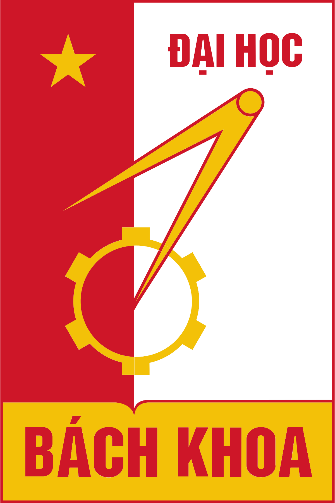
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**



**Báo cáo môn Project I**

**Ứng dụng Machine Learning trong phát hiện ảnh deepfake**

**Thành viên trong nhóm:**

|  |  |  |
| --- | --- | --- |
| Họ và tên thành viên | MSSV | Email |
| **Nguyễn Duy Khánh Linh** | 20204839 | [Linh.ndk204839@sis.hust.edu.vn](mailto:Linh.ndk204839@sis.hust.edu.vn) |

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | PGS.TS Lê Thanh Hương |
| **Bộ môn:**  **Mã lớp:** | Project I |
| **Trường:** | CNTT và TT. |

Mục lục

[**Lời cảm ơn** 3](#_Toc157583486)

[**Chương 1: Giới thiệu đề tài.** 4](#_Toc157583487)

[1.1 Đặt vấn đề: 4](#_Toc157583488)

[1.2 Bố cục đề tài: 4](#_Toc157583489)

[**Chương 2: Công nghệ tạo ra deepfake** 6](#_Toc157583490)

[2.1 Các thuật toán tạo ra deepfake: 6](#_Toc157583491)

[2.2 Quy trình chung tạo ra deepfake: 6](#_Toc157583492)

[**Chương 3: Phát hiện Deepfake** 8](#_Toc157583493)

[3.1 Cốt lõi của phương pháp: 8](#_Toc157583494)

[3.2 Bộ dữ liệu huấn luyện: 8](#_Toc157583495)

[3.3 Ưu điểm: 8](#_Toc157583496)

[3.4 Quá trình tạo ra negative sample: 8](#_Toc157583497)

[**Chương 4: Mô hình VGG16** 10](#_Toc157583498)

[4.1 Tổng quan về mô hình CNN: 10](#_Toc157583499)

[4.2 Lớp tích chập (Convolutional) 10](#_Toc157583500)

[4.3 Hàm kích hoạt phi tuyến ReLU: 11](#_Toc157583501)

[4.4 Lớp rút gọn (Pooling) 12](#_Toc157583502)

[4.5 Lớp kết nối đầy đủ (Fully connected) 12](#_Toc157583503)

[4.6 Hàm softmax: 13](#_Toc157583504)

[4.7 Cấu trúc của mạng VGG-16: 13](#_Toc157583505)

[**Chương 5: Huấn luyện Model** 15](#_Toc157583506)

[5.1 Dữ liệu huấn luyện: 15](#_Toc157583507)

[5.2 Hàm mất mát (Loss function): 15](#_Toc157583508)

[5.3 Bộ tối ưu hóa SGD: 15](#_Toc157583509)

[**Chương 6: Đánh giá mô hình** 17](#_Toc157583510)

[**Chương 7: Tài liệu tham khảo** 18](#_Toc157583511)

**Lời cảm ơn**

Trước hết, em xin gửi lời cảm ơn đến người hướng dẫn dự án của em, cô Lê Thanh Hương, vì sự hướng dẫn, hỗ trợ và kiến thức chuyên môn quý báu của cô trong suốt quá trình thực hiện dự án. Kiến thức sâu rộng và sự hướng dẫn của cô đã đóng một vai trò quan trọng trong định hình dự án của em và giúp em vượt qua những thách thức. Từ đó, học hỏi và rút kinh nghiệm cho các dự án tiếp theo.

**Chương 1: Giới thiệu đề tài.**

1. Đặt vấn đề:

Trong những năm gần đây, công nghệ Deepfake đã thu hút được sự quan tâm của cộng đồng nghiên cứu và người dùng trên toàn thế giới. Deepfake sử dụng trí tuệ nhân tạo và kỹ thuật học sâu để tạo ra những nội dung giả mạo. Ban đầu, công nghệ này được phát triển với mục tiêu giải trí và sáng tạo, tạo ra những nội dung giả hài hước hoặc kỹ xảo trong ngành điện ảnh. Tuy nhiên, việc sử dụng deepfake có thể dẫn đến những hệ quả tiêu cực.

Trong một thế giới nơi video và hình ảnh có thể được chỉnh sửa một cách tinh vi và khó phân biệt với nội dung thật, việc xác định sự chính xác và tin cậy của thông tin trở nên khó khăn. Những video và hình ảnh deepfake có thể được sử dụng để lan truyền thông tin sai lệch, gây ảnh hưởng tiêu cực đến xã hội và cá nhân.

Nhận thức về những tác hại tiềm ẩn của deepfake, tôi đã quyết định thực hiện đề tài "Phát hiện deepfake" dưới sự hướng dẫn của cô Lê Thanh Hương. Đề tài của tôi cũng sẽ tập trung vào việc phát triển một mạng CNN có thể phát hiện deepfake. Mạng này có mục tiêu giúp người dùng nhận biết và phát hiện các nội dung deepfake, đảm bảo tính xác thực và tin cậy của hình ảnh. Tôi hy vọng rằng qua việc nghiên cứu và phát triển ứng dụng này, tôi có thể đóng góp vào việc giảm thiểu tác động tiêu cực của deepfake và bảo vệ người dùng khỏi sự lừa đảo và xâm phạm quyền riêng tư.

1. Bố cục đề tài:

Trong báo cáo này, em sẽ chia thành 7 chương, ứng với 7 nội dung khác nhau.

Chương 2: Công nghệ tạo ra deepfake

Chương 2 sẽ giới thiệu tổng quan cách tạo ra một hình ảnh giả mạo, làm cơ sở để hiểu về phương pháp phát hiện deepfake.

Chương 3: Phát hiện Deepfake

Trong chương này, em tập trung vào việc phát hiện deepfake bao gồm giải thích các phương pháp và đánh giá ưu điểm của phương pháp.

Chương 4: Mô hình VGG16

Chương này giới thiệu mô hình VGG16, mạng nerual tích chập nổi tiếng trong lĩnh vực xử lý hình ảnh. Báo cáo giải thích kiến trúc của VGG16 và cách nó được sử dụng để trích xuất đặc trưng hình ảnh.

Chương 5: Huấn luyện Model

Chương 5 trình bày về quá trình huấn luyện mô hình để phát hiện deepfake. Tôi giới thiệu về việc chuẩn bị dữ liệu huấn luyện, bao gồm việc xử lý và tiền xử lý dữ liệu. Chúng tôi trình bày cách xây dựng mô hình học máy, bao gồm các tham số cần thiết để huấn luyện mô hình.

Chương 6: Đánh giá mô hình

Dựa trên mô hình được huấn luyện, tiến hành đánh giá độ chính xác và mất mát của mô hình bằng bộ dữ liệu khác nhau.

Chương 7: Tài liệu tham khảo

Trong chương này, em đề cập đến các tài liệu và nguồn tham khảo quan trọng mà em đã sử dụng trong quá trình nghiên cứu và làm đồ án.

**Chương 2: Công nghệ tạo ra deepfake**

1. Các thuật toán tạo ra deepfake:

Dựa vào trên mục đích và phương pháp thực hiện, các thuật toán tạo ra deepfake ảnh được chia thành hai loại: hoán đổi gương mặt (Face-swapping) và tái tạo gương mặt (Face-reenactment).

Hoán đổi gương mặt nhắm tới việc hoán đổi danh tính của người trong ảnh. Còn thuật toán tái hiện khuôn mặt lại cố gắng kiểm soát biểu cảm của mọi người trong ảnh để tạo ra ảnh về ai đó làm điều gì đó không tồn tại.

1. Quy trình chung tạo ra deepfake:

Để tạo deepfake ảnh thông thường sẽ có 5 bước, bao gồm: phát hiện vùng khuôn mặt (Face area detection), trích xuất các đặc điểm khuôn mặt (Landmarks extraction), áp dụng các thuật toán deepfake (Face alignment), biến đổi affine (Affine warp), xử lý hậu kỳ (Post-processing).

A collage of a person's face

Description automatically generated

* Phát hiện vùng khuôn mặt: Chúng ta cần phát hiện vùng khuôn mặt trong mỗi ảnh. Điều này giúp xác định vị trí và kích thước của khuôn mặt trong ảnh.
* Trích xuất đặc điểm khuôn mặt: Sau khi phát hiện được khuôn mặt, chúng ta trích xuất các đặc điểm quan trọng của khuôn mặt, chẳng hạn như mắt, mũi, miệng. Điều này giúp chúng ta xác định các phần khác nhau trong khuôn mặt để thực hiện quá trình đổi khuôn mặt.
* Áp dụng thuật toán deepfake: Bộ mã hóa tự động bao gồm hai thành phần: bộ mã hóa (encoder) và bộ giải mã (decoder). Bộ mã hóa giúp trích xuất các đặc trưng chung trong khuôn mặt nguồn và khuôn mặt mục tiêu. Để thực hiện tái hiện khuôn mặt, các thông số nhận dạng và chiếu sáng cảnh được giữ nguyên trong khi các thông số về tư thế đầu, biểu cảm và ánh mắt được thay đổi. Sau đó, hình ảnh tổng hợp của tác nhân mục tiêu được tái tạo dựa trên các tham số đã sửa đổi.

A diagram of a computer process

Description automatically generated

* Biến đổi Affine: Để giảm nhiễu và làm cho kết quả tổng hợp trông tự nhiên hơn, chúng ta sẽ sử dụng các điểm mốc của lông mày trái, phải và miệng dưới để xác định vùng cần được giữ lại. Mặt nạ này chỉ giữ lại nội dung bên trong và loại bỏ phần không cần thiết. Sử dụng các phép biến đổi affine để điều chỉnh hướng đầu, biểu cảm và các yếu tố khác của khuôn mặt tổng hợp để phù hợp với ảnh gốc. Kết hợp với hình ảnh gốc bằng cách chỉ giữ lại phần bên trong của vùng đã xác định và loại bỏ, hoặc làm mờ các vùng còn lại.
* Xử lý hậu xử lý: Cuối cùng, chúng ta áp dụng các phép biến đổi hậu xử lý để làm cho ảnh đổi khuôn mặt trông thực tế hơn. Các phép biến đổi này có thể bao gồm mờ Gaussian để làm mịn các biên của mặt nạ và điều chỉnh màu sắc để đảm bảo sự nhất quán giữa khuôn mặt tổng hợp và hình ảnh gốc.

**Chương 3: Phát hiện Deepfake**

Trong báo cáo này, em tập trung tìm hiểu về phương pháp dựa trên các hiện tượng hình ảnh (Visual artifacts-based methods), cụ thể là phương pháp dựa vào hiện tượng cong vênh gương mặt (Face warping artefacts) theo nghiên cứu của tác giả Li, Y., Lyu, S.

1. Cốt lõi của phương pháp:

Do hạn chế về tài nguyên tính toán và thời gian sản xuất, thuật toán DeepFake chỉ có thể tổng hợp hình ảnh khuôn mặt có kích thước cố định và chúng phải trải qua một quá trình biến đổi affine để phù hợp với cấu hình khuôn mặt nguồn. Quá trình biến đổi này để lại các dấu vết đặc trưng do không nhất quán về độ phân giải giữa khu vực khuôn mặt bị biến đổi và bối cảnh xung quanh. Do đó, các dấu vết này có thể được sử dụng để phát hiện ảnh DeepFake bằng cách sử dụng một mạng neural CNN đặc biệt.

1. Bộ dữ liệu huấn luyện:

Trong quá trình huấn luyện, cần có một tập dữ liệu gồm các mẫu không phải deepfake (positive samples) và các mẫu deepfake (negative samples). Các mẫu deepfake được tạo ra bằng cách sử dụng các biến đổi affine và làm mờ Gaussian. Các mẫu không phải deepfake thường được lựa chọn từ các nguồn dữ liệu đáng tin cậy, chẳng hạn như các hình ảnh từ các nguồn tin tức, phim, hoặc các video được tạo ra bởi người dùng thông thường.

Mô hình phát hiện deepfake được huấn luyện bằng cách sử dụng cả các mẫu deepfake và mẫu không phải deepfake. Quá trình huấn luyện sẽ giúp mô hình học cách nhận biết các dấu hiệu và biểu hiện không tự nhiên của deepfake. Bằng cách so sánh các ảnh kiểm tra với mô hình đã được huấn luyện, mô hình có thể đưa ra dự đoán xem ảnh đó có phải là deepfake hay không.

1. Ưu điểm:

So với các phương pháp trước đây cần sử dụng một lượng lớn ảnh thật và ảnh DeepFake được tạo ra để huấn luyện bộ phân loại CNN, phương pháp này nhắm đến các dấu vết trong biến đổi affine trên khuôn mặt là đặc trưng đặc biệt để phân biệt ảnh thật và giả tạo. Những dấu vết như vậy có thể được mô phỏng trực tiếp bằng các hoạt động xử lý ảnh đơn giản trên một hình ảnh để biến nó thành negative sample. Vì vậy, phương pháp này sẽ tiết kiệm được rất nhiều thời gian và tài nguyên cho việc huấn luyện một mô hình DeepFake để tạo ra các negative sample tốn thời gian và tài nguyên.

Thêm vào đó, vì những dấu vết như vậy tồn tại chung trong các ảnh DeepFake từ các nguồn khác nhau, phương pháp này có tính ổn định hơn so với các phương pháp khác.

1. Quá trình tạo ra negative sample:

Việc huấn luyện mô hình CNN dựa trên các hình ảnh khuôn mặt được thu thập từ Internet. Trong khi các positive sample được lấy từ các nguồn uy tín thì các negative sample sẽ được tạo ra thông qua quá trình mô phỏng về độ phân giải không nhất quán trong các biến đổi affine trên khuôn mặt thật.

A person's face with different facial features

Description automatically generated

Quá trình gồm bước:

* Phát hiện khuôn mặt trong các hình ảnh gốc và trích xuất vùng khuôn mặt
* Căn chỉnh khuôn mặt vào nhiều tỷ lệ và chọn một tỷ lệ ngẫu nhiên, sau đó làm mờ bằng Gaussian blur với kích thước kernel (5 × 5). Quá trình này nhằm tạo ra nhiều trường hợp độ phân giải khác nhau trong các khuôn mặt bị biến dạng affine, từ đó có thể giảm độ nét, độ nhiễu của ảnh, mô phỏng tốt hơn các loại không đồng nhất về độ phân giải khác nhau được giới thiệu trong quá trình biến dạng khuôn mặt affine.
* Khuôn mặt đã được làm mờ trải qua một phép biến dạng affine trở lại cùng kích thước của các khuôn mặt gốc để mô phỏng các hiện tượng trong quy trình sản xuất DeepFake.

Ngoài ra, hình dạng của khu vực khuôn mặt bị biến dạng affine có thể được xử lý tiếp theo dựa trên các điểm định vị khuôn mặt. Hình 3(d) chỉ ra rằng một hình dạng đa giác lồi được tạo ra dựa trên các điểm định vị của lông mày và đáy miệng.

A collage of a person's face

Description automatically generated

**Chương 4: Mô hình VGG16**

Mô hình VGG16 là một kiến trúc mạng neural tích chập (CNN) nổi tiếng được giới thiệu bởi nhóm nghiên cứu Visual Geometry Group (VGG) tại Đại học Oxford. VGG-16 được công bố năm 2014, phổ biến trong lĩnh vực xử lý hình ảnh, được sử dụng trong nhiều ứng dụng như phân loại hình ảnh, trích xuất đặc trưng, transfer learning, nhận diện đối tượng và xử lý ảnh y sinh. Trong thực tế mô hình VGG-16 đã đạt được thành công đáng kể trong các nhiệm vụ phân loại hình ảnh.

1. Tổng quan về mô hình CNN:

Mô hình CNN, viết tắt của Convolution Neural Network – mạng neural tích chập, là mô hình mạng Deep learning phổ biến, ứng dụng trong nhận dạng và phân loại hình ảnh với độ chính xác cao.

Trong lĩnh vực xử lý ảnh, input của mô hình CNN thường là hình ảnh kích thước cố định 224 x 224 với ba kênh màu - R, G và B. Ảnh đầu vào sẽ được tiến hành chuẩn hóa các giá trị RGB cho mỗi pixel. Điều này đạt được bằng cách trừ đi giá trị trung bình từ mỗi pixel.

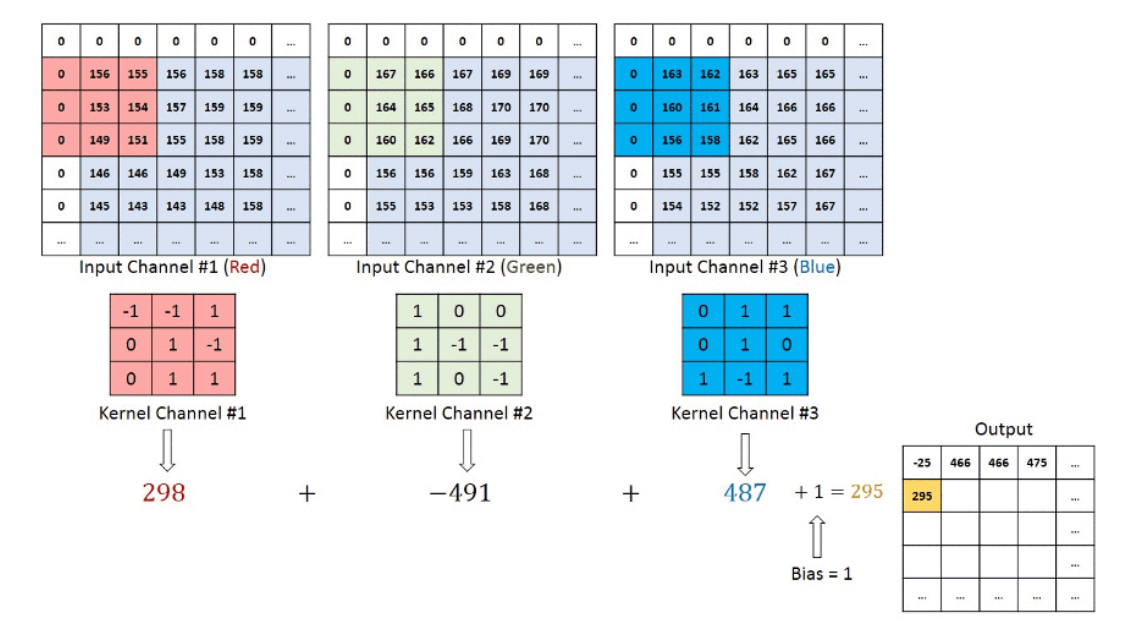
Các lớp cơ bản của một mô hình CNN bao gồm: lớp tích chập (Convolutional); lớp lấy mẫu (Pooling); lớp kết nối đầy đủ (Fully connected)

1. Lớp tích chập (Convolutional)

Đây là thành phần quan trọng nhất trong mạng CNN, thể hiện sự liên kết cục bộ giữa các điểm ảnh. Các liên kết này được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc filters có kích thước tương đương.

A diagram of a diagram of a number

Description automatically generated with medium confidence



Một số yếu tố quan trong trong lớp tích chập:

* Stride: sự dịch chuyển của bộ lọc khi quét qua dữ liệu đầu vào, đại diện cho số pixel bị bỏ qua giữa các vị trí quét.
* Padding: quá trình thêm các giá trị 0 vào cạnh dữ liệu đầu vào trước khi áp dụng bộ lọc. Quá trình này giúp giữ nguyên kích thước dữ liệu đầu vào sau khi áp dụng phép tích chập

A close-up of several images

Description automatically generated

Ở những Convolutional Layers đầu tiên, các bộ lọc phát hiện được các chi tiết chung dưới dạng các nét ngang, dọc và các cạnh của ảnh. Đây là những đặc trưng bậc thấp (low level feature) và khá chung chung. Chúng ta chưa thể nhận biết được vật thể dựa trên những đường nét này. Ở những Convolutional Layers cuối cùng là những đặc trưng bậc cao (high level feature) được tổng hợp từ đặc trưng bậc thấp. Đây là những đặc trưng tốt và có sức mạnh phân loại các classes.

1. Hàm kích hoạt phi tuyến ReLU:

Hàm ReLU (Rectified Linear Unit), với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự, trừ các giá trị âm đã bị loại bỏ.

𝑓(𝑥) = 𝑚𝑎𝑥(0, 𝑥)

Việc áp dụng hàm ReLU đã được quan sát là thúc đẩy quá trình hội tụ nhanh hơn trong quá trình huấn luyện. Điều này có thể do ReLU không bị bão hoà ở 2 đầu như Sigmoid và Tanh. Ngoài ra, hàm ReLU tính toán nhanh hơn. Tanh và Sigmoid sử dụng hàm exp và công thức phức tạp hơn ReLU rất nhiều do vậy sẽ tốn nhiều chi phí hơn để tính toán.

Khi giá trị đầu vào của một node là âm, hàm ReLU chuyển đầu ra thành 0. Điều này có nghĩa là nút đó không có đóng góp vào quá trình học và truyền ngược gradient. Nếu một số nút hoặc một phần lớn mạng neural bị "kích hoạt" thành 0, nghĩa là chúng không hoạt động, điều này có thể làm giảm khả năng học và hiệu suất của mạng.

1. Lớp rút gọn (Pooling)

Các tầng càng về sau khi trích xuất đặc trưng sẽ cần số lượng tham số lớn do chiều sâu được qui định bởi số lượng các kênh ở các tầng sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng tham số và khối lượng tính toán trong mạng neural. Do đó để giảm tải tính toán chúng ta sẽ cần giảm kích thước các chiều của khối ma trận đầu vào hoặc giảm số đơn vị của tầng. Vì mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng 1 bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là rút gọn nhằm mục đích giảm kích thước dài, rộng.

Lớp rút gọn cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh như lớp tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập, sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hai phương thức lấy mẫu phổ biến:

* Max Pooling : lấy giá trị điểm ảnh lớn nhất
* Avarage Pooling: lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ

A diagram of a pool

Description automatically generated

1. Lớp kết nối đầy đủ (Fully connected)

A diagram of a network

Description automatically generated

Đầu tiên, ma trận đầu vào sẽ được tiến hành trải phẳng (flatten) trở thành một vector. Các nút trong lớp fully connected sẽ được kết nối với mọi phần tử trong vector, đầu ra bằng tổng các phần tử nhân với trọng số tương ứng. Các nút trong lớp fully connected tiếp theo lại được kết nối với mọi nút của lớp trước đó. Đầu ra được xác định bằng cách nhân các giá trị nút đầu vào với một ma trận trọng số tương ứng, sử dụng hàm ReLU để loại bỏ các giá trị âm.

1. Hàm softmax:

Cuối cùng, hàm softmax được áp dụng để phân loại đa lớp. Hàm softmax chuyển đổi các giá trị đầu vào thành xác suất bằng cách lũy thừa chúng và chuẩn hóa kết quả.

Công thức tính phân phối xác xuất trong hàm softmax:

* Tính giá trị exponentials (hàm lũy thừa cơ số e) cho từng thành phần của đầu vào:

exp\_i = exp(z\_i)

* Tính tổng các giá trị exponentials:

sum\_exp = sum(exp\_i) for all i

* Tính giá trị đầu ra softmax cho mỗi thành phần:

softmax\_i = exp\_i / sum\_exp for all i

Ưu điểm của hàm softmax:

* Hiểu được dưới dạng xác suất: Hàm softmax tạo ra các đầu ra có thể được hiểu trực tiếp là xác suất của các lớp. Điều này làm cho nó phù hợp cho các nhiệm vụ phân loại đa lớp, nơi mục tiêu là gán xác suất cho mỗi lớp.
* Chuẩn hóa đầu ra: Hàm softmax chuẩn hóa các xác suất đầu ra, đảm bảo tổng của chúng là 1. Tính chuẩn hóa này cho phép so sánh và xếp hạng trực tiếp các xác suất dự đoán giữa các lớp khác nhau.
* Đặc tính đạo hàm của softmax: Hàm softmax có đạo hàm được xác định rõ ràng và tính toán hiệu quả, điều này rất quan trọng trong quá trình huấn luyện. Đạo hàm có thể được tính toán hiệu quả trong quá trình lan truyền ngược, giúp tối ưu hóa hiệu quả các tham số mạng.

1. Cấu trúc của mạng VGG-16:

A diagram of a graph

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| Trong mô hình VGG16, có tổng cộng 13 lớp tích chập (convolutional), 4 lớp rút gọn (pooling), và 3 lớp kết nối đầy đủ (fully connected).  Các lớp tích chập có các kernel khác nhau nhưng kích thước của kernel đều là 3x3. Stride và padding bằng 1 pixel để độ phân giải được duy trì và kích thước đầu ra bằng kích thước đầu vào.  Các lớp rút gọn sử dụng phương pháp Max pooling (lấy giá trị pixel lớn nhất) trên cửa sổ pixel 2x2 với stride bằng 2 pixel. Điều này sẽ làm giảm một nửa kích thước đầu vào.  Sau lớp pooling cuối cùng, ma trận đầu vào sẽ được tiến hành trải phẳng (flatten) trở thành một vector, làm đầu vào cho các lớp kết nối đầy đủ tiếp theo. Hai lớp đầu tiên có mỗi lớp 4.096 đơn vị (n), và lớp kết nối đầy đủ cuối cùng được sử dụng làm lớp đầu ra và có 1.000 đơn vị tương ứng với 1.000 lớp khả thi cho tập dữ liệu ImageNet. Lớp đầu ra được tiếp theo bởi hàm kích hoạt Softmax được sử dụng cho phân loại hạng mục. Ở đây, output của mô hình bao gồm 2 đơn vị, tương ứng với positive sample và negative sample. |  |

**Chương 5: Huấn luyện Model**

1. Dữ liệu huấn luyện:

Mô hình này sử dụng bộ dữ liệu LFW. Labeled Faces in the Wild là một cơ sở dữ liệu chứa các bức ảnh khuôn mặt được thiết kế để nghiên cứu vấn đề nhận dạng khuôn mặt. Cơ sở dữ liệu này được tạo ra và duy trì bởi các nhà nghiên cứu tại Đại học Massachusetts, Amherst). Bộ dữ liệu có tổng cộng 13.233 hình ảnh của 5.749 người. Trong số những người được chụp hình, có 1.680 người có hai hoặc nhiều hơn hai bức ảnh khác nhau trong cơ sở dữ liệu.

Các bức ảnh trong bộ dữ liệu sẽ được sử dụng như các positive sample (các mẫu không phải deepfake) và các negative sample (các mẫu deepfake) sẽ được tạo ra thông qua quá trình mô phỏng về độ phân giải không nhất quán trong các biến đổi affine trên khuôn mặt thật ở mục 3.4. Tuy nhiên, thay vì tạo ra tất cả các negative sample từ trước quá trình huấn luyện, thì mô hình này sẽ sử dụng phương pháp động để tạo negative sample ngay trong quá trình huấn luyện. Cụ thể, trong mỗi batch huấn luyện, một nửa số positive sample được chọn ngẫu nhiên và chuyển đổi thành negative sample. Như vậy, dữ liệu huấn luyện sẽ đa dạng hơn.

1. Hàm mất mát (Loss function):

Hàm mất mát sẽ đo lường sự khác biệt giữa đầu ra dự đoán của mô hình và giá trị đúng tương ứng với dữ liệu huấn luyện. Từ đó, xác định cách đánh giá và tối ưu hóa mô hình. Do vậy, việc chọn hàm mất mát phù hợp là rất quan trọng.

Mô hình này sử dụng hàm mất mát Categorical Crossentropy. Đây là loss function phổ biến được sử dụng trong bài toán phân loại đa lớp. Nó đo lường sự khác biệt giữa phân phối xác suất dự đoán của mô hình và phân phối xác suất thực tế (nhãn đúng).

Công thức chung của hàm mất mát Categorical Crossentropy giữa hai phân phối xác suất P và Q được tính như sau:

Trong đó:

* C đại diện cho số lớp đầu ra,
* là xác suất thực tế (0 hoặc 1) đối với lớp i,
* là xác suất dự đoán của lớp i

Ưu điểm của hàm Categorical Crossentropy rất nhạy cảm với sự sai khác giữa và Nghĩa là nếu xác suất dự đoán gần với 1 cho lớp đúng và gần với 0 cho các lớp không đúng, hàm mất mát sẽ có giá trị nhỏ. Ngược lại, xác suất dự đoán và nhãn thực tế càng khác nhau, hàm mất mát sẽ có giá trị lớn.

1. Bộ tối ưu hóa SGD:

Stochastic Gradient Descent (SGD) là một phương pháp tối ưu hóa phổ biến được sử dụng trong quá trình huấn luyện mạng neural. Nó là một biến thể của Gradient Descent (GD), được thiết kế để tối thiểu hóa hàm mất mát bằng cách điều chỉnh các trọng số của mô hình dựa trên đạo hàm của hàm mất mát.

Trong SGD, quá trình cập nhật các trọng số của mạng neural được thực hiện cho từng ví dụ huấn luyện một cách ngẫu nhiên. Thay vì tính toán gradient trên toàn bộ tập dữ liệu huấn luyện (sau mỗi epoch như trong GD), SGD chỉ tính toán gradient trên một ví dụ huấn luyện duy nhất hoặc một mini-batch (một tập con nhỏ của tập dữ liệu huấn luyện). Do đó, SGD có thể xử lý các tập dữ liệu lớn một cách hiệu quả hơn trong khi GD không làm được.

SGD có cấu trúc đơn giản và dễ hiểu. Nó chỉ yêu cầu tính toán gradient và cập nhật trọng số dựa trên tốc độ học tập (learning rate). Điều này thể hiện trên quá trình cập nhật trong SGD được thực hiện theo công thức:

Trong đó:

* w(t+1) là trọng số được cập nhật tại bước thời gian t+1,
* w(t) là trọng số hiện tại,
* learning\_rate là tốc độ học tập,
* gradient là giá trị của đạo hàm riêng của hàm mất mát đối với trọng số hiện tại. Nó đo lường độ dốc và hướng tăng nhanh nhất của hàm mất mát tại điểm đang xét.

Tốc độ học tập (learning rate) quyết định kích thước bước nhảy được thực hiện trong quá trình cập nhật trọng số, ảnh hưởng đến tốc độ hội tụ và hiệu suất cuối cùng của mô hình. Việc lựa chọn một tốc độ học tập phù hợp là rất quan trọng. Một giá trị cao giúp tiến nhanh tới đích sau vài vòng lặp tuy nhiên thuật toán không hội tụ, quanh quẩn quanh đích vì bước nhảy quá lớn. Trong khi một giá trị quá thấp có thể dẫn đến hội tụ chậm. Thường cần thử nghiệm và điều chỉnh tốc độ học tập một cách tinh tế để tìm ra tốc độ học tập tối ưu cho một tác vụ và kiến trúc mô hình cụ thể.

A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

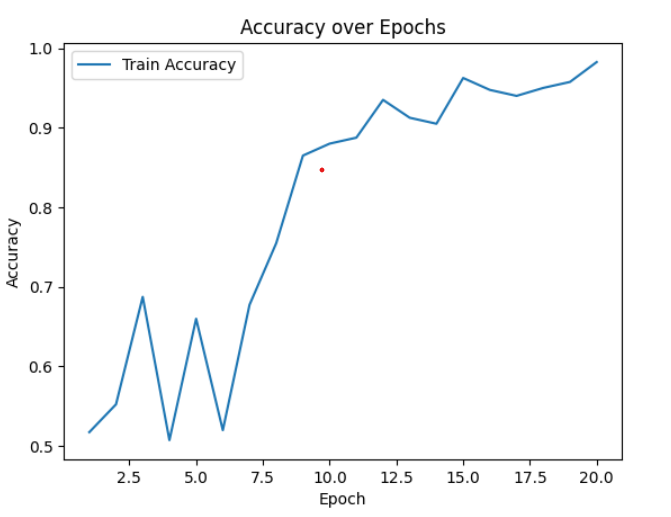
Description automatically generated

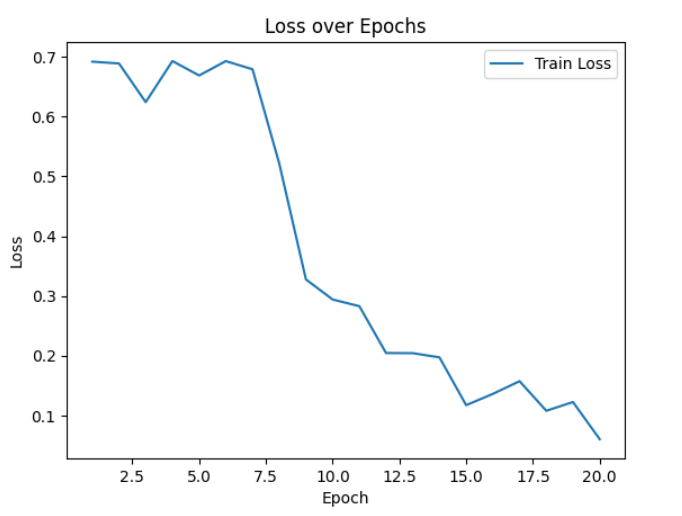
Ngoài ra, bộ tối ưu hóa SGD cho phép điều chỉnh tốc độ học tập (learning rate) một cách linh hoạt. Điều này cho phép tinh chỉnh learning rate để đạt được hiệu suất tốt nhất.

Trong mô hình này, chúng tôi sử dụng bộ tối ưu hóa SGD với kích thước của batch là 64, tốc độ học tập ban đầu bằng 0.001 và giảm dần 0.95 sau mỗi 1000 bước.

**Chương 6: Đánh giá mô hình**

Em thực hiện đánh giá trên chính bộ dữ liệu huấn luyện. Sau 20 epoch, kết quả tốt nhất thu được là mô hình với độ chính xác 96,52%, mất mát chỉ còn 0.067





**Chương 7: Tài liệu tham khảo**

1. Li, Y., Lyu, S.: *Exposing deepfake videos by detecting face warping artifacts*. arXiv preprint arXiv:1811.00656. (2018)
2. GitHub: [pbcquoc/cnn: cnn tutoral for beginer (github.com)](https://github.com/pbcquoc/cnn)
3. Athirasree Das; K.S Angel Viji; Linda Sebastian: *A Survey on Deepfake Video Detection Techniques Using Deep Learning.* (2022)
4. GitHub: [KhanhLinh2810/project\_20241](https://github.com/KhanhLinh2810/project_20241) – url to github project