BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CMC

BÁO CÁO THỰC TẬP NGHỀ NGHIỆP

TRÍ TUỆ NHÂN TẠO VÀ KHOA HỌC DỮ LIỆU

NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn: Nguyễn Thị Vân Anh

Sinh viên thực hiện: Nguyễn Thị Hải

Mã số sinh viên: BIT220048

Lớp: 22IT2

Khóa: 2022 – 2025

Hà Nội, năm 2024

TRƯỜNG ĐẠI HỌC CMC

KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 20 tháng 09 năm 2024

KẾ HOẠCH THỰC TẬP NGHỀ NGHIỆP

(Dành cho sinh viên)

Họ và tên: Nguyễn Thị Hải; Mã sinh viên: BIT220048

Ngày sinh: 29/08/2004; Lớp: 22IT2

Điện thoại liên lạc: 0394309545; Email: BIT220048@st.cmc-u.edu.vn

Khóa học: 2022-2025; Ngành học: Công nghệ thông tin.

Đơn vị thực tập: Viện Nghiên cứu ứng dụng Công Nghệ CMC (CMC ATI)

A. Thời gian thực tập:

Thực tập trong: 15 tuần, kể từ ngày 16/09/2024 đến ngày 29/12/2024

B. Nội dung công việc:

Tuần 1,2: Thực hiện ôn tập lại kiến thức (Python, Tensorflow, Pytorch, OpenCV, Keras) và các bài kiểm tra (do Chuyên viên hướng dẫn tạo).

Tuần 3 đến tuần 9: Thực hiện nghiên cứu và thử nghiệm các bài toán về Trí tuệ nhân tạo và Học máy.

Tuần 10 đến tuần 14: Nhận đề tài tốt nghiệp kỳ thực tập và thực hiện đề tài – theo nhóm sinh viên Lab IoT/Smart Devices.

Tuần 15: Viết báo cáo, đánh giá và nhận điểm số.

Xác nhận của giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

Sinh viên thực tập

(Ký và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC CMC

KHOA CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 27 tháng12 năm 2024

NHẬT KÝ THỰC TẬP NGHỀ NGHIỆP

Họ và tên: Nguyễn Thị Hải ; Mã sinh viên: BIT220048

Ngày sinh: 29/08/2004 ; Lớp: 22IT2

Điện thoại liên lạc: 0394309545 ; Email: BIT220048@st.cmcu.edu.vn

Khóa học: 2022-2025 ; Ngành học: Công nghệ thông tin

1. Đơn vị thực tập: Viện Nghiên cứu Ứng dụng Công nghệ CMC ATI

Thời gian thực tập: từ 16/09/2024 đến 29/12/2024.

2. Cán bộ hướng dẫn tại đơn vị: Trần Quang Đức

Nội dung nhật ký thực tập:

Buổi thứ	**Ngày/tháng/ năm**	**Nội dung thực tập (Kế hoạch)**
2	17/09/2024	Ôn tập Python cơ bản
3	18/09/2024	Ôn tập framework TensorFlow
4	19/09/2024	Ôn tập framework PyTorch
5	20/09/2024	Bắt đầu cuộc thi trên Kaggle. Thực hiện bài toán Face Mask Classification (FMC)
6	23/09/2024	Xử lý ảnh cơ bản với OpenCV

7-9	24-26/09/2024	Nghiên cứu cải thiện quá trình dự đoán mô hình với Pytorch, Thực hiện cuộc thi FMC.
10 - 12	27/09 - 01/10/2024	Áp dụng phương pháp cải thiện quá trình dự đoán cho FMC
13	02/10/2024	Tìm hiểu về Exploratory Data Analysis (EDA)
14	03/10/2024	Tìm hiểu về cách xử lý mất cân bằng dữ liệu (IBP)
15	04/10/2024	Áp dụng EDA và IBP vào FMC
16 - 19	07 - 10/10/2024	Làm quen với Object Detection. Triển khai, code from scratch Faster R-CNN theo tài liệu "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks"
20 - 22	11 -15/10/2024	Tìm hiểu về framework MLOps
23	16/10/2024	Tìm hiểu về ONNX, ONNX-Runtime và TensorRT, so sánh tài nguyên sử dụng khi infer bằng PyTorch, ONNX và TensorRT
24 - 29	17 - 24/10/2024	Tìm hiểu về NCCN. Triển khai mô hình Object Detection lên thiết bị Android
30	25/10/2024	Tìm hiểu về các nền tảng Vertex AI cho MLOps.
31	28/10/2024	Tìm hiểu về các nền tảng Vertex AI cho MLOps.
32 - 33	29 - 30/10/2024	Gán nhãn dữ liệu
34-44	31/10 – 15/11/2024	Tìm hiểu về Quantum Neural Network (QNN)
46	18/11/2024	Tìm hiểu, chọn đề đề tài cuối khóa thực tập
47 - 50	19-22/11/2024	Tìm hiểu, khảo sát về các phương pháp Face Anti-Spoofing (FAS) sau năm 2015. Làm báo cáo 1 - phần 1 - Giới thiệu.

51 - 59	25/11- 5/12/2024	Thu thập, đánh giá, khảo sát các tập dữ liệu public. Làm báo cáo 2 - phần 2 - Thu thập dữ liệu.
60- 61	06-09/12/2024	Làm data cho FAS. Làm báo cáo 3 - phần 3 - Các phương pháp tiêu biểu.
62	10-26/12/2024	Triển khai, code theo tài liệu "FLIP: Cross-domain Face Anti-spoofing with Language Guidance". Làm báo cáo 3 - phần 3 - Các phương pháp tiêu biểu.
75	27/12/2024	Hoàn thiện báo cáo, đánh giá và nhận kết quả.

Nhận xét của Cán bộ hướng dẫn:

……

……

……

……

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

Sinh viên thực tập

(Ký và ghi rõ họ tên)

Xác nhận của đơn vị thực tập

(Ký và ghi rõ họ tên)

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 27 tháng 12 năm 2024

PHIẾU TỰ ĐÁNH GIÁ THỰC TẬP

Họ và tên: Nguyễn Thị Hải ; Mã sinh viên: BIT220048

Ngày sinh: 29/08/2004 ; Lớp: 22IT2

Điện thoại liên lạc: 0394309545 ; Email: BIT220048@st.cmc-u.edu.vn

Khóa học: 2022 - 2024 ; Ngành học: Công nghệ thông tin

Khoa: Công nghệ thông tin và Truyền thông.

Đã hoàn thành đợt thực tập tại đơn vị thực tập từ ngày 16/09/2024 đến ngày 15/01/2025

Đánh giá quá trình thực tập tại đơn vị thực tập:

STT	Nội dung	Mức độ đánh giá					Ghi chú
		Rất kém	Kém	Trung bình	Tốt	Rất tốt	
1	Ý thức tổ chức kỷ luật				X		
2	Kiến thức chuyên môn				X		
3	Kỹ năng chuyên môn				X		
4	Kết quả thực tập					X	

Sinh viên thực tập

(Ký, ghi rõ họ tên)

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 27 tháng 12 năm 2024.

PHIẾU CHẤM ĐIỂM CỦA ĐƠN VỊ THỰC TẬP

Đơn vị thực tập: Viện Nghiên cứu ứng dụng Công Nghệ CMC (CMC ATI)

Họ và tên: Nguyễn Thị Hải; Mã sinh viên: BIT220048

Ngày sinh:29/08/2004; Lớp:22IT2

Khóa học: K1; Ngành học: Công nghệ thông tin

Trường: TRƯỜNG ĐẠI HỌC CMC

Tên đơn vị thực tập: Viện Nghiên cứu ứng dụng Công Nghệ CMC

Thời gian thực tập: 16/09/2024 – 15/01/2025

Họ và tên cán bộ hướng dẫn: Trần Quang Đức

Chức vụ: Chuyên viên

Nội dung đánh giá	**Điểm tối đa**	**Điểm chấm**
1. Hiệu quả công việc	**3,0**	
Khả năng hoàn thành nhiệm vụ đúng hạn và đạt chất lượng yêu cầu	1,5	
Mức độ đóng góp vào các dự án hoặc công việc của đơn vị	1,5	
2. Kỹ năng chuyên môn	**2,5**	
Kiến thức chuyên môn liên quan đến công việc	1,0	
Kỹ năng sử dụng công cụ, phần mềm, thiết bị	1,0	
Khả năng ứng dụng lý thuyết vào thực tế	0,5	
3. Kỹ năng mềm	**2,0**	

Kỹ năng giao tiếp và làm việc nhóm	1,0	
Khả năng giải quyết vấn đề và ra quyết định	1,0	
4. Thái độ làm việc	**1,5**	
Tinh thần trách nhiệm và chủ động trong công việc	1,0	
Tính kỷ luật và sự phù hợp với văn hóa doanh nghiệp	0,5	
5. Tự phát triển và học hỏi	**1,0**	
Khả năng học hỏi, tiếp thu kiến thức và kỹ năng mới	1,0	
Tổng điểm	**10**	

Tổng điểm: ……(có thể chấm lẻ tới 0,1 điểm).

Điểm bằng số: …….

Điểm bằng chữ: ……

Nhận xét chung của đơn vị thực tập đối với sinh viên thực tập (bắt buộc):

.....

.....

.....

.....

Đại diện đơn vị thực tập
(Ký và ghi rõ họ tên)

Cán bộ hướng dẫn
(Ký và ghi rõ họ tên)

TRƯỜNG ĐẠI HỌC CMC

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Hà Nội, ngày 27 tháng 12 năm 2024

PHIẾU CHẤM ĐIỂM CỦA GIẢNG VIÊN HƯỚNG DẪN

Khoa quản lý ngành: Công nghệ thông tin và Truyền thông

Họ và tên: Nguyễn Thị Hải ; Mã sinh viên: BIT220048

Ngày sinh: 29/08/2004 ; Lớp: 22IT2

Khóa học: 2022 – 2024 .; Ngành học: Công nghệ thông tin

Nội dung đánh giá	Điểm tối đa	Điểm chấm
1. Quá trình làm việc với giảng viên	**3,0**	
Có liên hệ để thông qua kế hoạch thực tập đúng thời hạn quy định	1,0	
Có trao đổi thường xuyên với giảng viên hướng dẫn	1,0	
Có thái độ đúng mực, biết tiếp thu các gợi ý của giảng viên hướng dẫn	1,0	
2. Quá trình thực tập tại đơn vị	**3,0**	
Chấp hành nội quy, quy định của đơn vị thực tập	1,0	
Nhật ký thực tập được đơn vị xác nhận	1,0	
Nhật ký thực tập được ghi chép đầy đủ theo từng buổi thực tập tại đơn vị	1,0	
3. Trình bày Báo cáo thực tập nghề nghiệp	**3,0**	
Báo cáo thực tập nghề nghiệp được nộp đúng hạn	1,0	
Báo cáo thực tập nghề nghiệp được trình bày đúng quy định	2,0	
4. Thái độ làm việc	**1,0**	
Thể hiện tinh thần trách nhiệm và sự chủ động trong công việc	1,0	
Tổng điểm	**10**	

Tổng điểm: ……(có thể chấm lẻ tới 0,1 điểm).

Điểm bằng số: ……

Điểm bằng chữ: ……

Nhận xét chung của giảng viên hướng dẫn:

.....

.....

.....

.....

Giảng viên hướng dẫn
(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Em xin chân thành cảm ơn Ban Giám hiệu Trường Đại học CMC và Viện nghiên cứu và ứng dụng công nghệ CMC-ATI đã tạo điều kiện để em có cơ hội tham gia kỳ thực tập bổ ích và ý nghĩa này.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến anh **Trần Quang Đức**, người đã trực tiếp hướng dẫn và hỗ trợ em trong suốt quá trình thực tập. Anh đã không chỉ tận tình chỉ dẫn mà còn chia sẻ những kiến thức chuyên môn sâu rộng, các kinh nghiệm thực tế quý báu, cũng như đưa ra những lời khuyên hữu ích giúp em định hướng rõ hơn về nghề nghiệp trong tương lai.

Bên cạnh đó, em xin dành lời cảm ơn chân thành đến thầy **Vũ Việt Vũ** và cô **Vân Anh**, người luôn đồng hành, động viên và hỗ trợ sinh viên trong mọi hoàn cảnh. Sự tận tâm và nhiệt huyết của thầy cô đã trở thành nguồn động lực lớn, giúp em vượt qua những khó khăn trong học tập và thực tập, đồng thời luôn cảm thấy được đồng hành và chia sẻ trong hành trình phát triển bản thân.

Kỳ thực tập này không chỉ mang lại cho em cơ hội học hỏi mà còn giúp em có cái nhìn toàn diện hơn về cách làm việc chuyên nghiệp trong môi trường thực tế. Những bài học từ anh Trần Quang Đức, sự quan tâm của cô Vân Anh và các đồng nghiệp tại CMC-ATI đã giúp em rèn luyện kỹ năng, phát triển tư duy sáng tạo, và trau dồi thêm nhiều kiến thức bổ ích.

Em tin rằng những trải nghiệm này sẽ là nền tảng vững chắc để em tiếp tục nỗ lực, hoàn thiện bản thân, và đóng góp tích cực vào lĩnh vực mà em theo đuổi.

Em xin chân thành cảm ơn và kính chúc Quý nhà trường, Viện CMC-ATI, thầy Vũ, cô Vân Anh, cùng anh Trần Quang Đức luôn phát triển bền vững, đạt được nhiều thành tựu và thành công trong mọi lĩnh vực.

Trân trọng,

Nguyễn Thị Hải

Mục lục

PHẦN MỞ ĐẦU: GIỚI THIỆU

1. Lý do chọn CMC-ATI

- Uy tín và chuyên môn cao trong ngành: ATI CMC là công ty thành viên của Tập đoàn CMC, một trong những tập đoàn công nghệ lớn và có tiếng tại Việt Nam. Công ty hoạt động chủ yếu trong các lĩnh vực ngân hàng, tài chính, giáo dục và y tế, cung cấp các giải pháp công nghệ thông tin hiện đại.
- Cơ hội học hỏi và trải nghiệm các vị trí thuộc mảng Artificial Intelligent (AI) và Machine Learning (ML).
- Phát triển kỹ năng làm việc nhóm và giao tiếp: Trong quá trình thực tập, người học sẽ có cơ hội làm việc cùng các kỹ sư giàu kinh nghiệm, rèn luyện khả năng làm việc nhóm và giao tiếp trong môi trường thực tế, từ đó phát triển kỹ năng mềm cần thiết cho công việc sau này.

2. Mục tiêu thực tập

- Áp dụng kiến thức học tập vào thực tiễn.
- Nâng cao kiến thức về quy trình phát triển cũng như việc triển khai mô hình hiệu quả, tối ưu trong AI và ML.
- Tích lũy kinh nghiệm và kỹ năng làm việc trong môi trường doanh nghiệp.
- Đánh giá và định hướng nghề nghiệp.

3. Thông tin tổng quan về CMC-ATI

3.1. Lịch sử hình thành và phát triển

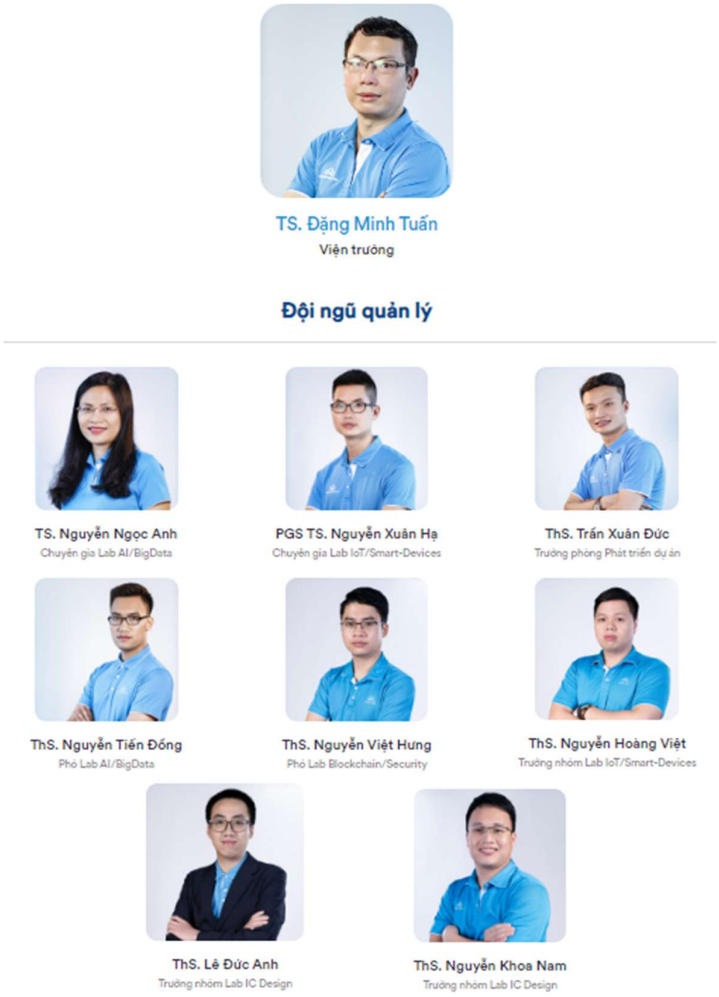
CMC ATI, Viện Nghiên cứu Ứng dụng Công nghệ thuộc Tập đoàn CMC, được thành lập năm 2014. Từ đó đến nay, công ty đã đóng góp không nhỏ vào việc phát triển các giải pháp công nghệ phục vụ chuyển đổi số, giúp nâng cao năng lực cạnh tranh cho các doanh nghiệp tại Việt Nam.

Tập đoàn CMC, chuyên nghiên cứu và phát triển các giải pháp công nghệ. ATI hướng đến việc ứng dụng các công nghệ tiên tiến trong những lĩnh vực như AI, Big Data, IoT, Blockchain, và An ninh mạng nhằm phục vụ quá trình chuyển đổi số cho các doanh nghiệp và tổ chức tại Việt Nam. Thông qua các giải pháp này, ATI hỗ trợ nâng cao năng lực cạnh tranh của khách hàng trong thời kỳ công nghệ 4.0 hiện nay.

CMC ATI cũng đã gặt hái được nhiều giải thưởng, bao gồm các thành tựu về giải pháp trí tuệ nhân tạo tại các cuộc thi và hội thảo trong nước, nhờ các sản phẩm và sáng kiến đột phá trong lĩnh vực nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, và tự động hóa quy trình số hóa tài liệu.

3.2. Quy mô, cơ cấu, tổ chức

CMC ATI hoạt động với các phòng lab chuyên biệt và đội ngũ nhân sự đa dạng từ các trường đại học hàng đầu như Đại học Bách Khoa Hà Nội và Đại học Quốc gia Hà Nội. Các chuyên gia chủ chốt của viện bao gồm những nhà nghiên cứu, tiến sĩ và kỹ sư có kinh nghiệm trong lĩnh vực công nghệ. Cấu trúc tổ chức của CMC ATI tập trung vào các lĩnh vực chuyên môn, mỗi lĩnh vực được điều hành bởi một chuyên gia phụ trách, nhằm tối ưu hóa hiệu quả trong nghiên cứu và triển khai sản phẩm.



3.3. Lĩnh vực hoạt động

CMC ATI hoạt động chủ yếu trong các lĩnh vực nghiên cứu và ứng dụng công nghệ số, bao gồm Trí tuệ Nhân tạo, Dữ liệu Lớn, IoT, Blockchain và Bảo mật thông tin. CMC ATI không chỉ cung cấp các sản phẩm công nghệ cho thị trường trong nước mà còn hướng tới các ứng dụng công nghệ đa lĩnh vực trên toàn cầu.

3.4. Sản phẩm, dịch vụ

- CIVAMS (Hệ thống nhận diện khuôn mặt và kiểm soát ra vào),
- C-OCR (Công nghệ nhận dạng ký tự quang học),
- C-VOICE (Hệ thống chuyển đổi văn bản thành giọng nói),
- C-SODA (Giải pháp tự động hóa dữ liệu).

Ngoài ra, CMC ATI còn cung cấp dịch vụ tư vấn và triển khai các giải pháp số hóa phù hợp cho doanh nghiệp, giúp khách hàng tăng cường năng lực cạnh tranh trong kỷ nguyên số.

4. Văn hóa doanh nghiệp

CMC ATI xây dựng văn hóa doanh nghiệp xoay quanh các giá trị cốt lõi: **sáng tạo, chuyên nghiệp và đồng đội**. Công ty luôn khuyến khích nhân viên tìm tòi, sáng tạo và phát triển các ý tưởng mới trong công việc nhằm giải quyết các thách thức công nghệ. Môi trường làm việc ở đây không chỉ là nơi nhân viên được tự do thể hiện quan điểm, mà còn là không gian để mỗi cá nhân học hỏi và cập nhật các kiến thức tiên tiến từ các đồng nghiệp và chuyên gia.

CMC ATI cũng đề cao trách nhiệm và sự hỗ trợ lẫn nhau trong các nhóm làm việc, giúp nhân viên rèn luyện kỹ năng làm việc nhóm và tinh thần đồng đội. Công ty thường xuyên tổ chức các buổi hội thảo, workshop và các chương trình phát triển cá nhân để nâng cao năng lực chuyên môn và kỹ năng mềm của nhân viên.

Ngoài ra, với môi trường trẻ trung, năng động và thân thiện, CMC ATI tạo điều kiện để mỗi cá nhân phát triển trong không gian làm việc cởi mở. Nhân

viên có thể dễ dàng trao đổi ý tưởng với các cấp quản lý, tạo ra một văn hóa tương tác hiệu quả và hỗ trợ lẫn nhau.

PHẦN NỘI DUNG VÀ KẾT QUẢ THỰC TẬP

Theo phân công, em được vào Lab IoT/Smart Devices thực tập dưới sự hướng dẫn của anh Trần Quang Đức với từng tuần thực tập được tóm tắt như sau:

Tuần	Mô tả công việc	Kết quả	Khó khăn và cách khắc phục
01	Ôn tập Python, Ôn tập framework tensorflow, Ôn tập framework Pytorch.	Về cơ bản là hoàn thành, ôn tập lại được các kiến thức.	Là tuần đầu tiên nên chưa thích nghi với môi trường.
02	Xử lý ảnh với OpenCV, Nghiên cứu cải thiện quá trình dự đoán mô hình, Thực hiện chủ đề: Face Mask Classification (FMC).	Học được các kiến thức xử lý ảnh cơ bản, Xây dựng các lớp giúp cải thiện quá trình với bài báo khoa học, Bắt đầu xây dựng bài toán FMC với nhiều mô hình khác nhau.	Vì xử lý ảnh và xây dựng mô hình dựa trên một mô hình có sẵn chưa được học và chưa làm nên có một chút hơi ngợp khi làm các bài tập. Tuy nhiên sau tìm hiểu và sự giúp đỡ của mentor thì đã hoàn thành công việc tốt.
03	Tìm hiểu về EDA, cách xử lý mất cân bằng dữ liệu và áp dụng vào cuộc thi FMC.	Hoàn thành cuộc thi với độ nhận diện là 94%. Hiểu được cách xử lý dữ liệu trước khi huấn luyện.	Ban đầu kết quả FMC khá thấp ~ 74%, tuy nhiên sau khi thực hiện nhiều kỹ thuật dữ liệu thì kết quả cao hơn.

04	Làm quen với Object Detection, Tìm hiểu về MLOps.	Thực hiện tìm hiểu và dùng *faster rcnn* cho bài toán Car Detection với độ chính xác ~98.6%, Hiểu về các công việc của MLOps.	Ban đầu nhận diện không chính xác – nhận diện tất cả các đối tượng có trong khung hình là xe ô tô, sau điều chỉnh thì đã đạt được độ chính xác cao.
05	Tìm hiểu về *onnx*, *onnxruntime* và *tensorrt*, so sánh tài nguyên sử dụng khi dự đoán bằng pytorch, onnx và tensorrt. Tìm hiểu về *ncnn*.	*ONNX Runtime* là sự lựa chọn trung gian với khả năng tối ưu tốt hơn trên cả CPU và GPU, tiết kiệm bộ nhớ và thời gian infer hơn PyTorch. Đồng thời, thực hiện triển khai các mô hình.	Có nhiều nguồn, phải bỏ nhiều thời gian để chọn lọc và đọc hiểu cũng như là triển khai mô hình.
06	Tìm hiểu về *ncnn*, sau đó triển khai mô hình ở format *ncnn* lên android (CoreML đối với hàng của apple) với bài toán Object Detection đã làm.	Tìm hiểu về *ncnn* và cách triển khai lên Android/IOS, Đã thực hiển triển khai *ncnn* lên android với bài toán Car Detection.	Khâu triển khai còn thiếu kinh nghiệm. Đã lên các web để học về cách triển khai thực tế một mô hình lên thiết bị vật lý.
07 + 08	Gán nhãn dữ liệu, Tìm hiểu về các nền tảng Vertex AI cho MLOps.	Gán nhãn 10078 hình ảnh, Thực hiện tìm hiểu và trải nghiệm về các nền tảng SageMaker, Vertex AI, Azure ML, ML Flow, Kubeflow.	MLOps là một lĩnh vực mới, khó, và cần nhiều tài nguyên nên cần đầu tư nhiều thời gian và công sức.

08 + 09	Tìm hiểu về Quantum Neural Network (QNN).	Hoàn thành tìm hiểu và triển khai một mạng QNN nhỏ.	Các bài toán với QNN khá khó và tiêu tốn nhiều tài nguyên.
10	Chọn đề tài bài tập cuối khóa thực tập, Khảo sát về các phương pháp Face Anti Spoofing từ 2015 làm báo cáo số 1.	Đề tài Face Anti Spoofing, Tìm đọc các bài báo khoa học về đề tài này từ năm 2015 đến nay để hiểu được các kiểu tấn công phổ biến cũng như phương pháp chống lại.	Đa dạng các loại tấn công, nhiều bài báo nên khá khó trong việc chọn lọc cũng như tổng hợp thông tin.
11	Thu thập dữ liệu, và làm báo cáo số 2.	Thu thập được 4 bộ dữ liệu với đa dạng các thuộc tính và thực hiện EDA các tập dữ liệu.	Không.
12	Thu thập dữ liệu, và làm báo cáo số 2.	Thu thập được 4 bộ dữ liệu với đa dạng các thuộc tính và thực hiện EDA các tập dữ liệu.	Không.
13 + 14	Kỹ thuật dữ liệu, huấn luyện mô hình và làm báo cáo số 3.	Thực hiện EDA dữ liệu, Xây dựng và huấn luyện mô hình.	Không.
15	Trình bày bài tập lớn và làm báo cáo kỳ thực tập.	Hoàn thành bài tập lớn.	Không

1. Cơ sở lý thuyết

1.1. Nghiên cứu, ôn tập về các framework cơ bản

- **Python**:
    - Ôn tập các khái niệm: cấu trúc dữ liệu (list, dictionary, set…), class và module.

- Thực hành các bài toán về xử lý chuỗi, tính toán ma trận, và lập trình hướng đối tượng.
- Xây dựng các class như `AI`, `DeepLearning` và module `model.py` để làm quen với lập trình nâng cao.

- **TensorFlow & PyTorch**:
  - Lý thuyết: Tìm hiểu sự khác biệt giữa `fit` và `fit_generator`, các thành phần chính của mô hình TensorFlow, và các bước huấn luyện mô hình.
  - Thực hành: Tạo tensor, thực hiện các phép tính với ma trận, xây dựng mạng deep learning để huấn luyện và đánh giá với dữ liệu huấn luyện/kiểm tra được chuẩn hóa.
- **OpenCV**:
  - Lý thuyết: Khái niệm phép tích chập và các bộ lọc số.
  - Thực hành: Làm mờ, làm sắc nét, phát hiện biên ảnh, và xử lý nhiễu trên dữ liệu hình ảnh sử dụng các hàm tích hợp sẵn của OpenCV.

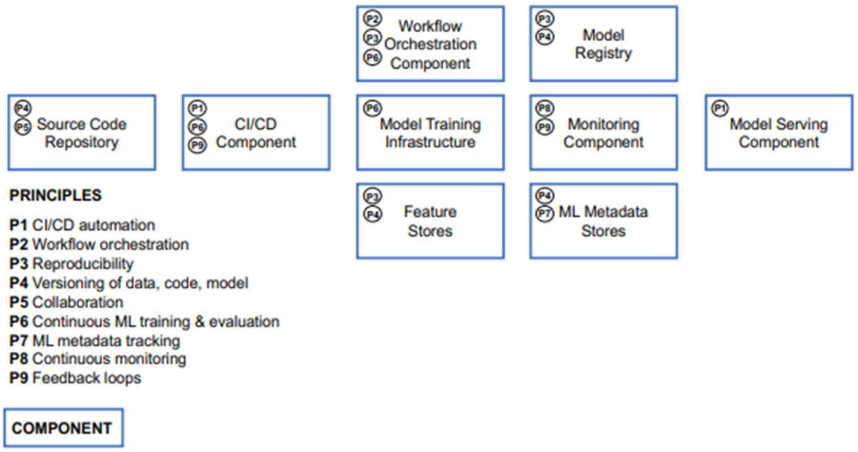
1.2. Lý thuyết về các phương pháp học máy

- **EDA (Exploratory Data Analysis)**:
  - Mục tiêu: Khai phá, trực quan hóa dữ liệu (heatmap, biểu đồ phân bố) và kiểm tra dữ liệu thiếu.
  - Thực hành: Chuẩn hóa dữ liệu bằng `MinMaxScaler`, phát hiện mối tương quan giữa các thuộc tính, và áp dụng các kỹ thuật xử lý mất cân bằng dữ liệu như SMOTE và undersampling.
- **Xử lý mất cân bằng dữ liệu**:
  - Các phương pháp: Oversampling, undersampling, focal loss, data augmentation.
  - Ứng dụng vào bài toán Face Mask Classification (FMC) giúp cải thiện độ chính xác từ 74% lên 94%.
- **Object Detection**:
  - Mô hình: Faster R-CNN, YOLOv5, SSD.

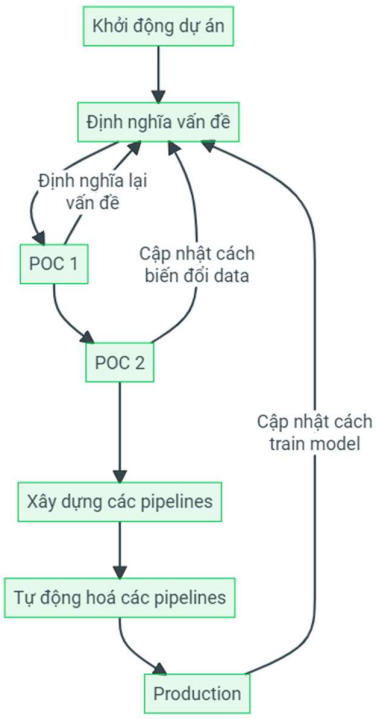
- Lý thuyết: Quy trình hoạt động của Faster R-CNN, các lớp chính trong kiến trúc mạng, và các kỹ thuật cải thiện hiệu suất như transfer learning.

1.3. Các nền tảng và công nghệ MLOps

+ **Yêu cầu:** Tìm hiểu các khái niệm, nguyên tắc, thành phần, quy trình và các nền tảng hỗ trợ MLOps.
+ **Phương pháp:** Thực hiện tìm hiểu từ các bài báo khoa học và chắt lọc thông tin cần tìm hiểu về MLOps.
+ **Kết quả:**
  - Khái niệm:
    - MLOps là mô hình kết hợp các thực tiễn tốt nhất (best practices), khái niệm, văn hóa làm việc trong quá trình phát triển, triển khai và giám sát hệ thống Machine Learning (ML).
    - MLOps kết hợp ba lĩnh vực: Machine Learning, Kỹ thuật phần mềm (DevOps) và Kỹ thuật dữ liệu (Data Engineering).
  - Nguyên tắc:
    - Kiểm soát phiên bản: Theo dõi sự thay đổi của mã nguồn, dữ liệu và mô hình ML để có thể tái hiện lại phiên bản trước.
    - Tự động hóa: Tự động hóa pipeline ML để lặp lại, nhất quán và mở rộng. Tự động huấn luyện, kiểm thử và triển khai mô hình.
    - Sự liên tục (Continuous X): Bao gồm tích hợp liên tục (CI), triển khai liên tục (CD), huấn luyện liên tục (CT), và giám sát liên tục (CM) để đảm bảo mô hình luôn cập nhật.
    - Quản trị mô hình: Quản lý sự hợp tác giữa các bên liên quan, bảo vệ dữ liệu, đảm bảo tuân thủ và tính công bằng trong mô hình.
  - Thành phần: 5 thành phần cơ bản.



- Lợi ích:
    - Tự động hóa quy trình: Giảm thiểu can thiệp thủ công trong huấn luyện, triển khai và cập nhật mô hình.
    - Quản lý mô hình: Theo dõi, quản lý và bảo trì nhiều phiên bản mô hình.
    - Khả năng mở rộng: Triển khai và quản lý mô hình trên quy mô lớn.
    - Tăng tính tái sử dụng: Tạo ra pipeline tái sử dụng được trong các dự án ML khác.
    - Quản lý rủi ro: Giảm rủi ro nhờ khả năng giám sát và kiểm soát mô hình.
- Thách thức:
    - Phức tạp: Yêu cầu tích hợp các công cụ và quy trình từ nhiều lĩnh vực.
    - Khả năng tái sản xuất: Đảm bảo kết quả mô hình có thể được tái sản xuất trên nhiều môi trường khác nhau.
    - Chi phí: Chi phí lớn về hạ tầng và nguồn lực để xây dựng hệ thống MLOps đầy đủ.
- Quy trình chung:



- Nền tảng hỗ trợ:
  - **Thương mại**:
    - **AWS SageMaker**: Cung cấp công cụ quản lý vòng đời ML, tích hợp tốt với hệ sinh thái AWS.
    - **Azure Machine Learning**: Hỗ trợ AutoML, tích hợp với dịch vụ Azure.
    - **Google Vertex AI**: Hợp nhất AutoML và công cụ AI, tích hợp tốt với BigQuery.
    - **DataRobot**: Tự động hóa AI từ chuẩn bị dữ liệu đến triển khai.
    - **Databricks**: Tích hợp tốt với MLFlow, tối ưu cho phân tích dữ liệu lớn.
  - **Mã nguồn mở**:
    - **KubeFlow**: Hỗ trợ triển khai và quản lý mô hình ML trên hạ tầng container hóa Kubernetes.
    - **MLflow**: Quản lý vòng đời mô hình, hỗ trợ theo dõi, quản lý và triển khai mô hình.

- **H2O.ai**: Hỗ trợ AutoML, phù hợp với phân tích dữ liệu lớn.
- **Flyte**: Hỗ trợ workflows phức tạp, phân phối công việc ML.
- **DAGsHub**: Hỗ trợ quản lý phiên bản dữ liệu và mô hình, tích hợp với MLFlow và DVC.

- Tìm hiểu về nền tảng Vertex AI cho MLOps: Tài liệu

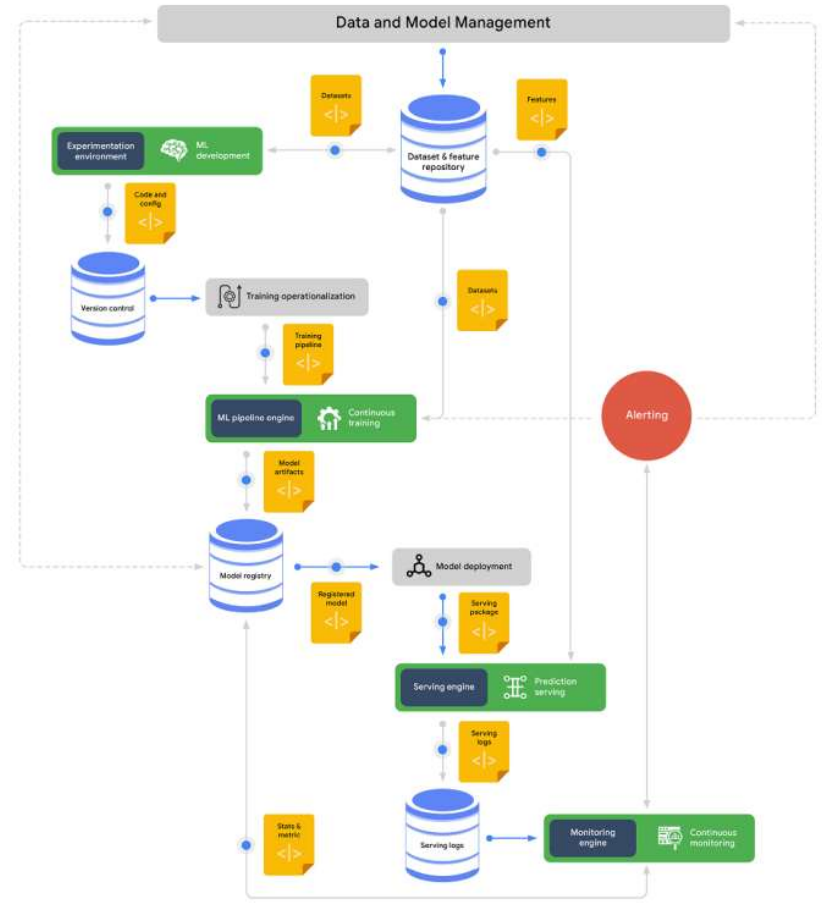


Figure 15. End-to-end MLOps workflow

+ Data and Model Management
  - Input: Tập dữ liệu và các đặc trưng (features).
  - Output: Kho lưu trữ dữ liệu và đặc trưng (Dataset & Feature Repository).
  - Mô tả: Đây là nơi lưu trữ các tập dữ liệu và đặc trưng để hỗ trợ quá trình huấn luyện và triển khai.
+ Experimentation Environment

- Input:
    - Mã nguồn và cấu hình (Code and Config).
    - Dữ liệu từ Dataset & Feature Repository.
- Output: Mô hình ban đầu.
- Mô tả: Đây là giai đoạn nghiên cứu và phát triển mô hình trong môi trường thí nghiệm.

+ Training Operationalization
    - Input:
        - Mô hình từ giai đoạn phát triển (ML Development).
        - Dữ liệu từ Dataset & Feature Repository.
    - Output: Pipeline huấn luyện.
    - Mô tả: Tạo và chuẩn hóa pipeline huấn luyện để tái sử dụng và tự động hóa.
+ Continuous Training
    - Input:
        - Pipeline huấn luyện.
        - Dữ liệu mới từ Dataset & Feature Repository.
    - Output: Các mô hình đã được huấn luyện (Model Artifacts).
    - Mô tả: Huấn luyện lặp đi lặp lại dựa trên dữ liệu mới để cập nhật mô hình.
+ Model Registry
    - Input:
        - Mô hình đã huấn luyện (Model Artifacts).
    - Output:
        - Mô hình được đăng ký (Registered Models).
        - Trạng thái và metric của mô hình.
    - Mô tả: Lưu trữ các phiên bản mô hình cùng thông tin liên quan, giúp quản lý vòng đời mô hình.
+ Model Deployment
    - Input:
        - Mô hình được đăng ký từ Model Registry.
    - Output:

- Gói triển khai (Serving Package).
- Mô tả: Đưa mô hình đã kiểm định vào môi trường triển khai.

+ Prediction Serving
  - Input:
    - Gói triển khai từ Model Deployment.
  - Output:
    - Dự đoán từ mô hình (Prediction Results).
    - Log phục vụ (Serving Logs).
  - Mô tả: Xử lý yêu cầu dự đoán từ hệ thống hoặc người dùng.
+ Monitoring Engine
  - Input:
    - Log phục vụ từ Prediction Serving.
  - Output:
    - Trạng thái và metric.
    - Cảnh báo (Alerting).
  - Mô tả: Theo dõi hoạt động mô hình trong môi trường sản xuất và gửi cảnh báo nếu có vấn đề.
+ Continuous Monitoring
  - Input:
    - Trạng thái và metric từ Monitoring Engine.
  - Output:
    - Cập nhật dữ liệu và mô hình trong vòng lặp.
  - Mô tả: Bảo trì liên tục để đảm bảo mô hình hoạt động hiệu quả và chính xác.

➢ Luồng dữ liệu chính:
  - Dữ liệu đầu vào (Input):
    - Tập dữ liệu (Datasets) và đặc trưng (Features).
    - Mã nguồn và cấu hình (Code and Config).
  - Dữ liệu đầu ra (Output):
    - Mô hình được triển khai (Deployed Model).
    - Dự đoán từ mô hình (Prediction Results).

- Cảnh báo và thông tin giám sát (Alerting, Metrics).

1.4. Lược khảo sát các bài toán Face Anti-Spoofing (FAS)

- Nghiên cứu các phương pháp chống giả mạo khuôn mặt (Face Anti-Spoofing - FAS) nhằm bảo vệ hệ thống nhận diện khuôn mặt khỏi các tấn công như sử dụng ảnh, video hoặc mặt nạ. Các tiêu chí đánh giá chính gồm: **khả năng khái quát hóa** (tương thích với nhiều điều kiện dữ liệu), **khả năng chống lại tấn công chưa biết**, và **khả năng triển khai trên thiết bị biên** (gọn nhẹ, hiệu quả).
- Dự án sử dụng các bộ dữ liệu nổi bật như CelebA-Spoof, Zalo AI Challenge 2022, NUAAAA và LCC FASD.
    - **CelebA-Spoof**: 625.537 hình ảnh (live: 156.384, spoof: 469.153), đa dạng về điều kiện ánh sáng, môi trường trong nhà/ngoài trời và nhiều loại tấn công (in phẳng, in cong, mặt nạ 2D, 3D).
    - **Zalo AI Challenge 2022**: 1.168 video (live: 598, spoof: 570), tập trung vào tín hiệu động với các thông số như độ phân giải, FPS, và mức độ chuyển động.
    - **NUAAAA**: Tập trung vào tấn công in phẳng và in cong dưới điều kiện môi trường được kiểm soát.
    - **LCC FASD**: Dữ liệu thu thập từ nhiều thiết bị, nhấn mạnh vào tấn công phát lại (replay) và mặt nạ 3D.

2. Thực nghiệm

2.1. Ôn tập và luyện tập

- **Thực hành Python:**

Phần	Mô tả	Phương pháp	Bài tập
1. Cấu trúc dữ liệu			
Sắp xếp và thao tác với Dictionary	Sắp xếp dictionary theo giá trị, tạo dictionary mới bằng cách đảo	Sử dụng `sorted()` với lambda function để sắp xếp. Sử dụng dictionary comprehension để tạo dictionary mới.	Sắp xếp dictionary `points` theo điểm số tăng dần, tạo dictionary mới với key là điểm số và value là tên.

Phần	Mô tả	Phương pháp	Bài tập
	ngược key và value.		
Thao tác với List	Tìm các giá trị duy nhất, đếm số lần xuất hiện của mỗi phần tử, tạo list mới chỉ chứa các phần tử duy nhất.	Sử dụng `set()` để tìm giá trị duy nhất. Sử dụng dictionary comprehension để đếm số lần xuất hiện. Sử dụng vòng lặp `for` để tạo list mới.	Tìm các giá trị duy nhất trong list `target`, đếm số lần xuất hiện của mỗi phần tử, tạo list mới chỉ chứa các phần tử duy nhất.
Xử lý Chuỗi	Phân tách chuỗi thành list các từ, đảo ngược thứ tự các từ trong list.	Sử dụng `split()` để phân tách chuỗi. Sử dụng `reverse()` để đảo ngược thứ tự các từ.	Phân tách một câu văn thành list các từ, đảo ngược thứ tự các từ trong list.
Tính toán	Tính giai thừa, tích các số chẵn và tích các số lẻ liên tiếp.	Sử dụng vòng lặp `for` và các phép toán.	Tính giai thừa của một số, tích các số chẵn liên tiếp nhỏ hơn `n`, tích các số lẻ liên tiếp nhỏ hơn `n`.
2. Class và Module			
Định nghĩa và sử dụng Class	Định nghĩa class, tạo constructor, định nghĩa thuộc tính và phương thức, khởi tạo object, gọi phương thức.	Sử dụng từ khóa `class`, hàm `__init__`, định nghĩa thuộc tính và phương thức, khởi tạo object bằng tên class, gọi phương thức bằng dấu chấm (`.`).	Xây dựng class `AI` với các thuộc tính `algorithm` và `model_type`, các phương thức `fit` và `predict`. Khởi tạo object cho class `AI` và gọi các phương thức `fit` và `predict`.
Kế thừa	Tạo class con kế thừa từ class cha, mở rộng class con, ghi đè phương thức.	Sử dụng cú pháp `class ChildClass(ParentClass)`, thêm thuộc tính và phương thức mới vào class con, định nghĩa lại phương thức của class cha trong class con.	Xây dựng class `DeepLearning` kế thừa từ class `AI`, mở rộng class `DeepLearning` bằng cách thêm phương thức `train_on_epoch` và ghi đè phương thức `fit` của class `AI`.

Phần	Mô tả	Phương pháp	Bài tập
Tổ chức code thành Module	Chia code thành các module riêng biệt, import và sử dụng các class từ các module khác.	Tạo các file Python riêng biệt cho mỗi module, sử dụng `import` để import các class.	Tạo các file `model.py`, `dataset.py`, `train.py`. Import các class `DeepLearning`, `AI`, `Dataset` và sử dụng chúng trong file `train.py`.
3. Ma trận và vector			
Thực hành với PyTorch và TensorFlow	Khởi tạo, tính toán, thao tác với số vô hướng, vector, ma trận.	Sử dụng các hàm và phương thức của PyTorch và TensorFlow.	Khởi tạo số vô hướng, vector, ma trận; tính tích, tổng, chuẩn; `reshape` kích thước.
Lý thuyết	Chứng minh các bài toán về ma trận và vector.	Sử dụng các định nghĩa và tính chất của ma trận và vector.	Chứng minh các mệnh đề về ma trận đối xứng, tính chất kết hợp của phép nhân ma trận, trace của ma trận, phép nhân Hadamard.

- **Thực hành Tensorflow:**
  - **Về mặt lý thuyết**, bao gồm các câu hỏi về:
    - Khởi tạo model
    - Sự khác biệt giữa fit và fit_generator
    - Các modules chính trong TensorFlow để xây dựng và huấn luyện model
    - Các bước trong huấn luyện model
    - Ý nghĩa của epochs và batch_size
  - **Về mặt thực hành**, bao gồm các bài tập về:
    - Khởi tạo tensor: Sử dụng hàm tf.random.normal để tạo tensor ngẫu nhiên với kích thước và định dạng yêu cầu.

- Truy xuất các ma trận ảnh tương ứng với kênh R, G, B: Sử dụng slicing để truy xuất các ma trận con tương ứng với từng kênh màu.
- Thực hiện tích Hadamard và tích thông thường giữa ma trận: Sử dụng toán tử * cho tích Hadamard và toán tử @ cho tích thông thường.
- Chuẩn hóa dữ liệu và phân chia tập train/test:
  - Sử dụng pandas để đọc dữ liệu từ file CSV.
  - Sử dụng sklearn.preprocessing.StandardScaler để chuẩn hóa dữ liệu.
  - Sử dụng sklearn.model_selection.train_test_split để phân chia dữ liệu train/test.
- Xây dựng mạng deep-neural-network để huấn luyện và đánh giá model:
  - Sử dụng tensorflow.keras để xây dựng model Sequential.
  - Thêm các layers Dense với hàm kích hoạt relu và sigmoid.
  - Compile model với optimizer Adam, loss function binary_crossentropy và metrics accuracy.
  - Huấn luyện model với hàm fit, sử dụng dữ liệu train và validation.
  - Đánh giá model với hàm evaluate, sử dụng dữ liệu test.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
dense_4 (Dense)	(None, 64)	512
dense_5 (Dense)	(None, 32)	2,080
dense_6 (Dense)	(None, 16)	528

Total params: 3,120 (12.19 KB)
Trainable params: 3,120 (12.19 KB)
Non-trainable params: 0 (0.00 B)

```
[ ] loss, accuracy = model.evaluate(X_test, y_test)
    print(f'Độ chính xác trên tập test: {accuracy*100:.2f}%')
```

```
157/157 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.8195 - loss: 0.3961
Độ chính xác trên tập test: 82.52%
```

- **Thực hành Pytorch:**

- Phần lý thuyết và thực hành Pytorch tương tự với Tensorflow nhưng sử dụng các thành phần của Pytorch (torch.nn, torch.optim, torch.utils.data…) cùng với bộ dữ liệu trên cho độ chính xác 77.14%:

```
Tập huấn luyện: (20000, 7), Tập kiểm thử: (5000, 7)
Epoch [5/30], Loss: 0.6430
Epoch [10/30], Loss: 0.6285
Epoch [15/30], Loss: 0.6127
Epoch [20/30], Loss: 0.5943
Epoch [25/30], Loss: 0.5727
Epoch [30/30], Loss: 0.5478
Độ chính xác trên tập test: 77.14%
```

- Ngoài ra còn có phần thực hành với bài toán Dog&Cat liên quan tới việc tạo một iterator dataset cho bộ dữ liệu "Dog and Cat". Bạn sẽ cần thực hiện các phép biến đổi tương tự như rotate, horizontal flip, random crop,... trước khi lấy ra mỗi bức ảnh.
    - Nạp các thư viện cần thiết: os, torchvision, torch.utils.data, google.colab. Các thư viện này hỗ trợ cho việc xử lý dữ liệu hình ảnh, tạo DataLoader, và kết nối với Google Drive.
    - Kết nối với Google Drive: Đoạn mã sử dụng drive.mount để kết nối với Google Drive của bạn và os.chdir để chuyển thư mục làm việc đến một thư mục cụ thể trong Drive. Điều này cho phép truy cập vào bộ dữ liệu được lưu trữ trên Drive.
    - Xác định đường dẫn đến dữ liệu: Biến path được gán đường dẫn đến thư mục chứa dữ liệu huấn luyện (Train_Data/Train_Data).
    - Tạo các phép biến đổi hình ảnh: Sử dụng transforms.Compose, đoạn mã tạo ra một chuỗi các phép biến đổi sẽ được áp dụng cho mỗi hình ảnh trước khi đưa vào mô hình huấn luyện. Các phép biến đổi này bao gồm:

```
data_transforms = transforms.Compose([
    transforms.RandomRotation(degrees=15),  # Xoay ngẫu nhiên trong khoảng +-15 độ
    transforms.RandomHorizontalFlip(),      # Lật ngang ngẫu nhiên
    transforms.RandomResizedCrop(size=224, scale=(0.8, 1.0)),  # Cắt ngẫu nhiên sau đó tha
    transforms.ToTensor(),                  # Chuyển đổi hình ảnh thành tensor
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])  # Chuẩn hóa ảnh
])
```

        - RandomRotation: Xoay ngẫu nhiên hình ảnh trong một khoảng góc nhất định.

        - RandomHorizontalFlip: Lật ngang hình ảnh ngẫu nhiên.
        - RandomResizedCrop: Cắt ngẫu nhiên một phần của hình ảnh và thay đổi kích thước nó về kích thước mong muốn.
        - ToTensor: Chuyển đổi hình ảnh từ dạng PIL Image sang dạng Tensor mà PyTorch có thể sử dụng.
        - Normalize: Chuẩn hóa các giá trị pixel của hình ảnh về một khoảng nhất định.
    - Tạo dataset: Sử dụng datasets.ImageFolder, đoạn mã tạo ra một dataset từ thư mục chứa dữ liệu huấn luyện. ImageFolder tự động gán nhãn cho hình ảnh dựa trên cấu trúc thư mục.
    - Tạo DataLoader: DataLoader được tạo ra để lấy dữ liệu từ dataset theo từng batch. Nó cho phép bạn chỉ định kích thước batch, có nên xáo trộn dữ liệu hay không, và các tùy chọn khác.
    - In kích thước của dữ liệu: Đoạn mã cuối cùng lặp qua DataLoader và in ra kích thước của batch hình ảnh và batch nhãn.
- **Bài toán Face Mask Classification (FMC):**
    - Yêu cầu: Xây dựng một mô hình Deep Learning nhận diện khuôn mặt đeo khẩu trang, đáp ứng các tiêu chí sau:
        - Nhẹ: Kích thước mô hình nhỏ, độ phức tạp tính toán thấp để triển khai trên thiết bị IoT. Giới hạn ở ResNet-50 khoảng 25.6M params và ảnh đầu vào tối đa (224, 224).
        - Chính xác: Độ chính xác cao trên tập dữ liệu cung cấp.
        - Khái quát hóa tốt: Hoạt động tốt với dữ liệu mới và môi trường đa dạng.
    - Phương pháp:
        - Sử dụng ResNet-50: Là mô hình đã được huấn luyện trước trên ImageNet, giúp rút ngắn thời gian huấn luyện và cải thiện độ chính xác.
        - Fine-tuning: Điều chỉnh một số lớp cuối của ResNet-50 để phù hợp với bài toán nhận diện khẩu trang.

- Kỹ thuật chống overfitting: Áp dụng Batch Normalization, Dropout, và kernel_regularizer để giảm overfitting và tăng khả năng khái quát hóa.
- Tăng cường dữ liệu: Sử dụng ImageDataGenerator để tạo ra nhiều biến thể của ảnh huấn luyện, giúp mô hình học được nhiều đặc trưng hơn.

o Kết quả:

- Độ chính xác: Đạt 74.46% trên tập dữ liệu kiểm tra.

- Mô hình nhẹ: Sử dụng ResNet-50 với giới hạn kích thước, đáp ứng yêu cầu triển khai trên thiết bị IoT.

- **Bài toán Object Detection:**

+ **Yêu cầu**: Xây dựng mô hình học máy với Faster R-CNN (từ bài báo khoa học) có khả năng phát hiện xe hơi trong ảnh và video. Mô hình này được huấn luyện trên tập dữ liệu chứa hình ảnh xe hơi và đường bao quanh tương ứng.

+ **Phương pháp**:

- Chuẩn bị dữ liệu: Tập dữ liệu được tải và xử lý trước. Hình ảnh được thay đổi kích thước và chuẩn hóa, đường biên được chuyển đổi thành định dạng phù hợp cho mô hình. Tập dữ liệu được chia thành tập huấn luyện và tập kiểm tra.
- Xây dựng mô hình: Sử dụng mô hình Faster R-CNN với kiến trúc ResNet-50 làm bộ trích xuất đặc trưng. Bộ dự đoán đường biên được sửa đổi để dự đoán hai lớp: nền và xe hơi.
- Huấn luyện mô hình: Mô hình được huấn luyện bằng cách sử dụng thuật toán tối ưu hóa SGD. Quá trình huấn luyện được giám sát bằng cách theo dõi các hàm mất mát khác nhau.
- Đánh giá mô hình: Mô hình được đánh giá trên tập kiểm tra để đo độ chính xác của nó. Các chỉ số như độ chính xác trung bình (mAP) được sử dụng để đánh giá hiệu suất.

- Áp dụng mô hình: Mô hình được sử dụng để phát hiện xe hơi trong ảnh và video mới. Đường biên và điểm tin cậy được dự đoán cho mỗi xe hơi được phát hiện.

+ **Kết quả đạt được**: Đã huấn luyện thành công một mô hình phát hiện đối tượng có khả năng phát hiện xe hơi trong ảnh và video với độ chính xác cao (~98%). Mô hình này có thể được sử dụng cho các ứng dụng như lái xe tự động, giám sát giao thông và đếm xe.

2.2. Xử lý ảnh và dữ liệu

- Xử lý ảnh với OpenCV: Notebook
  - Yêu cầu:
    - Tính toán số lượng màu hiển thị được trong mô hình RGB 24-bit:
      - Yêu cầu đưa ra công thức tính toán và kết quả.
      - Kết quả mong đợi: Mô hình RGB 24-bit có thể hiển thị được 256^3 màu.
    - Trình bày về phép tích chập và các bộ lọc số:
      - Yêu cầu giải thích về khái niệm phép tích chập, cách thức hoạt động và ứng dụng của nó trong xử lý ảnh.
      - Yêu cầu giải thích về các bộ lọc số, bao gồm các loại bộ lọc, cách thức hoạt động và ứng dụng của chúng trong xử lý ảnh.
    - Các thao tác xử lý ảnh: Có thể bao gồm các thao tác như làm mờ, làm sắc nét, phát hiện biên, phân đoạn ảnh,...
  - Kết quả đạt được:

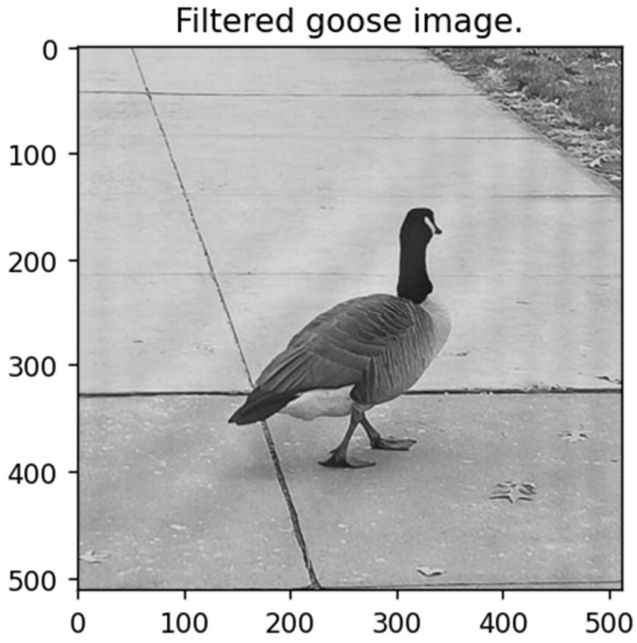
- Tính toán số lượng màu: Đã đưa ra công thức tính toán và kết quả chính xác cho số lượng màu hiển thị được trong mô hình RGB 24-bit.
- Trình bày về phép tích chập và các bộ lọc số: Đã giải thích rõ ràng về khái niệm phép tích chập và các bộ lọc số, bao gồm cách thức hoạt động và ứng dụng của chúng trong xử lý ảnh.

	Tích chập	Bộ lọc số / tuyến tính
Định nghĩa	Là một kỹ thuật toán học được sử dụng rộng rãi trong xử lý tín hiệu, hình ảnh, và học sâu. Phép tích chập được dùng để kết hợp hai hàm, trong đó một hàm thường là tín hiệu hoặc dữ liệu, còn hàm kia là bộ lọc. Tính chất: • Giao hoán • Kết hợp • Phân phối	Là các phương pháp sử dụng phép toán trên các giá trị dữ liệu số để loại bỏ nhiễu hoặc để làm mịn dữ liệu, tăng cường các đặc điểm nhất định. Có hai loại bộ lọc chính trong xử lý tín hiệu số: • Bộ lọc thông thấp (Low-pass filter) • Bộ lọc thông cao (High-pass filter) Các bộ lọc tuyến tính (linear filters) có thể được biểu diễn dưới dạng tích chập giữa tín hiệu đầu vào và một hạt nhân (kernel) tuyến tính. Một số bộ lọc phổ biến bao gồm: • Bộ lọc trung bình (Mean Filter) • Bộ lọc Gaussian • Bộ lọc Sobel Tính chất: • Tuyến tính • Dịch chuyển bất biến (Shift Invariant)
Công thức	• Tích chập - Trường hợp liên tục $g(x,y) = f(x,y) * h(x,y)$ $= \int_{\tau_1=-\infty}^{+\infty} \int_{\tau_2=-\infty}^{+\infty} f(\tau_1,\tau_2) h(x-\tau_1, y-\tau_2) d\tau_1 d\tau_2$	Tích chập - Trường hợp rời rạc $g[x,y] = f[x,y] * h[x,y]$ $= \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} f[n_1,n_2] h[x-n_1, y-n_2]$
Ứng dụng	**Xử lý ảnh**: Phát hiện biên cạnh, làm mịn, nâng cao độ nét. **Học sâu (CNN)**: Phát hiện đặc trưng trong nhận diện hình ảnh, phân loại đối tượng. **Âm thanh**: Loại bỏ nhiễu, thêm hiệu ứng âm thanh (echo, reverb). **Phân tích tín hiệu thời gian thực**: Xử lý dữ liệu từ sóng âm, EEG, dữ liệu cảm biến.	**Xử lý tín hiệu số**: Lọc nhiễu, cải thiện chất lượng âm thanh/tín hiệu. **Xử lý ảnh**: Làm mờ, giảm nhiễu, nén dữ liệu (JPEG, MP3).

- Các thao tác xử lý ảnh: Đã thực hiện một số các phép xử lý ảnh theo yêu cầu: làm mờ, làm sắc nét, lọc, xóa nhiễu….
    - Ghép các tấm ảnh với stitched:

- Sử dụng các phép xoay, lật, nghiêng…

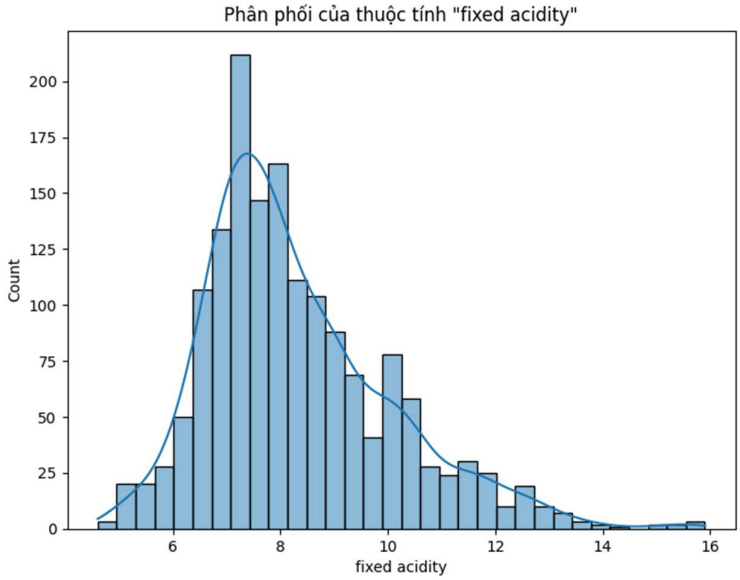
- Xóa nhiễu:



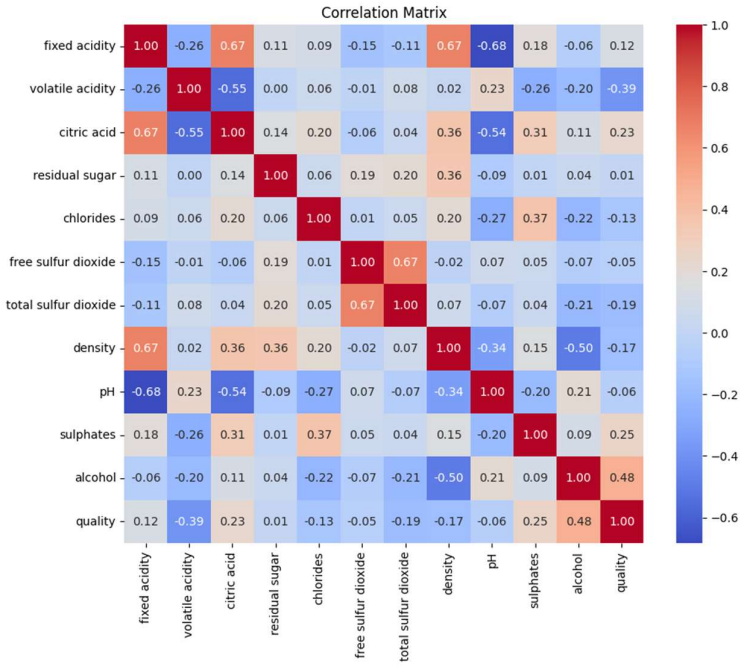
- Khai phá dữ liệu (EDA):
    - Yêu cầu: Tìm hiểu về Khai phá dữ liệu và thực hiện EDA bộ dữ liệu red_wine_quality.
    - Kết quả đạt được:
        - Dữ liệu được nạp và tiền xử lý thành công: Dữ liệu từ file winequality-red.csv đã được nạp vào DataFrame df và được xử lý:
            - Kiểm tra dữ liệu thiếu: df.isnull().sum() cho thấy không có dữ liệu thiếu trong tập dữ liệu.
            - Chuẩn hóa dữ liệu: Dữ liệu đã được chuẩn hóa bằng MinMaxScaler để đưa các biến về cùng một phạm vi, giúp cải thiện hiệu suất của mô hình Machine Learning.
        - Khám phá dữ liệu: Thông qua việc sử dụng các hàm head(), info(), describe() và trực quan hóa dữ liệu bằng biểu đồ phân phối và heatmap, ta đã có được cái nhìn tổng quan về tập dữ liệu, bao gồm:
            - Cấu trúc dữ liệu: Tập dữ liệu bao gồm các thông tin về các thuộc tính lý hóa của rượu vang đỏ và chất lượng tương ứng.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978

- Phân bố dữ liệu: Biểu đồ phân phối cho thấy phân bố của từng thuộc tính, ví dụ: fixed acidity có phân bố lệch phải.



- Mối tương quan giữa các biến: Heatmap cho thấy mối tương quan giữa các biến, ví dụ: density và alcohol có tương quan âm khá mạnh.



- Dữ liệu được chuẩn bị sẵn sàng để huấn luyện mô hình: Tập dữ liệu đã được chia thành tập huấn luyện và tập kiểm tra bằng train_test_split, với biến mục tiêu là quality và các biến còn lại là biến đầu vào. Điều này cho phép ta huấn luyện và đánh giá mô hình Machine Learning trên tập dữ liệu này.

- Cân bằng dữ liệu:
  - Resampling:
    - Oversampling: Tăng mẫu lớp thiểu số bằng cách nhân bản hoặc tạo mẫu mới (SMOTE, ADASYN).
    - Undersampling: Giảm mẫu lớp chiếm đa số để cân bằng.
  - Class Weights: Điều chỉnh trọng số trong hàm mất mát để mô hình ưu tiên lớp thiểu số.
  - Threshold Tuning: Điều chỉnh ngưỡng dự đoán để cân bằng giữa các lớp.
  - Ensemble Methods: Sử dụng kỹ thuật Bagging, Boosting để cải thiện độ chính xác.
  - Anomaly Detection: Đối với dữ liệu hiếm, xem như bài toán phát hiện dị thường.

- Focal Loss: Điều chỉnh hàm mất mát để giảm trọng số của các mẫu dễ phân loại và tăng trọng số cho các mẫu khó, phù hợp cho các bài toán mất cân bằng lớn.
- Data Augmentation: Tăng cường dữ liệu bằng cách biến đổi dữ liệu hiện tại (xoay, cắt, thay đổi độ sáng) để tăng lượng dữ liệu của lớp thiểu số.

Với bài toán FMC thì có thể sử dụng các phương pháp như thêm dữ liệu, Data Augmentation, Class Weights và sử dụng **BatchNormalization**, **Dropout** để tăng độ ổn định và giảm overfitting. Kết quả tăng từ 68% lên 77%.

2.3. Tối ưu và cải thiện quá trình dự đoán

- **Cải thiện quá trình dự đoán mô hình:**
  + **Yêu cầu:**

    Bài toán tập trung vào việc chuyển đổi một kiến trúc mạng phức tạp sang dạng đơn giản hơn, trong khi vẫn giữ được độ chính xác cao, để phù hợp với thiết bị biên (edge devices).

    - Kiến trúc train phase:
      - Sử dụng mạng ResNet hoặc tương tự, với nhiều nhánh (branches) bao gồm:
      - Nhánh Conv3x3 + BatchNorm
      - Nhánh Conv1x1 + BatchNorm
      - Nhánh Identity (nếu số lượng kênh đầu vào bằng đầu ra).
    - Kiến trúc inference phase: Gộp (merge) tất cả các nhánh thành một nhánh duy nhất, cụ thể là một convolution 3x3 với bias đã được kết hợp.
    - Phương pháp:
      - Gộp từng nhánh (Conv + BatchNorm) thành một convolution duy nhất.

- Kết hợp kernel của Conv1x1 và nhánh Identity (nếu có) vào kernel trung tâm của Conv3x3.
- Tổng hợp tất cả các vector bias từ các nhánh.

+ **Phương pháp chi tiết:**
  - **Gộp Conv + BatchNorm:**

    Sử dụng công thức:

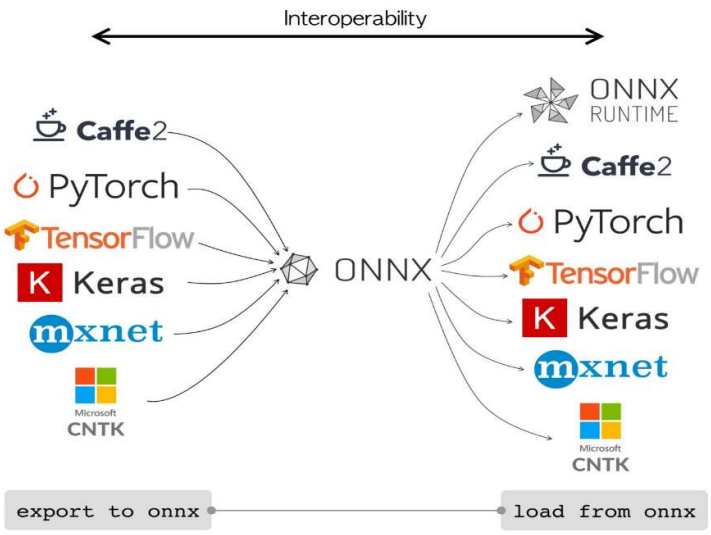
$$W_{fused} = W_{conv} \cdot \frac{\gamma_{BN}}{\sqrt{\sigma_{BN}^2 + \epsilon}}$$

$$b_{fused} = \beta_{BN} - \frac{\gamma_{BN} \cdot \mu_{BN}}{\sqrt{\sigma_{BN}^2 + \epsilon}}$$

  - **Nhánh Identity**:
    - Nhánh này được coi là Conv1x1 với kernel là ma trận đơn vị và bias là vector 0.
    - Sau đó áp dụng phương pháp gộp tương tự như Conv + BatchNorm.
  - **Kết hợp kernel**: Kernel của Conv1x1 và nhánh Identity được mở rộng vào trung tâm của kernel Conv3x3.
  - **Kết quả cuối cùng**: Kernel và bias sau khi gộp được sử dụng để thay thế kernel của Conv3x3, các nhánh khác sẽ bị xóa.
+ **Kết quả đạt được**:
  - **Tối ưu hóa inference**: Code
    - Sau khi gộp, kiến trúc trở nên đơn giản hơn với chỉ một lớp convolution 3x3 duy nhất, giúp giảm đáng kể chi phí tính toán trên thiết bị biên.
    - Mô hình vẫn giữ được độ chính xác cao nhờ cách gộp kernel và bias chính xác.
  - **Hiệu năng cải thiện**:
    - Mô hình gọn nhẹ hơn, giảm thiểu yêu cầu về bộ nhớ và khả năng tính toán.
    - Thời gian chạy inference nhanh hơn, phù hợp với thiết bị biên hạn chế tài nguyên.

- **Kiến thức ứng dụng**:
    - Kỹ thuật này mở rộng cho các kiến trúc phức tạp khác, không chỉ ResNet, như VGG hoặc các mạng tương tự với nhiều nhánh.

- **ONNX và TensorRT: Tìm hiểu về onnx, onnxruntime và tensorrt, so sánh tài nguyên sử dụng khi dự đoán bằng pytorch, onnx và tensorrt**:
    + Yêu cầu:
        - Hiểu khái niệm và chức năng của ONNX, ONNX Runtime và TensorRT.
        - So sánh tài nguyên sử dụng (CPU, bộ nhớ, thời gian) khi thực hiện dự đoán (inference) với mô hình PyTorch, ONNX và TensorRT.
    + Phương pháp:
        - Nghiên cứu tài liệu chính thức và các bài viết liên quan để hiểu rõ về các công nghệ này.



        - ONNX: Định dạng trung gian để biểu diễn mô hình deep learning.
        - ONNX Runtime: Bộ thực thi để chạy mô hình ONNX trên nhiều nền tảng.
        - TensorRT: Bộ công cụ tối ưu hóa và thực thi mô hình deep learning của NVIDIA.

- Chuyển đổi mô hình PyTorch sang ONNX: Sử dụng hàm torch.onnx.export để xuất mô hình PyTorch sang định dạng ONNX.
- Chạy dự đoán với mô hình PyTorch, ONNX (sử dụng ONNX Runtime) và TensorRT.
- Đo lường thời gian thực thi, mức sử dụng CPU và bộ nhớ trong quá trình dự đoán.
- Phân tích dữ liệu đo lường để so sánh hiệu năng và tài nguyên sử dụng của ba phương pháp.

+ Kết quả: Tài liệu

```
PyTorch FPS: 512.36
 TensorRT FPS: 2155.14
 Speedup: 4.21x
```

- PyTorch dễ sử dụng nhưng không tối ưu cho inference, tốn nhiều CPU/GPU và bộ nhớ hơn.
- ONNX Runtime là sự lựa chọn trung gian với khả năng tối ưu tốt hơn trên cả CPU và GPU, tiết kiệm bộ nhớ và thời gian infer hơn PyTorch.
- TensorRT vượt trội về hiệu năng trên GPU, sử dụng ít tài nguyên hơn và tối ưu hóa sâu về bộ nhớ và thời gian inference, nhưng phức tạp hơn trong triển khai và cần phần cứng NVIDIA.

- **Triển khai mô hình Faster R-CNN lên Android qua NCNN:**
  + Yêu cầu:
    - Mục tiêu: Triển khai mô hình object detection trên thiết bị di động Android sử dụng framework NCNN. Đối với thiết bị Apple, có thể sử dụng CoreML.
    - Mô hình: Sử dụng mô hình object detection đã được huấn luyện trước (pre-trained).

- Đánh giá: Đánh giá hiệu năng của mô hình trước và sau khi chuyển đổi sang định dạng NCNN.
- Triển khai: Triển khai mô hình trên Android để thực hiện object detection.

+ Phương pháp:
  - Tìm hiểu về NCNN: NCNN là một framework inference hiệu năng cao được thiết kế cho các thiết bị di động. Tìm hiểu về kiến trúc, cách sử dụng và các công cụ của NCNN.
  - Chuyển đổi mô hình: Chuyển đổi mô hình object detection đã huấn luyện sang định dạng NCNN. Có thể sử dụng các công cụ chuyển đổi được cung cấp bởi NCNN hoặc các công cụ của bên thứ ba.
  - Đánh giá mô hình: Đánh giá hiệu năng của mô hình trước và sau khi chuyển đổi, sử dụng các chỉ số như độ chính xác, tốc độ inference, dung lượng bộ nhớ.
  - Triển khai trên Android: Tích hợp mô hình NCNN vào ứng dụng Android. Sử dụng Android NDK và SDK để xây dựng ứng dụng và giao tiếp với mô hình.
  - Kiểm thử: Kiểm thử ứng dụng trên thiết bị Android để đảm bảo hoạt động chính xác và hiệu quả.
+ Kết quả:
  - Mô hình object detection được chuyển đổi sang định dạng NCNN.
  - Ứng dụng Android được phát triển để thực hiện object detection sử dụng mô hình NCNN.
  - Hiệu năng của mô hình được đánh giá trước và sau khi chuyển đổi.

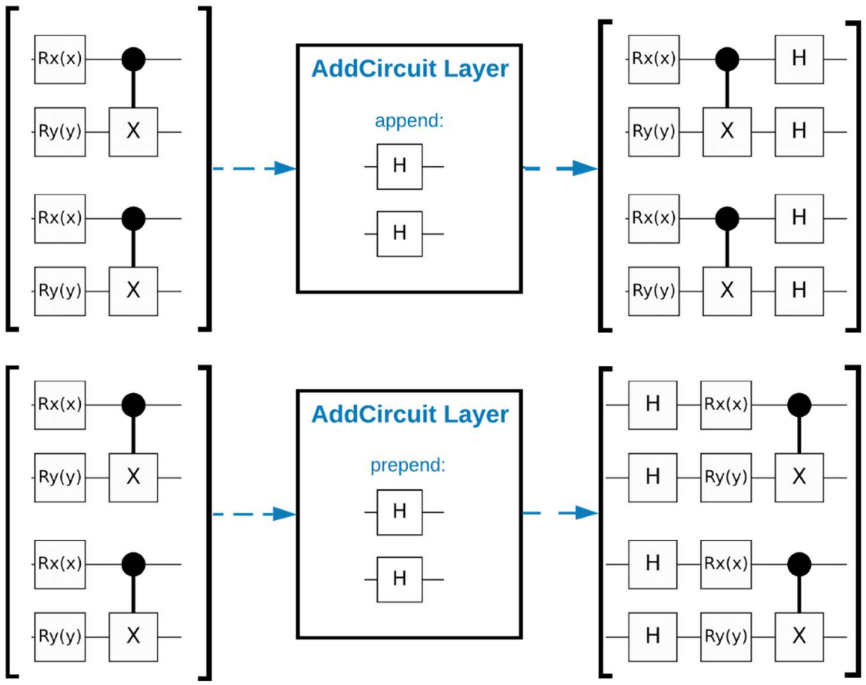
	Model	Param File Size (KB)	Bin File Size (KB)
0	Original YOLOv5 (NCNN)	21.801758	99728.617188
1	Optimized YOLOv5 (NCNN)	16.535156	49918.242188

2.4. Mạng nơ-ron lượng tử (QCNN)

- **Yêu cầu:**

+ Thực hiện một mạng nơ-ron tích chập lượng tử (QCNN) đơn giản, một mạng nơ-ron tương tự lượng tử được đề xuất cho mạng nơ-ron tích chập cổ điển, đồng thời bất biến tịnh tiến.
+ Mục tiêu là để phát hiện các thuộc tính nhất định của một nguồn dữ liệu lượng tử, chẳng hạn như cảm biến lượng tử hoặc mô phỏng phức tạp từ một thiết bị. Nguồn dữ liệu lượng tử là trạng thái cluster có thể có hoặc không có kích thích - điều mà QCNN sẽ học để phát hiện.

- **Phương pháp**:



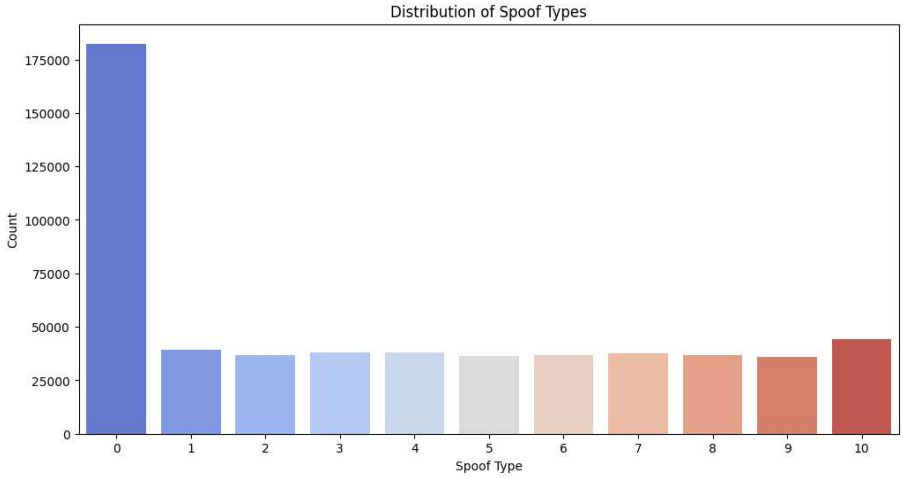
+ Xây dựng QCNN:
    - Lắp ráp các mạch trong biểu đồ TensorFlow bằng cách sử dụng tfq.layers.AddCircuit để thêm các mạch lượng tử vào mô hình.
    - Chuẩn bị trạng thái cluster và huấn luyện bộ phân loại lượng tử để phát hiện xem nó có bị "kích thích" hay không. Kích thích được biểu diễn bằng cổng cirq.rx.
    - Xác định các lớp cho trạng thái cluster, tích chập lượng tử (QConv) và gộp lượng tử (QPool).

        - Xác định mô hình bằng cách sử dụng các lớp đã xác định, bắt đầu với tám qubit, gộp xuống một…
    + Huấn luyện mô hình:
        - Tạo dữ liệu huấn luyện bao gồm các kích thích cho trạng thái cluster và các nhãn tương ứng.
        - Biên dịch mô hình bằng cách sử dụng trình tối ưu hóa Adam, hàm mất mát MSE và chỉ số chính xác tùy chỉnh.
        - Huấn luyện mô hình trên toàn bộ batch và theo dõi mất mát và độ chính xác.
    + Mô hình lai:
        - Khám phá các mô hình lai lượng tử-cổ điển bằng cách kết hợp các lớp tích chập lượng tử với các mạng nơ-ron cổ điển dày đặc.
        - Xác định và huấn luyện mô hình lai với một bộ lọc lượng tử duy nhất và với nhiều bộ lọc lượng tử.
        - So sánh hiệu suất của QCNN thuần túy và các mô hình lai.
- **Kết quả**:
    + Trình bày cách xây dựng và huấn luyện QCNN để phát hiện các trạng thái cluster bị kích thích.
    + Các mô hình lai, kết hợp các thành phần lượng tử và cổ điển, có thể hội tụ nhanh hơn so với các QCNN thuần túy.
    + Sử dụng TensorFlow Quantum và Cirq để tạo và huấn luyện các mô hình học máy lượng tử.
    + Đồ thị mất mát và độ chính xác cho thấy hiệu suất của mô hình trong quá trình huấn luyện.
    + Khám phá các kiến trúc lai khác nhau, bao gồm các mô hình với một bộ lọc lượng tử duy nhất và nhiều bộ lọc lượng tử.

Tóm lại, qua phần tìm hiểu về QNN cho thấy những góc nhìn tổng quan cũng như các phương pháp triển khai một mạng QNN. Tuy nhiên, trong quá trình triển khai gặp nhiều khó khăn về tài nguyên nên dự án chưa để hoàn thành đạt kỳ vọng.

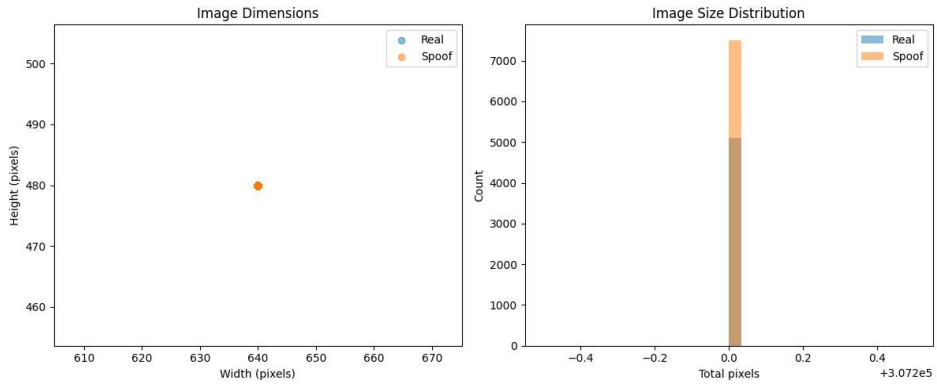
2.5. Khảo sát và triển khai các bài toán Face Anti-Spoofing

- **Khai phá dữ liệu:**
  - **CelebA-Spoof**: 625.537 hình ảnh (live: 156.384, spoof: 469.153), đa dạng về điều kiện ánh sáng, môi trường trong nhà/ngoài trời và nhiều loại tấn công (in phẳng, in cong, mặt nạ 2D, 3D).

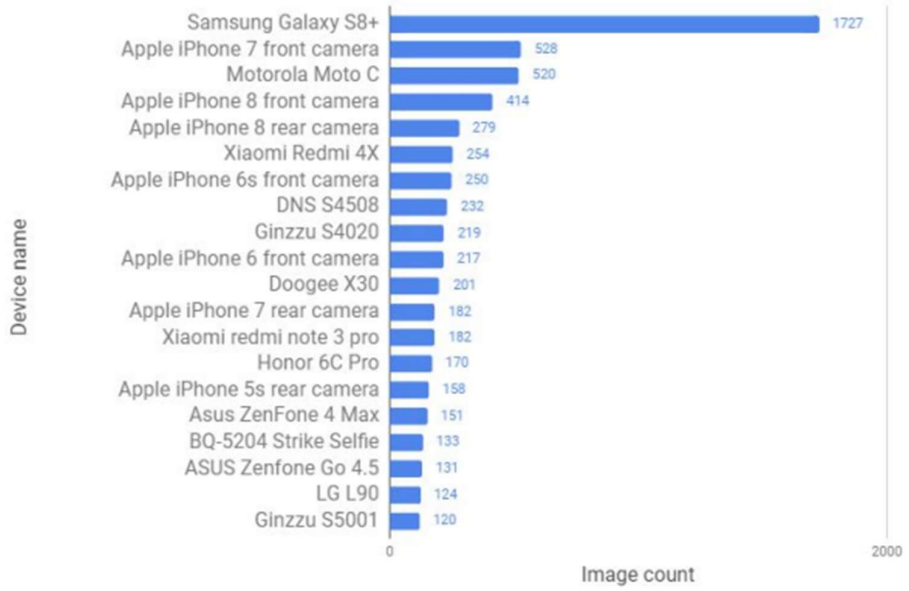


  - **Zalo AI Challenge 2022**: 1.168 video (live: 598, spoof: 570), tập trung vào tín hiệu động với các thông số như độ phân giải, FPS, và mức độ chuyển động.

  - **NUAAAA**: Tập trung vào tấn công in phẳng và in cong dưới điều kiện môi trường được kiểm soát.



- **LCC FASD**: Dữ liệu thu thập từ nhiều thiết bị, nhấn mạnh vào tấn công phát lại (replay) và mặt nạ 3D.



- **Triển khai phương pháp FLIP (Face Anti-Spoofing with Language-Image Pretraining)**

  - **Tổng quan**

    Phương pháp FLIP (Face Anti-Spoofing with Language-Image Pretraining) giải quyết bài toán **Cross-domain Face Anti-Spoofing (FAS)** bằng cách sử dụng kiến trúc Vision Transformer (ViT) với trọng số tiền huấn luyện từ mô hình CLIP (Contrastive Language-Image Pretraining). Điểm nổi bật của FLIP:

    - Sử dụng trọng số tiền huấn luyện từ mô hình ngôn ngữ-hình ảnh.

- Áp dụng chiến lược học tương phản (contrastive learning) và liên kết giữa ngữ nghĩa của văn bản và hình ảnh để cải thiện khả năng tổng quát hóa.

- Thách thức trong Cross-domain FAS:
  - Domain Shift: Sự khác biệt giữa các domain, bao gồm cảm biến camera, điều kiện ánh sáng, công cụ tấn công (ảnh in, video phát lại, mặt nạ 3D).
  - Dữ liệu giới hạn: Các tập dữ liệu hiện tại thường có kích thước nhỏ, dẫn đến việc mô hình dễ bị overfit vào domain nguồn.
  - Khả năng zero-shot: Phần lớn các phương pháp trước yêu cầu một lượng dữ liệu từ domain mục tiêu (labeled hoặc unlabeled), điều này không khả thi trong nhiều trường hợp thực 2tế.
- Mục tiêu của FLIP:
  - Tăng khả năng tổng quát hóa cho FAS qua việc tận dụng các đặc trưng học từ ngôn ngữ-hình ảnh.
  - Đạt hiệu năng cao trong bối cảnh dữ liệu thấp (low-data regimes).
  - Cải thiện khả năng phân biệt giữa khuôn mặt thật và giả mà không cần dữ liệu từ domain mục tiêu.
- Kiến trúc của FLIP

Phương pháp FLIP gồm ba biến thể chính, mỗi biến thể sử dụng một chiến lược tối ưu hóa khác nhau:

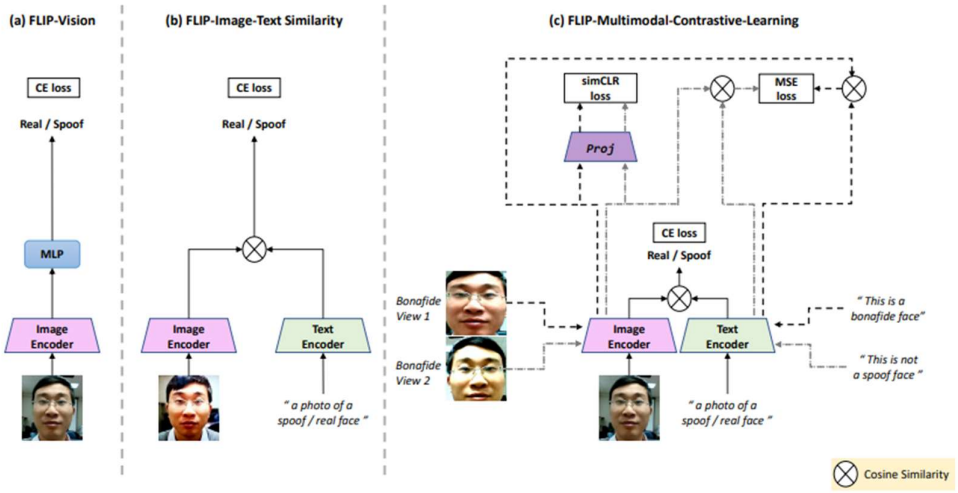


Figure 2. Overview of the proposed FLIP framework for cross-domain face anti-spoofing.

1. FLIP-Vision (FLIP-V):

- **Ý tưởng:** Fine-tune bộ mã hóa hình ảnh (image encoder) của CLIP để phân loại live/spoof.
- **Hoạt động:**
  - Dữ liệu hình ảnh được chia thành các patch cố định và truyền qua bộ mã hóa ViT.
  - Sử dụng một lớp MLP để phân loại spoof/real.
  - Loss: Cross-Entropy (CE) Loss.

2. FLIP-Image-Text Similarity (FLIP-IT):

- **Ý tưởng:** Liên kết hình ảnh với các mô tả ngữ nghĩa của văn bản (text prompts).
- **Hoạt động:**
  - Dùng bộ mã hóa văn bản (text encoder) của CLIP để mã hóa các mô tả ngữ nghĩa của hai lớp real và spoof (ví dụ: "This is a real face").
  - Tính độ tương đồng cosine giữa biểu diễn hình ảnh và văn bản.
  - Loss: Cross-Entropy Loss tính trên độ tương đồng cosine.
- **Ưu điểm:** Cải thiện hiệu năng bằng cách tận dụng thông tin ngữ nghĩa từ văn bản.

3. FLIP-Multimodal-Contrastive-Learning (FLIP-MCL):

- **Ý tưởng:** Kết hợp học tương phản (contrastive learning) và các phép đo tự giám sát để tăng khả năng tổng quát hóa.
- **Hoạt động:**
  - Loss bao gồm:
    - **Cross-Entropy Loss (Lce):** Phân loại spoof/real.
    - **SimCLR Loss (LsimCLR):** Áp dụng trên hai view khác nhau của cùng một hình ảnh.
    - **Mean Squared Error Loss (Lmse):** Đảm bảo tính nhất quán giữa các cặp hình ảnh-văn bản.
- **Ưu điểm:** Đặc biệt hiệu quả trong bối cảnh zero-shot và dữ liệu hạn chế.

- Kết quả thực nghiệm

1. Dataset và Protocols:

- **Protocol 1:** Đánh giá trên các tập dữ liệu nhỏ (MSU-MFSD (280 video, 35 người), CASIA-MFSD (600 video, 50 người), Replay Attack (1300 video, 05 người), OULU-NPU(4.950 video, 55 người)).
  - Mục tiêu: Đánh giá khả năng zero-shot khi bỏ lại một domain để kiểm tra.
- **Protocol 2:** Đánh giá trên các tập dữ liệu lớn (CASIA-SURF(>21.000 video, 1000 người), CASIA-CeFA(4.000 video, 160 người), WMCA(2.904 video, 72 người)).
  - Mục tiêu: Đánh giá trên các domain lớn và giàu tính đa dạng.
- **Protocol 3:** Đánh giá single-source-to-single-target (12 kịch bản khác nhau giữa MSU, CASIA, Replay Attack và OULU).
  - Mục tiêu: Đánh giá khả năng tổng quát hóa từ một domain nguồn đến một domain mục tiêu.

2. Kết quả

- Các chỉ số đo lường được sử dụng:

- HTER (Half Total Error Rate)
  - Ý nghĩa: Trung bình cộng của False Rejection Rate (FRR) và False Acceptance Rate (FAR).
  - Công thức:

$$\text{HTER} = \frac{\text{FRR} + \text{FAR}}{2}$$

Trong đó:

FRR (False Rejection Rate): Tỉ lệ ảnh thật bị phân loại sai là giả mạo:

$$\text{FRR} = \frac{\text{False Negatives (FN)}}{\text{False Negatives (FN)} + \text{True Positives (TP)}}$$

FAR (False Acceptance Rate): Tỉ lệ ảnh giả mạo bị phân loại sai là thật:

$$\text{FAR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}}$$

- AUC (Area Under the ROC Curve):
  - **Ý nghĩa:** Diện tích dưới đường cong ROC (Receiver Operating Characteristic), biểu diễn mối quan hệ giữa TPR và FPR ở các ngưỡng khác nhau.
  - **Công thức:** AUC được tính bằng cách tích phân diện tích dưới đường cong ROC:

$$\text{AUC} = \int_0^1 \text{TPR} \cdot d(\text{FPR})$$

Trong đó:

- TPR (True Positive Rate)

$$\text{TPR} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

- FPR (False Positive Rate)

$$\text{FPR} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}}$$

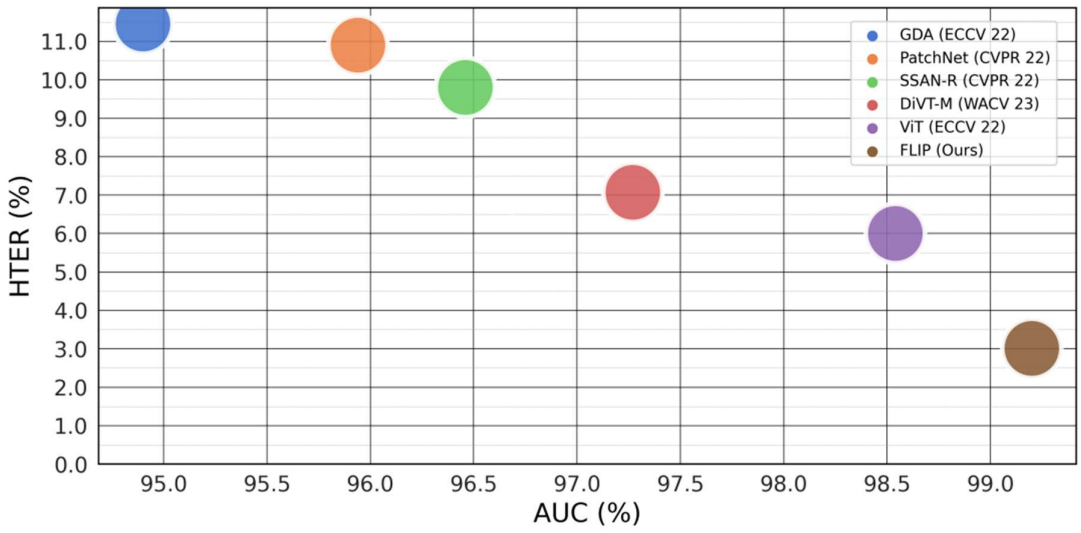
- TPR@FPR=x% (True Positive Rate at a Specific FPR)
  - **Ý nghĩa:** Tỉ lệ nhận dạng đúng ảnh thật (TPR) tại một mức FPR cố định, ví dụ 1%.
  - Công thức:

$$\text{TPR@FPR=x\%} = \frac{\text{TP tại ngưỡng FPR=x\%}}{\text{Tổng số Positive (TP + FN)}}$$

- EER (Equal Error Rate)
  - **Ý nghĩa:** Điểm mà FRR và FAR bằng nhau, thể hiện sự cân bằng giữa hai loại lỗi.
  - **Công thức:** EER được xác định tại ngưỡng τkhi:

$$\text{FRR}(\tau) = \text{FAR}(\tau)$$

- Bảng so sánh AUC và HTER giữa FLIP và SOTA.



Phương pháp	**Protocol 1**	**Protocol 2**	**Protocol 3**
FLIP-V	Hiệu năng vượt trội ở zero-shot nhưng phụ thuộc vào CE Loss.	Tốt trên các tập nhỏ.	Hiệu năng khá trong các kịch bản khó.
FLIP-IT	Tận dụng thông tin ngữ nghĩa, cải thiện trên 2/3 tập.	Tốt trên tập dữ liệu lớn.	Hiệu quả vượt trội khi có hỗ trợ từ text prompts.
FLIP-MCL	Hiệu năng tốt nhất, vượt cả 5-shot của SOTA.	Cải thiện mạnh mẽ trên tập lớn với HTER thấp hơn.	Kết quả trung bình tốt hơn SOTA +8.36%.

(Các phương pháp SOTA là những phương pháp được công bố tại các hội nghị hàng đầu như CVPR, ECCV, AAAI hoặc trên các tạp chí uy tín (TPAMI, IJCAI,

TIFS) và có hiệu năng cao nhất trong bài toán Cross-Domain Face Anti-Spoofing.)

Đánh giá chi tiết:

- Protocol 1: FLIP-MCL có AUC vượt trội so với SOTA ở ba trong bốn domain.

	Method	OCI → M			OMI → C			OCM → I			ICM → O			Avg.
		HTER	AUC	TPR@ FPR=1%	HTER	AUC	TPR@ FPR=1%	HTER	AUC	TPR@ FPR=1%	HTER	AUC	TPR@ FPR=1%	HTER
0-shot	MADDG (CVPR' 19) [38]	17.69	88.06	–	24.50	84.51	–	22.19	84.99	–	27.98	80.02	–	23.09
	MDDR (CVPR' 20) [44]	17.02	90.10	–	19.68	87.43	–	20.87	86.72	–	25.02	81.47	–	20.64
	NAS-FAS (TPAMI' 20) [53]	16.85	90.42	–	15.21	92.64	–	11.63	96.98	–	13.16	94.18	–	14.21
	RFMeta (AAAI' 20) [39]	13.89	93.98	–	20.27	88.16	–	17.30	90.48	–	16.45	91.16	–	16.97
	D^2AM (AAAI' 21) [6]	12.70	95.66	–	20.98	85.58	–	15.43	91.22	–	15.27	90.87	–	16.09
	DRDG (IJCAI' 21) [28]	12.43	95.81	–	19.05	88.79	–	15.56	91.79	–	15.63	91.75	–	15.66
	Self-DA (AAAI' 21) [46]	15.40	91.80	–	24.50	84.40	–	15.60	90.10	–	23.10	84.30	–	19.65
	ANRL (ACM MM' 21) [27]	10.83	96.75	–	17.85	89.26	–	16.03	91.04	–	15.67	91.90	–	15.09
	FGHV (AAAI' 21) [26]	9.17	96.92	–	12.47	93.47	–	16.29	90.11	–	13.58	93.55	–	12.87
	SSDG-R (CVPR' 20) [18]	7.38	97.17	–	10.44	95.94	–	11.71	96.59	–	15.61	91.54	–	11.28
	SSAN-R (CVPR' 22) [48]	6.67	98.75	–	10.00	96.67	–	8.88	96.79	–	13.72	93.63	–	9.80
	PatchNet (CVPR' 22) [42]	7.10	98.46	–	11.33	94.58	–	13.40	95.67	–	11.82	95.07	–	10.90
	GDA (ECCV' 22) [67]	9.20	98.00	–	12.20	93.00	–	10.00	96.00	–	14.40	92.60	–	11.45
0-shot	DiVT-M (WACV' 23) [23]	2.86	99.14	–	8.67	96.62	–	3.71	99.29	–	13.06	94.04	–	7.07
	ViT (ECCV' 22) [16]	**1.58**	**99.68**	**96.67**	5.70	98.91	88.57	9.25	97.15	51.54	7.47	98.42	69.30	6.00
5-shot	ViT (ECCV' 22) [16]	3.42	98.60	95.00	1.98	99.75	94.00	2.31	99.75	87.69	7.34	97.77	66.90	3.76
	ViTAF* (ECCV' 22) [16]	2.92	99.62	91.66	1.40	99.92	98.57	1.64	99.64	91.53	5.39	98.67	76.05	3.31
0-shot	FLIP-V	3.79	99.31	87.99	1.27	99.75	95.85	4.71	98.80	75.84	4.15	98.76	66.47	3.48
	FLIP-IT	5.27	98.41	79.33	**0.44**	**99.98**	99.86	**2.94**	**99.42**	**84.62**	3.61	99.15	84.76	3.06
	FLIP-MCL	4.95	98.11	74.67	0.54	**99.98**	**100.00**	4.25	99.07	**84.62**	**2.31**	**99.63**	**92.28**	**3.01**

- Protocol 2: FLIP-MCL giảm HTER trung bình và cải thiện mạnh mẽ TPR@FPR=1%.

	Method	CS → W			SW → C			CW → S			Avg.
		HTER	AUC	TPR@ FPR=1%	HTER	AUC	TPR@ FPR=1%	HTER	AUC	TPR@ FPR=1%	HTER
0-shot	ViT (ECCV' 22) [16]	7.98	97.97	73.61	11.13	95.46	47.59	13.35	94.13	49.97	10.82
5-shot	ViT (ECCV' 22) [16]	4.30	99.16	83.55	7.69	97.66	68.33	12.26	94.40	42.59	6.06
	ViTAF* (ECCV' 22) [16]	2.91	99.71	92.65	6.00	98.55	78.56	11.60	95.03	60.12	5.12
0-shot	FLIP-V	6.13	97.84	50.26	10.89	95.82	53.93	12.48	94.43	53.00	9.83
	FLIP-IT	4.89	98.65	59.14	10.04	96.48	59.4	15.68	91.83	43.27	10.2
	FLIP-MCL	**4.46**	**99.16**	**83.86**	**9.66**	**96.69**	**59.00**	**11.71**	**95.21**	**57.98**	**8.61**

- Protocol 3: Trong cài đặt khó nhất, FLIP-MCL vẫn vượt SOTA trung bình +8.36 HTER.

	Method	C → I	C → M	C → O	I → C	I → M	I → O	M → C	M → I	M → O	O → C	O → I	O → M	Avg.
0-shot	ADDA (CVPR' 17) [40]	41.8	36.6	-	49.8	35.1	-	39.0	35.2	-	-	-	-	39.6
	DRCN (ECCV' 16) [12]	44.4	27.6	-	48.9	42.0	-	28.9	36.8	-	-	-	-	38.1
	DupGAN (CVPR' 18) [15]	42.4	33.4	-	46.5	36.2	-	27.1	35.4	-	-	-	-	36.8
	KSA (TIFS' 18) [21]	39.3	15.1	-	12.3	33.3	-	9.1	34.9	-	-	-	-	24.0
	DR-UDA (TIFS' 20) [45]	15.6	9.0	28.7	34.2	29.0	38.5	16.8	3.0	30.2	19.5	25.4	27.4	23.1
	MDDR (CVPR' 20) [44]	26.1	20.2	24.7	39.2	23.2	33.6	34.3	8.7	31.7	21.8	27.6	22.0	26.1
	ADA (ICB' 19) [43]	17.5	9.3	29.1	41.5	30.5	39.6	17.7	5.1	31.2	19.8	26.8	31.5	25.0
	USDAN-Un (PR' 21) [19]	16.0	9.2	-	30.2	25.8	-	13.3	3.4	-	-	-	-	16.3
	GDA (ECCV' 22) [67]	15.10	**5.8**	-	29.7	20.8	-	12.2	2.5	-	-	-	-	14.4
	CDFTN-L (AAAI' 23) [56]	**1.7**	8.1	29.9	11.9	9.6	29.9	8.8	**1.3**	25.6	19.1	5.8	6.3	13.2
0-shot	FLIP-V	15.08	13.73	12.34	4.30	9.68	7.87	0.56	3.96	4.79	2.09	5.01	6.00	7.12
	FLIP-IT	12.33	15.18	7.98	1.12	8.37	6.98	**0.19**	5.21	4.96	**0.16**	**4.27**	**5.63**	6.03
	FLIP-MCL	10.57	7.15	**3.91**	**0.68**	**7.22**	**4.22**	**0.19**	5.88	**3.95**	0.19	5.69	8.40	**4.84**

3. Ưu điểm và hạn chế

- Ưu điểm:

- Khả năng tổng quát hóa tốt, ngay cả trong bối cảnh zero-shot.
- Hiệu quả trên dữ liệu hạn chế nhờ sử dụng học tương phản và liên kết văn bản-hình ảnh.
- Giải thích được thông qua các attention maps, mô hình tập trung vào các vùng spoof-specific (như kết cấu giấy, mép hình ảnh, moire patterns).

- Hạn chế:
  - Chi phí tính toán cao hơn do cần bộ mã hóa văn bản.
  - Một số trường hợp bị ảnh hưởng bởi độ phân giải thấp hoặc ánh sáng yếu.

4. Hướng phát triển

- Tối ưu hóa ngôn ngữ: Nghiên cứu phương pháp học prompt để cải thiện hiệu năng.
- Mở rộng ứng dụng: Áp dụng FLIP vào các bài toán khác như nhận diện khuôn mặt trong điều kiện ánh sáng yếu.
- Giảm độ phức tạp: Sử dụng kỹ thuật nén mô hình (quantization) để giảm chi phí tính toán.

3. Kết quả và đánh giá

3.1. Kết quả thực tập

- **Face Mask Classification (FMC)**:
  - Độ chính xác: 94%, cải thiện nhờ xử lý mất cân bằng dữ liệu.

- Object Detection: Độ chính xác 98% và đã được triển khai lên thiết bị di động.

- Face Anti-Spoofing (FAS): Triển khai được các mô hình FLIP với Vision Transformer trên dữ liệu CelebA-Spoof.
- Mạng nơ-ron lượng tử (QCNN): Thành công triển khai một mạng cơ bản để phát hiện trạng thái lượng tử kích thích.

3.2. Khó khăn và bài học

- **Tài nguyên hạn chế**:
  - QCNN yêu cầu phần cứng mạnh mẽ để xử lý lượng tử, nhưng môi trường thực tế lại hạn chế tài nguyên, dẫn đến thời gian huấn luyện kéo dài.
  - Các bước tối ưu đã được áp dụng, nhưng hiệu quả vẫn bị ảnh hưởng bởi hạn chế về GPU và RAM.
- **Thiếu kinh nghiệm triển khai**:
  - Trong quá trình triển khai mô hình trên thiết bị di động (Android), việc chuyển đổi mô hình từ TensorFlow hoặc PyTorch sang định dạng NCNN gặp khó khăn do thiếu tài liệu chi tiết.
  - Đã học được cách khắc phục thông qua thử nghiệm và nghiên cứu các nguồn mở khác nhau.
- **Quản lý pipeline MLOps**:
  - Khó khăn trong việc đồng bộ pipeline giữa các môi trường phát triển, thử nghiệm và sản xuất.
  - Sử dụng MLFlow và Kubeflow để giải quyết, nhưng vẫn cần nhiều thời gian để tinh chỉnh và tự động hóa hoàn toàn.

3.3. Đánh giá chung

- **Kiến thức và kỹ năng**:
  - Củng cố kiến thức về AI/ML và MLOps.
  - Nâng cao kỹ năng xử lý dữ liệu, tối ưu mô hình, và tích hợp công nghệ lượng tử vào AI cổ điển.
- **Ứng dụng thực tế**:
  - Áp dụng thành công các bài toán như Face Anti-Spoofing và Object Detection trên các nền tảng khác nhau.
  - Mở rộng nghiên cứu với mạng nơ-ron lượng tử để giải quyết các bài toán phức tạp hơn.
- **Hướng phát triển**:
  - **Nghiên cứu nâng cao**:
    - Tìm hiểu sâu hơn về các thuật toán lượng tử mới như Quantum GAN (Generative Adversarial Networks).
    - Thử nghiệm kết hợp mạng nơ-ron lượng tử với dữ liệu thực tế lớn hơn để đánh giá hiệu quả.
  - **Mở rộng ứng dụng**:
    - Ứng dụng mạng QCNN vào các bài toán trong xử lý ngôn ngữ tự nhiên (NLP).
    - Phát triển hệ thống bảo mật dựa trên mã hóa lượng tử kết hợp với AI.
  - **Hoàn thiện quy trình MLOps**:
    - Tích hợp các công cụ giám sát tự động cho pipeline.
    - Đẩy mạnh việc tối ưu hóa triển khai mô hình trên môi trường cloud-native để tiết kiệm tài nguyên và chi phí.

PHẦN KẾT LUẬN VÀ KIẾN NGHỊ

1.1. Kết luận

Trong quá trình thực tập, em đã tập trung vào các dự án liên quan đến thị giác máy tính, bao gồm huấn luyện mạng nơ-ron convolutional để phân loại ảnh mặt nạ với độ chính xác 95%, phát hiện đối tượng trong thời gian thực bằng mô hình YOLOv5 và triển khai mô hình lên nền tảng di động sử dụng ONNX Runtime. Qua các dự án này, em đã nâng cao kỹ năng lập trình bằng PyTorch, xử lý dữ liệu bất cân bằng bằng kỹ thuật oversampling và tối ưu hóa hyperparameter để cải thiện hiệu suất mô hình. Đặc biệt, việc triển khai mô hình lên thiết bị di động đã giúp em hiểu rõ hơn về các thách thức liên quan đến việc cân bằng giữa độ chính xác và tốc độ thực thi. Quá trình thực tập này không chỉ giúp em củng cố kiến thức lý thuyết mà còn trang bị cho em những kỹ năng thực tế cần thiết để ứng dụng học máy vào các bài toán thực tế.

1.2. Kiến nghị

- Về phía nhà trường:
  - + Em mong muốn trước mỗi kỳ thực tập nhà trường đưa thông tin chính xác về các đơn vị nhận sinh viên thực tập để tránh xảy ra sự hiểu lầm (ví dụ như việc sinh viên thực tập kỳ này tại CMC Global kết thúc vào ngày 13/12/2024 trong khi sinh viên thực tập tại các đơn vị khác kết thúc vào ngày 29/12/2024 (mặc dù trong quy chế là 16 tuần thực tập), hay thông tin sinh viên có chỗ bán thời gian/ toàn thời gian/ làm việc từ xa…).
  - + Đồng thời, em mong muốn nhà trường sẽ trao đổi rõ những vấn đề công việc thực tập với bên doanh nghiệp để đảm bảo sinh viên có thể đăng ký được các vị trí công việc phù hợp và chắc chắn được rằng sinh viên sẽ thực tập đúng các công việc của vị trí đó (ví dụ: có bao nhiêu vị trí thực tập về kiểm thử, AI, ML...).
- Về phía doanh nghiệp:
  - + Em thấy bên Viện CMC-ATI cần quan tâm tới sinh viên thực tập hơn:
    - Khi mới vào thì sinh viên thực tập cần được phổ biến về nội quy, quy định chấm điểm đối với sinh viên thực tập tập ở Viện (trên thực tế về quy định rõ ràng thì em chưa thấy bên nhân sự phổ biến – và chỉ nhận được hướng dẫn, nội quy từ chuyên viên hướng dẫn).
    - Ngày 15/11/2024, tại Viện có sự kiện và không có nhân viên nào ở lại nhưng sinh viên không nhận được bất kỳ thông báo nào và vẫn đi làm bình thường. Tuy nhiên vì sinh viên thực tập

không được cấp quyền mở cửa nên không thể vào bên trong thực hiện công việc.

+ Môi trường làm việc cần chuyên nghiệp hơn: Nếu sinh viên được làm việc trực tiếp với chuyên viên hướng dẫn thì sẽ học hỏi được nhiều kinh nghiệm hơn so với việc tất cả sinh viên thực tập chỉ ngồi ở phòng thực tập như đi học rồi đến ngày báo cáo thì báo cáo.