VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

CO3117 - HỌC MÁY

Báo cáo cuối kì

Dự đoán ảnh AI bằng các mô hình học máy

Advisor(s): Advisor Võ Thanh Hùng

Student(s): Nguyễn Anh Khoa ID 2211612

Nguyễn Hữu Huy Thịnh ID 2213291

HO CHI MINH CITY, APRIL 2025

Contents

List of Figures

List of Tables

Listings

1 Giới thiệu

Trong bối cảnh công nghệ số bùng nổ và nhu cầu kiểm chứng nguồn gốc hình ảnh ngày càng trở nên cấp thiết, việc phân biệt hình ảnh do AI tạo ra với hình ảnh thực tế trở thành một yêu cầu quan trọng trong nhiều lĩnh vực như báo chí, pháp lý, truyền thông và thương mại điện tử. Tuy nhiên, việc nhận diện bằng phương pháp thủ công thường mang tính chủ quan, thiếu chính xác và khó mở rộng trên quy mô lớn.

Với sự phát triển mạnh mẽ của trí tuệ nhân tạo và học máy, các mô hình phân loại và nhận diện hình ảnh đã và đang chứng minh được tính hiệu quả trong việc phát hiện hình ảnh do AI tạo ra. Đề tài này được thực hiện nhằm xây dựng các mô hình nhận diện ảnh AI dựa trên dữ liệu thực tế và các thuật toán học máy hiện đại, góp phần nâng cao độ chính xác trong việc phân loại hình ảnh, hỗ trợ các bên liên quan trong việc đảm bảo tính xác thực của thông tin hình ảnh.

Đề tài sử dụng tập dữ liệu AI vs. Human-Generated Images, bao gồm 79950 dòng dữ liệu và các hình ảnh thật được lấy mẫu từ nền tảng Shutterstock, trải rộng trên nhiều danh mục khác nhau, với tỷ lệ cân đối, trong đó khoảng một phần ba số hình ảnh có sự xuất hiện của con người. Các hình ảnh thật này được ghép cặp với các hình ảnh tương ứng được tạo ra bởi các mô hình sinh ảnh tiên tiến. Việc xây dựng bộ dữ liệu theo cặp như vậy cho phép so sánh trực tiếp giữa hình ảnh thật và hình ảnh do AI tạo ra, từ đó tạo nền tảng vững chắc cho việc huấn luyện và đánh giá các hệ thống nhận diện tính xác thực của hình ảnh. Phạm vi đề tài tập trung vào việc xử lý, huấn luyện và đánh giá mô hình phân loại ảnh, không đi sâu vào phân tích theo ngữ cảnh nội dung hay nguồn gốc cụ thể của từng mô hình sinh ảnh.

2 Tiền xử lý dữ liệu

2.1 Kiểm tra giá trị Null

Trước khi tiền xử lý, chúng tôi tiến hành kiểm tra dữ liệu để xác định xem có bất kỳ giá trị thiếu hoặc dữ liệu bị lỗi nào hay không. Cả hai tập dữ liệu huấn luyện và kiểm tra đều không chứa giá trị `null`, vì vậy không cần xử lý dữ liệu thiếu.

2.2 Kiểm tra sự tồn tại của tệp hình ảnh

Chúng tôi xác minh rằng tất cả các tệp hình ảnh được liệt kê trong các tệp CSV thực sự tồn tại trong thư mục dữ liệu. Kết quả cho thấy không có hình ảnh nào bị thiếu, đảm bảo tính toàn vẹn của bộ dữ liệu.

2.3 Thêm đường dẫn đầy đủ cho hình ảnh

Để thuận tiện cho việc tải hình ảnh trong quá trình huấn luyện, tất cả các tên tệp hình ảnh trong DataFrame được chuyển thành đường dẫn đầy đủ bằng cách kết hợp với thư mục dữ liệu gốc.

2.4 Tạo cặp hình ảnh

Mỗi cặp hình ảnh bao gồm một ảnh thật và một ảnh được tạo bởi AI. Chúng tôi sử dụng chỉ số dòng chia cho 2 để tạo ra cặp định danh `pair_id` cho mỗi cặp.

	id	label	pair_id
0	/root/.cache/kagglehub/datasets/alessandrasala...	1	0
1	/root/.cache/kagglehub/datasets/alessandrasala...	0	0
2	/root/.cache/kagglehub/datasets/alessandrasala...	1	1
3	/root/.cache/kagglehub/datasets/alessandrasala...	0	1
4	/root/.cache/kagglehub/datasets/alessandrasala...	1	2

Figure 2.1: Pair ID

2.5 Phân bố lớp

Chúng tôi kiểm tra sự phân bố của các nhãn để đảm bảo không có hiện tượng mất cân bằng lớp. Kết quả cho thấy số lượng ảnh trong hai lớp là cân bằng, do đó không cần áp dụng các kỹ thuật xử lý mất cân bằng như oversampling hay undersampling.

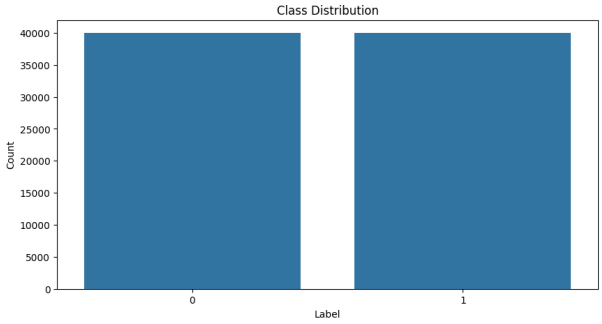

Figure 2.2: Phân bố lớp

2.6 Kiểm tra vi phạm cặp

Chúng tôi kiểm tra xem có cặp nào chứa hai hình ảnh cùng một nhãn hay không (ví dụ cả hai đều là ảnh thật hoặc ảnh AI). Số lượng vi phạm là rất thấp (hoặc bằng 0), cho thấy cặp dữ liệu được xây dựng chính xác.

2.7 Kiểm tra tính đầy đủ của cặp

Mỗi `pair_id` nên có chính xác 2 hình ảnh. Chúng tôi kiểm tra phân phối kích thước cặp và xác nhận rằng mọi cặp đều đầy đủ.

2.8 Phân tích siêu dữ liệu hình ảnh

Chúng tôi thực hiện phân tích siêu dữ liệu cho một mẫu ngẫu nhiên gồm 10.000 hình ảnh từ tập huấn luyện. Phân tích bao gồm:

- Kích thước hình ảnh: chiều rộng và chiều cao trung bình.

- Định dạng hình ảnh: phân bố định dạng JPEG.
- Kênh màu: hầu hết hình ảnh có 3 kênh (RGB), đảm bảo tính nhất quán.

2.9 Phân tích mẫu hình ảnh

Chúng tôi trực quan hóa một số ảnh đại diện từ cả hai lớp (thật và AI) để nhận diện mẫu trực quan, chất lượng hình ảnh và sự khác biệt đặc trưng giữa các lớp.

Figure 2.3: Phân tích mẫu

2.10 So sánh phân bố cường độ điểm ảnh

Cuối cùng, chúng tôi so sánh phân bố cường độ điểm ảnh (dưới dạng grayscale) giữa ảnh do con người tạo và ảnh do AI tạo. Kết quả cho thấy:

- Ảnh AI có xu hướng có một đỉnh lớn ở vùng cường độ cao (200-250), cho thấy có những vùng quá sáng.
- Ảnh thật có phân bố cường độ đều hơn trên toàn dải, phản ánh sự tự nhiên hơn về ánh sáng và độ tương phản.

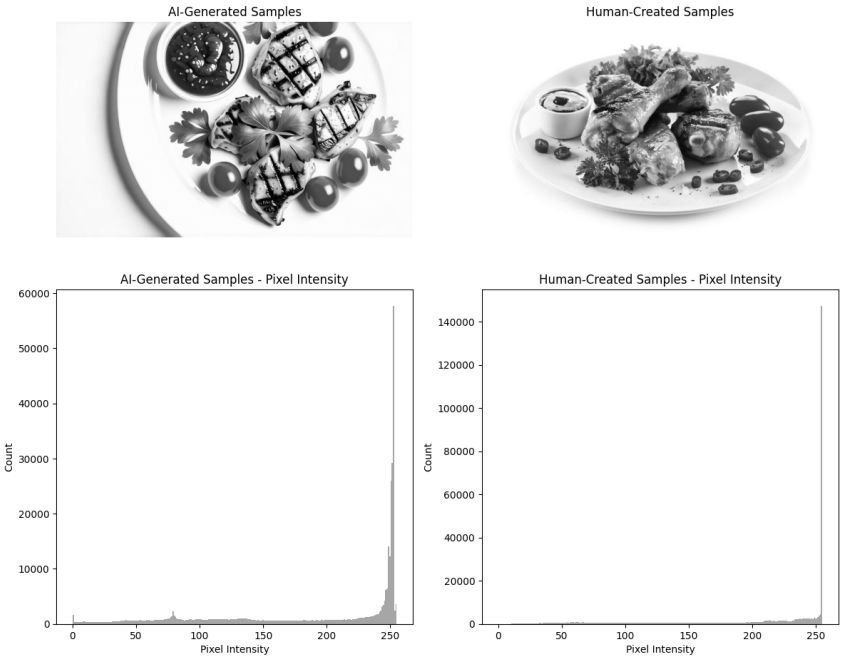

Figure 2.4: Cường độ ảnh

3 Xây dựng mô hình dự đoán

3.1 LBP và XGBOOST

3.1.1 Trích xuất đặc trưng bằng Local Binary Patterns (LBP)

Local Binary Patterns (LBP) là một kỹ thuật phân tích kết cấu hình ảnh phổ biến, cho phép mã hóa thông tin kết cấu cục bộ xung quanh mỗi điểm ảnh. Quy trình trích xuất đặc trưng LBP bao gồm các bước:

- Đọc ảnh dưới dạng ảnh xám.
- Resize ảnh về kích thước cố định 64×64 để đồng bộ dữ liệu.
- Áp dụng bộ lọc Gaussian để giảm nhiễu.
- Tính toán LBP cho từng pixel dựa trên số điểm lân cận và bán kính xác định.
- Xây dựng histogram chuẩn hóa của các giá trị LBP để làm vector đặc trưng đầu vào cho mô hình.

Hàm trích xuất đặc trưng LBP được định nghĩa như sau:

```
def extract_lbp_features(img_path, radius=3, points=24, method="uniform"):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    if img is None:
        return np.zeros(points + 2)
    img = cv2.resize(img, (64, 64))
    img = cv2.GaussianBlur(img, (5, 5), 0)
    lbp = local_binary_pattern(img, points, radius, method)
    (hist, _) = np.histogram(lbp.ravel(), bins=np.arange(0, points + 3), range=(0,
    hist = hist.astype("float")
    hist /= (hist.sum() + 1e-7)
    return hist
```

Vector đặc trưng này sau đó được sử dụng làm đầu vào cho mô hình phân loại.

3.1.2 XGBoost

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy mạnh mẽ thuộc nhóm ensemble learning, cụ thể là phương pháp boosting. Nó được thiết kế để tối ưu hiệu năng và tốc độ khi huấn luyện mô hình cây quyết định. Nguyên lý hoạt động: XGBoost xây dựng mô hình bằng cách kết hợp nhiều cây quyết định theo từng vòng lặp. Mỗi cây mới được huấn luyện nhằm sửa lỗi của cây trước đó, giúp mô hình học tốt hơn theo thời gian.

1. Hàm mục tiêu

Ở vòng boosting thứ t, mô hình XGBoost tối ưu hàm mục tiêu sau:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

Trong đó:

- l: hàm mất mát (thường dùng MSE: $l(y, \hat{y}) = (y - \hat{y})^2$)
- ${\hat{y}_i}^{(t-1)}$: dự đoán của mô hình tại bước t - 1
- f_t: cây quyết định mới tại bước t
- $\Omega(f_t)$: thành phần regularization giúp chống overfitting

$$\Omega(f)f_t = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Trong đó:

- T: số lá của cây
- w_j: trọng số lá thứ j
- γ, λ: hệ số điều chỉnh độ phức tạp mô hình

2. Xấp xỉ Taylor bậc hai

XGBoost sử dụng xấp xỉ Taylor bậc hai để làm trơn hàm mất mát:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

Trong đó: - $g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ (đạo hàm bậc 1) - $h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$ (đạo hàm bậc 2)

3. Tính giá trị tối ưu tại mỗi lá

Với mỗi lá j, tập mẫu thuộc lá là I_j. Tổng gradient và hessian tại lá:

$$G_j = \sum_{i \in I_j} g_i, \quad H_j = \sum_{i \in I_j} h_i$$

Giá trị đầu ra tối ưu tại lá:

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$

Giá trị hàm mục tiêu tại điểm cực tiểu:

$$\mathcal{L}^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T$$

4. Quá trình huấn luyện XGBoost bao gồm các bước chính sau:

1. **Khởi tạo mô hình:** Dự đoán ban đầu là xác suất.
2. **Tính toán gradient và hessian:** Các gradient (g_i) và hessian (h_i) được tính cho mỗi mẫu dựa trên hàm mất mát.
3. **Xây dựng cây quyết định:** Mỗi cây mới được xây dựng để tối ưu hóa hàm mất mát với việc sử dụng gradient và hessian.
4. **Cập nhật dự đoán:** Dự đoán của mô hình được cập nhật bằng cách cộng thêm giá trị từ cây mới.

5. **Regularization:** Thành phần regularization giúp kiểm soát độ phức tạp của mô hình và ngăn ngừa overfitting.
6. **Lặp lại quá trình:** Các bước trên được lặp lại cho đến khi đạt số vòng lặp tối đa hoặc không còn cải thiện đáng kể.

XGBoost là một lựa chọn mạnh mẽ cho bài toán phân loại nhờ vào khả năng xử lý các mối quan hệ phi tuyến tính phức tạp trong dữ liệu. Mô hình này sử dụng kỹ thuật boosting, giúp tăng cường độ chính xác qua từng vòng lặp bằng cách học từ các sai sót của các cây quyết định trước đó.

3.2 GridSearchCV

GridSearchCV là một phương pháp được sử dụng để tìm kiếm và tối ưu các siêu tham số (hyperparameters) của một mô hình học máy, nhằm cải thiện độ chính xác và hiệu suất của mô hình. Nó thực hiện thử tất cả các tổ hợp các giá trị siêu tham số có thể có từ một "lưới" các giá trị đã được chỉ định trước.

Nguyên lý hoạt động:

GridSearchCV tìm kiếm trong không gian siêu tham số bằng cách thử nghiệm với mọi giá trị của các tham số đã cho, sử dụng phương pháp k-fold cross-validation để đánh giá hiệu suất của mỗi kết hợp tham số. Mục tiêu là tìm ra bộ tham số có thể tối ưu hóa độ chính xác của mô hình.

Quá trình này có thể được mô tả như sau:

$$\text{Best Parameters} = \arg\max_{\theta} \left(\frac{1}{k} \sum_{i=1}^{k} \text{score}(X_{train}^{(i)}, y_{train}^{(i)}, \theta) \right)$$

Trong đó:

- θ là bộ tham số của mô hình.
- $X_{train}^{(i)}$ và $y_{train}^{(i)}$ là dữ liệu huấn luyện tại lần gập thứ i trong quá trình k-fold cross-validation.

- score(X, y, θ) là độ chính xác của mô hình với tham số θ trên bộ dữ liệu X và y.

Tối ưu siêu tham số với GridSearchCV

Trong bài toán này, nhóm sẽ kết hợp boosting cây quyết định, cụ thể là mô hình XGBoost, với kỹ thuật GridSearchCV để tối ưu hóa các tham số của mô hình. Phương pháp boosting được sử dụng với XGBoost, nhằm tăng cường hiệu quả dự đoán và giảm thiểu lỗi.

Mô hình XGBoost có các siêu tham số quan trọng có thể điều chỉnh để tối ưu hóa hiệu suất. Với GridSearchCV, nhóm sẽ tìm kiếm và chọn lựa các giá trị siêu tham số tối ưu thông qua việc thử nghiệm tất cả các kết hợp có thể của các tham số trong không gian tìm kiếm. Quá trình này giúp mô hình đạt được hiệu quả dự đoán tốt nhất cho bài toán cụ thể.

GridSearchCV sẽ giúp tìm ra các tham số tối ưu của mô hình thông qua việc đánh giá hiệu suất mô hình trên một tập kiểm tra (cross-validation) với các tham số khác nhau, từ đó chọn ra bộ tham số tối ưu giúp mô hình hoạt động hiệu quả nhất. Các tham số sẽ bao gồm:

- **XGBoost:** số lượng cây (*n_estimators*), tốc độ học (*learning_rate*), độ sâu của cây (*max_depth*), tỷ lệ mẫu (*subsample*), gamma.

Việc kết hợp giữa boosting và GridSearchCV giúp mô hình cải thiện khả năng dự đoán trong các bài toán phức tạp như dự báo giá nhà, từ đó giúp mô hình có khả năng học được các mối quan hệ phi tuyến tính và giảm thiểu overfitting.

3.3 Kiến Trúc ResNet-50

ResNet-50 là một mạng nơ-ron tích chập sâu sử dụng các kết nối residual (còn gọi là kết nối bỏ qua) để giải quyết vấn đề gradient biến mất. Ý tưởng chính của ResNet là cho phép mạng bỏ qua một hoặc nhiều lớp trong quá trình truyền tiến, giúp mạng học các phép biến đổi đồng nhất dễ dàng hơn và giúp đào tạo các mạng nơ-ron sâu hơn.

3.3.1 Học Residual

Trong các mạng nơ-ron truyền thống, khi độ sâu của mạng tăng lên, gradient trong quá trình lan truyền ngược có xu hướng rất nhỏ, khiến việc huấn luyện trở nên khó khăn. ResNet khắc phục vấn đề này bằng cách sử dụng học residual, trong đó đầu vào của một lớp được cộng thêm với đầu ra của lớp đó, giúp bỏ qua một số phép biến đổi.

Khối residual được định nghĩa như sau:

$$y = F(x, \{W_i\}) + x$$

Trong đó $F(x, \{W_i\})$ đại diện cho các phép toán trong khối residual, và x là đầu vào. Phép cộng này giúp duy trì gradient trong quá trình lan truyền ngược, làm cho mạng dễ dàng huấn luyện hơn.

3.3.2 Kiến Trúc Mạng ResNet-50

Kiến trúc ResNet-50 bao gồm 50 lớp, được tổ chức thành nhiều giai đoạn, mỗi giai đoạn chứa một số khối residual. Các thành phần chính của mạng bao gồm:

- **Các Lớp Convolutional:** Mạng bắt đầu với một loạt các lớp convolutional để trích xuất đặc trưng từ hình ảnh đầu vào. Lớp convolution đầu tiên sử dụng kernel 7x7, theo sau là lớp max-pooling 3x3.
- **Các Khối Residual:** Các khối residual là thành phần cốt lõi của ResNet, mỗi khối chứa hai hoặc ba lớp convolutional. Các khối này được xếp chồng lên nhau để tạo thành các lớp sâu hơn của mạng.
- **Kiến Trúc Bottleneck:** ResNet-50 sử dụng kiến trúc bottleneck cho các khối residual, giúp giảm số lượng tham số và làm cho mạng hiệu quả hơn.
- **Lớp Kết Nối Hoàn Toàn:** Sau các lớp convolutional và residual, đầu ra được truyền qua một lớp pooling trung bình toàn cục và sau đó là một lớp kết nối hoàn toàn (fully connected) để phân loại.

3.3.3 Sửa Đổi Cho Phân Loại Nhị Phân

Để giải quyết bài toán phân loại hình ảnh là hình ảnh người tạo hoặc hình ảnh AI tạo, chúng ta sửa đổi lớp fully connected (FC) cuối cùng của ResNet-50. Lớp FC gốc của ResNet-50 đầu ra 1000 lớp (cho phân loại ImageNet). Tuy nhiên, đối với bài toán phân loại nhị phân của chúng ta, chúng ta thay thế lớp này bằng một lớp FC mới, xuất ra hai lớp: Hình ảnh Người tạo và Hình ảnh AI tạo.

Sửa đổi này được thực hiện bằng cách thay thế lớp FC bằng mã sau:

```
model.fc = nn.Linear(ftrs, 2)
```

Trong đó $ftrs$ là số lượng đặc trưng đầu vào của lớp FC, và 2 là số lớp đầu ra, tương ứng với hai loại hình ảnh.

4 Huấn Luyện

Ta tiến hành chia tập thành 2 phần, dữ liệu train và dữ liệu test theo tỷ lệ 80% cho tập train và 20% cho tập test.

4.1 LBP và XGBoost

Trong giai đoạn huấn luyện, phương pháp kết hợp giữa trích xuất đặc trưng bằng LBP và phân loại bằng XGBoost được áp dụng.

Đầu tiên, LBP được sử dụng để trích xuất đặc trưng từ ảnh đầu vào, chuyển đổi mỗi ảnh thành một vector histogram. Các vector histogram này sau đó được sử dụng làm đầu vào cho mô hình XGBoost.

Để tối ưu hóa mô hình, GridSearchCV được sử dụng với chiến lược cross-validation gồm 5 lần gập (5-fold) và phương thức đánh giá là `log_loss`. Các siêu tham số được tìm kiếm bao gồm:

- `n_estimators`: [100, 300, 500]

- `max_depth`: [3, 6, 9]
- `learning_rate`: [0.01, 0.1, 0.2]
- `gamma`: [0, 0.1, 0.2]
- `subsample`: [0.7, 0.8, 1.0]

Sau khi tìm kiếm, mô hình tìm được bộ siêu tham số tốt nhất như sau:

- `gamma`: 0.2
- `learning_rate`: 0.1
- `max_depth`: 6
- `n_estimators`: 300
- `subsample`: 1.0

4.2 Resnet-50

Trong giai đoạn huấn luyện với ResNet-50, mô hình được tải trọng số đã fine-tune từ một checkpoint đã lưu trước đó và chuyển sang GPU nếu khả dụng.

Mô hình được huấn luyện trong 5 epoch sử dụng bộ tối ưu hóa AdamW với tốc độ học (`learning rate`) là 0.001 và hàm mất mát là `CrossEntropyLoss`, phù hợp với bài toán phân loại.

5 Đánh giá và So sánh Hai Mô Hình

Trong phần này, chúng ta sẽ đánh giá và so sánh hiệu suất của hai mô hình: **LBP + XGBoost** và **ResNet (CNN)**. Cả hai mô hình đều được áp dụng trên bộ dữ liệu phân loại ảnh thật và ảnh do AI tạo ra. Các chỉ số đánh giá như độ chính xác, độ chính xác (precision), độ hồi phục (recall), và điểm F1 sẽ được tính toán và so sánh.

5.1 Đánh giá Mô Hình XGBoost với LBP

Trước tiên, chúng ta sẽ đánh giá mô hình XGBoost sử dụng đặc trưng Local Binary Pattern (LBP). Kết quả được tính toán trên bộ dữ liệu kiểm tra.

- **Độ chính xác (Accuracy)**: 72%
- **Precision, Recall, F1-Score**: Cả ba chỉ số đều đạt 0.72 cho cả hai lớp.
- **Quan sát**: Mô hình này hoạt động ổn định giữa các lớp, tuy nhiên hiệu suất tổng thể vẫn khá khiêm tốn. Điều này cho thấy mặc dù XGBoost với đặc trưng LBP có thể nắm bắt được một số mẫu, nhưng khả năng của nó trong việc trích xuất các đặc trưng phức tạp từ dữ liệu là hạn chế.

5.2 Đánh giá Mô Hình ResNet (CNN)

Tiếp theo, chúng ta sẽ đánh giá mô hình ResNet (mạng nơ-ron học sâu) trên cùng bộ dữ liệu. ResNet đã được tinh chỉnh với một lớp fully connected để phân loại giữa hai lớp (thật và AI).

- **Độ chính xác (Accuracy)**: 89%
- **Precision, Recall, F1-Score**:
  - **Lớp 0 (Thật)**: Precision = 0.91, Recall = 0.87, F1 = 0.89
  - **Lớp 1 (AI)**: Precision = 0.87, Recall = 0.92, F1 = 0.90

- **Quan sát**: Mô hình ResNet vượt trội hơn hẳn XGBoost trên mọi chỉ số. Mô hình này tổng quát tốt qua cả hai lớp, với sự cân bằng giữa độ chính xác và độ hồi phục. Điểm F1 cao cho thấy mô hình không chỉ chính xác mà còn đáng tin cậy trong việc phân loại và phát hiện.

5.3 Ma Trận Nhầm Lẫn

Tiếp theo, chúng ta sẽ hiển thị ma trận nhầm lẫn của cả hai mô hình để cung cấp một cái nhìn trực quan hơn về hiệu suất của chúng.

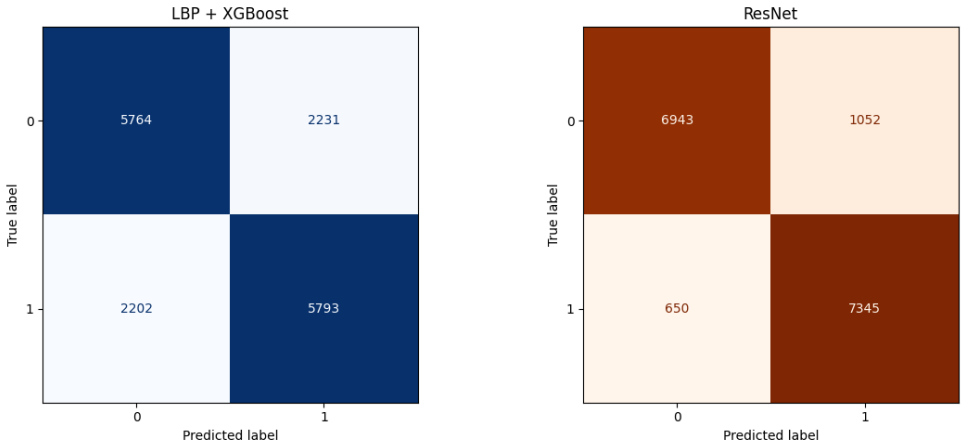

Figure 5.1: Ma Trận Nhầm Lẫn giữa LBP + XGBoost và ResNet

5.4 So sánh Độ Chính Xác, Precision, Recall và F1-Score

Cuối cùng, chúng ta sẽ vẽ biểu đồ so sánh độ chính xác, precision, recall và F1-score của hai mô hình để có cái nhìn trực quan về sự khác biệt giữa chúng.

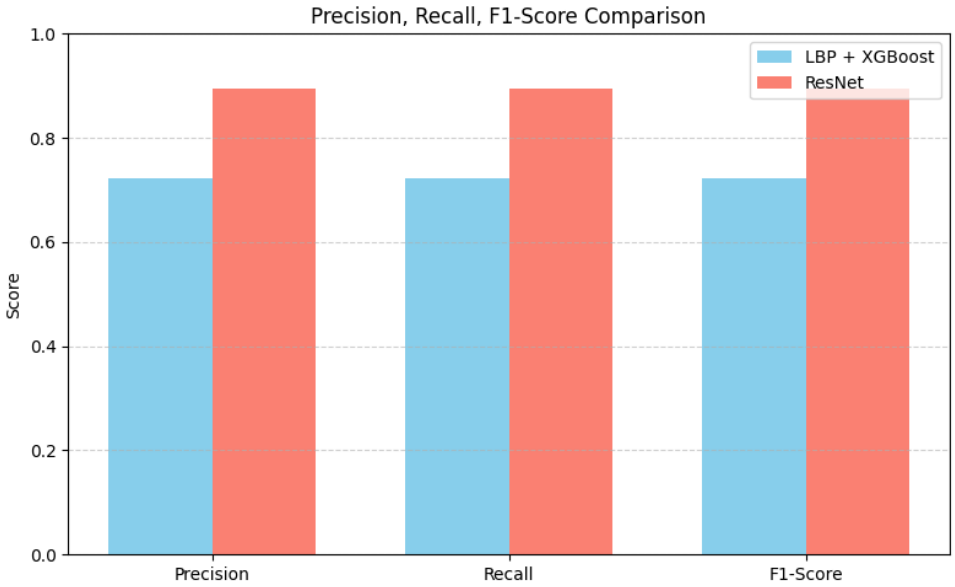

Figure 5.2: So Sánh Precision, Recall và F1-Score

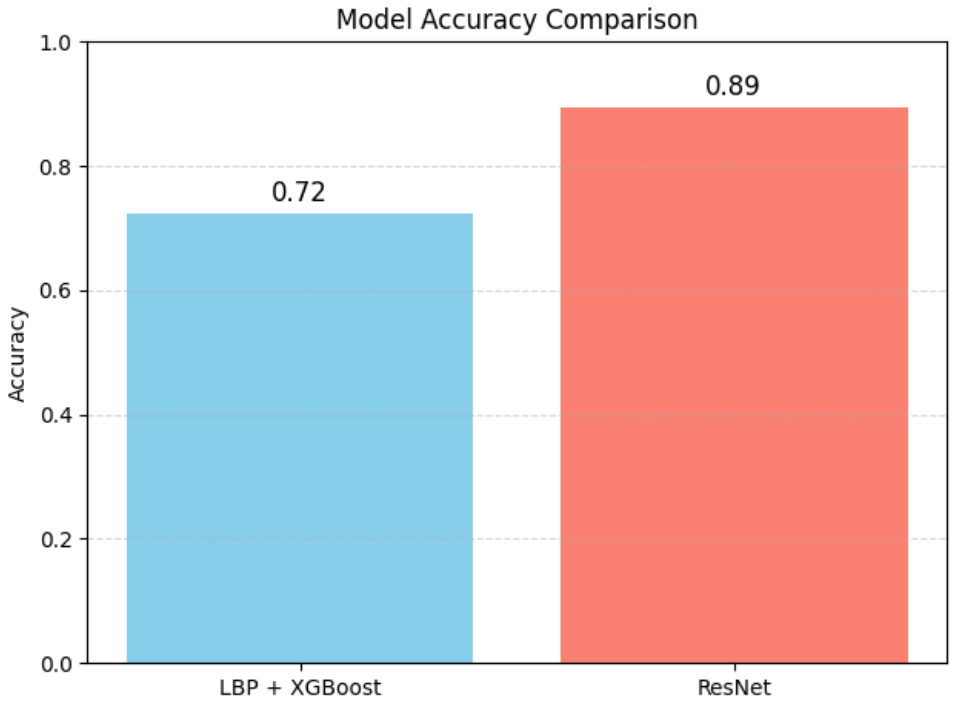

Figure 5.3: So Sánh Accuracy

5.5 Kết luận

Trong bài toán nhận diện ảnh AI, hai phương pháp tiếp cận khác nhau đã được nghiên cứu gồm LBP kết hợp với XGBoost và mô hình ResNet-50. Mỗi phương pháp mang lại những kết quả và đặc trưng riêng biệt trong quá trình nhận diện.

Phương pháp sử dụng LBP và XGBoost cho thấy khả năng nhận diện ảnh AI ở mức khá, đặc biệt đối với những ảnh AI có kết cấu khác biệt rõ rệt so với ảnh thật. Tuy nhiên, do LBP chủ yếu dựa trên việc trích xuất đặc trưng kết cấu bề mặt, nên phương pháp này gặp nhiều khó khăn trước các ảnh AI do các mô hình thế hệ mới tạo ra. Những ảnh này có kết cấu tinh vi và gần gũi hơn với ảnh thật, khiến hiệu quả của LBP suy giảm rõ rệt.

Ngược lại, mô hình ResNet-50 cho phép hệ thống tự động học các đặc trưng phức tạp và cấp cao từ dữ liệu, thay vì phải dựa vào các kỹ thuật trích xuất thủ công. Nhờ khả năng học sâu, ResNet-50 đạt được độ chính xác cao hơn và thể hiện khả năng tổng quát hóa tốt hơn khi đối mặt với các ảnh AI tinh vi.

Từ những kết quả đạt được, có thể kết luận rằng việc áp dụng các mô hình học sâu như ResNet-50 mang lại lợi thế vượt trội trong bài toán nhận diện ảnh AI. So với các phương pháp truyền thống dựa trên trích xuất đặc trưng thủ công, học sâu cho phép mô hình khai thác các biểu diễn đặc trưng mạnh mẽ và đa dạng hơn, giúp nâng cao hiệu quả nhận diện trong bối cảnh ảnh AI ngày càng tinh vi.

6 Tổng kết

Bài tập lớn này nhằm mục tiêu nghiên cứu các phương pháp hiệu quả để phát hiện sự khác biệt giữa hình ảnh thật và hình ảnh do AI tạo ra, nhằm đối phó với những lo ngại ngày càng gia tăng về deepfake và các phương tiện truyền thông tổng hợp. Trong bối cảnh công nghệ trí tuệ nhân tạo phát triển nhanh chóng, việc nhận diện chính xác nội dung giả mạo trở nên cấp thiết hơn bao giờ hết để bảo vệ tính minh bạch, an toàn thông tin và niềm tin của người dùng.

Trong quá trình nghiên cứu, hai mô hình XGBoost và ResNet đã được lựa chọn và triển khai cho bài toán phân loại này. Kết quả thực nghiệm cho thấy mô hình XGBoost đạt được hiệu quả ở mức khá, có khả năng phát hiện các ảnh AI ở mức độ nhất định, đặc biệt trong các trường hợp mà sự khác biệt về đặc trưng bề mặt còn rõ rệt. Tuy nhiên, với sự phát triển của các mô hình tạo ảnh ngày càng tinh vi, phương pháp này gặp nhiều hạn chế do phụ thuộc chủ yếu vào các đặc trưng thủ công.

Ngược lại, mô hình ResNet, một kiến trúc học sâu điển hình, đã chứng minh khả năng vượt trội trong việc phân loại chính xác giữa hình ảnh thật và hình ảnh do AI tạo ra. Nhờ vào kiến trúc nhiều tầng và khả năng tự động học các đặc trưng phức tạp từ dữ liệu, ResNet có thể khai thác được những yếu tố tinh tế hơn mà các phương pháp truyền thống khó nắm bắt. Điều này giúp mô hình không chỉ đạt được độ chính xác cao hơn mà còn thể hiện khả năng tổng quát hóa tốt hơn trong việc nhận diện các dạng nội dung giả mạo mới.

Trong tương lai, hướng mở rộng của nghiên cứu có thể bao gồm việc áp dụng các phương pháp này lên các loại phương tiện truyền thông đa dạng hơn, chẳng hạn như video, nhằm tăng cường khả năng phát hiện và phòng chống nội dung do AI tạo ra trên nhiều nền tảng khác nhau.

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *arXiv:1603.02754*, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.

[3] The scikit-learn developers. Gridsearchcv — scikit-learn 0.24.2 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, 2021.