|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

**Tên đề tài: Xây dựng mô hình phân loại hình ảnh do AI tạo và hình ảnh do người thật tạo**

*Sinh viên thực hiện*: Thái Văn Sáng

Vũ Thị Như Quỳnh

Đồng Anh Quân

Nguyễn Thị Quỳnh Anh

*Lớp*: 64TTNT1

*Giáo viên hướng dẫn*: Th.s Nguyễn Đắc Phương Thảo

**TÓM TẮT ĐỀ TÀI**

Sự ra đời của hình ảnh AI tạo ra đã phần nào phá vỡ thế giới nghệ thuật. Phân biệt hình ảnh do AI tạo ra với nghệ thuật của con người là một vấn đề đầy thách thức mà tác động của nó ngày càng tăng theo thời gian [1]. Đề tài này nhằm xây dựng một mô hình học sâu có khả năng phân loại hình ảnh thành hai nhóm: ảnh do con người chụp (thật) và ảnh do AI tạo ra.

Lí do chọn đề tài:

* Sự ra đời của hình ảnh AI tạo ra đã phần nào phá vỡ thế giới nghệ thuật [1].
* Việc hình ảnh AI tạo ra cho phép kẻ xấu lừa đảo những cá nhân trả phí bảo hiểm cho nghệ thuật của con người và các công ty có chính sách cấm hình ảnh AI [1].
* Ảnh đóng vai trò là cách để con người ghi lại những gì họ trải nhiệm trong cuộc sống hằng ngày và chúng thường được coi là nguồn thông tin đáng tin cây. Tuy nhiên, có một số mối lo ngại ngày càng tang rằng sự tiếng bộ của trí tuệ nhân tạo (AI) công nghệ có thể tạo ra những bức ảnh giải, có thể gây nhầm lẫn và làm giảm tin cậy vào các bước ảnh [2].
* Hình ảnh tạo ra bằng AI có thể dẫn đến việc phát tán thông tin sai lệch ở khắp mọi nơi[3].

**CÁC MỤC TIÊU CHÍNH**

1. Thu thập và tạo tập dữ liệu
2. Tiền xử lý dữ liệu và tăng cường dữ liệu
3. Xây dựng mô hình (ConvNext, ViT).

4. Phân tích so sánh độ chính xác giữa các mô hình khác nhau để lựa chọn mô hình tối ưu.

**KẾT QUẢ DỰ KIẾN**

* Mô hình phân loại được hình ảnh do AI tạo và hình ảnh do người thật tạo.
* Chọn được mô hình tốt nhất

**TIẾN ĐỘ THỰC HIỆN**

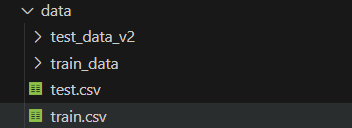
| STT | Thời gian | Nội dung công việc | Kết quả dự kiến đạt được |
| --- | --- | --- | --- |
| 1 | Tuần 1 | Xác định đề tài, lập kế hoạch thực hiện | Hoàn thiện đề cương, xác định hướng nghiên cứu |
| 2 | Tuần 2 | Lấy dữ liệu hình ảnh trên Kaggle | Có bộ dữ liệu phục vụ xây dựng mô hình |
| 3 | Tuần 3 | Tiền xử lý dữ liệu, tăng cường dữ liệu | Dữ liệu sẵn sàng đưa vào mô hình |
| 4 | Tuần 4 | Chọn mô hình học sâu (ConvNext, ViT) | Mô hình có các trọng số phù hợp để phân loại bộ dữ liệu |
| 5 | Tuần 5 | Triển khai các mô hình và fintune mô hình | Chạy mô hình và fintune để phù hợp với dữ liệu của bài |
| 6 | Tuần 6 | Đánh giá mô hình, so sánh độ chính xác | Lựa chọn mô hình tốt nhất |
| 7 | Tuần 7 | Viết báo cáo, làm ptt | Báo cáo đầy đủ nội dung, hoàn thiện ppt |
| 8 | Tuần 8 | Hoàn thiện chuyên đề | Nội dung hoàn chỉnh, chuẩn bị thuyết trình |

**1. Thu thập và tạo tập dữ liệu**

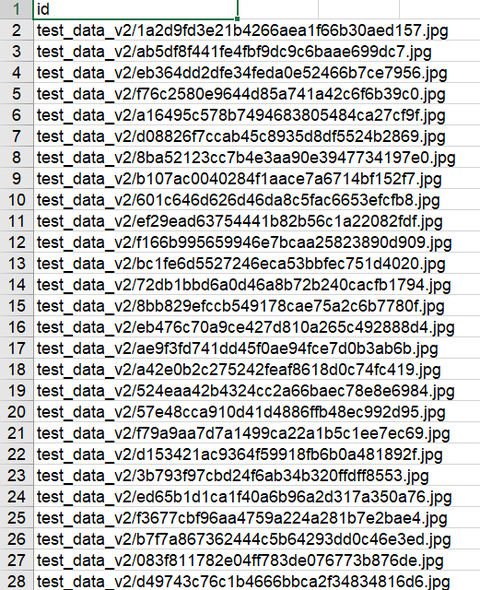
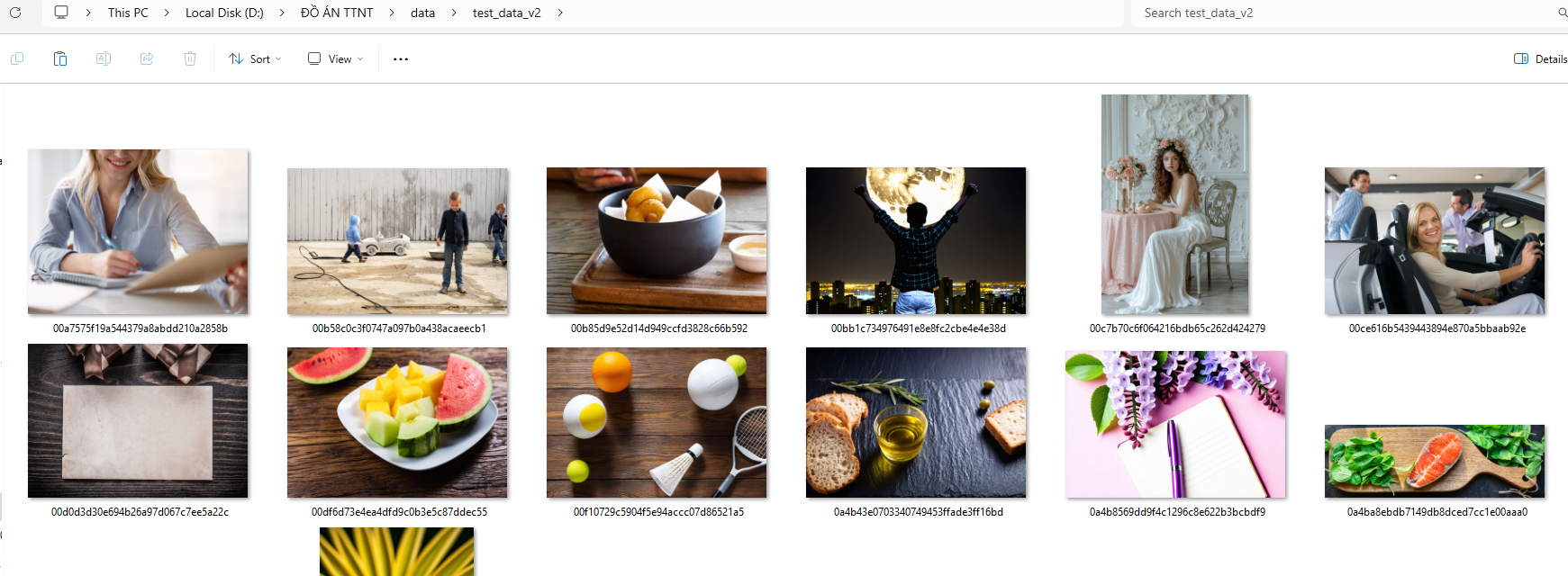
Trong dự án này, nhóm sử dụng bộ dữ liệu mang tên **"**[AI vs Human Generated Dataset](https://www.kaggle.com/datasets/alessandrasala79/ai-vs-human-generated-dataset)**"**, được cung cấp trên nền tảng Kaggle. Bộ dữ liệu có mục tiêu rõ ràng: giúp huấn luyện mô hình có khả năng phân biệt giữa hình ảnh do con người chụp và hình ảnh được tạo ra bởi trí tuệ nhân tạo (AI).

* 1. ***Cấu trúc tập dữ liệu***

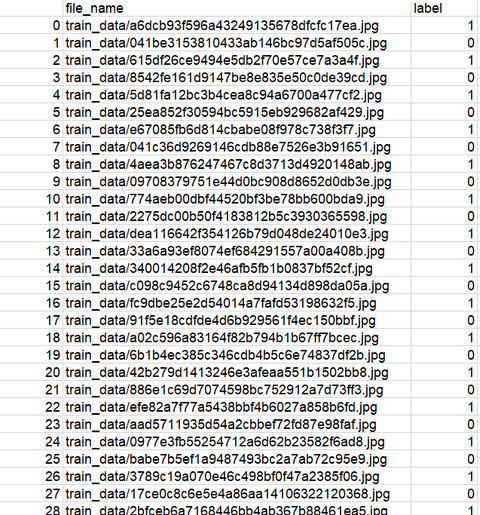
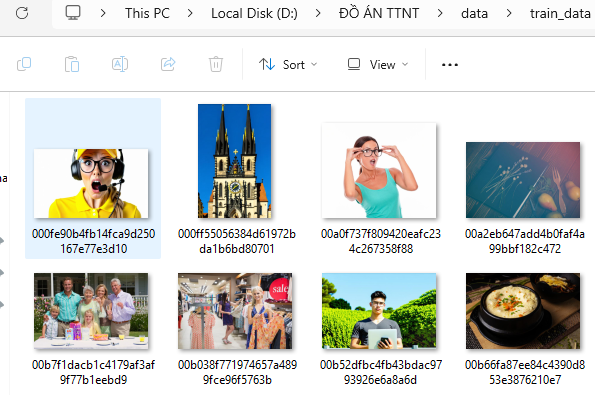
Quá trình xử lý dữ liệu bắt đầu bằng việc phân tích cấu trúc của bộ dữ liệu được cung cấp. Cụ thể, bộ dữ liệu bao gồm 2 thành phần chính: tập huấn luyện (training set) và tập kiểm tra (test set).



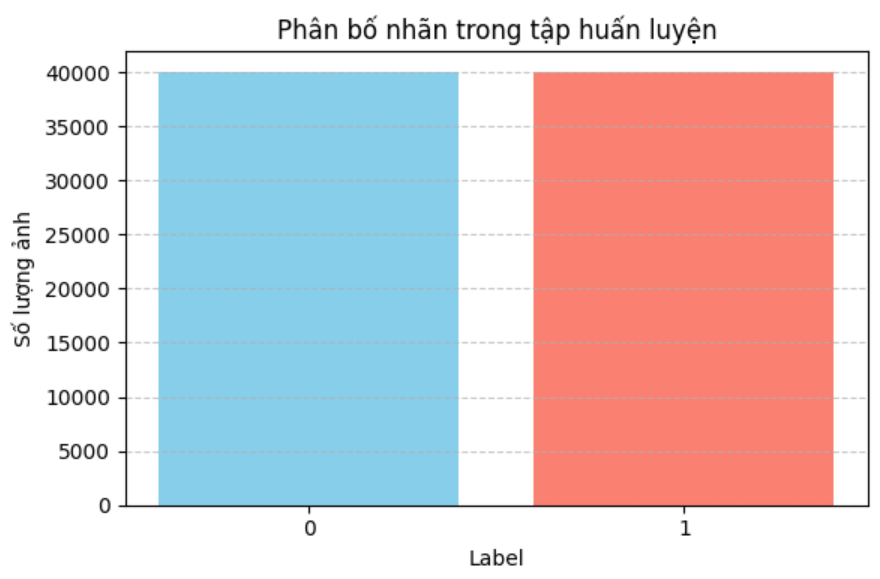
* + Thư mục test\_data\_V2/ là nơi lưu trữ hình ảnh kiểm tra, dùng để đánh giá khả năng tổng quát của mô hình. Thư mục này đi kèm với tệp test.csv, chứa:



* 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:



* 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
  + Trực quan hóa:



* + 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 ảnh do Người tạo (Sample 0) và do AI tạo (Sample 1)

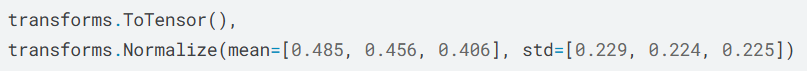
**2. Tiền xử lý dữ liệu và tăng cường dữ liệu**

- 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.



**3. Xây dựng mô hình (ConvNext, ViT).**

**3.1 ConvNext**

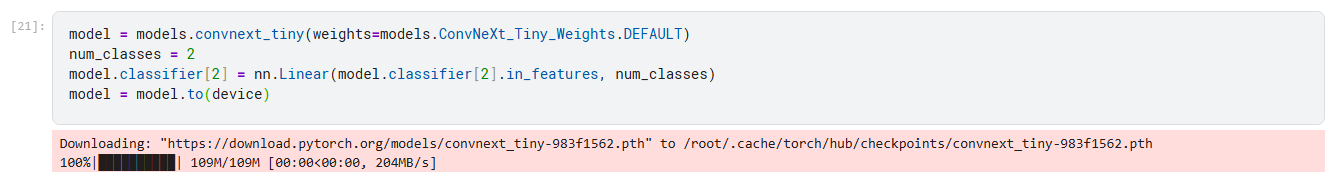
* + 1. ***Lý thuyết ConvNext:***

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].

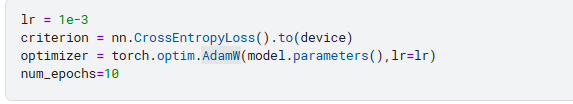
* + 1. ***Cơ chế hoạt động ConvNext***

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:

* Stage 1: Patchify đầu vào (Conv 4×4 stride 4) để giảm kích thước.
* Stage 2 – 4: Stack các ConvNeXt block.
* Cuối cùng: Global Average Pooling + Linear Classifier.
  + 1. ***Áp dụng mô hình ConvNext***

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

- Huấn luyện mô hình:



+ Khởi tạo hàm mất mát (loss function) là CrossEntropyLoss

+ Khởi tạo bộ tối ưu hóa (optimizer) là AdamW

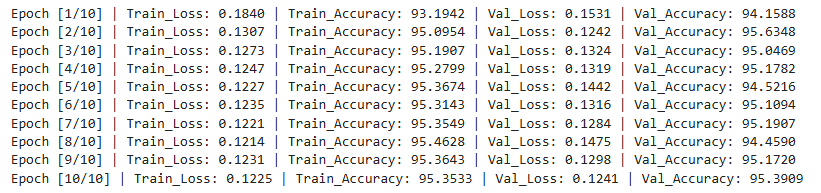
- Vòng lặp huấn luyện (training loop) và đánh giá (validation loop)



- Các độ đo đánh giá mô hình:

+ Train\_Loss, Train\_Accuracy: độ lỗi và độ chính xác trên tập huấn luyện

+ Val\_Loss, Val\_Accuracy: độ lỗi và độ chính xác trên tập kiểm định



+ Train\_Loss: Giảm từ 0.1840 xuống ~0.1225 → cho thấy mô hình học tốt trên tập huấn luyện.

+ Train\_Accuracy: Tăng từ 93.19% lên ~95.35% → độ chính xác cao và ổn định.

+ Val\_Loss: Giao động trong khoảng 0.1241 – 0.1531 → không tăng mạnh → chưa có dấu hiệu overfitting.

+ Val\_Accuracy: Giao động trong khoảng 94.15% – 95.63% → mô hình giữ được khả năng tổng quát hóa tốt.

* 1. **Vision Transformer (ViT)**
     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].

* + 1. ***Cơ chế hoạt động ViT***

Linear Embedding

+ Positional Encoding

Chia patch

Ảnh đầu vào

Thêm [CLS] token

Transformer Encoder × L lần

Output [CLS] token

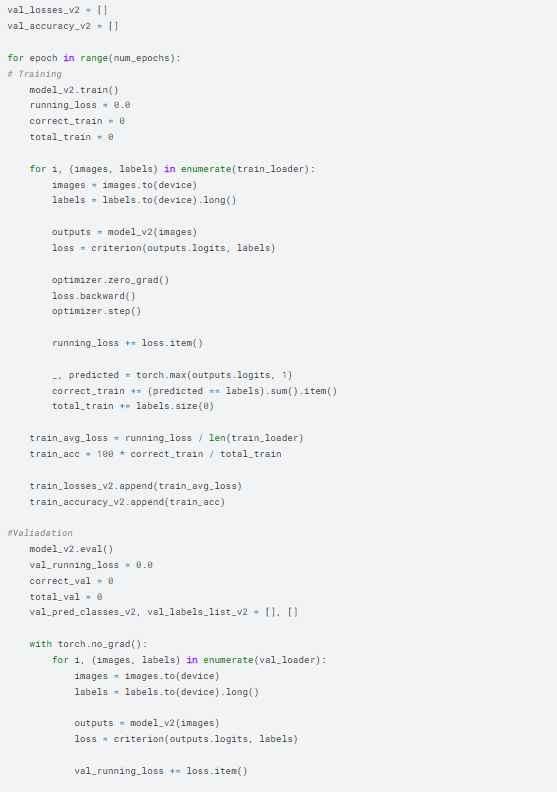
MLP Head

+ Dự đoán nhãn

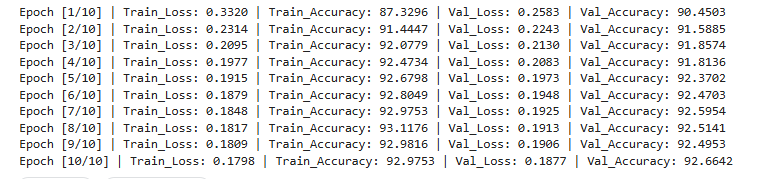
* Ảnh đầu vào: Là bức ảnh gốc.
* Chia patch: Ảnh được chia thành nhiều mảnh nhỏ để xử lý.
* Linear Embedding + Positional Encoding: Biến mỗi patch thành vector có kèm thông tin vị trí.
* Thêm [CLS] Token: Token đặc biệt dùng để tổng hợp thông tin cho ảnh.
* Transformer Encoder × L lần: Dãy token được 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.
  + 1. ***Áp dụng mô hình ViT***
  + Tải mô hình Vision Transformer (ViT):



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



* + Các độ đo đánh giá mô hình:
* Train\_Loss, Train\_Accuracy: độ lỗi và độ chính xác trên tập huấn luyện
* Val\_Loss, Val\_Accuracy: độ lỗi và độ chính xác trên tập kiểm định

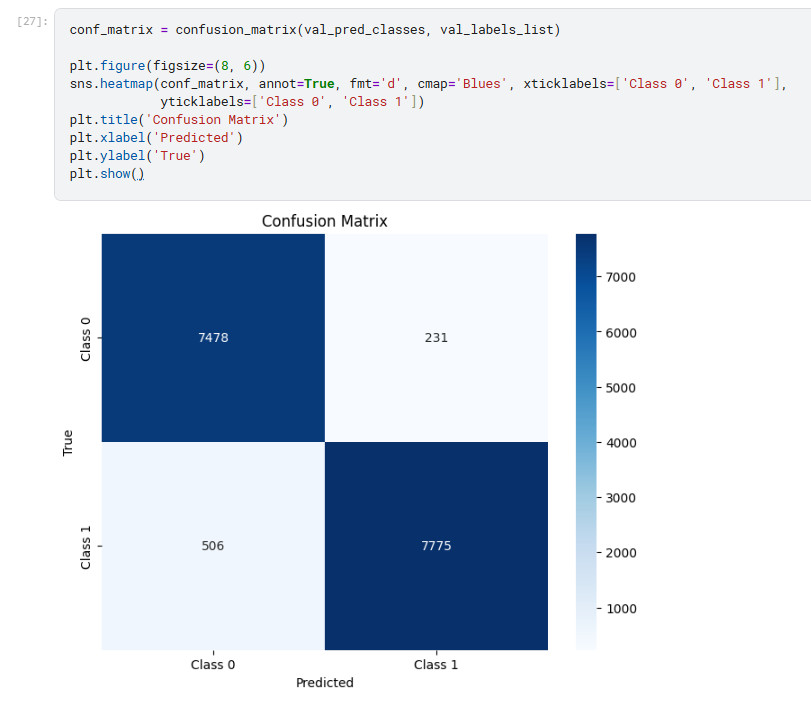


* **Loss giảm đều**:
  + Train\_Loss: Từ **0.3320 → 0.1798**.
  + Val\_Loss: Từ **0.2583 → 0.1877**.
* **Accuracy tăng mạnh**:
  + Train\_Accuracy: Từ **87.33% → 92.97%**.
  + Val\_Accuracy: Từ **90.45% → 92.66%**.

=> Mô hình đang **học hiệu quả**, không bị overfitting hay underfitting

***3.3 Ma trận nhầm lẫn***

* + Ma trận nhầm lẫn:



* **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.

**4. Phân tích so sánh độ chính xác giữa các mô hình khác nhau để lựa chọn mô hình tối ưu.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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. |  |

=> Nên chọn mô hình ConvNext vì nó hoạt động tốt hơn trên cả 2 tập. Tuy nhiên, ta có thể chọn ViT trong khi cần tối ưu chi phí hơn

**TÀI LIỆU THAM KHẢO**

[1] Anna Yoo Jeong Ha, Josephine Passananti, Ronik Bhaskar, Shawn Shan, Reid Southen, Haitao Zheng, và Ben Y. Zhao, “Organic or Diffused: Can We Distinguish Human Art from AI-generated Images?”, trong \*Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS ’24)\*, Salt Lake City, Hoa Kỳ, tháng 10 năm 2024. DOI: https://doi.org/10.1145/3658644.3670306

[2] Z. Lu, D. Huang, L. Bai, J. Qu, C. Wu, X. Liu, and W. Ouyang, *“Seeing is not always believing: Benchmarking Human and Model Perception of AI-Generated Images,”* in *NeurIPS 2023 Datasets and Benchmarks Track*, 2023. [Online]. Available: <https://github.com/Inf-imagine/Sentry>

[3] N. Zhong, Y. Xu, S. Li, Z. Qian, and X. Zhang, "PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection," *arXiv preprint arXiv:2311.12397*, Mar. 2024. [Online]. Available: <https://arxiv.org/abs/2311.12397>  
[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in \*Proc. of the International Conference on Learning Representations (ICLR)\*, 2021.

[5] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," \*arXiv preprint arXiv:2201.03545\*, Jan. 2022. [Online]. Available: <https://arxiv.org/abs/2201.03545>  
[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” \*arXiv preprint arXiv:2010.11929\*, Oct. 2020. [Online]. Available: https://arxiv.org/abs/2010.11929