TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

MÔN : ĐỒ ÁN TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:

ỨNG DỤNG CONVNEXT VÀ VISION TRANSFORMER TRONG PHÂN LOẠI HÌNH ẢNH DO AI TẠO VÀ HÌNH ẢNH DO CON NGƯỜI TẠO RA

Giảng viên hướng dẫn: Ths. Nguyễn Đắc Phương Thảo

Nhóm sinh viên thực hiện: Nhóm 3_64TTNT1

Họ tên sinh viên	Mã sinh viên
Thái Văn Sáng	2251262634
Vũ Thị Như Quỳnh	2251262633
Nguyễn Thị Quỳnh Anh	2251262657
Đồng Anh Quân	2251262626

MỤC LỤC

Chương I. Giới thiệu bài toán

1. Mô tả bài toán

Sự phát triển nhanh chóng của các mô hình trí tuệ nhân tạo tạo sinh (generative AI) trong lĩnh vực thị giác máy tính đã mang đến những thay đổi sâu sắc trong việc tạo ra nội dung hình ảnh. Các mô hình như: DALL-E 2 **[1]**, Midjourney, và Stable Diffusion **[2]** có khả năng sinh ảnh từ văn bản với độ chân thực cao, thường khó phân biệt bằng mắt thường với các ảnh do con người chụp hoặc vẽ. Khả năng tạo ảnh chất lượng cao một cách dễ dàng đã mở ra nhiều ứng dụng sáng tạo trong nghệ thuật, truyền thông và thiết kế, nhưng đồng thời cũng đặt ra những rủi ro lớn liên quan đến thông tin sai lệch, giả mạo hình ảnh và thao túng dư luận **[3]**. Trong bối cảnh này, việc phát triển các phương pháp phân biệt hình ảnh do AI tạo với hình ảnh do con người tạo ra trở thành một yêu cầu cấp thiết. Các nghiên cứu gần đây đã chỉ ra rằng hình ảnh do AI tạo thường mang một số đặc điểm riêng biệt như: hiện tượng artefacts, bất thường trong kết cấu hoặc bố cục không hợp lý mà mô hình học sâu có thể khai thác **[4]**. Tuy nhiên, khi chất lượng hình ảnh từ các mô hình như Stable Diffusion 2.1 hay Midjourney ngày càng tiệm cận ảnh thật, việc phân biệt trở nên khó khăn hơn, đòi hỏi các kiến trúc mô hình mạnh mẽ và có khả năng trích xuất đặc trưng sâu.Trong nghiên cứu này, chúng tôi xây dựng một hệ thống phân loại hình ảnh dựa trên hai kiến trúc học sâu hiện đại: Vision Transformer (ViT) **[5]** và ConvNeXt **[6]**. Hai mô hình đại diện cho hai hướng tiếp cận khác nhau trong học sâu, một bên dựa trên cơ chế attention và một bên dựa trên cơ chế tích chập. Chúng tôi sử dụng một tập dữ liệu có sẵn bao gồm các hình ảnh do AI tạo và hình ảnh do con người tạo, đồng thời huấn luyện và đánh giá các mô hình trên tập dữ liệu này.

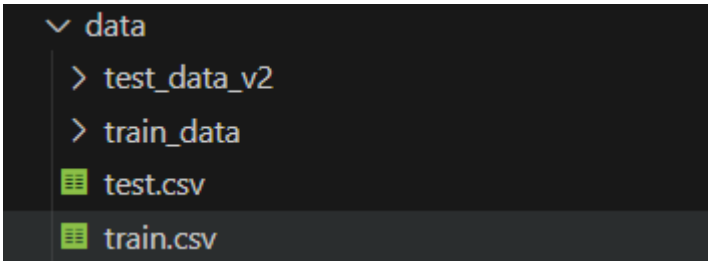
2. Giới thiệu bộ dữ liệu

Trong dự án này, nhóm sử dụng bộ dữ liệu mang tên "AI vs Human Generated Dataset", được cung cấp trên nền tảng Kaggle. Bộ dữ liệu có mục tiêu rõ ràng: giúp huấn luyện mô hình có khả năng phân biệt giữa hình ảnh do con người chụp và hình ảnh được tạo ra bởi trí tuệ nhân tạo (AI).

Chương II. Chuẩn bị dữ liệu

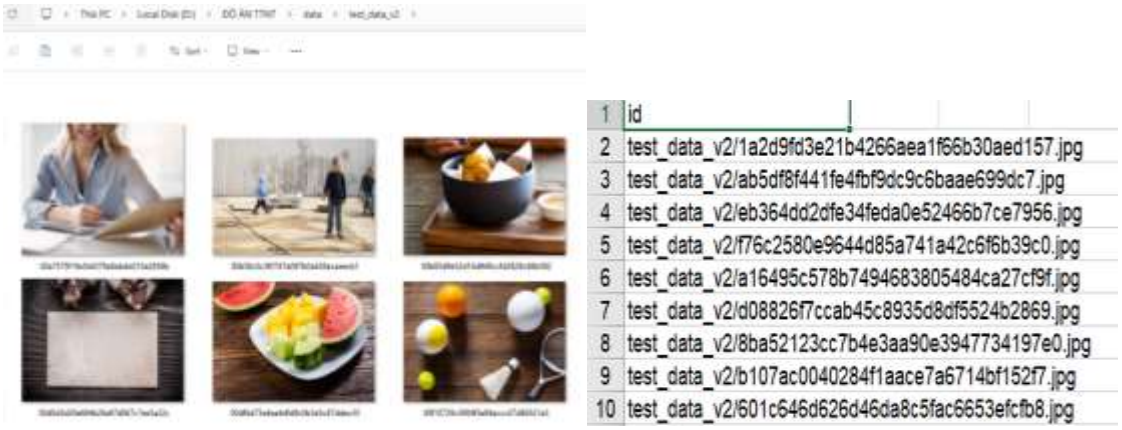
1. Khám phá dữ liệu

Quá trình xử lý dữ liệu bắt đầu bằng việc phân tích cấu trúc của bộ dữ liệu được cung cấp. Cụ thể, bộ dữ liệu bao gồm 2 thành phần chính: tập huấn luyện (training set) và tập kiểm tra (test set).



Hình 2.1.1: Các thành phần trong bộ dữ liệu

- **Thư mục test_data_v2** là nơi lưu trữ hình ảnh kiểm tra, dùng để đánh giá khả năng tổng quát của mô hình. Thư mục này đi kèm với tệp test.csv, chứa:

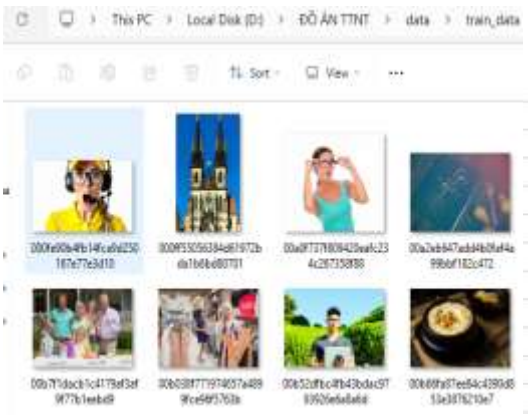


Hình 2.1.2: Dữ liệu trong thư mục test_data_v2

+ Cột id: tên các tệp ảnh nằm trong thư mục test_data_V2/

+ Tổng cộng có **5.540 ảnh**, không có nhãn (label) – điều này phù hợp với mục tiêu kiểm tra mô hình sau khi huấn luyện.

- **Thư mục train_data** chứa toàn bộ hình ảnh dùng cho việc huấn luyện mô hình. Thông tin mô tả của những bức ảnh này được lưu trong tệp train.csv, gồm:



	file_name	label
0	train_data/a6dcb93f596a43249135678dfcfc17ea.jpg	1
1	train_data/041be3153810433ab146bc97d5af505c.jpg	0
2	train_data/615df26ce9494e5db2f70e57ce7a3a4f.jpg	1
3	train_data/8542fe161d9147be8e835e50c0de39cd.jpg	0
4	train_data/5d81fa12bc3b4cea8c94a6700a477cf2.jpg	1
5	train_data/25ea852f30594bc5915eb929682af429.jpg	0
6	train_data/e67085fb6d814cbabe08f978c738f3f7.jpg	1
7	train_data/041c36d9269146cdb88e7526e3b91651.jpg	0
8	train_data/4aea3b876247467c8d3713d4920148ab.jpg	1
9	train_data/09708379751e44d0bc908d8652d0db3e.jpg	0
10	train_data/774aeb00dbf44520bf3be78bb600bda9.jpg	1

Hình 2.1.3: Dữ liệu trong thư mục train_data

+ Tổng cộng có **79.949 ảnh**

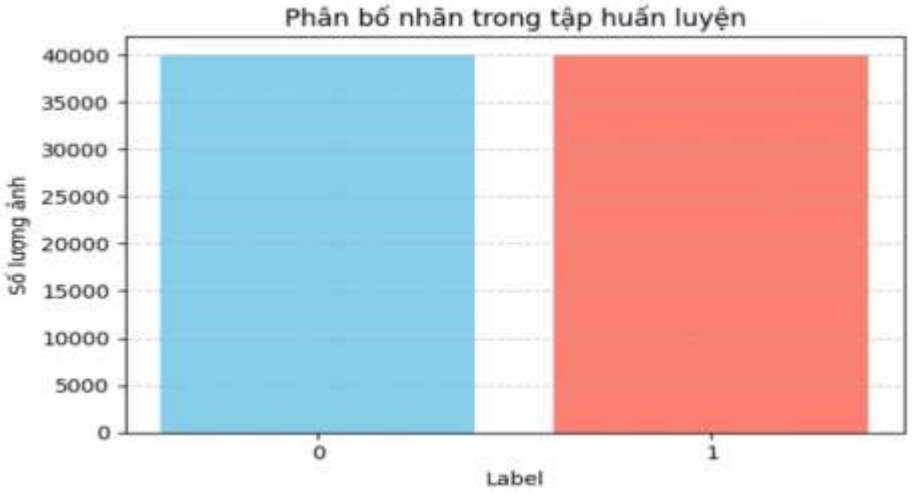
+ Cột file_name: đường dẫn ảnh trong thư mục train_data/

+ Cột label: giá trị nhị phân thể hiện bản chất của ảnh:

0: Ảnh thật – do con người chụp

1: Ảnh do AI tạo ra

2. Trực quan hóa dữ liệu



Hình 2.2.1: Biểu đồ phân bố nhãn trong tập huấn luyện

- Biểu đồ trên thể hiện số lượng ảnh tương ứng với từng nhãn (label) trong tập huấn luyện. Tập dữ liệu được chia thành hai nhãn: Label 0 và Label 1, mỗi nhãn có khoảng 40.000 ảnh.

=> Sự phân bố này cho thấy dữ liệu được cân bằng tốt giữa hai lớpxử lý dữ liệu và tăng cường dữ liệu.

Hình 2.2.2: Dữ liệu hình ảnh do Người tạo (Sample 0) và do AI tạo (Sample 1)

3. Tiền xử lý dữ liệu và tăng cường dữ liệu

- Tiền xử lý dữ liệu:

+ Xử lý kích thước ảnh: đưa tất cả ảnh về kích thước 224x224

+ Chuẩn hóa ảnh trên từng kênh màu RGB

- Tăng cường dữ liệu:

+ Cắt ảnh: một vùng trong ảnh gốc, sau đó **resize** vùng đó thành kích thước 224x224.

+ Lật ảnh: Lật ảnh theo chiều ngang hoặc dọc để tạo ra các biến thể mới.

Chương III. Tổng quan về mô hình

1. ConvNext

1.1 Lý thuyết ConvNext:

ConvNext là một kiến trúc CNN hiện đại được thiết kế để cạnh tranh với các mô hình Transformer trong thị giác máy tính. ConvNeXt được xây dựng bằng cách hiện đại hóa ResNet, bao gồm các cải tiến như sử dụng các kernel tích chập lớn hơn, áp dụng chuẩn hóa Layer Normalization và sử dụng các kỹ thuật như stochastic depth và activation GELU. Kết quả là ConvNext đạt được hiệu suất cao trên các tác vụ như phân loại ảnh ImageNet, phát hiện đối tượng COCO và phân đoạn ngữ nghĩa ADE20K[5].

1.2 Cơ chế hoạt động ConvNext

ConvNext cải tiến như sử dụng các kernel tích chập lớn hơn, áp dụng chuẩn hóa Layer Normalization và sử dụng các kỹ thuật như stochastic depth và activation GELU ConvNeXt được tổ chức thành 4 stage, tương tự ResNet:

+ Stem: Conv 4×4, stride = 4 để giảm nhanh kích thước ảnh.
+ Stage 1 → Stage 4: Mỗi stage gồm nhiều ConvNeXt Block, xen giữa là các downsampling layer để giảm kích thước không gian và tăng số chiều kênh.
+ Cuối cùng: Global Average Pooling → Linear Classifier.

1.3 Ưu điểm của ConvNeXt

+ Hiệu suất cao: ConvNeXt có thể đạt top-1 accuracy > 85% trên ImageNet.
+ Khả năng tổng quát tốt: Nhờ thiết kế hiện đại và mạnh mẽ.
+ Tương thích tốt với GPU hiện có: Nhờ giữ nguyên bản chất CNN, ConvNeXt dễ triển khai hơn ViT trên phần cứng hiện nay.
+ Hiệu quả tính toán cao hơn Transformer trong nhiều tác vụ nếu tài nguyên hạn chế.

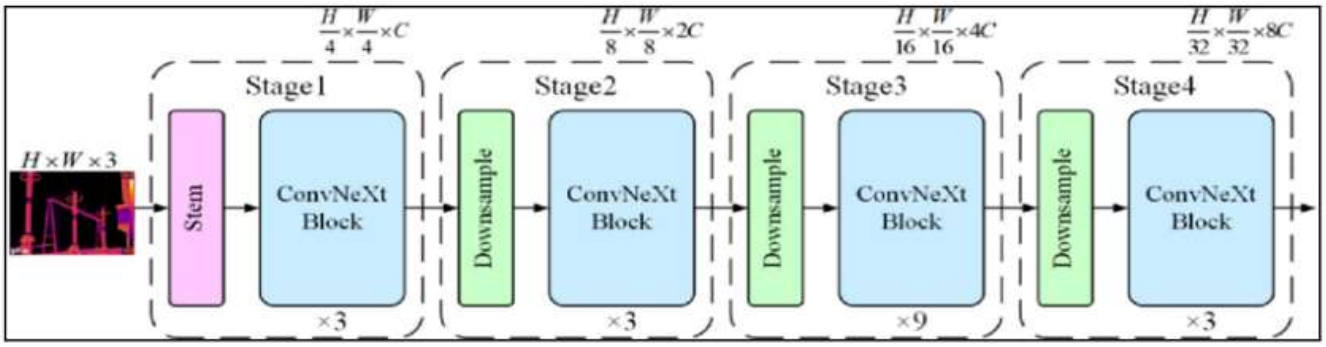
1.4 Hạn chế của ConvNeXt

+ Không khai thác tốt quan hệ toàn cục như ViT.
+ Phụ thuộc vào kiến trúc sâu, cần nhiều tài nguyên tính toán.
+ Thiếu linh hoạt trong các tác vụ đa nhiệm hoặc đa mô thức.
+ Không có cơ chế positional encoding rõ ràng.
+ Kém hiệu quả khi áp dụng cho ảnh bị biến dạng hoặc không gian phi tuyến.

1.5 Ứng dụng

ConvNeXt có thể áp dụng hiệu quả trong nhiều nhiệm vụ thị giác máy tính:

+ Phân loại hình ảnh
+ Nhận dạng đối tượng
+ Phân đoạn ảnh (segmentation)
+ Phát hiện ảnh giả mạo (deepfake detection)
+ Phân biệt hình ảnh do con người chụp và ảnh do AI tạo ra



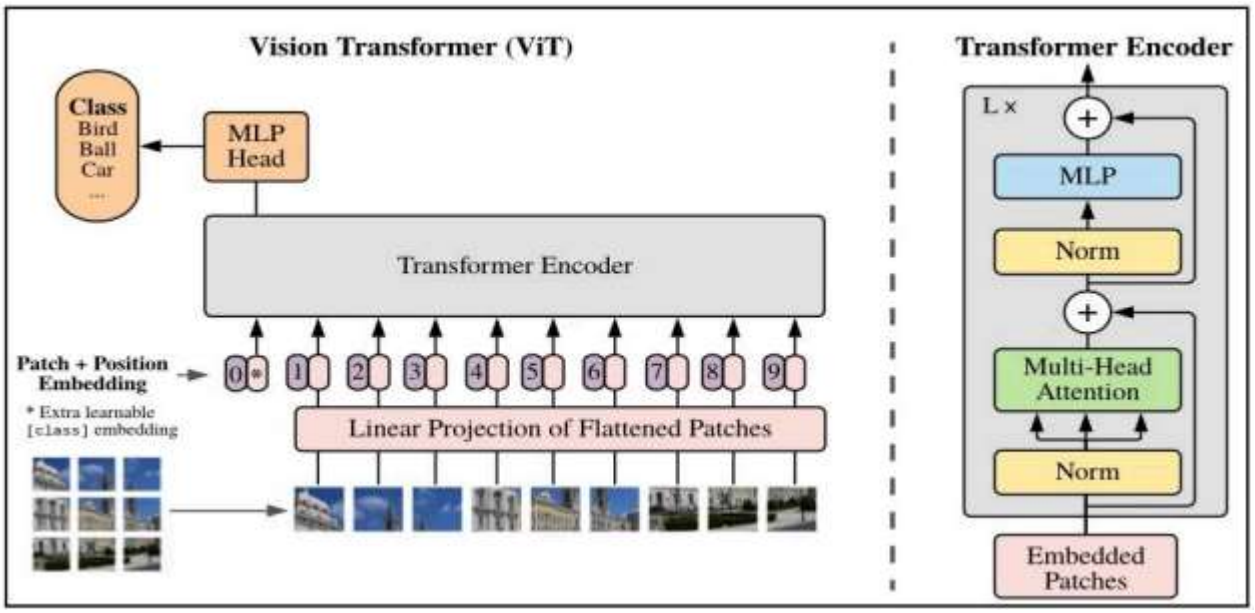
Hình 3.1: Tổng quan về kiến trúc ConvNeXt **[9]**

2. Vision Transformer

2.1 Lý thuyết ViT

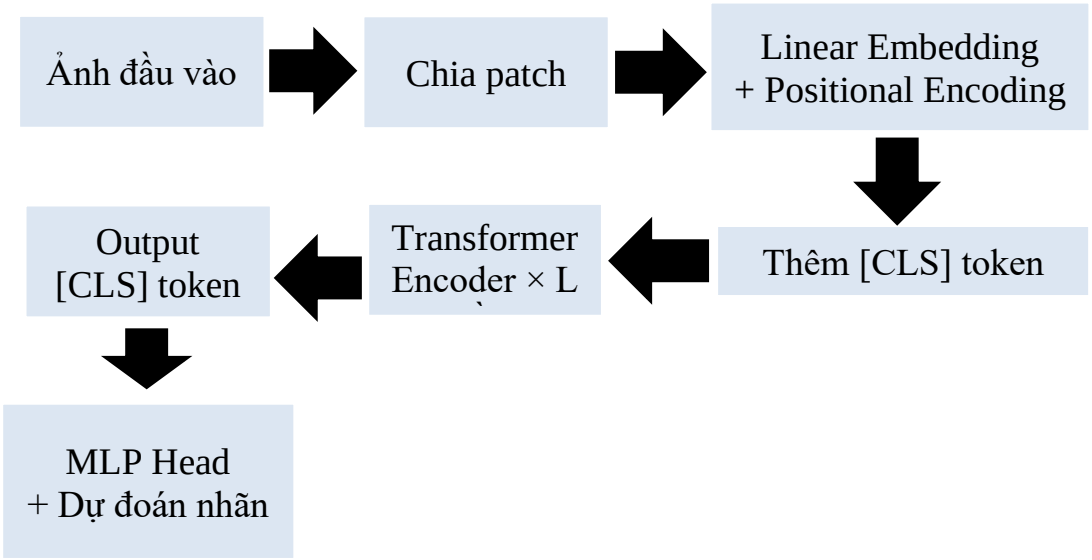
Vision Transformer (ViT) áp dụng mô hình Transformer từ xử lý ngôn ngữ tự nhiên vào thị giác máy tính. ViT chia hình ảnh thành các patch cố định, biến chúng thành vector và xử lý như một chuỗi đầu vào cho Transformer. Khi được

huấn luyện trên các tập dữ liệu lớn như JFT-300M hoặc ImageNet-21k, ViT đạt hiệu suất vượt trội so với các mô hình CNN truyền thống như ResNet trên các benchmark như ImageNet, CIFAR-100 và VTAB **[6]**.



Hình 3.2: Tổng quan của kiến trúc ViT **[10]**

2.2 Cơ chế hoạt động ViT



+ Ảnh đầu vào: Là bức ảnh gốc.

+ Chia patch: Ảnh được chia thành nhiều mảnh nhỏ để xử lý.

+ Linear Embedding + Positional Encoding: Biến mỗi patch thành vector có kèm thông tin vị trí.
+ Thêm [CLS] Token: Token đặc biệt dùng để tổng hợp thông tin cho ảnh.
+ Transformer Encoder × L lần: Dãy token được xử lý qua nhiều lớp Transformer để trích xuất đặc trưng.
+ Output [CLS] Token: Token [CLS] sau khi được xử lý chứa đặc trưng tổng hợp.
+ MLP Head: Mạng nơ-ron đa tầng dùng để phân loại ảnh dựa trên đặc trưng tổng hợp.

2.3 Ưu điểm của ViT

+ Học được mối quan hệ toàn cục (global dependencies) từ sớm.
+ Kiến trúc đơn giản, dễ mở rộng.
+ Tách rời phần học biểu diễn và kiến trúc xử lý.
+ Cho kết quả rất tốt khi huấn luyện với dữ liệu lớn.

2.4 Hạn chế

+ Phụ thuộc nhiều vào dữ liệu lớn và cần huấn luyện lâu.
+ Không nắm tốt thông tin cục bộ như CNN khi dữ liệu ít.
+ Cần GPU mạnh do số lượng tính toán lớn (quadratic complexity của attention).

2.5 Ứng dụng ViT

- ViT được ứng dụng rộng rãi trong:
  + Phân loại ảnh (image classification)
  + Nhận diện ảnh giả (deepfake detection)
  + Phân đoạn ảnh (semantic segmentation)
  + Trích xuất đặc trưng ảnh
  + Ứng dụng trong y học, vệ tinh, và xử lý video (TimeSformer)

Chương IV. Xây dựng mô hình

1. ConvNext

- Khởi tạo mô hình ConvNeXt-Tiny và tùy chỉnh lớp phân loại cuối cùng để huấn luyện

```
model = models.convnext_tiny(weights=models.ConvNeXt_Tiny_Weights.DEFAULT)
num_classes = 2
model.classifier[2] = nn.Linear(model.classifier[2].in_features, num_classes)
model = model.to(device)
```

```
Downloading: "https://download.pytorch.org/models/convnext_tiny-983f1562.pth" to /root/.cache/torch/hub/checkpoints/convnext_tiny-983f1562.pth
100%|██████████| 109M/109M [00:00<00:00, 204MB/s]
```

Hình 4.1.1: Khởi tạo mô hình ConvNeXt-Tiny

- Huấn luyện mô hình:

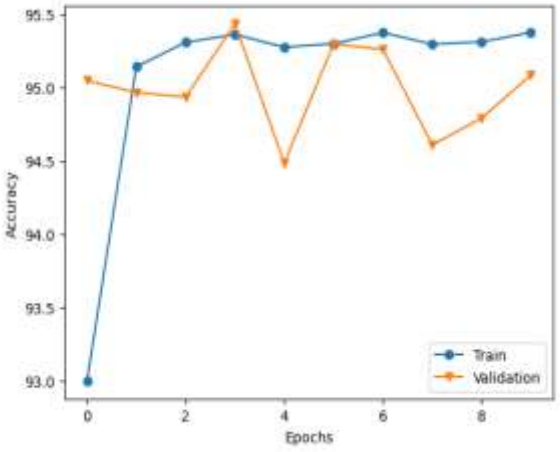
+ Cấu hình huấn luyện: Sử dụng 10 epochs để huấn luyện và đánh giá , dùng CrossEntropy để tính Loss và optimizer Adam để tối ưu và cập nhật trọng số.

```
lr = 1e-3
criterion = nn.CrossEntropyLoss().to(device)
optimizer = torch.optim.AdamW(model.parameters(),lr=lr)
num_epochs=10
```

Hình 4.1.2: Huấn luyện mô hình sử dụng 10 epochs

+ Đánh giá qua các tiêu chí**:** Loss, Accuracy và Ma trận nhầm lẫn

- Accuracy

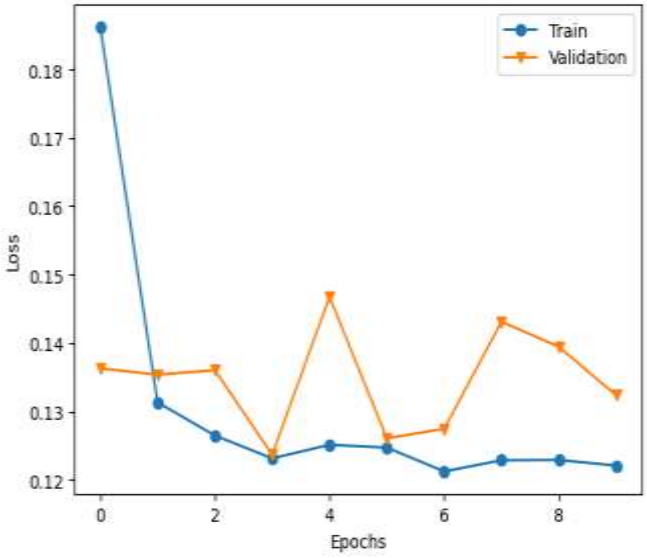


Hình 4.1.3: Biểu đồ Accuracy

+ Train accuracy tăng nhanh và ổn định: Bắt đầu từ khoảng 93.0%, accuracy tăng nhanh đến trên 95.0% chỉ sau 1–2 epoch. Từ epoch 3 trở đi, accuracy duy trì quanh mức 95.0% – 95.4%.

+ Validation accuracy dao động nhẹ: Dao động quanh mức 94.5%–95.5%, nhưng có những điểm tăng giảm đột ngột (epoch 4 và 7). Tuy có xu hướng giảm dần, nhưng độ dao động này phản ánh mô hình chưa tổng quát hóa thực sự tốt, có thể do dữ liệu validation chưa đồng đều.

- Loss



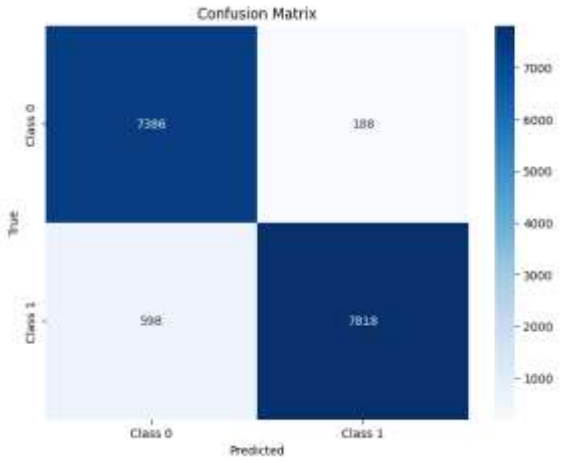
Hình 4.1.4: Biểu đồ Loss

+ Train Loss giảm ổn định: Sau epoch 0, loss giảm mạnh và dao động nhẹ quanh 0.12 từ epoch 2 trở đi → cho thấy mô hình học ổn định trên tập huấn luyện.

+ Validation Loss dao động nhiều: Dễ thấy validation loss dao động không đều, có lúc tăng vọt (ví dụ epoch 4 và 7). Điều này phản ánh mô hình có thể chưa tổng quát hóa tốt trên tập validation, hoặc dữ liệu kiểm định có tính đa dạng hoặc nhiễu cao.

- Nhận xét: Từ epoch 2–9, khoảng cách giữa hai đường loss có lúc khá lớn → nguy cơ mô hình overfitting nhẹ, học tốt trên train nhưng kém ổn định trên validation.

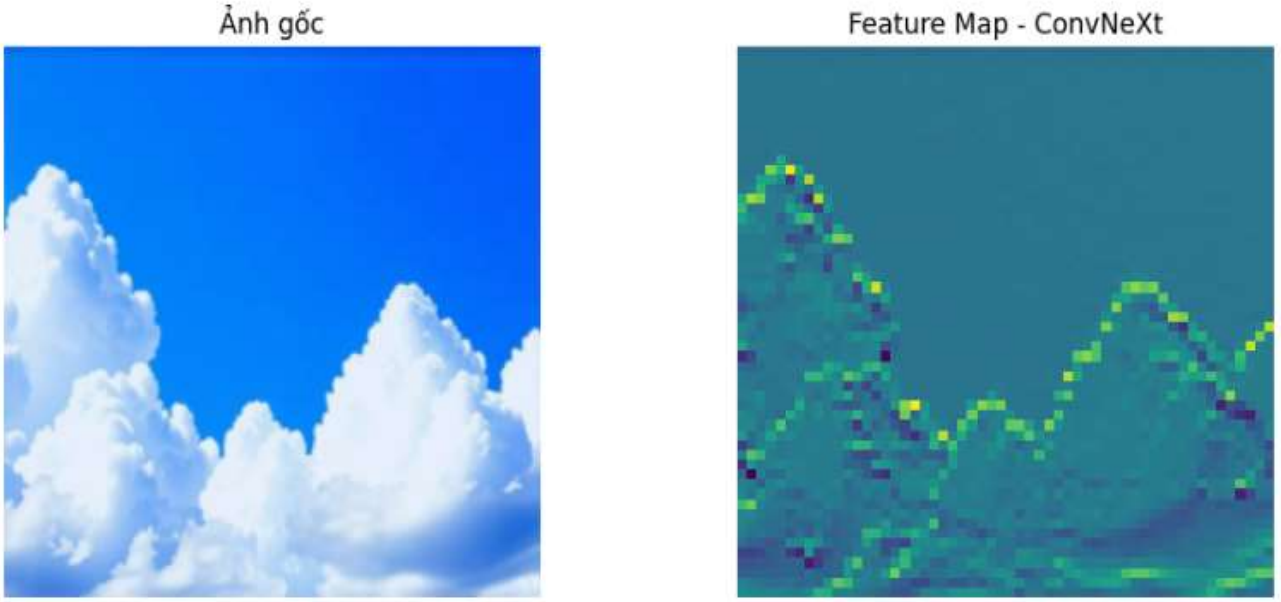
- Ma trận nhầm lẫn



Hình 4.1.5: Biểu đồ heatmap

+ Mô hình này **rất hiệu quả tổng thể**, đặc biệt trong việc **phát hiện ảnh AI**

+ Tuy nhiên, **còn 598 ảnh AI bị nhầm là thật**, điều này có thể đáng lo trong các bối cảnh nghiêm trọng (như deepfake).



Hình 4.1.6: Đặc trưng đầu ra của mô hình ConvNeXt

2. Vision Transformer

- Tải mô hình Vision Transformer (ViT):

```
from transformers import ViTForImageClassification


model_v2 = ViTForImageClassification.from_pretrained(
    'google/vit-base-patch16-224-in21k',
    num_labels=2
)

model_v2 = model_v2.to(device)
for param in model_v2.vit.parameters():
    param.requires_grad = False
```

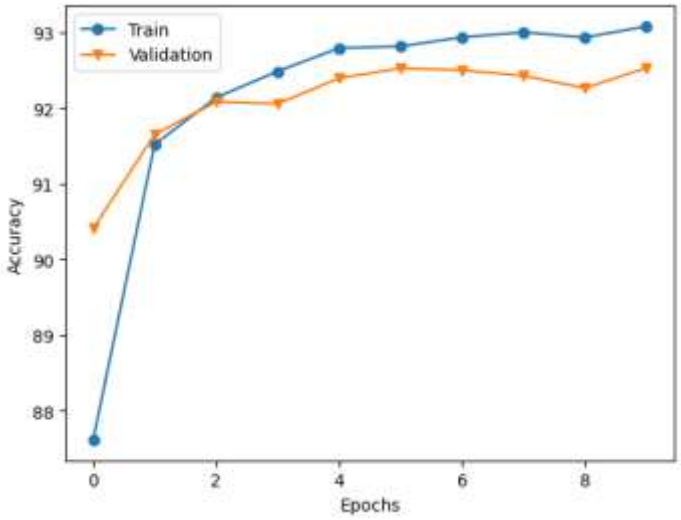
Hình 4.2.1: Tải mô hình ViT

- Vòng lặp huấn luyện (training loop) và đánh giá (validation loop)

+ Cấu hình huấn luyện: Sử dụng 10 epochs để huấn luyện và đánh giá , dùng CrossEntropy để tính Loss và optimizer Adam để tối ưu và cập nhật trọng số.

+ Đánh giá qua các tiêu chí: Loss, Accuracy và Ma trận nhầm lẫn

- Accuracy



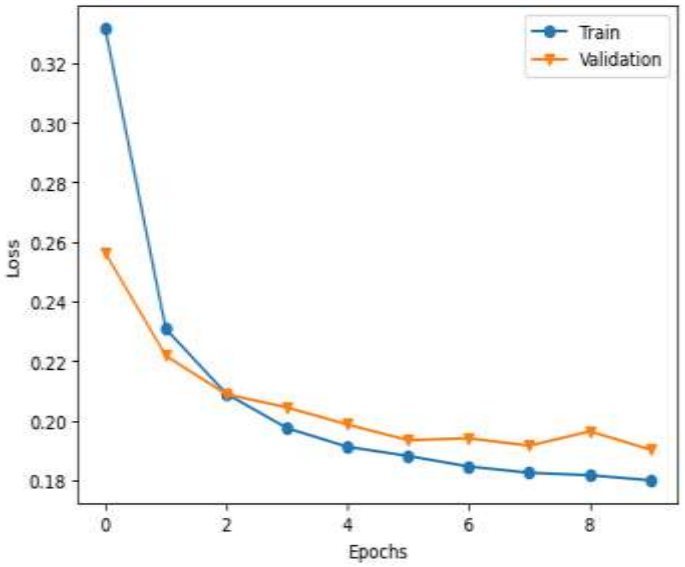
Hình 4.2.2: Biểu đồ Accuracy

+ Tăng nhanh ban đầu: Từ epoch 0 đến epoch 2, cả train và validation accuracy tăng nhanh chóng. Điều này cho thấy mô hình học được đặc trưng rất tốt từ đầu.

+ Ổn định sau epoch 3: Sau epoch 3, độ chính xác train dao động nhẹ quanh **92.9% – 93.1%.** Validation ổn định quanh **92.3% – 92.6%**, không còn tăng nhiều.

=> Train và validation accuracy chênh lệch rất nhỏ (< 1%), cho thấy mô hình **tổng quát tốt**.

- Loss

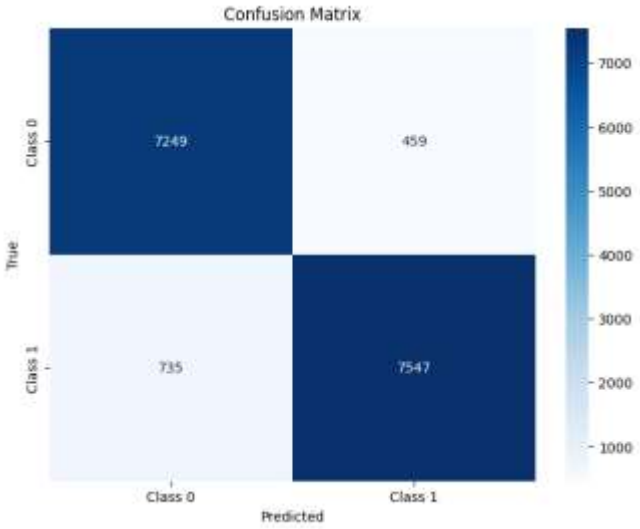


Hình 4.2.3: Biểu đồ Loss

+ Giảm loss rõ rệt ở giai đoạn đầu (epoch 0 → 2): Cả hai đường đều giảm rất nhanh, cho thấy mô hình học được đặc trưng tốt từ dữ liệu ngay từ đầu.

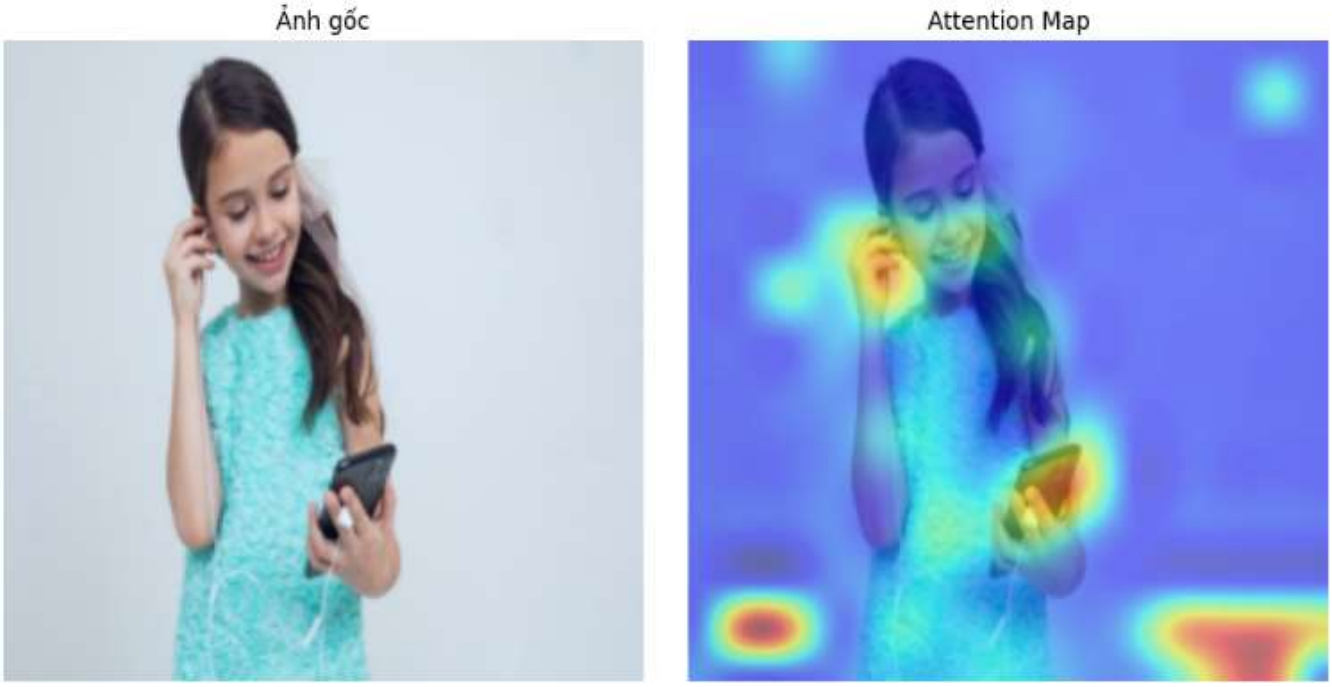
+ Từ epoch 2 trở đi: Loss tiếp tục giảm nhưng chậm dần, đặc biệt ở tập huấn luyện. Tập validation dao động nhẹ quanh mức **0.19 - 0.20**, cho thấy mô hình ổn định.

- Ma trận nhầm lẫn



Hình 4.2.4: Biểu đồ heatmap

+ Mô hình hoạt động rất tốt: Số lượng dự đoán đúng rất cao. Số lượng lỗi dự đoán sai khá thấp -> cho thấy độ chính xác và độ tin cậy cao.
+ Sai lệch: 506 > 231: do dữ liệu không cân bằng hoặc chưa tối ưu.



Hình 4.2.5: Đặc trưng đầu ra của mô hình ViT

3. Đánh giá mô hình và so sánh

Mô hình	Train Accuracy	Validation Accuracy	Nhận xét
ConvNext	~95.35%	~ 95.63%	ConvNext hoạt động tốt hơn ViT trên cả tập Train và Validation, cho thấy mô hình này học hiệu quả hơn và tổng quát tốt hơn.
ViT	~ 92.97%.	~ 92.66%.	ViT vẫn đạt được độ chính xác khá cao và ổn định giữa hai tập, cho thấy khả năng tổng quát hóa tốt.

KẾT LUẬN

Trong báo cáo này, nhóm đã thực hiện xây dựng và đánh giá hai mô hình học sâu hiện đại là ConvNeXt và Vision Transformer (ViT) nhằm giải quyết bài toán phân loại hình ảnh do AI tạo và hình ảnh do con người chụp. Kết quả thực nghiệm cho thấy mô hình ConvNeXt đạt hiệu suất cao hơn với độ chính xác 95.35% trên tập huấn luyện và 95.63% trên tập kiểm định, trong khi ViT đạt lần lượt 92.97% và 92.66%.

Thông qua quá trình triển khai, nhóm nhận thấy ba yếu tố then chốt ảnh hưởng mạnh đến hiệu quả mô hình gồm: chất lượng dữ liệu, kiến trúc mạng được lựa chọn, và chiến lược khởi tạo mô hình. Việc sử dụng mô hình tiền huấn luyện và lựa chọn tập dữ liệu huấn luyện phù hợp đã góp phần nâng cao tính ổn định và khả năng khái quát của hệ thống.

Kết quả đạt được cho thấy hướng tiếp cận này là khả thi và có tiềm năng được mở rộng, từ đó hỗ trợ các ứng dụng thực tiễn như kiểm duyệt nội dung hình ảnh, phát hiện hình ảnh giả trên mạng xã hội hoặc truyền thông.

TÀI LIỆU THAM KHẢO

[1] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," *arXiv preprint arXiv:2204.06125*, 2022.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10684–10695.

[3] Y. Mirsky and T. Mahler, "The creation and detection of deepfakes: A survey," *ACM Comput. Surv.*, vol. 54, no. 1, pp. 1–41, 2021.

[4] S. Wang, X. Yu, and J. Li, "AI-generated image detection based on texture and frequency domain analysis," *J. Vis. Commun. Image Represent.*, vol. 89, p. 103750, 2023.

[5] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2021.

[6] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "ConvNeXt: Revisiting ConvNets for Image Recognition," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12076–12086.

[7] Y. Zhang, P. Sun, H. Qi, and Y. Ma, "Detecting GAN-generated fake images using co-occurrence matrices," *arXiv preprint arXiv:1903.06836*, 2019.

[8] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Detection of GAN-generated fake images over social networks," in *Proc. IEEE Conf. Multimedia Information Processing and Retrieval (MIPR)*, 2018.

[9] The architecture of ConvNeXt-Tiny. | Download Scientific Diagram

[10] Model Architecture Vision Transformer