Đồ án cuối môn hướng AI/NLP
Đề tài: Liveness Detection

MÔN HỌC: PHÂN TÍCH DỮ LIỆU CHUYÊN BIỆT

GIÁO VIÊN: PGS.TS. Nguyễn Đình Thúc
HỌC VIÊN: 22C01026 - Nguyễn Ngọc Thảo Uyên

1. Bài toán Liveness Detection

Trong những năm gần đây, chúng ta đã chứng kiến một sự gia tăng đáng kể của bài toán Liveness Detection hay còn gọi là nhận diện và xác thực khuôn mặt. Điều này thể hiện rõ qua các ứng dụng tiêu biểu như việc sử dụng khuôn mặt để mở khóa điện thoại trên các smartphone hoặc yêu cầu xác thực khuôn mặt khi đăng ký tài khoản ngân hàng trực tuyến.

Tuy nhiên, với sự tiện lợi đến từ việc sử dụng hệ thống nhận diện khuôn mặt, xuất hiện cũng những thách thức đáng kể. Hệ thống nhận diện khuôn mặt có thể bị "đánh lừa" thông qua việc sử dụng ảnh in hoặc video chứa khuôn mặt của người đó. Điều này đặt ra một thách thức nghiêm trọng và yêu cầu sự chú ý của cộng đồng nghiên cứu và các nhà phát triển. Trong báo cáo này, một phương pháp được đề xuất chính là sử dụng mô hình BEIT cho việc nhận diện và xác thực khuôn mặt.

2. Giới Thiệu Về Mô Hình BEIT

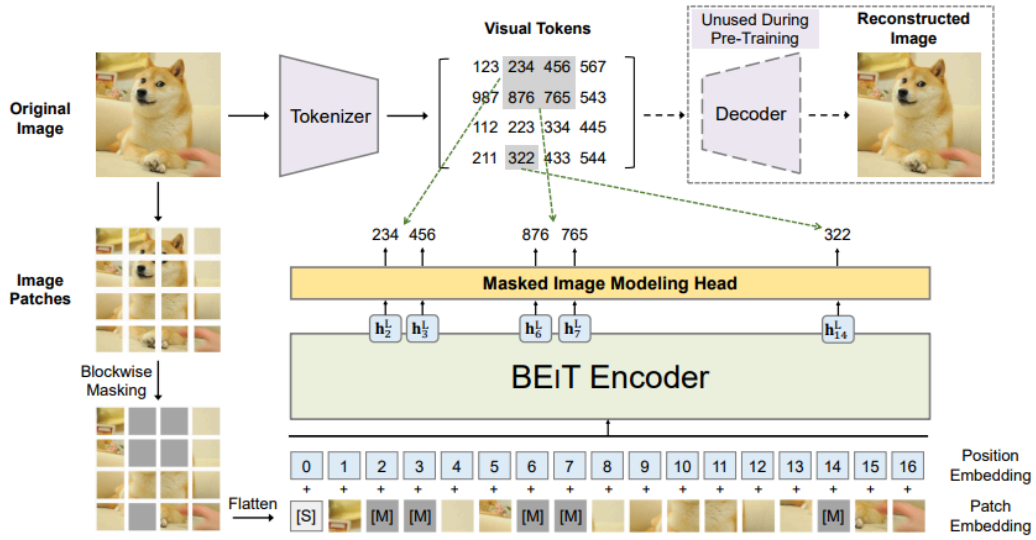
2.1. Giới thiệu

Mô hình BEIT *(Bidirectional Encoder representation from Image Transformers)* là một mô hình biểu diễn thị giác tự giám sát *(Self-supervised)* được thiết kế dựa trên cơ sở kiến trúc BERT *(Bidirectional Encoder Representation from Transformers)* - sử dụng kiến trúc mã hóa hai chiều từ Transformer. Mô hình BEIT nhằm tạo ra biểu diễn đa chiều cho hình ảnh, cho phép hiểu và biểu diễn thông tin từ cả hai hướng.

2.2. Kiến trúc mô hình

BEIT sử dụng kiến trúc Transformer, bao gồm nhiều lớp Encoder và Decoder. Lớp Encoder trong BEIT chịu trách nhiệm biểu diễn hình ảnh và trích xuất các đặc trưng quan trọng. Lớp Decoder được sử dụng để tạo ra biểu diễn đa chiều từ các đặc trưng được trích xuất. Lấy cảm hứng từ BERT, BEIT sử dụng một phương pháp tiền huấn luyện mới, gọi là mô hình

hóa ảnh bị che khuất (Masked Image Modeling - MIM). Trong MIM, một phần của hình ảnh được che đi và mô hình cố gắng dự đoán nội dung của phần che.



Hình 1: Quá trình tiền huấn luyện của BEIT

Trước tiên, chúng ta thực hiện quá trình gọi là "tái tạo tự mã hóa," nơi một hình ảnh được chia thành các biểu tượng thị giác (visual tokens) riêng lẻ dựa trên learned vocabulary. Mục đích là tạo ra một bộ mã hóa có khả năng biểu diễn hình ảnh theo cách hiểu được bởi mô hình.

Trong quá trình tiền huấn luyện, mỗi hình ảnh được xem xét từ hai góc độ khác nhau: một là thông qua các đoạn ảnh (image patches) và hai là thông qua biểu tượng thị giác (visual tokens). Chúng ta ngẫu nhiên che đi một phần nhất định của các đoạn ảnh trong hình ảnh (những đoạn ảnh màu xám trong hình 1) và thay thế chúng bằng một biểu tượng đặc biệt được ký hiệu là **[M]**. Điều này tạo ra một dạng hình ảnh "bị hỏng" hay bị che phủ. Các đoạn ảnh đã được xử lý được đưa vào một mô hình cốt lõi của thị giác, có nhiệm vụ tạo ra các vector mã hóa biểu diễn cho hình ảnh. Mục tiêu cuối cùng của quá trình tiền huấn luyện là dự đoán lại các biểu tượng thị giác của hình ảnh gốc dựa trên các vector mã hóa của hình ảnh đã bị "hỏng.

Sau khi đã được tiền huấn luyện, BEIT có thể được sử dụng cho các công việc cụ thể như phân loại hình ảnh và phân đoạn ngữ nghĩa. Để làm điều này, chúng ta thêm các lớp công việc cụ thể lên trên BEIT đã được tiền huấn luyện và điều chỉnh các tham số trên các bộ dữ liệu cụ thể của nhiệm vụ đó.

Kết quả thực nghiệm cho thấy rằng BEIT đóng một vai trò quan trọng đối với hiệu quả của việc tiền huấn luyện theo kiểu BERT cho dữ liệu hình ảnh. Ngoài hiệu suất, cải thiện về tốc độ hội tụ và sự ổn định của việc điều chỉnh giảm chi phí đào tạo cho các nhiệm vụ cuối cùng.

2.2. Ứng dụng của BEIT

BEIT được thiết kế đặc biệt cho việc xử lý hình ảnh và thị giác máy tính. Một số ứng dụng của mô hình BEiT như:

- **Phân loại Ảnh:** BEiT có thể được sử dụng để phân loại các đối tượng trong ảnh với độ chính xác cao, nhờ khả năng học biểu diễn tự giám sát và khả năng hiểu bối cảnh của nó.
- **Giảm chiều dữ liệu (Dimensionality Reduction):** Mô hình BEiT có khả năng giảm chiều dữ liệu mà vẫn giữ được thông tin quan trọng, giúp tăng tốc quá trình xử lý và giảm bộ nhớ yêu cầu.
- **Phân loại Video:** BEiT có thể được mở rộng để phân loại và hiểu nội dung video, từ đó hỗ trợ ứng dụng trong lĩnh vực nhận diện hành động và phân tích video tự động.
- **Ứng dụng trong Y học:** BEiT có thể hỗ trợ trong việc phân loại và đánh giá hình ảnh y tế, giúp chẩn đoán và theo dõi các bệnh lý.
- **Tìm kiếm ảnh:** BEiT có thể được tích hợp vào hệ thống tìm kiếm ảnh để cải thiện khả năng tìm kiếm dựa trên nội dung hình ảnh.
- **Xử lý Ngôn ngữ Tự nhiên và Hình ảnh Kết hợp:** BEiT có thể được sử dụng trong các ứng dụng yêu cầu kết hợp thông tin từ cả ngôn ngữ tự nhiên và hình ảnh, như trong các dự án trí tuệ nhân tạo đa modality.

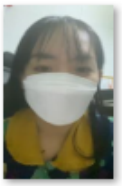
3. Xây dựng mô hình Liveness Detection sử dụng BEIT

3.1. Chuẩn bị Dữ Liệu

Bộ dữ liệu được sử dụng cho quá trình huấn luyện là Liveness Detection của Zalo AI Challenge 2022 được thiết kế để đánh giá khả năng của các mô hình nhận diện sự sống (liveness detection) trong ứng dụng nhận diện khuôn mặt. Bao gồm:

- 1168 video khuôn mặt đeo khẩu trang, với 598 video có nhãn là khuôn mặt thật (Real) and 570 video có nhãn là khuôn mặt giả mạo (Fake).
- File nhãn (Label.CSV), trong đó mỗi hàng cung cấp tên video và nhãn của video đó (0 = Fake, 1 = Real).


1.mp4


2.mp4


14.mp4


18.mp4

Các bước xử lý dữ liệu:

- **Xử lý dữ liệu hình ảnh:** Đối với video, ta trích xuất các khung hình chính xác để giảm kích thước dữ liệu.
- **Chia tập dữ liệu:** Phân chia tập dữ liệu thành tập huấn luyện, tập validation để đánh giá hiệu suất mô hình.

3.2. Huấn Luyện Mô Hình

Chuẩn bị môi trường:

- Để huấn luyện mô hình, ta sử dụng mã nguồn từ GitHub repository của Microsoft với phiên bản cụ thể.

Thiết lập tham số huấn luyện:

- Số lượng lớp *(NUM_CLASSES)* được đặt là **2** cho bài toán phân loại liveness (real và fake).
- Sử dụng mô hình được fine-tune từ một checkpoint trước đó.

Huấn luyện mô hình:

- Sử dụng mô hình cơ sở BEiT với kiến trúc ***beit_base_patch16_224*** với kích thước patch 16x16 và kích thước ảnh đầu vào là 224x224.
- Batch size được đặt là 64, learning rate là 2e-3, và các tham số khác được xác định.
- Kết quả của quá trình huấn luyện và đánh giá được ghi vào tệp out.log.

Quá trình huấn luyện diễn ra trên 2 epochs với sự sử dụng của deepspeed để tối ưu hóa hiệu suất. Thông qua quá trình fine-tune, mô hình cố gắng học biểu diễn đặc trưng cho việc phân loại liveness video.

3.3. Load Mô Hình để Tinh Chỉnh và Dự Đoán

Load Mô Hình và Tinh Chỉnh:

Quá trình load mô hình được thực hiện sao cho đảm bảo rằng trọng số từ checkpoint được tích hợp chính xác vào mô hình BEiT. Sử dụng argparse để tạo các tham số như tên mô hình, kích thước ảnh đầu vào, số lớp, và đường dẫn fine-tune. Nếu có yêu cầu fine-tune, tiến hành load trọng số từ checkpoint đã được huấn luyện trước đó.
Xử lý các khóa trọng số không khớp giữa mô hình hiện tại và checkpoint. Sau khi đã điều chỉnh và load đầy đủ các trọng số, mô hình được chuyển lên GPU để thực hiện các phép tính nhanh chóng.

Sau khi load mô hình và tinh chỉnh, ta wrap up lại mô hình để chạy dự đoán trên tập dữ liệu validation. Bằng cách wrap up mô hình vào một lớp riêng, ta có thể dễ dàng thực hiện bất kỳ xử lý nào trước khi đưa dữ liệu vào mô hình. Sử dụng hàm khởi tạo *__init__*, mô hình nhận một mô hình con đã được huấn luyện *(model_s)* và chuyển nó sang trạng thái đánh giá để không tính toán gradient trong quá trình dự đoán.

Dự Đoán Với Mô Hình:

Sử dụng hàm forward, ảnh đầu vào được di chuyển đến GPU và được mở rộng thành một tensor 4 chiều. Sử dụng mô hình con *(model_s)* để tính toán logits và sau đó áp dụng *softmax* để lấy xác suất của các lớp. Kết quả được trả về dưới dạng một numpy array.

Dự Đoán Trên Tập Dữ Liệu:

Sử dụng *torchvision.datasets.ImageFolder* để tải tập validation. Dùng một vòng lặp để dự đoán xác suất cho từng hình ảnh trong tập dữ liệu. Lưu trữ xác suất dự đoán, hình ảnh và nhãn thực tế vào các danh sách tương ứng.

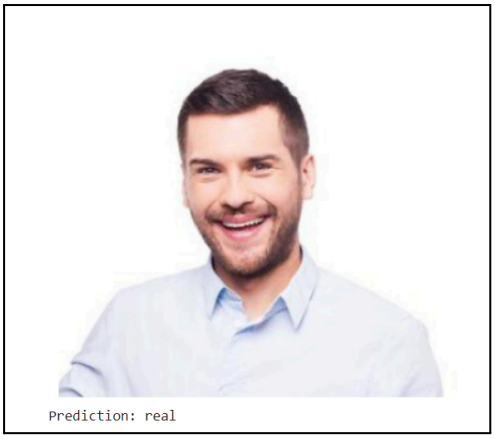
Đánh Giá Kết Quả:

Tính toán độ chính xác trên tập dữ liệu bằng cách so sánh nhãn dự đoán với nhãn thực tế và tính trung bình của các dự đoán chính xác.

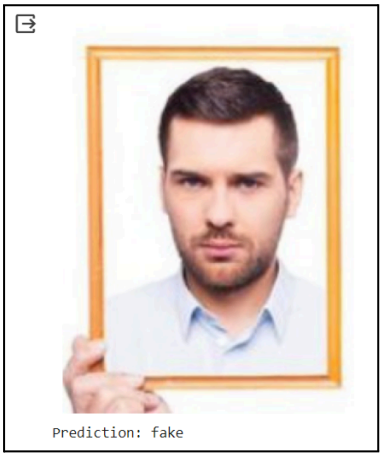
3.4. Nhận xét Và Hướng Phát Triển

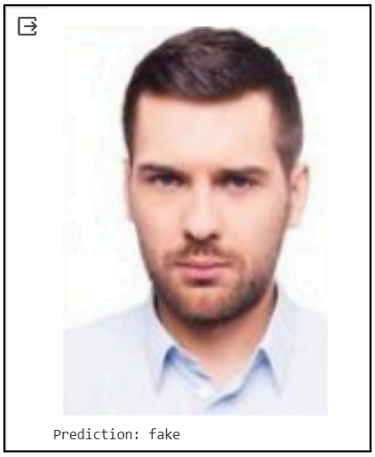
Kết quả nhận được sau khi huấn luyện mô hình có độ chính xác 93,92%. Khi kiểm tra trên một hình ảnh bất kỳ ta nhận được kết quả khá khả quan. Một ví dụ như sau:

Hình trên gồm hai đối tượng thật và giả. Khi kiểm tra với đối tượng thật, mô hình cho ra kết quả chính xác.



Khi kiểm tra với đối tượng là giả, mô hình cho ra kết quả đúng.





Kể cả khi cắt bỏ phần viền của đối tượng giả, mô hình vẫn cho ra kết quả đúng.

Độ chính xác khá cao trên tập dữ liệu validation cho thấy mô hình có khả năng phân loại một cách hiệu quả giữa video chính thống và video giả mạo, đạt được mức độ tin cậy đáng kể. Trong tương lai, để nâng cao hiệu suất có thể cần mở rộng tập dữ liệu để bao gồm nhiều điều kiện chiếu sáng, góc chụp, và loại giả mạo khác nhau.