

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT HÀ NỘI



BÁO CÁO HỌC THUẬT

**“Ứng dụng trí tuệ nhân tạo trong nhận diện hình ảnh
phục vụ cho việc giảng dạy môn Trí tuệ nhân tạo”.**

GV. TS. NGÔ THỊ PHƯƠNG THẢO

Hà Nội, tháng 06/2025

MỤC LỤC

MỤC LỤC	2
MỞ ĐẦU.....	3
CHƯƠNG 1: Cơ sở lý thuyết.....	5
1.1. Khái niệm và vai trò của AI trong thời đại hiện nay.....	5
1.2. Nhận diện hình ảnh là gì? Ứng dụng thực tế.....	5
1.3. Vị trí của nhận diện hình ảnh trong giảng dạy AI	5
1.4. Các mô hình AI tiên tiến trong nhận diện ảnh.....	6
CHƯƠNG 2: Xây dựng hệ thống nhận diện hình ảnh hỗ trợ giảng dạy.....	9
2.1. Quy trình xây dựng hệ thống nhận diện ảnh hỗ trợ giảng dạy.....	9
2.2. Triển khai mô hình trên giao diện web/app đơn giản.....	19
2.3. Trình diễn công cụ nhận diện hình ảnh.....	20
Kết luận.....	21

MỞ ĐẦU

Sự phát triển vượt bậc của trí tuệ nhân tạo (AI) đang tạo ra những đổi thay sâu rộng trong lĩnh vực giáo dục, đặc biệt là trong việc giảng dạy các môn học liên quan đến chính bản thân AI. Việc kết hợp AI vào quá trình giảng dạy không chỉ giúp nâng cao hiệu quả truyền đạt kiến thức mà còn mang đến trải nghiệm học tập sinh động, trực quan và mang tính tương tác cao cho người học.

Hiện nay, việc giảng dạy môn Trí tuệ nhân tạo tại các cơ sở đào tạo chủ yếu vẫn dựa vào bài giảng lý thuyết, các thuật toán mô tả qua slide và ví dụ minh họa mang tính khái quát. Tuy nhiên, những phương pháp truyền thống này còn nhiều hạn chế trong việc giúp sinh viên hình dung được quá trình xử lý thực tế của các mô hình AI, đặc biệt là trong lĩnh vực thị giác máy tính (Computer Vision) – một nhánh quan trọng của AI. Việc ứng dụng các mô hình AI vào nhận diện hình ảnh mang lại nhiều lợi ích thiết thực cho cả giảng viên và sinh viên, cụ thể như sau:

❖ Lợi ích đối với giảng viên:

- **Trực quan hóa nội dung giảng dạy:** Thay vì mô tả trừu tượng, giảng viên có thể sử dụng công cụ nhận diện hình ảnh để minh họa trực tiếp hoạt động của các mô hình AI.
- **Hỗ trợ xây dựng bài giảng tương tác:** AI giúp tạo ra các bài học minh họa sinh động thông qua hình ảnh, giúp sinh viên dễ tiếp thu và ghi nhớ.
- **Tự động phân tích hình ảnh mẫu:** Giảng viên có thể sử dụng AI để phân tích hình ảnh do sinh viên cung cấp và đưa ra phản hồi ngay trong lớp.
- **Khuyến khích sinh viên nghiên cứu ứng dụng:** Tạo điều kiện cho sinh viên thực hành trực tiếp trên các mô hình AI, thúc đẩy tư duy nghiên cứu.

❖ Lợi ích đối với sinh viên

- **Tiếp cận mô hình AI thực tế:** Sinh viên có thể trực tiếp tương tác với công cụ AI, quan sát cách hệ thống phân loại và nhận diện hình ảnh.
- **Thực hành thuật toán AI thông qua hình ảnh:** Việc thao tác với hình ảnh giúp sinh viên hiểu rõ hơn về các khái niệm như học sâu, mạng nơ-ron tích chập, tăng cường dữ liệu,...

- **Phát triển kỹ năng triển khai ứng dụng AI:** Sinh viên không chỉ học lý thuyết mà còn được hướng dẫn xây dựng, huấn luyện và triển khai mô hình AI đơn giản.
- **Tăng khả năng làm việc nhóm và sáng tạo:** Các bài tập nhóm gắn với xử lý hình ảnh thực tế giúp sinh viên rèn luyện kỹ năng cộng tác và giải quyết vấn đề.

Tuy nhiên, việc ứng dụng AI trong nhận diện hình ảnh phục vụ giảng dạy cũng đặt ra một số thách thức như yêu cầu thiết bị có khả năng xử lý cao, sự am hiểu về công nghệ từ phía giảng viên, và việc lựa chọn công cụ phù hợp với trình độ sinh viên. Do đó, nghiên cứu này hướng đến việc triển khai một mô hình AI đơn giản, dễ tích hợp vào bài giảng và phù hợp với môi trường giáo dục, qua đó giúp nâng cao hiệu quả giảng dạy và học tập môn Trí tuệ nhân tạo.

❖ Mục tiêu nghiên cứu

Mục tiêu chính của đề tài là xây dựng và đề xuất một giải pháp ứng dụng các mô hình trí tuệ nhân tạo hiện đại trong nhận diện và phân loại hình ảnh nhằm hỗ trợ hiệu quả cho quá trình giảng dạy và học tập môn Trí tuệ nhân tạo.

Cụ thể, đề tài hướng đến các mục tiêu sau:

1. **Khảo sát và đánh giá các mô hình AI hiện đại** trong lĩnh vực thị giác máy tính, đặc biệt là các mô hình học sâu như CNN (Convolutional Neural Network), YOLO (You Only Look Once), ResNet, EfficientNet,... để chọn ra mô hình phù hợp với môi trường giáo dục.
2. **Xây dựng một hệ thống nhận diện hình ảnh minh họa**, có khả năng phân loại các đối tượng trong hình ảnh, phục vụ minh họa trực quan cho bài giảng môn Trí tuệ nhân tạo.
3. **Thiết kế bài giảng tích hợp AI**, sử dụng hình ảnh đầu vào thực tế, giúp sinh viên hiểu rõ cách hoạt động của các mô hình học máy và học sâu thông qua minh họa trực tiếp và các ví dụ thực hành.
4. **Đánh giá hiệu quả sử dụng công cụ AI trong giảng dạy**, thông qua thử nghiệm thực tế và phản hồi từ sinh viên và giảng viên nhằm điều chỉnh, hoàn thiện giải pháp.
5. **Khuyến khích sinh viên tiếp cận và ứng dụng AI**, tạo tiền đề cho việc phát triển kỹ năng nghiên cứu, sáng tạo và triển khai mô hình trí tuệ nhân tạo vào các lĩnh vực khác trong tương lai.

CHƯƠNG 1: Cơ sở lý thuyết

1.1. Khái niệm và vai trò của AI trong thời đại hiện nay

Trí tuệ nhân tạo (Artificial Intelligence – AI) là lĩnh vực nghiên cứu và phát triển các hệ thống máy tính có khả năng thực hiện những nhiệm vụ mà trước đây đòi hỏi trí thông minh của con người, như nhận thức, lập luận, học tập và ra quyết định. AI bao gồm nhiều nhánh như học máy (Machine Learning), học sâu (Deep Learning), xử lý ngôn ngữ tự nhiên (NLP), thị giác máy tính (Computer Vision), và các hệ thống chuyên gia.

Trong thời đại chuyển đổi số, AI đóng vai trò then chốt trong việc nâng cao hiệu quả sản xuất, tối ưu hóa vận hành, cải thiện trải nghiệm người dùng và hỗ trợ ra quyết định thông minh. Các ứng dụng AI ngày càng trở nên phổ biến trong nhiều lĩnh vực như y tế, tài chính, giao thông, thương mại điện tử và đặc biệt là giáo dục.

1.2. Nhận diện hình ảnh là gì? Ứng dụng thực tế

Nhận diện hình ảnh (Image Recognition) là quá trình sử dụng các mô hình học máy hoặc học sâu để xác định và phân loại đối tượng, đặc điểm, hoặc mẫu hình trong một bức ảnh hoặc chuỗi ảnh. Đây là một trong những ứng dụng tiêu biểu và quan trọng nhất của thị giác máy tính – một nhánh của AI.

Các ứng dụng thực tế của nhận diện hình ảnh bao gồm:

- Nhận diện khuôn mặt trong hệ thống an ninh và điện thoại thông minh.
- Phân loại sản phẩm trong thương mại điện tử.
- Phát hiện vật thể trong xe tự hành.
- Chẩn đoán hình ảnh y tế như X-quang, MRI.
- Kiểm tra chất lượng sản phẩm trong công nghiệp.
- Đếm người, phân tích hành vi trong các hệ thống giám sát.

1.3. Vị trí của nhận diện hình ảnh trong giảng dạy AI

Nhận diện hình ảnh là một chủ đề nền tảng và trực quan trong chương trình giảng dạy môn Trí tuệ nhân tạo. Thông qua các bài học và dự án liên quan đến phân loại và nhận diện hình ảnh, sinh viên có thể dễ dàng tiếp cận và hiểu rõ các khái niệm quan trọng như:

- Tiền xử lý dữ liệu ảnh.

- Huấn luyện và đánh giá mô hình học sâu (CNN).
- Overfitting, augmentation và tối ưu hóa mô hình.
- Giao diện ứng dụng AI vào các hệ thống thực tế.

Việc sử dụng công cụ nhận diện hình ảnh trong giảng dạy giúp sinh viên tiếp cận AI một cách sinh động, trực quan và gần gũi hơn. Thay vì chỉ học lý thuyết, sinh viên được quan sát quá trình xử lý hình ảnh thực tế, từ đó hình thành tư duy ứng dụng và sáng tạo. Đây cũng là bước khởi đầu quan trọng để đào tạo nguồn nhân lực AI chất lượng cao trong tương lai.

1.4. Các mô hình AI tiên tiến trong nhận diện ảnh

1.4.1. Giới thiệu các mô hình

a. CNN (Convolutional Neural Networks)

CNN là một trong những kiến trúc mạng học sâu phổ biến và được ứng dụng rộng rãi nhất trong xử lý ảnh. CNN hoạt động bằng cách sử dụng các lớp tích chập (convolutional layers) để trích xuất đặc trưng từ hình ảnh đầu vào. Cấu trúc của CNN giúp giảm số lượng tham số cần học, đồng thời tăng khả năng nhận diện các đặc trưng không gian trong ảnh như cạnh, góc, hình khối...

Ứng dụng: Nhận diện chữ viết tay, phân loại hình ảnh, phát hiện bệnh trong ảnh y tế, kiểm tra chất lượng sản phẩm.

b. YOLO (You Only Look Once)

YOLO là một mô hình phát hiện đối tượng (object detection) theo thời gian thực nổi bật vì khả năng dự đoán vị trí và phân loại các đối tượng trong ảnh chỉ qua một lần "nhìn" toàn bộ ảnh. Các phiên bản mới như YOLOv4, YOLOv5, YOLOv8 mang lại cải tiến đáng kể về tốc độ và độ chính xác. YOLO rất phù hợp cho các ứng dụng yêu cầu thời gian phản hồi nhanh như camera giám sát, xe tự hành, robot.

Ứng dụng: Giám sát giao thông, nhận diện khuôn mặt, phát hiện hành vi, kiểm tra công nghiệp tự động.

c. ViT (Vision Transformer)

ViT là một mô hình mới nổi trong thị giác máy tính, sử dụng kiến trúc transformer – vốn nổi bật trong xử lý ngôn ngữ tự nhiên – để xử lý ảnh. Thay vì sử dụng tích chập như CNN, ViT chia hình ảnh thành các “patch” và xử lý chúng như chuỗi đầu vào. Mô hình này đã đạt nhiều thành tích vượt trội trong các benchmark về phân loại ảnh và có tiềm năng mở rộng mạnh mẽ khi dữ liệu lớn và tài nguyên tính toán dồi dào.

Ứng dụng: Phân loại ảnh, phân đoạn ảnh (image segmentation), và kết hợp nhiều mô thức dữ liệu (multimodal AI).

1.4.2. So sánh hiệu năng, tốc độ, độ chính xác

Tiêu chí	CNN	YOLO	ViT
Tốc độ xử lý	Nhanh (phân loại ảnh đơn)	Rất nhanh (real-time object detection)	Trung bình đến chậm (phụ thuộc phần cứng)
Độ chính xác	Cao (trong phân loại)	Cao (cân bằng tốc độ và độ chính xác)	Rất cao (khi được huấn luyện với tập dữ liệu lớn)
Khả năng mở rộng	Tốt (trong phạm vi bài toán đơn giản)	Rất tốt (trong các ứng dụng thực tế đa đối tượng)	Xuất sắc (phù hợp với bài toán phức tạp và mô hình đa nhiệm)
Yêu cầu phần cứng	Trung bình	Thấp đến trung bình	Cao (đặc biệt với mô hình ViT lớn)
Ứng dụng phù hợp	Phân loại ảnh, nhận diện đơn giản	Nhận diện đối tượng trong thời gian thực	Phân tích hình ảnh lớn, học sâu kết hợp mô hình hóa phức tạp

❖ Nhận xét:

- CNN phù hợp cho việc giảng dạy khái niệm cơ bản về thị giác máy tính và xử lý ảnh.

- **YOLO** là lựa chọn lý tưởng cho các bài giảng minh họa khả năng phát hiện đối tượng trong thực tế, với phản hồi nhanh và trực quan.
- **ViT** là công cụ mạnh mẽ trong các khóa học nâng cao, giúp sinh viên tiếp cận xu hướng AI hiện đại và nghiên cứu chuyên sâu.

CHƯƠNG 2: Xây dựng hệ thống nhận diện hình ảnh hỗ trợ giảng dạy

2.1. Quy trình xây dựng hệ thống nhận diện ảnh hỗ trợ giảng dạy

2.1.1. Chuẩn bị dữ liệu hình ảnh

Việc chuẩn bị dữ liệu là bước nền tảng trong quá trình xây dựng hệ thống AI. Dữ liệu hình ảnh cần phản ánh đúng các đối tượng mà mô hình sẽ nhận diện, đồng thời đảm bảo độ đa dạng và tính đại diện. Có thể sử dụng hai cách tiếp cận chính:

- **Thu thập từ nguồn có sẵn:** Sử dụng các kho dữ liệu công khai như ImageNet, COCO, Open Images Dataset. Các nguồn này thường đi kèm nhãn dữ liệu sẵn, tiết kiệm thời gian.
- **Tự xây dựng tập dữ liệu:** Với các nội dung giảng dạy đặc thù, có thể tự chụp ảnh, thu thập từ mạng Internet và sắp xếp vào các nhóm như: hình học cơ bản, biểu cảm gương mặt, con vật, đồ vật học đường...

Trong bước này, cũng cần thực hiện kiểm tra chất lượng ảnh, loại bỏ những ảnh mờ, trùng lặp, và thay đổi kích thước hình ảnh cho phù hợp với yêu cầu mô hình (thường là 224×224 hoặc 416×416 pixel).

Ta có thể sử dụng các nguồn mở với mục đích học tập:

1. ImageNet: ImageNet là một trong những bộ dữ liệu lớn nhất và phổ biến nhất trong lĩnh vực thị giác máy tính, bao gồm hơn 1,2 triệu hình ảnh thuộc 1.000 lớp đối tượng. Ứng dụng: Thường được sử dụng trong các bài toán phân loại hình ảnh và là nền tảng cho nhiều mô hình học sâu như AlexNet, ResNet. Truy cập và tải về: Để tải bộ dữ liệu ImageNet, bạn cần đăng ký tài khoản và đăng nhập tại trang chính thức: ImageNet Download . image-net.org

2. COCO (Common Objects in Context): COCO là bộ dữ liệu lớn chứa hơn 330.000 hình ảnh với chú thích chi tiết, bao gồm 80 danh mục đối tượng, hỗ trợ các tác vụ như phát hiện đối tượng, phân đoạn ảnh và chú thích hình ảnh. docs.ultralytics.com +1 huggingface.co +1.

Ứng dụng: Phù hợp cho các bài toán phát hiện và phân đoạn đối tượng trong ngữ cảnh thực tế.

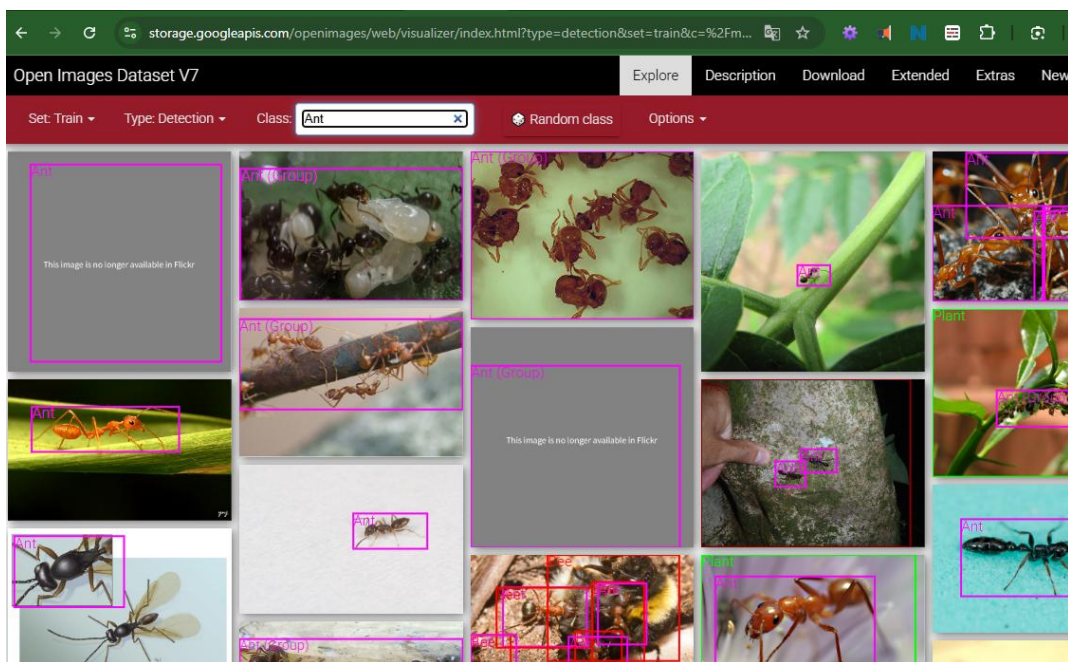
Truy cập và tải về: Có thể tải bộ dữ liệu COCO tại trang chính thức: COCO Dataset . docs.ultralytics.com+6 datasets.activeloop.ai +6 cocodataset.org +6

3. Open Images Dataset: Open Images là bộ dữ liệu hình ảnh lớn do Google phát triển, chứa hơn 9 triệu hình ảnh với hơn 6.000 danh mục, bao gồm các chú thích như nhãn lớp, hộp giới hạn và phân đoạn đối tượng. github.com +2 research.google +2 kaggle.com +2.

Ứng dụng: Thích hợp cho các bài toán nhận diện và phân loại đối tượng trong hình ảnh.

Truy cập và tải về: Có thể truy cập và tải bộ dữ liệu Open Images tại trang chính thức: Open Images Dataset . storage.googleapis.com +1 github.com +1

Trong báo cáo này dùng “Open Images Dataset”



Hình 2.1: “Open Images Dataset” nhận diện kiến

2.1.2. Hướng dẫn huấn luyện YOLO với bộ dữ liệu Kiến từ Open Images Dataset

- ❖ Chuẩn bị môi trường
 - Cài đặt Python 3.8+ và các thư viện cần thiết.
 - Tốt nhất dùng máy có GPU (NVIDIA, có CUDA).
 - Cài đặt PyTorch (theo hướng dẫn pytorch.org).

```
C:\Users\daoli>pip --version
pip 22.3.1 from C:\Users\daoli\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)

C:\Users\daoli>pip install torch torchvision torchaudio
Collecting torch
  Downloading torch-2.7.1-cp310-cp310-win_amd64.whl (216.1 MB)
    158.1/216.1 MB 4.1 MB/s eta 0:00:15

C:\WINDOWS\system32\cmd.exe
Downloading fsspec-2025.5.1-py3-none-any.whl (199 kB)
    199.1/199.1 MB 4.1 MB/s eta 0:00:00
Collecting filelock
  Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
Collecting jinja2
  Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
    134.9/134.9 MB 7.8 MB/s eta 0:00:00
Collecting networkx
  Downloading networkx-3.4.2-py3-none-any.whl (1.7 MB)
    1.7/1.7 MB 4.1 MB/s eta 0:00:00
Collecting numpy
  Downloading numpy-2.2.6-cp310-cp310-win_amd64.whl (12.9 MB)
    12.9/12.9 MB 3.2 MB/s eta 0:00:00
Collecting pillow!=8.3.*,>=5.3.0
  Downloading pillow-11.2.1-cp310-cp310-win_amd64.whl (2.7 MB)
    2.7/2.7 MB 3.9 MB/s eta 0:00:00
Collecting mpmath<1.4,>=1.1.0
  Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
    536.2/536.2 kB 4.8 MB/s eta 0:00:00
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-3.0.2-cp310-cp310-win_amd64.whl (15 kB)
Installing collected packages: mpmath, typing-extensions, sympy, pillow, numpy, networkx, MarkupSafe, fsspec, filelock,
jinja2, torch, torchvision, torchaudio
Successfully installed MarkupSafe-3.0.2 filelock-3.18.0 fsspec-2025.5.1 jinja2-3.1.6 mpmath-1.3.0 networkx-3.4.2 numpy-2
.2.6 pillow-11.2.1 sympy-1.14.0 torch-2.7.1 torchaudio-2.7.1 torchvision-0.22.1 typing-extensions-4.14.0

[notice] A new release of pip available: 22.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\daoli>python

Command Prompt
Downloading pillow-11.2.1-cp312-cp312-win_amd64.whl (2.7 MB)
    2.7/2.7 MB 56.7 MB/s eta 0:00:00
Downloading sympy-1.14.0-py3-none-any.whl (6.3 MB)
    6.3/6.3 MB 50.3 MB/s eta 0:00:00
Downloading typing_extensions-4.14.0-py3-none-any.whl (43 kB)
    43.8/43.8 kB ? eta 0:00:00
Downloading filelock-3.18.0-py3-none-any.whl (16 kB)
Downloading fsspec-2025.5.1-py3-none-any.whl (199 kB)
    199.1/199.1 kB ? eta 0:00:00
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
    134.9/134.9 kB 8.3 MB/s eta 0:00:00
Downloading networkx-3.5-py3-none-any.whl (2.0 MB)
    2.0/2.0 MB 65.3 MB/s eta 0:00:00
Downloading numpy-2.3.0-cp312-cp312-win_amd64.whl (12.7 MB)
    12.7/12.7 MB 43.7 MB/s eta 0:00:00
Downloading setuptools-80.9.0-py3-none-any.whl (1.2 MB)
    1.2/1.2 MB 38.4 MB/s eta 0:00:00
Downloading MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl (15 kB)
Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
    536.2/536.2 kB ? eta 0:00:00
Installing collected packages: mpmath, typing-extensions, sympy, setuptools, pillow, numpy, networkx, MarkupSafe, fsspec
, filelock, jinja2, torch, torchvision, torchaudio
Successfully installed MarkupSafe-3.0.2 filelock-3.18.0 fsspec-2025.5.1 jinja2-3.1.6 mpmath-1.3.0 networkx-3.5 numpy-2.3
.0 pillow-11.2.1 setuptools-80.9.0 sympy-1.14.0 torch-2.7.1 torchaudio-2.7.1 torchvision-0.22.1 typing-extensions-4.14.0

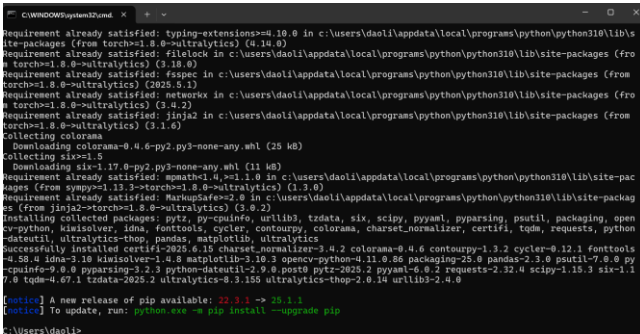
[notice] A new release of pip is available: 23.2.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

F:\2024_2025\BaoCaoHT\NDienanh>
```

Hình 2.2: Cài Pytorch

- Cài YOLOv5:

```
git clone https://github.com/ultralytics/yolov5.git
cd yolov5
pip install -r requirements.txt
```



12

```
Administrator: YOLO5
ts.txt (line 15)) (3.17.0)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\programdata\miniconda3\lib\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (4.12.2)
Requirement already satisfied: sympy>=1.13.3 in c:\programdata\miniconda3\lib\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (1.14.0)
Requirement already satisfied: networkx in c:\programdata\miniconda3\lib\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (2.8.0)
Requirement already satisfied: Jinja2 in c:\programdata\miniconda3\lib\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.1.5)
Requirement already satisfied: fsspec in c:\programdata\miniconda3\lib\site-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (2024.3.1)
Requirement already satisfied: colorama in c:\programdata\miniconda3\lib\site-packages (from tqdm>=4.66.3->-r requirements.txt (line 17)) (0.4.6)
Requirement already satisfied: py-cpuinfo in c:\programdata\miniconda3\lib\site-packages (from ultralytics>=8.2.34->-r requirements.txt (line 18)) (9.0.0)
Requirement already satisfied: ultralytics-thop>=2.0.0 in c:\programdata\miniconda3\lib\site-packages (from ultralytics>=8.2.34->-r requirements.txt (line 18)) (2.0.14)
Requirement already satisfied: pytz>=2020.1 in c:\programdata\miniconda3\lib\site-packages (from pandas>=1.1.4->-r requirements.txt (line 27)) (2025.1)
Requirement already satisfied: zipp>=3.1.0 in c:\programdata\miniconda3\lib\site-packages (from importlib-resources>=3.2.0->matplotlib>=3.3->-r requirements.txt (line 6)) (3.21.0)
Requirement already satisfied: six>=1.5 in c:\programdata\miniconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.3->-r requirements.txt (line 6)) (1.16.0)
Requirement already satisfied: ipmath>=1.4.0-1.11.0 in c:\programdata\miniconda3\lib\site-packages (from sympy>=1.13.3->torch>=1.8.0->-r requirements.txt (line 15)) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\miniconda3\lib\site-packages (from Jinja2->torch>=1.8.0->-r requirements.txt (line 15)) (2.1.5)
K:\bom\mon\Bao cao HT 2024\codepython\YOLO5\yolov5>pause
Press any key to continue . . .
```

Kiểm tra

```
C:\WINDOWS\system32\cmd. x + v
C:\Users\daoli>yolo
Creating new Ultralytics Settings v0.0.6 file
View Ultralytics Settings with 'yolo settings' or at 'C:\Users\daoli\AppData\Roaming\Ultralytics\settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.

Arguments received: ['yolo']. Ultralytics 'yolo' commands use the following syntax:

yolo TASK MODE ARGS

Where TASK (optional) is one of ['classify', 'obb', 'pose', 'segment', 'detect']
MODE (required) is one of ['train', 'predict', 'val', 'export', 'track', 'benchmark']
ARGS (optional) are any number of custom 'arg=value' pairs like 'imgsz=320' that override defaults.
See all ARGS at https://docs.ultralytics.com/usage/cfg or with 'yolo cfg'

1. Train a detection model for 10 epochs with an initial learning_rate of 0.01
yolo train data=coco8.yaml model=yololln.pt epochs=10 lr0=0.01

2. Predict a YouTube video using a pretrained segmentation model at image size 320:
yolo predict model=yololln-seg.pt source=https://youtu.be/LNm0DJKvt4' imgsz=320

3. Val a pretrained detection model at batch-size 1 and image size 640:
yolo val model=yololln.pt data=coco8.yaml batch=1 imgsz=640

4. Export a YOLO11n classification model to ONNX format at image size 224 by 128 (no TASK required)
yolo export model=yololln-cls.pt format=onnx imgsz=224,128

5. Ultralytics solutions usage
yolo solutions count or in ['crop', 'blur', 'workout', 'heatmap', 'isegment', 'visioneye', 'speed', 'queue', 'analytics', 'inference', 'trackzone'] source=path/to/video.mp4

6. Run special commands:
yolo help
yolo checks
yolo version
yolo settings
```

Đảm bảo phiên bản đã đúng:

```
pip show pillow
```

Kết quả nên có dòng:

```
makefile
```

```
Version: 10.2.0
```

```
F:\2024_2025\BaoCaoHT\NDienanh\yolov5>pip show pillow
Name: pillow
Version: 11.2.1
Summary: Python Imaging Library (Fork)
Home-page:
Author:
Author-email: "Jeffrey A. Clark" <aclark@aclark.net>
License:
Location: C:\Users\Asus\AppData\Local\Programs\Python\Python312\Lib\site-packages
Requires:
Required-by: matplotlib, torchvision, ultralytics
F:\2024_2025\BaoCaoHT\NDienanh\yolov5>
```

Nếu cần kiểm tra toàn bộ môi trường AI (các gói như torch, opencv, gradio...) để tránh lỗi tương lai, chỉ cần gõ:

```
pip list
```

```
requests      2.32.4
scipy         1.15.3
seaborn       0.13.2
setuptools    80.9.0
six           1.17.0
smmap         5.0.2
sympy         1.14.0
thop          0.1.1.post2209072238
torch         2.7.1
torchaudio    2.7.1
torchvision   0.22.1
tqdm          4.67.1
typing_extensions 4.14.0
tzdata        2025.2
ultralytics   8.3.153
ultralytics-thop 2.0.14
urllib3       2.4.0

[notice] A new release of pip is available: 23.2.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

- Cài YOLOv8:

```
pip install ultralytics
```

- Thu thập dữ liệu ảnh kiến từ Open Images Dataset

Dùng script oidv6 (Open Images Downloader):

```
pip install openimages
```

- ***Sử dụng OIDv6 Toolkit để tải ảnh và nhãn***

1. Cài đặt OIDv6 Toolkit

```
git clone
```

```
https://github.com/NanoCode012/OIDv6_ToolKit_Download_Open_Images_Support_Yolo_Format.git
```

```
cd OIDv6_ToolKit_Download_Open_Images_Support_Yolo_Format
```

```
pip install -r requirements.txt
```

2. Tạo thư mục lưu dữ liệu

```
mkdir OID
```

```
mkdir OID/Dataset
```

3. Tải ảnh và nhãn cho lớp "Ant"

```
python main.py downloader --classes Ant --type_csv train --limit 200
```

```
Administrator: Taianhkien
K:\bo mon\Bao cao HT 2024\codepython\YOLO5\OIDv6_ToolKit_Download_Open_Images_Support_Yolo_Format>title Taianhkien
K:\bo mon\Bao cao HT 2024\codepython\YOLO5\OIDv6_ToolKit_Download_Open_Images_Support_Yolo_Format>rem mkdir OID/Dataset
K:\bo mon\Bao cao HT 2024\codepython\YOLO5\OIDv6_ToolKit_Download_Open_Images_Support_Yolo_Format>python main.py downloa
der --classes Ant --type csv train --limit 200
[INFO] | saving dataset configurations at ./OID\Dataset\config.json

@OIDv4 W
Downloader

[INFO] | Downloading Ant.
[ERROR] | Missing the train-annotations-bbox.csv file.
[DOWNLOAD] | Do you want to download the missing file? [Y/n] y
...2%, 23 MB, 3120 KB/s, 7 seconds passed
```

Lệnh trên sẽ tải 200 ảnh thuộc lớp "Ant" cùng với nhãn tương ứng. Có thể điều chỉnh số lượng ảnh bằng cách thay đổi giá trị của `--limit`.

Ví dụ: `openimages_download --classes Ant --type_box all --limit 1000 --download_dir ./ant_dataset`

Hoặc tải thủ công trên trang Open Images bằng filter nhãn "Ant" rồi tải.

4. Chuẩn bị dữ liệu theo định dạng YOLO

YOLO yêu cầu dữ liệu:

Cấu trúc thư mục:

```
dataset/
  images/
    train/
    val/
  labels/
    train/
    val/
```

Mỗi ảnh có file label .txt cùng tên, nội dung mỗi dòng:

`class_id x_center y_center width height`

(tất cả tọa độ chuẩn hóa từ 0 đến 1)

Chuyển dữ liệu Open Images sang định dạng YOLO

Open Images có file nhãn ở dạng .csv (bounding boxes tọa độ tuyệt đối).

5. Tạo file cấu hình dataset cho YOLO

Ví dụ file ant.yaml cho YOLOv5/yolov8:

```
train: dataset/images/train
val: dataset/images/val
nc: 1
names: ['ant']
```

6. Huấn luyện mô hình YOLO

Với YOLOv5

```
python train.py --img 640 --batch 16 --epochs 50 --data ant.yaml --weights yolov5s.pt --cache
```

Với YOLOv8 (ultralytics)

```
yolo task=detect mode=train model=yolov8n.pt data=ant.yaml epochs=50 imgsz=640
```

7. Đánh giá và thử nghiệm mô hình

Kiểm tra kết quả training, các chỉ số mAP, precision, recall.

Thử chạy inference trên ảnh mới:

Ví dụ YOLOv5:

```
python detect.py --weights runs/train/exp/weights/best.pt --img 640 --conf 0.25 --source
path_to_test_images/
```

Ví dụ YOLOv8:

```
yolo task=detect mode=predict model=runs/detect/train/weights/best.pt
source=path_to_test_images/
```

2.1.3. Gán nhãn và xử lý dữ liệu

Sau khi thu thập ảnh, việc **gán nhãn** giúp mô hình học được mối quan hệ giữa hình ảnh và thông tin mô tả. Các công cụ như LabelImg, CVAT hoặc Roboflow có thể được sử dụng để:

- Vẽ khung bao (bounding box) đối với bài toán phát hiện đối tượng (object detection).
- Gán nhãn lớp đối với bài toán phân loại (classification).
- Phân vùng ảnh đối với bài toán phân đoạn (segmentation).

Ngoài ra, cần thực hiện **xử lý dữ liệu** (data preprocessing) như:

- Chuẩn hóa ảnh (normalize pixel values).

- Tăng cường dữ liệu (data augmentation): xoay ảnh, lật ảnh, thay đổi độ sáng nhằm giúp mô hình học tốt hơn và tránh overfitting.

2.1.4. Huấn luyện mô hình

Sau khi có dữ liệu đã xử lý, bước tiếp theo là huấn luyện mô hình. Một số mô hình tiên tiến có thể sử dụng gồm:

- **CNN (Convolutional Neural Network):** Phù hợp với bài toán phân loại hình ảnh đơn giản.
- **YOLO (You Only Look Once):** Phù hợp với phát hiện nhiều đối tượng trong một ảnh, tốc độ nhanh, hiệu quả cao.
- **ViT (Vision Transformer):** Áp dụng cơ chế Attention, phù hợp với bài toán phức tạp và dữ liệu lớn.

Quá trình huấn luyện được thực hiện trên nền tảng Python với các thư viện như TensorFlow, PyTorch hoặc sử dụng môi trường Google Colab để tận dụng GPU miễn phí.

```

C:\WINDOWS\system32\cmd. X + -
val: New cache created: D:\NCKH2025\logo\dataset\labels\val.cache
WARNING: Labels are missing or empty in D:\NCKH2025\logo\dataset\labels\val.cache, training may not work correctly. See h
https://docs.ultralytics.com/datasets for dataset formatting guidance
C:\Users\daoli\AppData\Local\Programs\Python\Python310\lib\site-packages\torch\utils\data\loader.py:665: UserWarning
: 'pin_memory' argument is set as true but no accelerator is found, then device pinned memory won't be used.
warnings.warn(warn_msg)
Plotting labels to runs\detect\train1\labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr' and
'momentum' automatically...
optimizer: Adam(lr=0.002, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(de
cay=0.0)
Image sizes 640 train, 640 val
Using 0 dataloader workers
Logging results to runs\detect\train1
Starting training for 50 epochs...

Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/50      0G       1.993      8.369      1.724      16          640 100% | 6/6 [0:13<00:00, 18.58
Class Images Instances Box(s) 0 0 0 0 0 0
C:\Users\daoli\AppData\Local\Programs\Python\Python310\lib\site-packages\ultralytics\utils\metrics.py:765: RuntimeWarning
g: Mean of empty slice.
  i = smoothen(cfl_curve.mean(0), 0.1).argmax() # max F1 index
C:\Users\daoli\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\core\_methods.py:137: RuntimeWarning: in
valid value encountered in divide
  ret = un.true_divide(
all 16 0 0 0 0 0
WARNING: no labels found in detect set, can not compute metrics without labels
Epoch   GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
2/50      0G       1.759      4.165      1.566      21          640 83% | 5/6 [0:32<00:10, 10.56

```

```

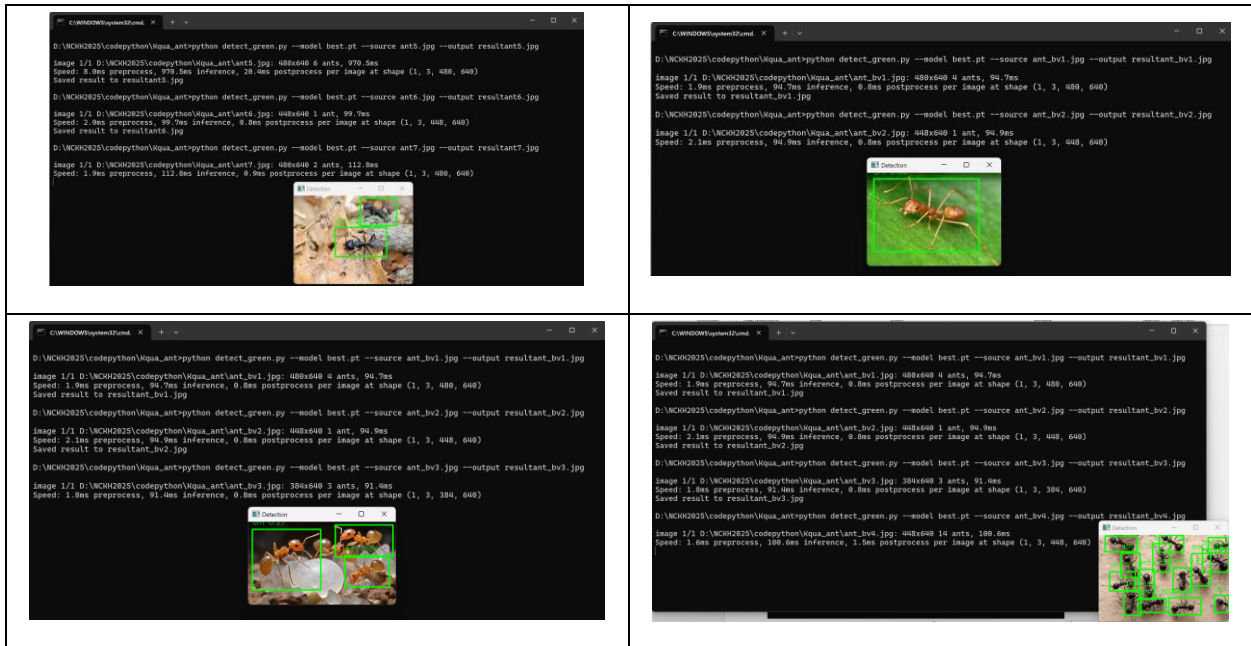
D:\NCKH2025\logo>yolo task=detect mode=train model=yolov8s.pt data=D:\NCKH2025\logo\dataset\data.yaml epochs=50 device=c
pu
Ultralytics 8.3.155 Python-3.10.10 torch-2.7.1+cpu CPU (13th Gen Intel Core(TM) i5-13500H)
engine=trainer: agnostic_nms=False, amp=True, augment=False, auto_augment=randaugment, batch=16, bgr=0.0, box=7.5, cache
=False, cfg=None, classes=None, close_mosaic=10, cls=0.5, conf=None, copy_paste=0.0, copy_paste_mode=flip, cos_lr=False,
cutmix=0.9, data=D:\NCKH2025\logo\dataset\data.yaml, degrees=0.0, deterministic=True, device=cpu, dfl=1.5, dnn=False, d
ropout=0.0, dynamic=False, embed=None, epochs=50, erasing=0.4, exist_ok=False, flip_lr=0.5, flipud=0.0, format=torchscrip
t, fraction=1.0, freeze=None, half=False, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, imgsz=640, int8=False, iou=0.7, keras=False
, kobj=1.0, line_width=None, lr=0.01, lrf=0.01, mask_ratio=4, max_det=300, mixup=0.0, mode=train, model=yolov8s.pt, mom
entum=0.937, mosaic=1.0, multi_scale=False, name=train11, nbs=64, nms=False, opset=None, optimize=False, optimizer=auto,
overlap_mask=True, patience=100, perspective=0.0, plots=True, pose=12.0, pretrained=True, profiles=False, project=None,
rect=False, resume=False, retina_masks=False, save=True, save_conf=False, save_crop=False, save_dir=runs\detect\train11,
save_frames=False, save_json=False, save_period=1, save_txt=False, scale=0.5, seed=0, shear=0.0, show=False, show boxe
s=True, show_conf=True, show_labels=True, simplify=True, single_cls=False, source=None, split=val, stream_buffer=False,
task=detect, time=None, tracker=botssort.yaml, translate=0.1, val=True, verbose=True, vid_stride=1, visualize=False, warn
up_bias_lr=0.1, warmup_epochs=3.0, warmup_momentum=0.0, weight_decay=0.0005, workers=8, workspace=None
Matplotlib is using the font cache; this may take a moment.
Downloading https://ultralytics.com/assets/Arial.ttf to 'C:\Users\daoli\AppData\Roaming\Ultralytics\Arial.ttf'...
100%|██████████| 755k/755k [00:01<00:00, 589kB/s]
Overriding model.yaml nc=80 with nc=1

```

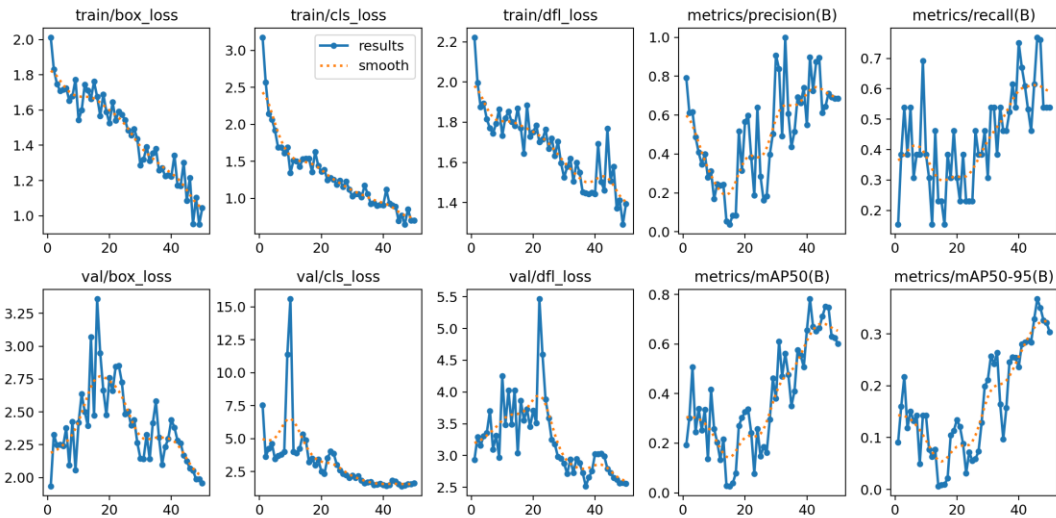
Results saved to runs\detect\train11
 Learn more at <https://docs.ultralytics.com/modes/train>
 D:\NCKH2025\logo>pause
 Press any key to continue . . . |

Hình 2.3. Quá trình huấn luyện

2.1.5. Kết quả nhận diện hình ảnh:



Mô hình sau khi huấn luyện được lưu thành file (ví dụ `model.pt` hoặc `model.h5`) và đánh giá dựa trên độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (precision) và thời gian suy luận.



Hình 2.4. Đánh giá độ chính xác

- ❖ Mô hình học được khá tốt. Các chỉ số quan trọng đều đang cải thiện theo thời gian, đặc biệt là:
 - **Precision (độ chính xác):** tăng dần đều, đạt tới ~0.8
 - **Recall (độ bao phủ):** tăng tới ~0.7

- **mAP50 và mAP50-95:** đều tăng và đạt mức khá (mAP50 ~ 0.75 , mAP50-95 ~ 0.35)

⇒ Đây là một mô hình tốt, đặc biệt nếu bài toán không quá phức tạp.

❖ Huấn luyện (train)

train/box_loss : Giảm đều từ ~ 2.0 xuống ~ 1.0 ⇒ mô hình học vị trí bbox tốt

train/cls_loss : Giảm rõ rệt từ 3.0 xuống < 1.0 ⇒ mô hình học phân loại tốt

train/df_l_loss: Giảm từ ~ 2.2 xuống ~ 1.3 ⇒ học được phân phối tọa độ tốt

⇒ Các loss đều giảm ⇒ mô hình học ổn định

❖ Validation (val)

val/box_loss: Dao động, nhưng xu hướng giảm từ ~ 3.2 về ~ 2.0

val/cls_loss: Có đỉnh cao bất thường (~ 15), sau giảm ổn định về ~ 2.5

val/df_l_loss: Giảm ổn định từ ~ 5.5 về ~ 2.5

⇒ **Chú ý:** Giai đoạn đầu có dao động lớn → có thể do batch size nhỏ hoặc dữ liệu chưa đều.

❖ Hiệu suất mô hình (metrics)

metrics/precision(B): Tăng từ ~ 0.3 lên ~ 0.8 ⇒ mô hình ít báo sai

metrics/recall(B): Tăng từ ~ 0.2 lên ~ 0.7 ⇒ mô hình tìm được nhiều đối tượng

metrics/mAP50(B): Tăng tốt đến ~ 0.75 ⇒ khả năng phát hiện rất tốt

metrics/mAP50-95(B): Tăng đều đến ~ 0.35 ⇒ chất lượng bbox chấp nhận được

⇒ Những chỉ số này cho thấy mô hình **phát hiện ổn định và chính xác**.

❖ Kết luận đánh giá

Tiêu chí	Mức độ	Đánh giá
Loss train	Giảm ổn	Tốt
Loss validation	Giảm nhưng dao động	Trung bình

Tiêu chí	Mức độ	Đánh giá
Precision	~0.8	Rất tốt
Recall	~0.7	Tốt
mAP@0.5	~0.75	Cao
mAP@0.5:0.95	~0.35	Trung bình

=> **Mô hình tốt để sử dụng thực tế.** Nếu muốn cải thiện thêm:

- Dùng mô hình lớn hơn (yolov8m.pt, yolov8l.pt)
- Làm sạch và tăng số lượng dữ liệu huấn luyện
- Tăng epochs đến 100 hoặc dùng early stopping

2.2. Triển khai mô hình trên giao diện web/app đơn giản

Sau khi huấn luyện thành công, mô hình cần được triển khai để sử dụng dễ dàng trong giảng dạy.

Có thể xây dựng giao diện đơn giản bằng:

- **Flask/Django (Python):** Đưa mô hình lên một ứng dụng web có giao diện tải ảnh, nhận kết quả.
- **Streamlit/Gradio:** Các thư viện Python giúp dựng web demo AI nhanh chóng mà không cần kỹ năng lập trình giao diện.
- **App Android đơn giản:** Có thể sử dụng TensorFlow Lite hoặc ML Kit để tích hợp mô hình vào ứng dụng di động.

Việc triển khai giúp giảng viên dễ dàng tương tác với mô hình trong lớp học, tạo thành công cụ trực quan cho bài giảng.

2.3. Trình diễn công cụ nhận diện hình ảnh

❖ Demo trực tiếp mô hình nhận diện các đối tượng

Trong phần trình diễn, hệ thống sẽ nhận diện các đối tượng quen thuộc như:

- **Hình khối học toán:** nhận dạng hình vuông, tam giác, tròn.
- **Biểu cảm gương mặt:** vui, buồn, ngạc nhiên, tức giận.
- **Động vật:** mèo, chó, cá, chim.
- **Vật dụng học đường:** bút, sách, balo, thước.

Việc sử dụng hình ảnh thực tế hoặc webcam ngay tại lớp học giúp sinh viên thấy rõ quá trình suy luận của AI và quan sát phản hồi của mô hình theo thời gian thực.

❖ *Tích hợp kết quả vào bài giảng*

Các kết quả đầu ra của mô hình có thể tích hợp vào nội dung bài giảng theo nhiều cách:

- **Ví dụ minh họa cho thuật toán CNN, YOLO, ViT.**
- **Biểu diễn dữ liệu:** minh họa ma trận trọng số, lớp convolution, biểu đồ độ chính xác theo từng epoch.
- **Kiểm thử mô hình:** sinh viên thử thay ảnh và đánh giá độ chính xác mô hình, từ đó hiểu rõ yếu tố ảnh hưởng đến AI.

Giảng viên có thể thiết kế bài tập thực hành yêu cầu sinh viên tải ảnh, phân tích đầu ra, và đề xuất cải tiến mô hình, qua đó phát triển tư duy phản biện.

❖ *Phân tích ưu điểm và hạn chế của hệ thống*

Ưu điểm:

- Giao diện đơn giản, dễ sử dụng trong môi trường giảng dạy.
- Mô hình hoạt động theo thời gian thực, tạo hứng thú học tập.
- Dễ dàng tùy biến dữ liệu, giúp giảng viên xây dựng nội dung phù hợp.

Hạn chế:

- Độ chính xác mô hình phụ thuộc vào chất lượng và số lượng dữ liệu huấn luyện.
- Các mô hình lớn như ViT cần cấu hình máy cao hoặc dịch vụ tính toán đám mây.
- Cần thời gian chuẩn bị dữ liệu và kiến thức nền về AI để giảng viên tự vận hành.

KẾT LUẬN

Trong bối cảnh giáo dục ngày càng gắn liền với công nghệ, việc ứng dụng trí tuệ nhân tạo, đặc biệt là nhận diện hình ảnh vào giảng dạy đã và đang mở ra những hướng đi mới, giúp tăng tính trực quan, sinh động và hiệu quả cho bài học.

Nội dung của đề tài đã thực hiện được các mục tiêu chính như:

- Trình bày tổng quan về trí tuệ nhân tạo và vai trò của nhận diện hình ảnh trong giáo dục hiện đại.
- Tìm hiểu và so sánh một số mô hình AI tiên tiến như CNN, YOLO, và ViT.
- Xây dựng quy trình đầy đủ để thiết kế một hệ thống nhận diện hình ảnh gồm: chuẩn bị dữ liệu, gán nhãn, huấn luyện, triển khai và tích hợp vào giảng dạy.
- Trình diễn mô hình AI qua các ví dụ thực tế, tạo điều kiện để sinh viên quan sát, thử nghiệm và phát triển tư duy ứng dụng công nghệ.

Hệ thống được xây dựng có khả năng nhận diện tốt các đối tượng phổ biến như hình khối, vật dụng, biểu cảm gương mặt... và dễ dàng tích hợp vào nội dung bài giảng môn học Trí tuệ nhân tạo hoặc Thị giác máy tính. Qua đó, đề tài đã góp phần tạo nên một công cụ hỗ trợ giảng dạy hiệu quả, có khả năng mở rộng ứng dụng cho các bối cảnh học tập khác nhau.

TÀI LIỆU THAM KHẢO

- [1]. Trường Đại học Công nghiệp Hà Nội (HaUI), “Ứng dụng xử lý ảnh và công nghệ trí tuệ nhân tạo”
- [2]. Andre Esteva¹, Brett Kuper¹, Roberto A. Novoa, “Dermatologist-level classification of skin cancer with deep neural networks,” [Online]. Available: <https://www.nature.com/articles/nature21056.epdf>
- [3]. Alexis Cook, SudKul Sudhanshu Kulshrestha, Abhi Ojha, Hugues Vinzant “dermatologist-ai”. [Online]. Available: <https://github.com/udacity/dermatologist-ai>.
- [4] Tahiya Chowdhury (2025), “Computational Thinking with Computer Vision: Developing AI Competency in an Introductory Computer Science Course”
- [5] Yu Zhang (2024), Analysis of the Application of Artificial Intelligence in Information Courses in Primary and Secondary Schools”