đơn giản.

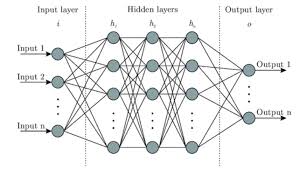


* Mạnh đối với dữ liệu nhiễu: Thuật toán KNN mạnh đối với dữ liệu nhiễu vì nó không đưa ra bất kỳ giả định nào về phân phối dữ liệu cơ bản. Nó có thể xử lý các giá trị ngoại lai và các điểm dữ liệu nhiễu bằng cách lấy trung bình các giá trị của các hàng xóm gần nhất.
* Không cần kỹ thuật tính năng: Thuật toán KNN không yêu cầu kỹ thuật tính năng. Nó có thể hoạt động với bất kỳ loại dữ liệu đầu vào nào, bao gồm dữ liệu phân loại và số.
* Độ chính xác cao: Thuật toán KNN có thể đạt độ chính xác cao, đặc biệt khi giá trị của k được chọn cẩn thận. Nó thường được sử dụng làm thuật toán chuẩn cho nhiều tác vụ học máy.
* Nhược điểm:
* Đắt về mặt tính toán: Thuật toán KNN rất tốn kém về mặt tính toán, đặc biệt là khi xử lý các tập dữ liệu lớn. Đối với mỗi điểm dữ liệu mới, thuật toán phải tính toán khoảng cách đến tất cả các điểm dữ liệu khác trong tập dữ liệu, điều này có thể tốn thời gian.
* Sử dụng nhiều bộ nhớ: Thuật toán KNN sử dụng nhiều bộ nhớ vì nó phải lưu trữ tất cả dữ liệu đào tạo trong bộ nhớ. Đây có thể là một vấn đề đối với các tập dữ liệu lớn, đặc biệt là khi làm việc với dữ liệu nhiều chiều.
* Độ nhạy với tỷ lệ tính năng: Thuật toán KNN nhạy cảm với tỷ lệ tính năng. Nếu các tính năng có tỷ lệ khác nhau, thì thuật toán có thể tăng trọng số cho các tính năng có tỷ lệ lớn hơn. Đây có thể là một vấn đề khi xử lý dữ liệu nhiều chiều hoặc khi các tính năng có các đơn vị đo lường khác nhau.
* Nhạy cảm với lựa chọn k(Phải tìm giá trị k tối ưu): Thuật toán KNN nhạy cảm với lựa chọn k, số lượng hàng xóm gần nhất để xem xét. Nếu k quá nhỏ thì thuật toán có thể quá nhạy cảm với nhiễu trong dữ liệu. Nếu k quá lớn thì thuật toán có thể không nắm bắt được cấu trúc cục bộ của dữ liệu.
* Thời gian dự đoán có thể chậm: Vì thuật toán KNN không có giai đoạn huấn luyện và yêu cầu tính toán khoảng cách cho mỗi điểm mới nên thời gian dự đoán có thể chậm đối với các tập dữ liệu lớn.
* Nguy cơ về chiều: Khi số lượng tính năng tăng lên, thuật toán KNN trở nên kém hiệu quả hơn do khoảng cách giữa các điểm dữ liệu ngày càng giống nhau. Điều này được gọi là lời nguyền của chiều và có thể gây khó khăn cho việc tìm kiếm những người hàng xóm gần nhất có ý nghĩa.

### **1.2. Mô hình mạng nơ-ron**

Sử dụng một mô hình mạng nơ-ron để phân loại hình ảnh. Cụ thể hơn, chúng ta sử dụng một mô hình ResNet-18 đã được huấn luyện trước trên tập dữ liệu ImageNet để phân loại hình ảnh của các loài động vật.

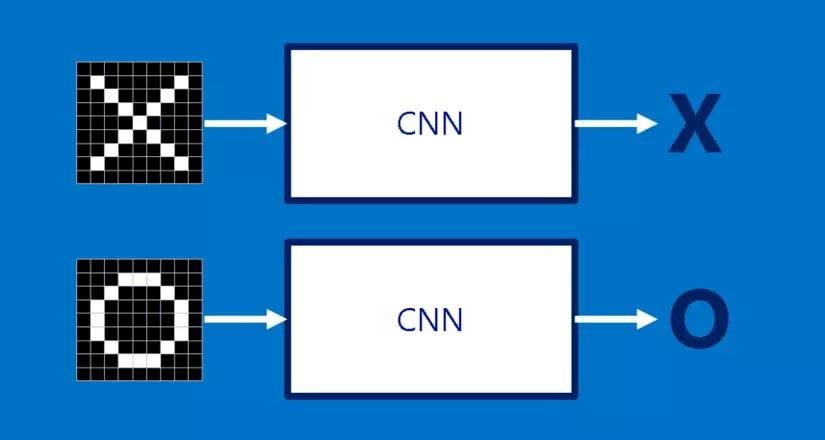
Mô hình ResNet-18 là một mô hình mạng nơ-ron tích chập sâu, được thiết kế để giải quyết các bài toán phân loại hình ảnh. Mô hình này bao gồm nhiều lớp tích chập, lớp kết nối đầy đủ và các lớp khác để trích xuất đặc trưng từ hình ảnh và dự đoán nhãn cho hình ảnh.



**Mô tả về mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)**

Mạng nơ-ron tích chập (CNN) là một mô hình Deep Learning được thiết kế để xử lý dữ liệu có cấu trúc dạng lưới, như hình ảnh [3]. CNN bao gồm ba thành phần chính: lớp tích chập (convolutional layer) để trích xuất đặc trưng từ các vùng cục bộ trong ảnh, lớp gộp (pooling layer) để giảm kích thước dữ liệu và tăng tính bền vững của mô hình với biến dạng, và lớp kết nối đầy đủ (fully connected layer) để thực hiện dự đoán cuối cùng. Một đặc điểm nổi bật của CNN là khả năng tự động học các đặc trưng quan trọng từ dữ liệu, thay vì phải trích xuất thủ công như trong Machine Learning truyền thống [3]. Nhờ đó, CNN đặc biệt phù hợp với bài toán nhận diện chữ số viết tay, nơi mà dữ liệu đầu vào có nhiều biến dạng về kích thước, góc độ và cách viết. Khả năng học tốt các đặc trưng không gian và bền bỉ với nhiễu của CNN giúp mô hình đạt độ chính xác cao hơn so với các phương pháp khác.

Ví dụ đơn giản về cách **Mạng nơ-ron tích chập (Convolutional Neural Network - CNN)** có thể được sử dụng để phân loại hình ảnh.



**Dữ liệu đầu vào:**

- Hình ảnh được đưa vào là các ký tự như "X" và "O".

- Những hình ảnh này được biểu diễn dưới dạng ma trận các pixel (điểm ảnh) trắng và đen.

**Quá trình qua CNN:**

CNN là một loại mạng học sâu chuyên xử lý và phân loại dữ liệu hình ảnh. CNN tự động trích xuất các đặc trưng từ hình ảnh thông qua các lớp:

Lớp tích chập (Convolution Layer): Trích xuất các đặc trưng như cạnh, đường nét.

Lớp gộp (Pooling Layer): Giảm kích thước dữ liệu để tăng hiệu quả tính toán.

Lớp dày đặc (Dense Layer): Đưa ra dự đoán cuối cùng.

**Dự đoán:**

Hình ảnh ký tự "X": CNN xử lý và dự đoán rằng đây là ký tự "X".

Hình ảnh ký tự "O": CNN xử lý và dự đoán rằng đây là ký tự "O".

### **2. Học không giám sát**

Học máy không giám sát (Unsupervised Learning): là một phương pháp học máy trong đó mô hình được huấn luyện trên dữ liệu chưa được gán nhãn. Thay vì dựa vào các cặp dữ liệu đầu vào và đầu ra (input-output), học máy không giám sát cố gắng phát hiện các cấu trúc ẩn, mối quan hệ hoặc mẫu trong dữ liệu.

Học không giám sát có thể được nhóm lại thành 2 loại chính:

* Phân cụm (Clustering)

Là phương pháp học không giám sát, trong đó các đối tượng dữ liệu được nhóm lại thành các cụm (clusters) dựa trên tính tương đồng hoặc khoảng cách giữa chúng. Các điểm trong cùng một cụm có nhiều đặc điểm giống nhau hơn so với các điểm trong các cụm khác.

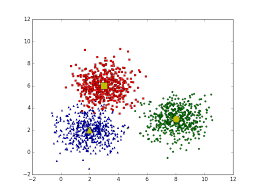
* Liên kết (Association)

Là một phương pháp học không giám sát, tập trung vào việc tìm kiếm các mối quan hệ hoặc quy luật tiềm ẩn trong dữ liệu. Các quy luật liên kết giúp xác định các mẫu thường xuất hiện cùng nhau trong một tập hợp dữ liệu.

### **2.1. K – Means**

Phân cụm K-Means là một thuật toán Học máy không giám sát, thuật toán này nhóm tập dữ liệu chưa được gắn nhãn thành các cụm khác nhau. Học máy không giám sát là quá trình dạy máy tính sử dụng dữ liệu chưa được gắn nhãn, chưa được phân loại và cho phép thuật toán hoạt động trên dữ liệu đó mà không cần giám sát. Không có bất kỳ đào tạo dữ liệu nào trước đó, công việc của học máy trong trường hợp này là sắp xếp dữ liệu chưa được sắp xếp theo các điểm tương đồng, mẫu và biến thể.

Trong thuật toán K-means clustering, chúng ta không biết nhãn (label) của từng điểm dữ liệu. Mục đích là làm thể nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.



* **Các bước của thuật toán K-Means Clustering**

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K.

Đầu ra: Các center M và label vector cho từng điểm dữ liệu Y.

1. Chọn K điểm bất kỳ làm các center ban đầu.

2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.

3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

5. Quay lại bước 2.

* **Mục đích**

Từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra tâm của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng

Trong thuật toán K-means, độ đo khoảng cách được sử dụng để xác định độ tương đồng giữa các phần tử và ảnh hưởng đến hình dạng của các cụm. Độ đo khoảng cách được sử dụng phải đáp ứng được một số yêu cầu nhất định. Nó phải là một hàm số phi tuyến, không âm và có tính đối xứng.

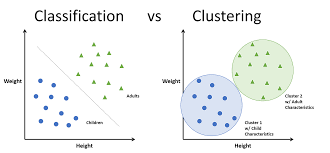
Phương pháp để đo khoảng cách như: Euclidean trường hợp phổ biến nhất là xác định khoảng cách giữa hai điểm. Nếu chúng ta có điểm p và điểm q, khoảng cách euclidean là một đường thẳng bình thường. Đó là khoảng cách giữa hai điểm trong không gian Euclide.



### **2.2. Fuzzy C – Means**

Fuzzy C-Means (FCM) là một thuật toán học máy không giám sát, được sử dụng để phân cụm dữ liệu thành các cụm dựa trên mức độ thành viên (membership degree). Không giống như K-Means, trong đó mỗi điểm dữ liệu chỉ thuộc về một cụm, FCM cho phép mỗi điểm dữ liệu thuộc về nhiều cụm với mức độ thành viên khác nhau (giá trị từ 0 đến 1).

Học máy không giám sát là quá trình dạy máy tính sử dụng dữ liệu chưa được gắn nhãn hoặc phân loại, từ đó cho phép thuật toán tự tìm ra các cấu trúc hoặc mẫu trong dữ liệu.



Trong trường hợp của FCM, thuật toán sẽ phân chia dữ liệu vào các cụm dựa trên sự tương đồng, nhưng mỗi điểm có thể có mối liên hệ với nhiều cụm.

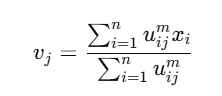
* **Các bước của thuật toán Fuzzy C- Means**

Đầu vào: Dữ liệu X và số lượng cụm cần tìm C.

Đầu ra: Ma trận thành viên U và tâm cụm V.

1. Khởi tạo ngẫu nhiên một ma trận thành viên U có kích thước n × C (với n là số lượng điểm dữ liệu và C là số lượng cụm). Các giá trị trong ma trận này năm trong khoảng [0,1], sao cho tổng mức độ thành viên của mỗi điểm dữ liệu với tất cả các cụm bằng 1.

2. Sử dụng ma trận U để tính tâm của các cụm theo.



* : Tâm cụm thứ j.
* : Mức độ thành viên của điểm dữ liệu đối với cụm j.
* m: Tham số mờ, thường là 2.

3. Cập nhật ma trận thành viên U.

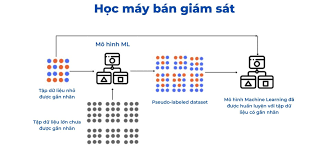
4. Kiểm tra hội tụ, dừng thuật toán nếu sự thay đổi của ma trận U giữa hai lần lặp liên tiếp nhỏ hơn một ngưỡng đã định trước.

5. Quay lại bước 2, nếu chưa hội tụ.

* **Mục đích:** Từ dữ liệu đầu vào và số lượng cụm cần tìm, thuật toán FCM xác định tâm của mỗi cụm và mức độ thành viên của từng điểm dữ liệu đối với các cụm đó. Điều này cho phép phân chia dữ liệu một cách linh hoạt hơn so với phương pháp K-Means.
* **Đặc điểm:**
* Mỗi điểm dữ liệu có thể thuộc về nhiều cụm, phù hợp với dữ liệu có sự chồng lấn (overlap) giữa các cụm.
* Phương pháp tiếp cận mờ giúp thuật toán linh hoạt hơn trong các bài toán phức tạp.
* Tâm cụm và mức độ thành viên được cập nhật liên tục để tối ưu hóa việc phân cụm.
* Trong thuật toán FCM, khoảng cách giữa các điểm dữ liệu và tâm cụm được sử dụng để xác định mức độ thành viên. Khoảng cách thường được đo bằng Euclidean.

### **3. Học bán giám sát**

* **Học bán giám sát**
* Học bán giám sát là một phương pháp học máy thu hẹp khoảng cách giữa học có giám sát và học không giám sát bằng cách sử dụng cả dữ liệu có nhãn và không có nhãn.
* Phương pháp này tận dụng lợi thế của dữ liệu không có nhãn phong phú trong khi giảm thiểu yêu cầu đối với các trường hợp có nhãn, thường tốn kém và mất thời gian để có được.
* **Tầm quan trọng và sự liên quan**
* Trong các kịch bản học máy thông thường, học có giám sát phụ thuộc rất nhiều vào dữ liệu được gắn nhãn, trong đó mỗi đầu vào được ghép nối với một đầu ra chính xác. Mặt khác, học không giám sát không sử dụng bất kỳ nhãn nào.



* Học bán giám sát tạo ra sự cân bằng bằng cách sử dụng một phần nhỏ dữ liệu được gắn nhãn cùng với một tập dữ liệu không được gắn nhãn lớn hơn.
* Cách tiếp cận này đặc biệt có lợi khi việc gắn nhãn dữ liệu tốn kém hoặc không thực tế nhưng việc thu thập một lượng lớn dữ liệu thô là khả thi.
* **Cách thức hoạt động**
* Các mô hình học bán giám sát thường được xây dựng bằng cách ban đầu đào tạo trên một tập dữ liệu có nhãn nhỏ hơn.
* Sau khi mô hình cơ bản được xây dựng, dữ liệu không có nhãn được kết hợp để tinh chỉnh và cải thiện mô hình hơn nữa.
* Các kỹ thuật như tự đào tạo, đồng đào tạo và phương pháp dựa trên đồ thị thường được sử dụng:
* Tự đào tạo : Mô hình dự đoán nhãn cho dữ liệu chưa có nhãn và những dự đoán này được sử dụng như thể chúng là nhãn thực trong vòng đào tạo tiếp theo.
* Đào tạo chung : Hai bộ phân loại được đào tạo trên các góc nhìn khác nhau của dữ liệu, hỗ trợ lẫn nhau bằng cách dán nhãn dữ liệu chưa được dán nhãn.
* Phương pháp dựa trên đồ thị : Các trường hợp dữ liệu được biểu diễn dưới dạng các nút trong đồ thị và các thuật toán học bán giám sát sẽ truyền thông tin nhãn qua các cạnh.
* **Ứng dụng trong thế giới thực**
* Học bán giám sát được sử dụng trong nhiều lĩnh vực khác nhau:
* Chăm sóc sức khỏe : Trong hình ảnh y tế, chẳng hạn như chụp MRI hoặc CT, thường tốn nhiều công sức để chú thích tất cả hình ảnh. Một số hình ảnh được các chuyên gia dán nhãn và mô hình học hỏi từ cả hình ảnh có nhãn và không có nhãn để giảm bớt gánh nặng chú thích.
* Xử lý ngôn ngữ tự nhiên (NLP) : Bằng cách giảm lượng dữ liệu văn bản được gắn nhãn thủ công, học bán giám sát có thể hỗ trợ các tác vụ như phân loại văn bản và phân tích tình cảm.
* **Sự khác biệt chính từ các khái niệm liên quan**
* Học bán giám sát không nên bị nhầm lẫn với các khái niệm tương tự như học chủ động và học chuyển giao.
* [Học chủ động](https://www.ultralytics.com/vi/glossary/active-learning) liên quan đến việc lựa chọn các mẫu thông tin nhất để dán nhãn nhằm nâng cao hiệu quả học tập.
* Trong khi đó, [học chuyển giao](https://www.ultralytics.com/vi/glossary/transfer-learning) liên quan đến việc chuyển giao kiến thức từ một lĩnh vực để cải thiện hiệu suất trong lĩnh vực khác.
* **Thách thức và cân nhắc**
* Học bán giám sát phải đối mặt với những thách thức như đảm bảo độ tin cậy của các nhãn dự đoán và xử lý các phân phối dữ liệu đa dạng.
* Điều này đòi hỏi phải lựa chọn cẩn thận các kỹ thuật thuật toán và đôi khi là xác thực bổ sung để đảm bảo rằng dữ liệu chưa được gắn nhãn được sử dụng hiệu quả mà không gây hiểu lầm cho quá trình học.
* **Kết thúc**
* Học bán giám sát là một công cụ mạnh mẽ trong bộ công cụ AI tận dụng sự phong phú của dữ liệu chưa được gắn nhãn để xây dựng các mô hình dự đoán hiệu quả hơn.
* Nó có ứng dụng trong nhiều lĩnh vực, từ chăm sóc sức khỏe đến NLP, cho phép các mô hình mạnh mẽ hơn với ít ví dụ được gắn nhãn hơn.
* Hiểu và khám phá khái niệm này có thể dẫn đến các giải pháp sáng tạo khi tồn tại các nút thắt cổ chai trong việc gắn nhãn dữ liệu.

Khám phá cách Ultralytics HUB có thể hỗ trợ các nỗ lực AI và học máy tương tự bằng cách cung cấp nền tảng đa năng để đào tạo và triển khai mô hình.

# CHƯƠNG 2 - XÂY DỰNG HỆ THỐNG PHÁT TRIỂN MÔ HÌNH PHÂN LOẠI NHẬN DIỆN CÁC LOÀI ĐỘNG VẬT

## 1 Mô tả bài toán

### **1.1. Bối cảnh**

Công nghệ nhận diện hình ảnh ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực như bảo tồn động vật, giáo dục, và nghiên cứu sinh học. Để giải quyết bài toán nhận diện các loài động vật, chúng ta cần một hệ thống có khả năng phân loại chính xác các loài từ ảnh đầu vào. Trong bài toán này, mô hình ResNet-18, một kiến trúc học sâu hiệu quả, sẽ được sử dụng để xây dựng hệ thống.

### **1.2. Mục tiêu**

Phát triển một hệ thống có khả năng:

* Nhận diện chính xác các loài động vật từ ảnh đầu vào.
* Cung cấp dự đoán nhanh chóng, với độ chính xác cao.
* Dễ dàng mở rộng để nhận diện thêm các loài động vật khác.

### **1.3. Đặc điểm bài toán**

* Đầu vào:Ảnh đầu vào chứa một loài động vật, định dạng JPEG, PNG.
* Đầu ra:Một nhãn (tên loài động vật) kèm theo xác suất dự đoán.
* Tập dữ liệu:
* Bộ dữ liệu gồm hình ảnh của nhiều loài động vật với nhãn tương ứng .
* Tập dữ liệu mẫu: tự thu thập.

## 2. Xây dựng hệ thống phát triển mô hình phân loại nhận diện các loài động vật

### **2.1 Mô hình sử dụng**

- ResNet-18 là một kiến trúc mạng nơ-ron tích chập (CNN) được đề xuất trong bài báo nổi tiếng *"Deep Residual Learning for Image Recognition"* (He et al., 2015). Đây là một trong các mô hình **ResNet** (Residual Network), nổi bật nhờ khả năng giải quyết vấn đề "gradient vanishing" khi độ sâu mạng tăng.

**ResNet-18** bao gồm **18 lớp trọng số học được**, cụ thể là:

* 1 lớp Convolution đầu tiên
* 16 lớp Convolution trong các khối residual
* 1 lớp Fully Connected (FC) cuối cùng.

+ **Tính chất nổi bật của ResNet-18**

**Khả năng huấn luyện mạng sâu:**

* Skip Connection làm giảm lỗi gradient vanishing.
* Mô hình dễ huấn luyện hơn các mạng CNN sâu truyền thống.

**Hiệu quả cao:**

* ResNet-18 cân bằng giữa độ sâu và chi phí tính toán.
* Thích hợp cho các bài toán với dữ liệu trung bình (hình ảnh nhỏ hoặc vừa).

**Ứng dụng đa dạng:**

* Nhận diện đối tượng (Image Classification).
* Phân đoạn ảnh (Image Segmentation).
* Chuyển giao học (Transfer Learning).

**-Thư viện:**

PyTorch là một thư viện mã nguồn mở hỗ trợ học sâu (deep learning) được phát triển bởi Facebook's AI Research lab (FAIR). PyTorch được sử dụng rộng rãi trong cộng đồng nghiên cứu và phát triển trí tuệ nhân tạo, nhờ vào tính linh hoạt, dễ sử dụng, và khả năng tích hợp tốt với nhiều công cụ khác.

**+Ứng dụng của PyTorch:**

* Học sâu (Deep Learning): Xây dựng và huấn luyện các mô hình mạng neuron nhân tạo (ANNs), từ cơ bản đến phức tạp.
* Xử lý ngôn ngữ tự nhiên (NLP): PyTorch thường được sử dụng trong các mô hình ngôn ngữ lớn như GPT hoặc BERT.
* Xử lý ảnh và video: Tạo các mô hình phân loại, phát hiện đối tượng, hoặc tạo ảnh.
* Reinforcement Learning (Học tăng cường): PyTorch hỗ trợ tốt cho việc triển khai các thuật toán học tăng cường.

**+Ưu điểm của PyTorch:**

* Dễ học và dễ sử dụng nhờ cú pháp trực quan, tương tự như Python.
* Khả năng tương thích cao với các thư viện khác của Python như NumPy, SciPy.
* Cộng đồng phát triển mạnh mẽ và tài liệu phong phú.

### **2.2 Ngôn ngữ**

**-Python** là một ngôn ngữ lập trình bậc cao, được thiết kế với mục tiêu dễ đọc, dễ học và dễ sử dụng. Nó được tạo ra bởi **Guido van Rossum** vào năm 1991 và hiện tại là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

**-Ưu điểm của Python**

* Học nhanh và dễ sử dụng cho người mới bắt đầu.
* Tính linh hoạt cao, sử dụng được trong nhiều lĩnh vực.
* Cộng đồng mạnh mẽ và hỗ trợ tốt.

**-Hạn chế của Python**

* Tốc độ chậm hơn so với ngôn ngữ như C, C++ do Python là ngôn ngữ thông dịch.
* Không phù hợp cho các ứng dụng đòi hỏi hiệu năng cao như xử lý thời gian thực.
* Sử dụng nhiều tài nguyên bộ nhớ hơn so với các ngôn ngữ như C.

### **2.3.Các bước thực hiện**

Chia tập dữ liệu ảnh thành 2 tập Train và Validation

Kiểm tra xem file mô hình có tồn tại không, không tồn tại thì thực hiện Training

Sử dụng mô hình đã được huấn luyện trước bằng cách sử dụng Resnet18, 5 lớp động vật

|  |
| --- |
| model = models.resnet18(pretrained=True)  # Sử dụng mô hình đã được huấn luyện trước      num\_ftrs = model.fc.in\_features      model.fc = nn.Linear(num\_ftrs, 5) |

- Tiền xử lý dữ liệu ảnh trước khi đưa vào mô hình học sâu

|  |
| --- |
| transform = transforms.Compose([          transforms.Resize(256),          transforms.CenterCrop(224),          transforms.RandomHorizontalFlip(),  # Phép biến đổi ngẫu nhiên          transforms.RandomRotation(10),  # Xoay hình ảnh ngẫu nhiên          transforms.ToTensor(),          transforms.Normalize(              mean=[0.485, 0.456, 0.406],              std=[0.229, 0.224, 0.225]          )      ]) |

- Tạo một bộ nạp dữ liệu để lấy dữ liệu ảnh để train theo từng batch, mỗi batch 32 ảnh

|  |
| --- |
| train\_dataset = datasets.ImageFolder(r'C:/Users/ASUS/Downloads/xla/animal-identification-main/animal-identification-main/data/animals/train', transform=transform)      train\_loader = DataLoader(train\_dataset, batch\_size=32, shuffle=True) |

- Huấn luyện mô hình trong 70 epoch, tính độ chính xác accuracy trong mỗi lần huấn luyện và thời gian huấn luyện

|  |
| --- |
| criterion = nn.CrossEntropyLoss()      optimizer = optim.Adam(model.parameters(), lr=0.001)      start\_time = time.time()  # Start time measurement      # Huấn luyện mô hình      for epoch in range(70):  # Huấn luyện trong 70 epoch          model.train()          correct = 0          total = 0          for inputs, labels in train\_loader:              optimizer.zero\_grad()              outputs = model(inputs)              loss = criterion(outputs, labels)              loss.backward()              optimizer.step()              # Calculate accuracy for the current batch              \_, predicted = torch.max(outputs.data, 1)  # Get the index of the predicted class              total += labels.size(0)              correct += (predicted == labels).sum().item()          # Print epoch loss and accuracy          train\_accuracy = 100 \* correct / total          print(f'Epoch [{epoch+1}/{70}] Loss: {loss.item():.4f} | Train Accuracy: {train\_accuracy:.2f}%')          # Calculate total training time      end\_time = time.time()      training\_time = end\_time - start\_time      print(f"Training completed in {training\_time:.2f} seconds.")      # Sau khi huấn luyện xong, lưu mô hình      torch.save(model.state\_dict(), 'animal\_classifier.pth')      print("Model trained and saved as 'animal\_classifier.pth'.") |

- Xác định danh sách các loài động vật

|  |
| --- |
| class\_names = ['capybara', 'chim-canh-cut', 'cho', 'ga', 'meo'] |

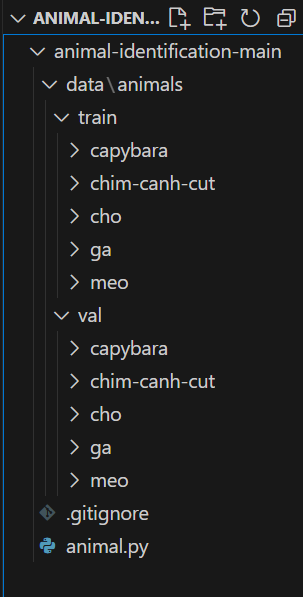
- Tải hình ảnh muốn test lên, áp dụng các phép biến đổi cho hình ảnh

|  |
| --- |
| # Tải hình ảnh từ đường dẫn              img = Image.open(file\_path)              # Áp dụng các phép biến đổi cho hình ảnh              img\_t = transform(img)              img\_t = img\_t.unsqueeze(0) |

- Dự đoán nhãn cho hình ảnh và dán nhãn lên

|  |
| --- |
| # Dự đoán nhãn cho hình ảnh              with torch.no\_grad():                  output = model(img\_t)                  predicted\_class = output.argmax(dim=1).item()              # Lấy nhãn được dự đoán              predicted\_label = class\_names[predicted\_class] |

### **2.4. Một số mã nguồn quan trọng**



Train: folder ảnh dữ liệu được sử dụng để huấn luyện mô hình

Val: folder ảnh dữ liệu được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện

Animal.py: code chương trình

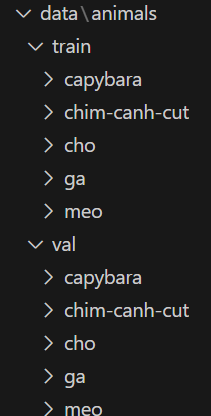
# CHƯƠNG 3 – THỰC NGHIỆM

## I. Dữ liệu

**1. Dữ liệu huấn luyện (Training Data)**

- Nguồn dữ liệu: Bộ dữ liệu được sử dụng để huấn luyện mô hình được lưu trữ tại đường dẫn D:/animal-identification-main/animal-identification-main/data/animals/train. Đây là một thư mục chứa hình ảnh của các loài động vật, được tổ chức theo cấu trúc sau:

* Mỗi lớp động vật (ví dụ: capybara, chim-cánh-cụt, chó, gà, mèo) được lưu trong một thư mục riêng. Tên thư mục là tên của lớp.



* Hình ảnh trong mỗi thư mục là các ảnh của loài động vật tương ứng.

- Phép biến đổi (Transformations): Dữ liệu huấn luyện được chuẩn hóa và thực hiện augmentations (làm tăng tính đa dạng của dữ liệu) thông qua các phép biến đổi:

* Resize(256): Chuyển đổi kích thước ảnh thành 256 pixel.
* CenterCrop(224): Cắt ảnh để tạo kích thước đầu vào cố định (224x224 pixel).
* RandomHorizontalFlip(): Lật ngang ngẫu nhiên hình ảnh.
* RandomRotation(10): Xoay ngẫu nhiên hình ảnh trong khoảng ±10 độ.
* ToTensor(): Chuyển đổi ảnh thành tensor.
* Normalize(mean, std): Chuẩn hóa dữ liệu dựa trên giá trị trung bình và độ lệch chuẩn của tập dữ liệu ImageNet.

**2. Dữ liệu dự đoán (Prediction Data)**

* Nguồn dữ liệu: Người dùng cung cấp một hình ảnh đầu vào thông qua giao diện (Giao diện được xây dựng bằng Tkinter).
* Đường dẫn: Hình ảnh được chọn từ hệ thống tệp của người dùng thông qua hộp thoại chọn file của Tkinter (filedialog.askopenfilename()).
* Phép biến đổi (Transformations): Hình ảnh được áp dụng cùng các phép biến đổi như trên (nhưng không bao gồm augmentations như lật và xoay).
* Đầu ra: Mô hình dự đoán hình ảnh đầu vào thuộc về một trong 5 lớp động vật: capybara, chim-canh-cut, cho, ga, meo.

## II. Độ đo đánh giá

### **1.Độ chính xác (Accuracy)**

**-Công thức: Accuracy ​**

* **(True Positive):** Dự đoán đúng là dương.
* **TN (True Negative):** Dự đoán đúng là âm.
* **FP (False Positive):** Dự đoán sai là dương.
* **FN (False Negative):** Dự đoán sai là âm.

**-Ý nghĩa:**

* Đo lường tỷ lệ mẫu được dự đoán đúng (cả dương và âm) so với tổng số mẫu.

**-Ứng dụng:**

* Hiệu quả trong các bài toán mà số lượng lớp cân đối (vd. số lượng ảnh mèo và chó gần bằng nhau).
* Không phù hợp khi dữ liệu mất cân đối (vd. 90% ảnh là mèo, 10% là chó).

### **2. Độ nhạy (Recall)**

**-Công thức:** Recall=

**-Ý nghĩa:** Tỷ lệ mẫu dương được phát hiện đúng trên tổng số mẫu dương.

**-Ứng dụng:** Quan trọng trong các bài toán cần phát hiện toàn bộ mẫu dương, ví dụ:

* Phát hiện ung thư (không bỏ sót bệnh nhân)
* Phát hiện lỗi trong ảnh sản phẩm.

### **3. Độ chính xác trung bình (Precision)**

**-Công thức:** Precision=

**-Ý nghĩa:** Tỷ lệ các dự đoán dương đúng trong tổng số các dự đoán dương.

**-Ứng dụng:** Quan trọng trong các bài toán cần hạn chế báo động giả, ví dụ:

* Hệ thống cảnh báo bảo mật (không được báo nhầm quá nhiều).
* Phát hiện vật cản trong xe tự hành.

### **4. Điểm F1 (F1 Score)**

**-Công thức:** F1= 2 x ​

**-Ý nghĩa:** Trung bình điều hòa giữa Precision và Recall, hữu ích trong trường hợp dữ liệu mất cân đối.

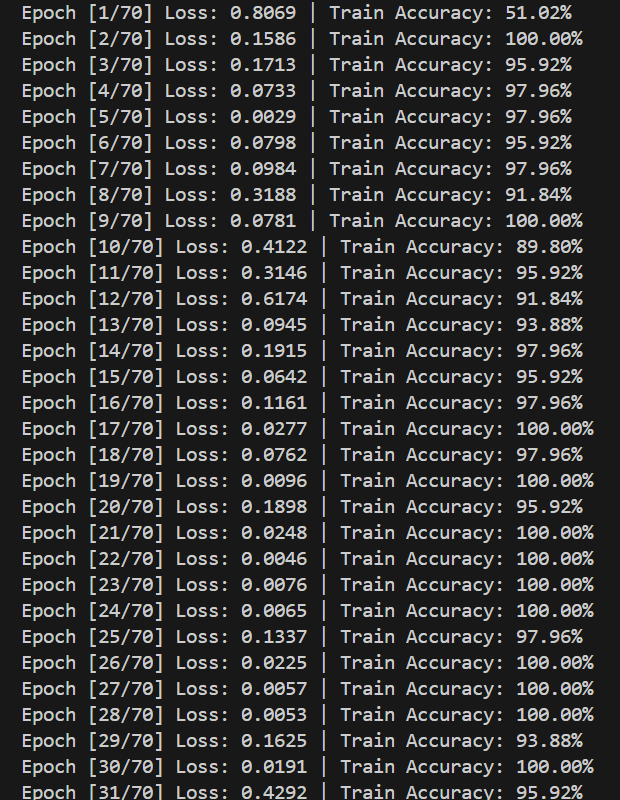
**-Ứng dụng:** Sử dụng khi cả Recall và Precision đều quan trọng, ví dụ:

* Phân loại ảnh y tế.
* Hệ thống nhận dạng khuôn mặt trong các môi trường phức tạp.

**Kết quả:**

- Tính độ chính xác accuracy sau mỗi lần huấn luyện:

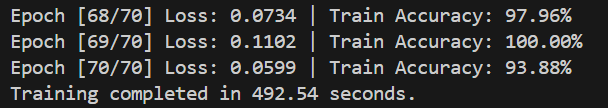
|  |
| --- |
| total += labels.size(0)              correct += (predicted == labels).sum().item()          # Print epoch loss and accuracy          train\_accuracy = 100 \* correct / total          print(f'Epoch [{epoch+1}/{70}] Loss: {loss.item():.4f} | Train Accuracy: {train\_accuracy:.2f}%') |



- Thời gian chạy mô hình mất 492.54 s:

|  |
| --- |
| start\_time = time.time()  # Start time measurement |

|  |
| --- |
| end\_time = time.time()      training\_time = end\_time - start\_time      print(f"Training completed in {training\_time:.2f} seconds.") |



### **Báo cáo phân loại (Classification Report)**

*Bảng 1: Bảng đo độ đánh giá của model*

|  |  |  |
| --- | --- | --- |
|  | Loss | Accuracy |
| Capybara | 0,0734 | 97,96% |
| Chim cánh cụt | 0,1102 | 100,00% |
| Cho | 0,0599 | 93,38% |
| Ga | 0,1204 | 100,00% |
| Meo | 0,0589 | 98,87% |

## III. Kết quả thực nghiệm

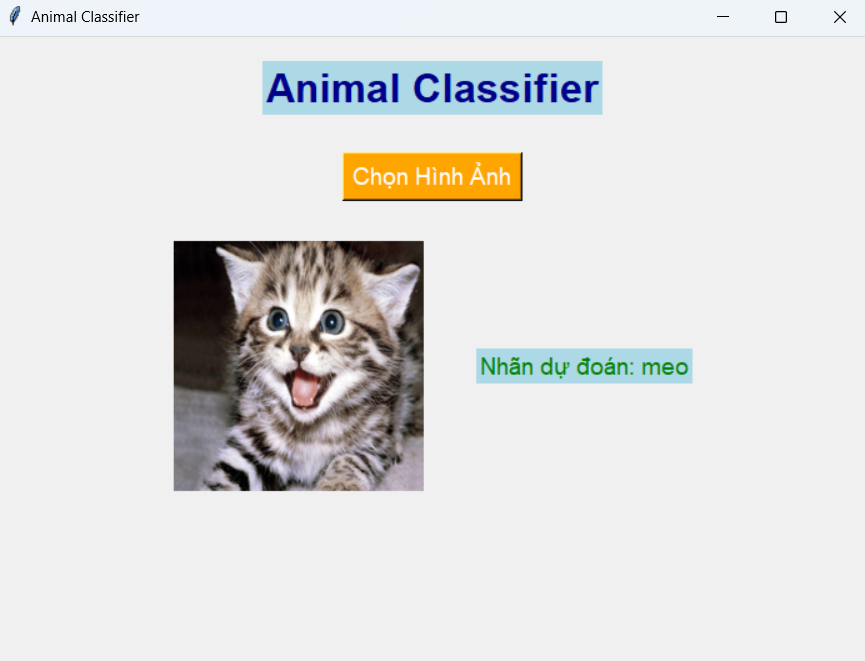
### **1. Hình ảnh minh họa**

* Tiến hành nhận dạng con chó



*Hình 1: Giao diện nhận dạng con chó*

* Tiến hành nhận dạng con mèo



*Hình 2: Giao diện nhận dạng con mèo*

* Tiến hành nhận diện con gà



*Hình 3: Giao diện nhận dạng con gà*

* Tiến hành nhận diện con capybara



*Hình 4: Giao diện nhận dạng con capybara*

* Tiến hành nhận dạng con chim cánh cụt



*Hình 5: Giao diện nhận dạng con chim cánh cụt*

### **2. Thời gian chạy**

- Thời gian xử lý trung bình: 50ms, việc sử dụng 70 epoch đồng nghĩa với việc mô hình sẽ phải trải qua 70 lần huấn luyện trên toàn bộ tập dữ liệu khiến thời gian chạy bài mất 1 khoảng thời gian để có thể load kịp.

- Có khả năng áp dụng với mô hình nhỏ như bài tập dựa trên tốc độ và độ chính xác.

### **3. Nhận xét chung**

- Tổng kết những kết quả đạt được: Mô hình có độ chính xác cao. Tốc độ xử lý nhanh, nhưng còn tồn tại hạn chế trong các trường hợp đặc biệt (Vd: góc nghiêng chụp ảnh con vật bị mờ, hình ảnh không rõ, góc chụp nghiêng,..)

- Hướng cải thiện:

* Sử dụng thêm các kỹ thuật tăng cường dữ liệu.
* Kết hợp mô hình với thuật toán tiền xử lý ảnh để xử lý tốt hơn trong các trường hợp ảnh bị nhiễu.

# KẾT LUẬN

Về kết quả đạt được, ứng dụng đã hoàn thành các yêu cầu được đặt ra, chương trình cung cấp tương đối đầy đủ các tính năng của một ứng dụng dạy và học, chương trình đã có được giao diện thân thiện, đẹp mắt dễ sử dụng. Client và server, liên kết rõ ràng, độc lập có thể dễ dàng trong công việc quản lý, phát triển. Cũng như nhóm chúng em đã nắm bắt được cũng như áp dụng kiến thức học tập lý thuyết trên lớp vào đồ án lần này.

Về hạn chế thì nhóm chúng em cũng rút ra được một số mặt hạn chế không đáng có do vấn đề thời gian hoàn thiện đề tài có chút gấp rút. Trong tương lai sắp tới nhóm chúng em sẽ cố gắng khắc phục những mặt hạn chế trong thời gian qua như chương trình chưa được sử dụng offline, tốc độ chương trình chưa được tối ưu hoàn toàn.

Cũng qua đồ án lần này nhóm chúng em rất biết ơn sự giảng dạy và dẫn dắt của giảng viên thầy *Lương Thị Hồng Lan* đã hướng dẫn tận tình chúng em trong thời gian vừa qua. Trong đồ án lần này nhóm chúng em vẫn còn một số mặt hạn chế và một chút sai sót không thể tránh khỏi, nhóm chúng em sẽ cố gắng nhiều hơn trong thời gian tới.

Nhóm chúng em xin chân thành cảm ơn thầy cô!

# TÀI LIỆU THAM KHẢO

[1] N. T. Tuan, Deep learning cơ bản, 2020.

[2] P. J. Braspenning, F. Thuijsman and A. J. M. M. Weijters, Artificial Neural

[3] Cornell University Library PSEC Documentation Committee, Feb. 2010.

[Online]. Available: *http://www.library.cornell.edu/resrch/citmanage/apa.*