

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA CÔNG NGHỆ THÔNG TIN**

Tel. (+84.0236) 3736949, Fax. (+84.0236) 3842771  
Website: <http://dut.udn.vn/khoacntt> , E-mail: cntt@dut.udn.vn



**BÁO CÁO**  
**THỰC TẬP TỐT NGHIỆP**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**  
**Image Colorization**

**CÔNG TY THỰC TẬP:**  
**Công ty Công nghệ thông tin VNPT (VNPT-IT)**  
**Trung tâm VNPT-IT Khu Vực III**

**SINH VIÊN : Vũ Hoàng Tín**  
**MÃ SINH VIÊN : 102210331**  
**LỚP SH : 21TCLC\_KHDL2**  
**NHÓM HP : 21.Nh15**  
**CBHD : Trần Quốc Nam**

**Đà Nẵng, 03/2024**

## LỜI CẢM ƠN

*Em xin gửi lời cảm ơn sâu sắc đến thầy Ly, cô Hạnh và các anh chị, cô chú ở Trung tâm VNPT-IT khu vực III đã tạo điều kiện cho em có cơ hội thực tập, học hỏi và trau dồi kiến thức trong thời gian qua.*

*Đặc biệt em xin cảm ơn đến anh Nam, người đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình thực tập. Sự chỉ bảo và những chia sẻ quý báu của anh là hành trang mà em sẽ mang theo và áp dụng trên con đường nghề nghiệp sau này.*

*Em cũng xin gửi lời cảm ơn đến các anh/chị tại Trung tâm VNPT khu vực III đã giúp đỡ, tạo điều kiện cho em có kỳ thực tập thành công. Môi trường làm việc chuyên nghiệp và thân thiện này sẽ mãi là kỷ niệm đáng nhớ của em.*

*Do giới hạn về thời gian và kiến thức còn hạn chế, báo cáo thực tập của em không tránh khỏi thiếu sót. Em mong nhận được những ý kiến đóng góp quý báu để có thể cải thiện và hoàn thiện hơn trong tương lai.*

*Em xin chân thành cảm ơn!*

**Họ và tên, chữ ký của sinh viên**

**PHIẾU ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP**

Họ và Tên sinh viên: *Vũ Hoàng Tín*    Lớp: *21TCLC\_KHDL2*    Nhóm: *21.Nh15*

Cơ quan/Đơn vị thực tập: *Trung tâm VNPT-IT khu vực III*

Địa chỉ: *344 đường 2/9 Đà Nẵng*

Thời gian thực tập từ *10/02/2025* đến *22/03/2025*

Người hướng dẫn: *Trần Quốc Nam*

Email: *vuhoangtin@gmail.com*                      Điện thoại: *0865316559*

**1. Đánh giá về năng lực chuyên môn**

Nội dung đánh giá	Xuất sắc	Tốt	Khá	T.Bình	Yếu
Năng lực chuyên môn đáp ứng công việc					
Hoàn thành các công việc được giao					
Khả năng sử dụng ngoại ngữ					
Ứng dụng kết quả thực tập cho cơ quan					

**2. Đánh giá về ý thức làm việc**

Nội dung đánh giá	Xuất sắc	Tốt	Khá	T.Bình	Yếu
Tinh thần, thái độ làm việc					
Tuân thủ kỷ luật (thời gian làm việc, báo nghỉ...)					
Giao tiếp, quan hệ với cán bộ, công nhân viên					

**3. Đánh giá kết quả công việc**

Nội dung đánh giá	Xuất sắc	Tốt	Khá	T.Bình	Yếu
Khả năng phân tích thiết kế hệ thống					
Kỹ năng lập trình					
Khả năng học hỏi, nắm bắt công nghệ mới					

**4. Các nhận xét khác (nếu có)**

.....  
5. **Điểm đánh giá.**                      Ghi bằng số:...../10                      Ghi bằng chữ:.....

*Xác nhận của cơ quan/đơn vị thực tập*  
(Ký, ghi rõ họ tên và đóng dấu)

Ngày ..... tháng ..... năm 20.....  
*Người hướng dẫn*  
(Ký và ghi rõ họ tên)

## This image shows a single page of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU CƠ QUAN THỰC TẬP.....	7
1.1. GIỚI THIỆU VỀ CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT-IT.....	7
1.2. VĂN HÓA VÀ CÁC QUY ĐỊNH CỦA CÔNG TY.....	10
1.3. KẾT CHƯƠng.....	12
CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG .....	13
2.1. MÔ TẢ CÔNG VIỆC.....	13
2.2. KẾT QUẢ CÔNG VIỆC .....	34
2.3. KẾT CHƯƠng.....	34
CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG .....	36
3.1. KẾT QUẢ TRIỂN KHAI.....	36
3.2. KẾT CHƯƠng.....	39

## DANH SÁCH HÌNH

Hình 1. Logo VNPT.....	7
Hình 2: Các mốc phát triển của VNPT. ....	8
Hình 3. Mô hình tổ chức của công ty VNPT-IT. ....	8
Hình 4: Chiến lược sản phẩm.....	9
Hình 5: Sản phẩm dịch vụ dẫn dắt thị trường. ....	9
Hình 6: Sản phẩm đạt nhiều giải thưởng.....	9
Hình 7. Cấu trúc mô hình U-Net.....	14
Hình 8. Encoder của U-Net.....	15
Hình 9. Encoder của U-Net.....	16
Hình 10. Tổng quan kiến trúc ResNet-18 .....	18
Hình 11. Tổng quan về mô hình GAN.....	21
Hình 12. Công thức hàm mất mát cGAN.....	23
Hình 13. Công thức hàm mất mát L1 .....	24
Hình 14. Công thức hàm mất mát kết hợp .....	25
Hình 15. Bảng thống kê kích thước các đối tượng được đánh dấu.....	26
Hình 16. Phân bố các đối tượng được đánh dấu .....	26
Hình 17. Số lượng các đối tượng trong mỗi ảnh.....	27
Hình 18. Không gian RGB (Red - Green - Blue).....	27
Hình 19. Không gian $L^*a^*b$ .....	28
Hình 20. Tải dữ liệu và chia dữ liệu.....	29
Hình 21. Xử lý dữ liệu .....	30
Hình 22. Xây dựng mô hình.....	30
Hình 23. Theo dõi loss. ....	31

Hình 24. Hàm huấn luyện. ....	31
Hình 24. Huấn luyện mô hình .....	32
Hình 25. Generator.....	32
Hình 26. Discriminator.....	32
Hình 27. Hàm mất mát GAN .....	33
Hình 28. Mô hình GAN .....	33
Hình 29. Sơ đồ tổng quát hệ thống Image Colorization .....	33
Hình 30. Kết quả huấn luyện mô hình Resnet19_Unet ở những epoch cuối.....	34
Hình 31. Training Loss_L1 Res18-Unet.....	36
Hình 32. Training Loss GAN.....	36
Hình 33. Kết quả khi train mô hình .....	37
Hình 34. Đưa vào các input.....	37
Hình 35. Kết quả trả về .....	37
Hình 36. So sánh kết quả trả về của mô hình Unet không/ có sử dụng GAN .....	38
Hình 37. Công thức tính chỉ số colorfulness.....	39

# DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
G	Generator
D	Discriminator
GAN	Generative Adversarial Network



# CHƯƠNG 1: GIỚI THIỆU CƠ QUAN THỰC TẬP

## 1.1. GIỚI THIỆU VỀ CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT-IT

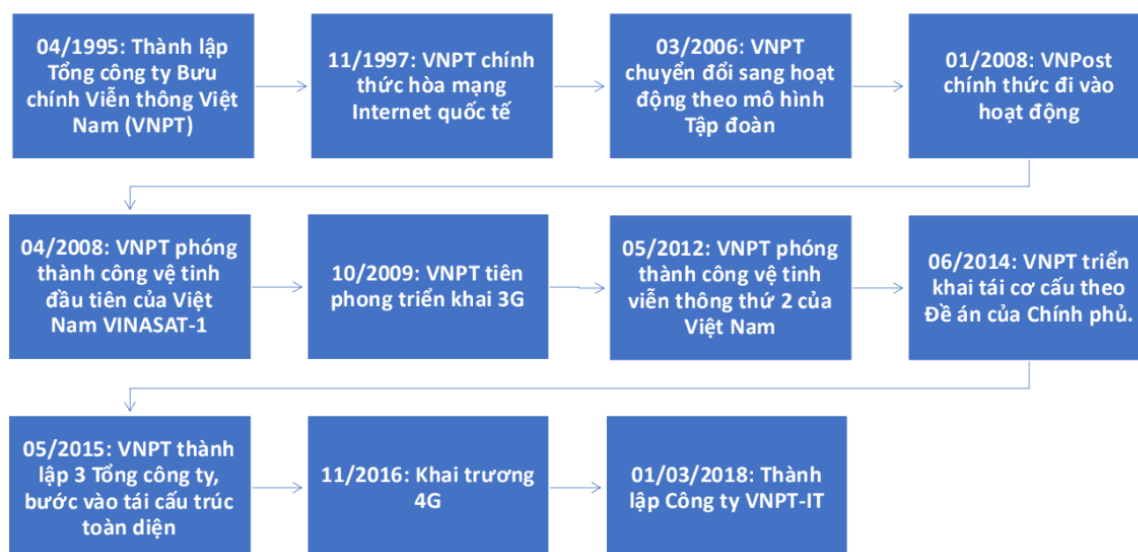
### 1.1.1. Thông tin chung

- Tên giao dịch bằng tiếng Việt: Công ty Công Nghệ thông tin VNPT.
- Tên giao dịch quốc tế: VNPT Information Technology Company.
- Tên viết tắt: VNPT-IT.
- Ngày thành lập: 01/03/2018.
- Trụ sở chính: Số 57 phố Huỳnh Thúc Kháng, Phường Láng Hạ, Quận Đống Đa, Hà Nội.
- Điện thoại: (84-24) 3385.5588 | Fax: (84-24) 3553.3388
- Website: <http://www.vnptit.vn>



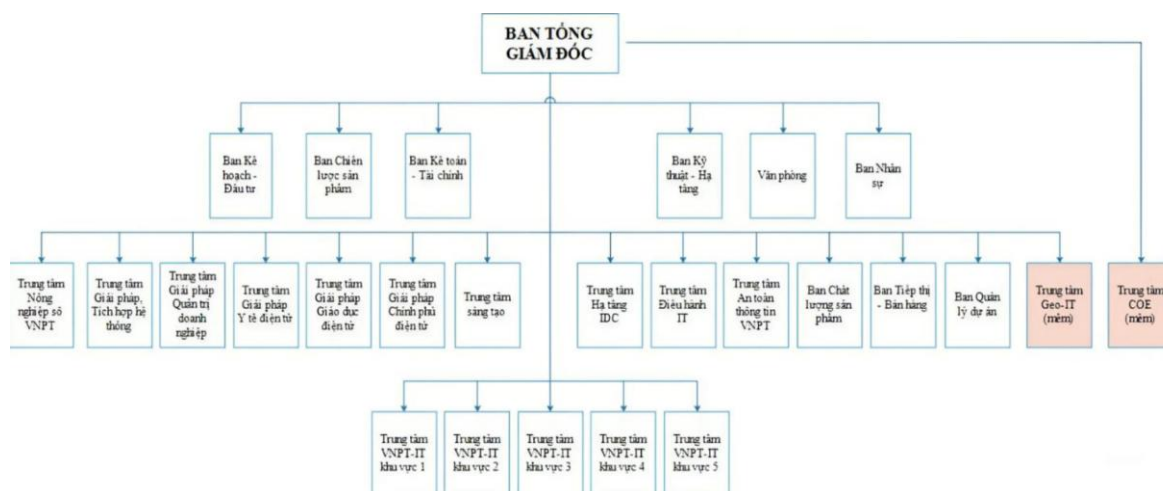
Hình 1. Logo VNPT.

### 1.1.2. Lịch sử hình thành và phát triển:



Hình 2: Các mốc phát triển của VNPT.

### 1.1.3. Mô hình tổ chức



Hình 3. Mô hình tổ chức của công ty VNPT-IT.

### 1.1.4. Sứ mệnh, tầm nhìn và giá trị cốt lõi:

#### ❖ Sứ mệnh

- Cung cấp cho khách hàng và đối tác các sản phẩm, dịch vụ Viễn thông - CNTT.
- CNTT – Truyền thông và Dịch vụ số chất lượng, đột phá, sáng tạo.
- Trở thành trung tâm số (Digital Hub) của khu vực châu Á.
- Tôn vinh và đánh giá giá trị đích thực của người lao động.
- Tiên phong trong các hoạt động về cộng đồng.

#### ❖ Tầm nhìn

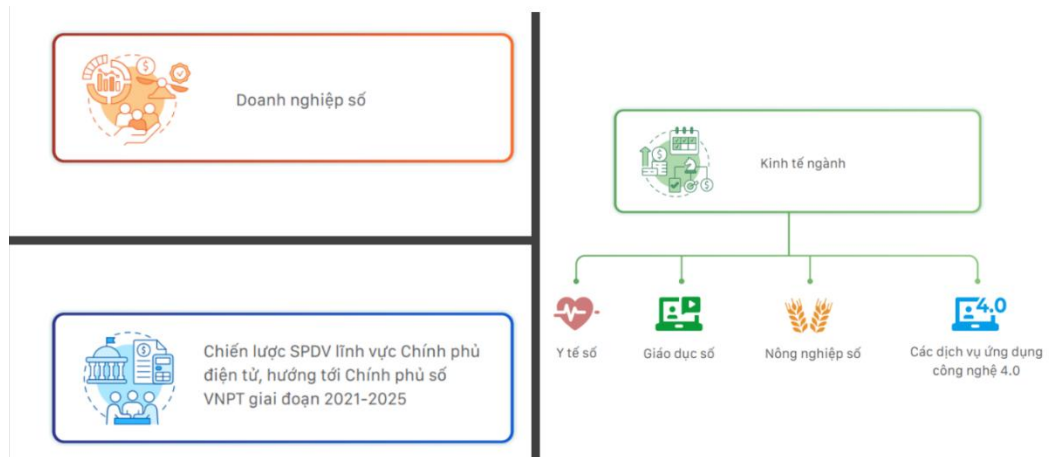
- Trở thành nhà cung cấp dịch vụ số hàng đầu Việt Nam vào năm 2025.
- Trở thành Trung tâm số (Digital Hub) của châu Á vào năm 2030.
- Trở thành lựa chọn số 1 của khách hàng sử dụng sản phẩm, dịch vụ Công nghệ thông tin – Viễn thông (ICT) tại thị trường.

#### ❖ Giá trị cốt lõi

- Con người là chìa khoá.
- Khách hàng là trung tâm.
- Sáng tạo không ngừng.
- Đối tác đáng tin cậy.

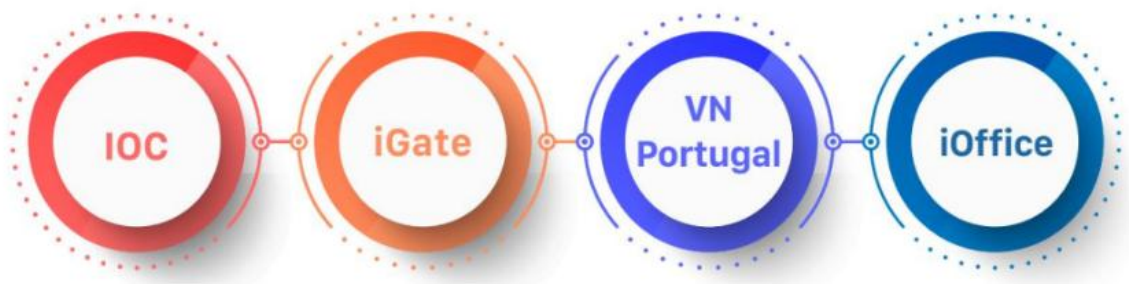
### 1.1.5. Lĩnh vực hoạt động

#### ❖ Chiến lược sản phẩm



Hình 4: Chiến lược sản phẩm.

#### ❖ Các sản phẩm dịch vụ



Hình 5: Sản phẩm dịch vụ dẫn dắt thị trường.



Hình 6: Sản phẩm đạt nhiều giải thưởng.

### 1.1.6. Giải thưởng, thành tựu đạt được

- ❖ Năm 2018: VNPT IT đạt 11 giải thưởng (Stevie Awards; Sao Khuê; Nhân tài đất việt; Top 3 doanh nghiệp có giá trị thương hiệu lớn nhất Việt Nam...).

- ❖ Năm 2019: VNPT IT đạt 23 giải thưởng (Stevie Awards; Nhân tài đất việt; Giải thưởng Chuyển đổi số Việt Nam 2019; Sao Khuê; Top 5 doanh nghiệp CNTT; Top 10 doanh nghiệp công nghệ uy tín...)
- ❖ Năm 2020: VNPT-IT đạt 54 giải thưởng (Giải Nhất sản phẩm công nghệ số Make in Vietnam; đạt giải Ba trong hạng mục Sản phẩm số xuất sắc; Đơn vị viễn thông sáng tạo nhất châu Á tại cuộc thi Asia Communication Awards...).
- ❖ Năm 2021: VNPT-IT đạt 62 giải thưởng (Globe Business Awards; Stevie Awards Châu Á – Thái Bình Dương 2021; Giải thưởng CNTT Thế giới 2021 (IT World Awards); Giải thưởng IBA Stevie Awards 2021...).
- ❖ Năm 2022: VNPT-IT đạt 40 giải thưởng: Sao Khuê; Stevie Awards; Make In Vietnam; Giải thưởng CNTT Thế giới 2022 (Information Technology World Awards);...
- ❖ Năm 2023: đạt 39 Giải thưởng: Công ty có môi trường công nghệ tốt nhất lĩnh vực Viễn thông (A14VN 2023); Vô địch đấu trường An toàn thông tin (Security Bootscamp); Giải 3 Make In Việt Nam.....

## **1.2. VĂN HÓA VÀ CÁC QUY ĐỊNH CỦA CÔNG TY**

### **1.2.1. Tầm nhìn**

- ❖ Nằm trong danh sách 10 nhà cung cấp dịch vụ CNTT hàng đầu trong khu vực và trên toàn châu Á.
- ❖ Là lựa chọn số 1 cho khách hàng trong ngành CNTT.
- ❖ Đến năm 2025, VNPT trở thành doanh nghiệp số thông minh, năng động, hiệu quả, có vai trò quan trọng trong Bản đồ số quốc gia của Việt Nam, đồng thời, thâm nhập thành công thị trường khu vực, quốc tế.
- ❖ Đến năm 2030, VNPT là Platform of Every Customer, sở hữu Ecosystem of Everything, có vai trò tiên phong dẫn dắt thực hiện chuyển đổi số quốc gia, là Trung tâm giao dịch số của khu vực Châu Á

### **1.2.2. Triết lý**

- ❖ Cách tiếp cận lấy khách hàng làm trung tâm: luôn đặt khách hàng làm cốt lõi trong chiến lược kinh doanh và phát triển.
- ❖ Phương pháp cam kết chất lượng: Chất lượng là trọng tâm trong công việc và giá trị cung cấp cho khách hàng.

- ❖ Phương pháp đảm bảo hiệu suất: coi năng suất là thước đo tăng trưởng, lợi thế cạnh tranh cũng như trách nhiệm của công ty đối với xã hội.

### **1.2.3. Sứ mệnh**

- ❖ Cung cấp một cơ sở hạ tầng CNTT hiện đại và mạnh mẽ cho sự phát triển kinh tế xã hội của đất nước.
- ❖ Đáp ứng nhu cầu về CNTT của khách hàng mọi lúc, mọi nơi.
- ❖ Tôn vinh giá trị đích thực của nhân viên trong môi trường kinh doanh mới.
- ❖ Tham gia các sáng kiến trong các hoạt động trách nhiệm xã hội.

### **1.2.4. Giá trị cốt lõi**

- ❖ Con người là chìa khoá.
- ❖ Khách hàng là trung tâm.
- ❖ Sáng tạo không ngừng.
- ❖ Đối tác đáng tin cậy.

### **1.2.5. Nguyên tắc nơi làm việc**

- ❖ Chế độ làm việc theo giờ hành chính: 08 giờ trong một ngày và 40 giờ/01 tuần.
- ❖ Thời gian làm việc từ thứ 2 đến thứ 6: Sáng từ 8h -12h, chiều từ 13h - 17h.
- ❖ Chế độ nghỉ phép: người lao động có đủ 12 tháng làm việc tại công ty thì được nghỉ tiêu chuẩn là 12 ngày, ngày nghỉ của người lao động được tăng thêm theo thâm niên làm việc, cứ 05 năm được nghỉ thêm 01 ngày.
- ❖ Đảm bảo an toàn thông tin cho hoạt động của các cơ sở hạ tầng thông tin, không được phép sao chép, lưu trữ, cung cấp, chia sẻ, phát tán các thông tin bảo mật.
- ❖ Bảo vệ tài sản, thương hiệu, uy tín và bí mật công nghệ, sở hữu kinh doanh, sở hữu trí tuệ của VNPT-IT.

### **1.2.6. Văn hóa nơi công sở**

- ❖ Trang phục văn minh lịch sự, phù hợp với nơi làm việc. Mặc đồng phục vào thứ 2 hàng tuần và trong các sự kiện của đơn vị. Đeo thẻ nhân viên trong thời gian làm việc.
- ❖ Giao tiếp: sử dụng ngôn ngữ phù hợp, niềm nở, tôn trọng KH và đồng nghiệp. Cúi chào KH, đối tác khi gặp hoặc ra về.

- ❖ Điện thoại, email: sử dụng số điện thoại Vinaphone trong quá trình xử lý công việc. Sử dụng email VNPT để giao tiếp với các đồng nghiệp, khách hàng, nhà cung cấp,...
- ❖ An toàn, VSLĐ: sắp xếp khoa học gọn gàng, giữ gìn sạch sẽ nơi làm việc. Tắt các thiết bị điện không cần thiết trước khi rời khỏi.

#### **1.2.7. Văn hóa ứng xử với công việc**

- ❖ Phong cách làm việc chuyên nghiệp, luôn đúng giờ, chủ động trong công việc, làm việc hết mình.
- ❖ Hợp tác trong công việc: chia sẻ thông tin và tăng cường làm việc nhóm để đem lại hiệu quả cao trong công việc.
- ❖ Có tinh thần thái độ với công việc: tôn trọng tự hào với công việc của mình mọi lúc mọi nơi.
- ❖ Học hỏi không ngừng, tư duy tích cực, linh hoạt, sẵn sàng đón nhận sự thất bại để hướng tới thành công

### **1.3. KẾT CHƯƠNG**

Trong chương này, em đã có dịp tìm hiểu chi tiết về Công ty Công nghệ Thông tin VNPT-IT – đơn vị mà em thực tập. Qua quá trình nghiên cứu, em đã nắm bắt được quá trình hình thành, cơ cấu tổ chức cũng như những thành tựu nổi bật mà công ty đã đạt được. Điều này giúp em hiểu rõ hơn về quy mô hoạt động và vai trò quan trọng của VNPT-IT trong ngành công nghệ thông tin tại Việt Nam.

Ngoài ra, việc tìm hiểu về văn hóa doanh nghiệp, các quy tắc làm việc và những giá trị cốt lõi của công ty đã mang lại cho em góc nhìn tổng quan về một môi trường làm việc chuyên nghiệp. Những kiến thức này không chỉ giúp ích cho em trong quá trình thực tập mà còn là nền tảng quan trọng để em áp dụng trong sự nghiệp sau này.

Trong những chương tiếp theo, em sẽ đi sâu vào những nhiệm vụ cụ thể mà em đã đảm nhận, chia sẻ những trải nghiệm thực tế và đúc kết những bài học quan trọng mà em có được trong suốt thời gian thực tập tại VNPT-IT.

## CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ VÀ TRIỂN KHAI HỆ THỐNG

### 2.1. MÔ TẢ CÔNG VIỆC

#### 2.1.1. Phương thức làm việc

- ❖ Trong quá trình thực tập, công việc được thực hiện theo phương thức nhóm và cá nhân. Cụ thể:
  - Làm việc nhóm: các thực tập sinh được chia vào các nhóm đúng chuyên ngành. Hàng tuần vào thứ 3 và thứ 6, nhóm thực tập sinh AI sẽ đến phòng họp để thảo luận cùng với người hướng dẫn để đề xuất ra phương pháp tiếp cận phù hợp. Ngoài thời gian cố định, các thực tập sinh có thể trao đổi và làm việc chung thông qua nền tảng trực tuyến.
  - Làm việc cá nhân: thực hiện việc thu thập tài liệu về đề tài, nghiên cứu đề tài, thực hiện triển khai mô hình đầy đủ bao gồm cả việc tìm kiếm và thu thập dữ liệu, xử lý dữ liệu, huấn luyện mô hình và đánh giá kết quả.
- ❖ Các công việc thực hiện của thực tập sinh đều được thông qua người hướng dẫn.
- ❖ Các nền tảng làm việc và chia sẻ thông tin nội bộ: Telegram, Zalo.

#### 2.1.2. Quy trình làm việc

Quá trình thực tập bao gồm các nhiệm vụ chính như sau:

- Tuần 1: Tìm hiểu về cơ quan thực tập, chuẩn bị đề tài thực tập và kế hoạch thực hiện.
- Tuần 2: Chốt đề tài thực hiện. Tiến hành khảo sát và thu thập dữ liệu, tài liệu về mô hình cần thiết, đề xuất giải pháp.
- Tuần 3, 4: Xây dựng mô hình.
- Tuần 5: Huấn luyện mô hình.
- Tuần 6: Đánh giá kết quả và viết báo cáo.

#### Đề tài thực tập

Tên đề tài: Image Colorization.

Tổng quan: đề tài nhằm phát triển một hệ thống hoặc mô hình trí tuệ nhân tạo có khả năng tự động chuyển đổi hình ảnh từ dạng đơn sắc (grayscale) sang hình ảnh đầy đủ màu sắc (colorful). Điều này không chỉ giúp tái hiện các hình ảnh cũ, đen trắng trở nên sống động và gần gũi với thực tế, mà còn ứng dụng trong nhiều lĩnh vực khác như phục chế hình ảnh lịch sử, cải thiện chất lượng hình ảnh y tế, và hỗ trợ chỉnh sửa ảnh trong nhiếp ảnh.

Mô hình được sử dụng: U-Net, ResNet-18, và GAN.

#### Kế hoạch thực hiện

- Tuần 2:
  - Thu thập các tài liệu cần thiết về U-net, Resnet-18 và GAN.
  - Thu tập dữ liệu.

- Khám phá dữ liệu.
- Tiền xử lý dữ liệu.
- Tuần 3: Xây dựng mô hình kết hợp của ResNet-18 và Unet.
- Tuần 4: Xây dựng mô hình cGAN.
- Tuần 5: Huấn luyện mô hình G-Net và cGAN.
- Tuần 6: Nhận xét và đánh giá kết quả, viết báo cáo.

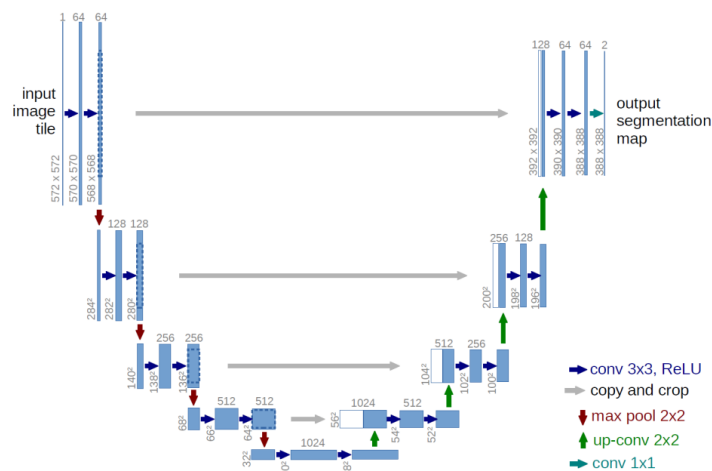
## 🚩 Tìm hiểu về U-Net, Resnet-18 và GAN

### U-Net

U-Net là một mô hình học sâu được phát triển lần đầu tiên bởi Ronneberger và cộng sự vào năm 2015, với mục đích chính là giải quyết các bài toán phân đoạn ảnh trong lĩnh vực y tế.

Điểm mạnh nổi bật của U-Net là khả năng hoạt động hiệu quả ngay cả trên các tập dữ liệu nhỏ, nhờ vào việc tận dụng tối đa thông tin từ ảnh thông qua các skip connections. Mô hình U-Net có cấu trúc hình chữ U, với các phần đối xứng liên kết giữa Encoder (bên trái) và Decoder (bên phải). Nhờ đó mô hình có khả năng phân đoạn hình ảnh mạnh mẽ.

Hiện nay, U-Net đã được ứng dụng rộng rãi không chỉ trong y tế mà còn trong nhiều lĩnh vực khác như xử lý ảnh vệ tinh, thị giác máy tính, và phân đoạn ảnh trong thực tế tăng cường.



Hình 7. Cấu trúc mô hình U-Net

### ***Ý tưởng chính:***

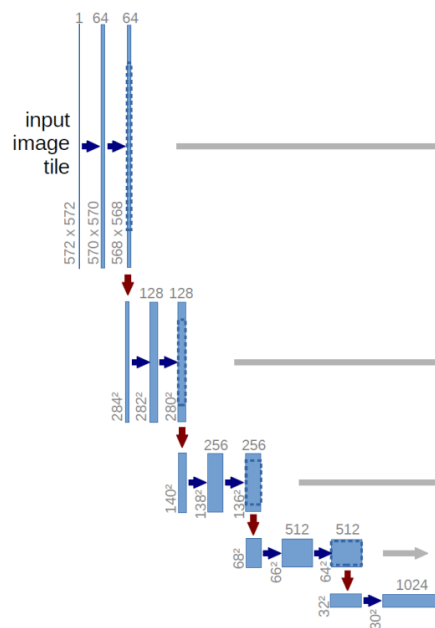
Từ 1 ảnh ban đầu, mô hình sử dụng các lớp Conv2D để trích xuất đặc trưng. Sau đó đặc trưng được downscale đưa về lớp fully connected. Tiếp theo, lớp Conv2D transpose sẽ upscale đặc trưng lại về kích thước ban đầu. Tuy nhiên mô hình này khá sâu nên để mô hình không bị quên đi các đặc trưng từ những lớp ban đầu thì mô hình sẽ đưa những đặc trưng từ lớp ban đầu vào quá trình huấn luyện của các lớp sau.

### ***Kiến trúc:***

#### ❖ Phần Encoder (Bộ mã hóa)



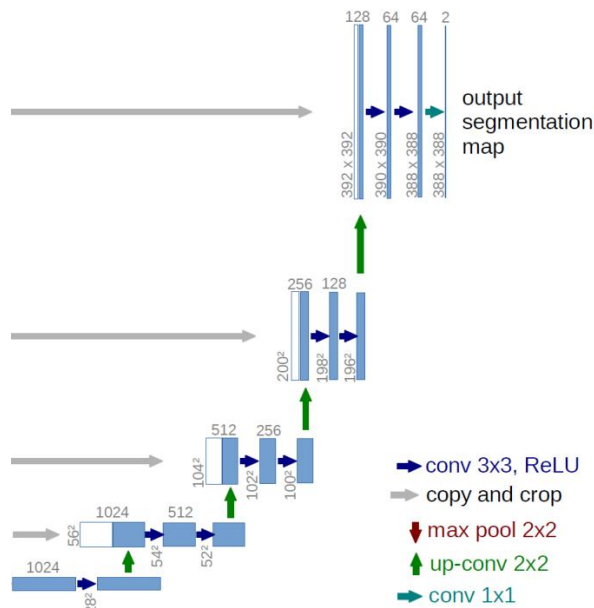
- Chức năng: Trích xuất đặc trưng từ ảnh đầu vào.
- Cấu trúc:
  - Bao gồm các khối convolution 2D với kích thước kernel nhỏ (3x3), stride = 1, và hàm kích hoạt ReLU.
  - Sau mỗi convolution, sử dụng max pooling (2x2) để giảm kích thước không gian và tăng số lượng kênh đặc trưng.
  - Số lượng filters tăng dần qua các tầng, thường theo bội số của 2 (ví dụ: 64, 128, 256,...).



Hình 8. Encoder của U-Net.

#### ❖ Phần Decoder (Bộ giải mã)

- Chức năng: Tái tạo lại kích thước không gian của ảnh đầu vào, đồng thời duy trì thông tin đặc trưng.
- Cấu trúc:
  - Sử dụng các khối up-sampling hoặc transpose convolution để tăng kích thước không gian.
  - Kết hợp với thông tin từ các tầng tương ứng của Encoder thông qua skip connections.
  - Sau mỗi bước tăng kích thước, áp dụng các lớp convolution để tinh chỉnh đặc trưng.



Hình 9. Encoder của U-Net.

### ❖ Skip Connections

#### ○ Vai trò:

- Kết nối thông tin giữa các tầng tương ứng của Encoder và Decoder, giúp mô hình giữ lại thông tin không gian bị mất trong quá trình pooling.
- Tăng hiệu quả trong việc học đặc trưng cục bộ và giảm thiểu hiện tượng vanishing gradient.

### ❖ Output Layer

- Chức năng: Tạo ra kết quả phân đoạn cuối cùng.
- Cấu trúc: Sử dụng convolution 1x1 với số lượng kênh bằng số lớp (classes) cần phân đoạn, kết hợp với hàm kích hoạt (softmax hoặc sigmoid) để tạo ra xác suất cho mỗi pixel.

### Nguyên lý hoạt động

#### ❖ Tiền xử lý dữ liệu

Trước khi đưa dữ liệu vào mô hình U-Net, việc tiền xử lý là bước cần thiết để đảm bảo dữ liệu có chất lượng tốt và phù hợp với yêu cầu của mô hình.

- Chuẩn hóa dữ liệu: Dữ liệu ảnh thường được chuẩn hóa để các giá trị pixel nằm trong khoảng  $[0, 1]$  hoặc  $[-1, 1]$ , giúp giảm độ lệch giữa các giá trị và tăng tốc độ hội tụ khi huấn luyện.

- Điều chỉnh kích thước ảnh: Ảnh đầu vào thường được điều chỉnh về một kích thước cố định (như  $128 \times 128$  hoặc  $256 \times 256$ ) để phù hợp với kiến trúc

U-Net.

- Tăng cường dữ liệu (Data Augmentation): Với các tập dữ liệu nhỏ, tăng cường dữ liệu được áp dụng để tạo ra nhiều biến thể khác nhau của ảnh gốc như xoay, lật, cắt xén, thay đổi độ sáng, hoặc co giãn. Điều này giúp mô hình học được các đặc trưng tốt hơn và tránh overfitting.

#### ❖ Giai đoạn mã hóa (Encoder)

Giai đoạn mã hóa là bước đầu tiên trong kiến trúc U-Net, nơi thông tin từ ảnh được trích xuất thành các đặc trưng quan trọng.

- Trích xuất đặc trưng cục bộ:
  - Ảnh đầu vào được đưa qua các tầng convolution 2D. Các convolution này sử dụng kernel nhỏ (kích thước 3x3) để phát hiện các đặc trưng cục bộ như cạnh, góc, hoặc kết cấu.
  - Sau mỗi bước convolution, hàm kích hoạt ReLU được sử dụng để tạo ra sự phi tuyến tính, giúp mô hình học được các đặc trưng phức tạp hơn.
- Giảm kích thước không gian: Max pooling (2x2) được sử dụng để giảm kích thước không gian của ảnh. Việc giảm kích thước này giúp loại bỏ các chi tiết không quan trọng và tập trung vào các đặc trưng chính
- Tăng chiều đặc trưng: Khi kích thước không gian giảm, số lượng filters (chiều đặc trưng) tăng dần để học được các đặc trưng trừu tượng hơn. Ví dụ, số lượng filters thường tăng từ 64 → 128 → 256 qua các tầng.

#### ❖ Giai đoạn giải mã (Decoder)

- Giai đoạn giải mã là bước tái dựng lại kích thước ảnh về đúng kích thước ban đầu, đồng thời sử dụng các đặc trưng học được từ Encoder để tạo ra kết quả phân đoạn.
  - Up-sampling để tái dựng kích thước không gian
    - Tại mỗi tầng Decoder, ảnh đặc trưng được tăng kích thước không gian thông qua các phép up-sampling (ví dụ: bilinear interpolation) hoặc transpose convolution.
  - Kết hợp thông tin từ skip connections
    - Skip connections giúp kết nối các tầng tương ứng giữa Encoder và Decoder. Thông tin từ tầng Encoder (kích

thước không gian lớn hơn) được kết hợp với thông tin của tầng Decoder (đang trong quá trình tái dựng).

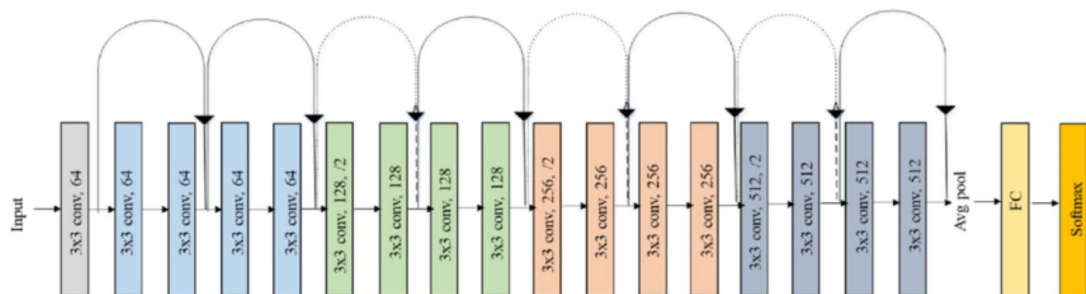
- Kết hợp này giúp mô hình duy trì thông tin chi tiết của ảnh ban đầu, tăng độ chính xác của dự đoán.
- Convolution tinh chỉnh: Sau mỗi bước up-sampling, các lớp convolution (3x3) được áp dụng để tinh chỉnh thông tin và giảm nhiễu, giúp kết quả phân đoạn rõ ràng hơn.

## **ResNet-18**

ResNet (Residual Neural Network) là một loại mạng nơ-ron sâu được thiết kế để giải quyết vấn đề suy giảm độ chính xác khi mạng trở nên quá sâu, thường được gọi là vấn đề vanishing gradient. ResNet-18 là một biến thể của ResNet, với độ sâu 18 lớp, được thiết kế để cân bằng giữa hiệu năng và tốc độ tính toán, phù hợp cho các ứng dụng yêu cầu thời gian thực hoặc tài nguyên hạn chế.

### ***Kiến trúc***

- ❖ ResNet-18 được chia thành 4 giai đoạn chính, mỗi giai đoạn bao gồm nhiều khối residual.
- ❖ Kích thước của ảnh được giảm dần thông qua các lớp pooling, trong khi chiều đặc trưng (số lượng kênh) tăng lên theo từng giai đoạn.
- ❖ Số lớp trọng số: 17 lớp convolution và 1 lớp fully connected cuối cùng.



*Hình 10. Tổng quan kiến trúc ResNet-18*

### ***Thành phần***

#### **❖ Convolutional Layers**

- Sử dụng các kernel nhỏ (3x3) để học đặc trưng, với padding để giữ nguyên kích thước không gian.
- Tích hợp Batch Normalization (BN) và hàm kích hoạt ReLU để tăng tốc độ hội tụ và giảm hiện tượng overfitting.

#### **❖ Residual Block (Khối dư)**

- Mỗi khối residual bao gồm hai lớp convolution và một skip connection.

- Skip connection cho phép tín hiệu đầu vào được cộng trực tiếp với tín hiệu đầu ra của khối, giúp duy trì thông tin gốc và cải thiện khả năng lan truyền gradient.
- Công thức chính

$$y = F(x, W) + x$$

Trong đó  $F(x, W)$  là kết quả của các khối convolution đầu tiên.

#### ❖ Pooling Layers

- Max Pooling: Giảm kích thước không gian sau các tầng convolution đầu tiên.
- - Global Average Pooling (GAP): Tích hợp thông tin từ toàn bộ bản đồ đặc trưng trước khi đưa vào lớp fully connected.

#### ❖ Fully Connected Layer

- Lớp cuối cùng dự đoán kết quả phân loại, với số đầu ra tương ứng với số lớp của bài toán.

### *Nguyên lý hoạt động*

#### ❖ Trích xuất đặc trưng

- Dữ liệu đầu vào (ảnh) được đưa qua lớp convolution đầu tiên để phát hiện các đặc trưng cơ bản như cạnh và góc.
- Các tầng tiếp theo trong mỗi giai đoạn học các đặc trưng ngày càng trừu tượng hơn, như hình dạng hoặc đối tượng trong ảnh.

#### ❖ Khối Residual

Skip connection trong các khối residual đóng vai trò quan trọng:

- Cải thiện khả năng học: Skip connection cho phép tín hiệu gốc (input) được cộng vào tín hiệu đã qua xử lý, giúp mô hình học các biến đổi nhỏ hơn (residual mapping) thay vì học toàn bộ phép biến đổi.
- Giảm vanishing gradient: Nhờ kết nối trực tiếp, gradient có thể truyền ngược qua skip connection mà không bị làm nhỏ đi đáng kể.

#### ❖ Giảm kích thước không gian

Max pooling và stride convolution được sử dụng để giảm kích thước không gian của bản đồ đặc trưng, giúp giảm số lượng tham số và tải nguyên tính toán.

#### ❖ Tổng hợp thông tin

- Trước khi dự đoán, ResNet-18 sử dụng global average pooling (GAP) để nén thông tin từ toàn bộ bản đồ đặc trưng thành một vector duy nhất.
- Lớp fully connected cuối cùng thực hiện phân loại dựa trên vector này

### **Tích hợp ResNet-18 vào phần Encoder của U-Net**

Trong bài toán Image Colorization, mục tiêu là chuyển đổi hình ảnh từ không gian màu xám (grayscale) sang hình ảnh đầy đủ màu sắc (colorful). Để đạt được điều này, mô hình cần học cách tái tạo thông tin màu sắc từ các đặc trưng trích xuất của ảnh đầu vào. Mô hình U-Net truyền thống với cấu trúc Encoder-Decoder, được chọn làm nền tảng vì khả năng học và tái dựng thông tin không gian tốt.

Tuy nhiên, để cải thiện chất lượng tái dựng màu sắc và khai thác tốt hơn các đặc trưng trích xuất từ ảnh xám, phần Encoder của U-Net đã được thay thế bằng kiến trúc ResNet-18. Với khả năng học đặc trưng sâu và hiệu quả của ResNet-18, mô hình mới không chỉ tái dựng chính xác các chi tiết không gian mà còn tạo ra những bức ảnh màu sắc sống động và tự nhiên hơn. Điều này là nhờ vào việc ResNet-18 có thể khai thác tốt hơn các mối quan hệ phức tạp giữa cường độ xám và thông tin màu sắc.

### ***Lý do chọn ResNet-18***

- ❖ *Học đặc trưng sâu:* ResNet-18 với các khối residual giúp học các đặc trưng trừu tượng từ dữ liệu đầu vào, hỗ trợ việc tái tạo thông tin màu sắc phức tạp.
- ❖ *Giảm hiện tượng mất thông tin:* Các skip connections trong ResNet-18 cho phép lan truyền gradient hiệu quả hơn trong quá trình huấn luyện, đảm bảo rằng thông tin quan trọng từ ảnh đầu vào được bảo toàn.
- ❖ *Hiệu suất cao và phù hợp với tài nguyên tính toán:* ResNet-18 có cấu trúc gọn nhẹ và hiệu quả so với các mạng lớn hơn như ResNet50, đảm bảo mô hình đạt chất lượng cao mà vẫn tiết kiệm tài nguyên tính toán.
- ❖ *Tối ưu hóa việc học thông tin không gian:* Các đặc trưng không gian được học tốt hơn nhờ khả năng trích xuất đa cấp độ của ResNet-18.

### ***Cách tích hợp ResNet-18 vào Encoder***

#### **❖ Cách tích hợp ResNet-18 vào Encoder**

- Phần Encoder truyền thống của U-Net được thay thế hoàn toàn bằng ResNet-18 để thực hiện trích xuất đặc trưng từ ảnh grayscale đầu vào.
- Các tầng Residual Blocks trong ResNet-18 giúp mô hình học được các đặc trưng sâu và giảm thiểu hiện tượng mất thông tin khi đi qua nhiều tầng.
- ResNet-18 vẫn giữ nguyên các skip connections nội tại để đảm bảo quá trình lan truyền gradient diễn ra ổn định.

#### **❖ Kết nối ResNet-18 với Decoder của U-Net**

- Các đặc trưng được trích xuất từ các tầng khác nhau của ResNet-18 sẽ được sử dụng trong skip connections và truyền đến phần Decoder để tái dựng ảnh màu.
- Up-sampling: Ở phần Decoder, sau khi kết hợp với đặc trưng từ ResNet-18 thông qua skip connections, thực hiện up-sampling (bằng bilinear interpolation hoặc transpose convolution) để tăng kích thước không gian của đặc trưng.

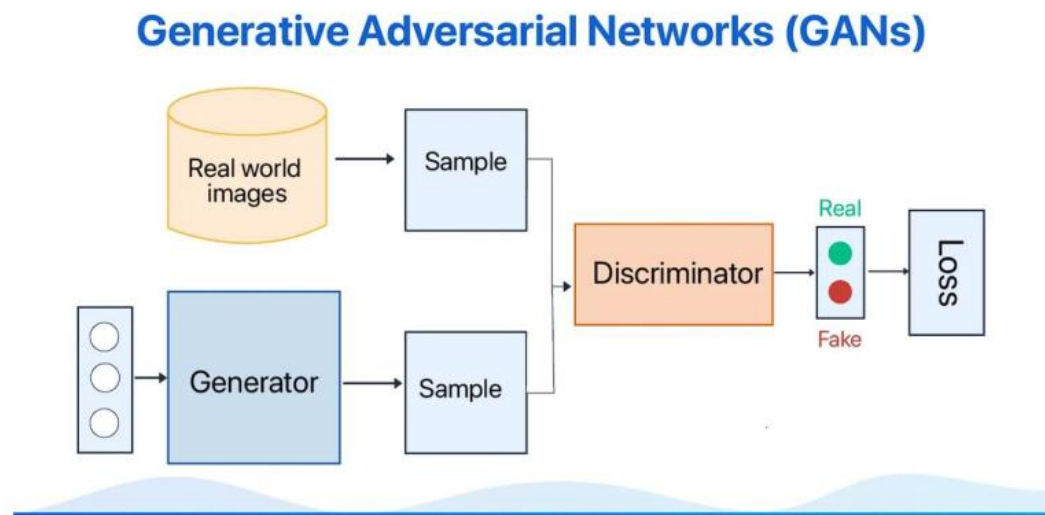
- Tầng đầu ra (Output Layer): Sau khi qua các khối residual, sử dụng một lớp convolution, với hàm kích hoạt sigmoid, để tái tạo ảnh màu cuối cùng trong không gian màu Lab hoặc RGB.

❖ **Ưu điểm sử dụng ResNet-18 thay thế Encoder**

- *Trích xuất đặc trưng mạnh mẽ*: ResNet-18 giúp học các đặc trưng trừu tượng và chi tiết, đặc biệt hữu ích cho việc tái dựng màu sắc phức tạp và chân thực.
- *Cải thiện chất lượng tái dựng màu sắc*: ResNet-18 cung cấp thông tin đặc trưng chính xác và hiệu quả cho Decoder, giúp tái tạo các chi tiết nhỏ và ranh giới màu sắc một cách tự nhiên hơn.
- *Hiệu quả huấn luyện*: Với các skip connections trong ResNet-18, hiện tượng vanishing gradient được giảm thiểu, giúp mô hình hội tụ nhanh và ổn định hơn.
- *Khả năng tổng quát hóa tốt*: ResNet-18 giúp mô hình có khả năng tổng quát tốt hơn khi áp dụng trên các tập dữ liệu mới, đảm bảo chất lượng màu sắc ngay cả với ảnh chưa thấy trong quá trình huấn luyện.

**Mô hình GAN**

GAN, viết tắt của Generative Adversarial Network, là một trong những mô hình học sâu tiên tiến được giới thiệu bởi Ian Goodfellow vào năm 2014. Mô hình này được thiết kế dựa trên ý tưởng học có giám sát đối kháng, trong đó hai mạng thần kinh nhân tạo – Generator (bộ sinh) và Discriminator (bộ phân biệt) – cạnh tranh với nhau để cải thiện hiệu suất.



Hình 11. Tổng quan về mô hình GAN

**Thành phần chính**

❖ **Generator (G):**

Generator là mạng tạo dữ liệu mới dựa trên nhiều đầu vào hoặc các điều kiện bổ sung. Nhiệm vụ của nó là học cách tạo ra dữ liệu giống với dữ liệu thật, đủ để đánh lừa Discriminator.

❖ **Discriminator (D):**

Discriminator là mạng phân biệt, có nhiệm vụ xác định xem dữ liệu đầu vào là

"thật" (từ tập dữ liệu gốc) hay "giả" (do Generator tạo ra).

***Ý tưởng hoạt động***

Quá trình huấn luyện GAN giống như một trò chơi giữa hai người chơi:

- **Generator** cố gắng tạo dữ liệu giống thật nhất có thể.
- **Discriminator** kiểm tra và đánh giá dữ liệu, trả về hai kết quả:
  - "Thật" nếu dữ liệu là từ tập gốc.
  - "Giả" nếu dữ liệu là từ Generator.

Qua các lần huấn luyện, Generator học cách tạo dữ liệu tốt hơn, trong khi Discriminator học cách phân biệt dữ liệu tốt hơn, cho đến khi Discriminator không còn phân biệt được đâu là thật, đâu là giả.

***Conditional GAN (cGAN)***

Conditional GAN là một biến thể của GAN, trong đó cả Generator và Discriminator đều được cung cấp thêm một thông tin điều kiện (condition) để hướng dẫn việc tạo và phân loại dữ liệu

❖ **Cách hoạt động**

- Condition: Là một dữ liệu bổ sung cung cấp thông tin liên quan, ví dụ: ảnh thang độ xám hoặc nhãn phân loại.
- Generator sử dụng condition để tạo dữ liệu mới phù hợp với điều kiện đầu vào.
- Discriminator cũng sử dụng condition để kiểm tra tính hợp lý giữa condition và dữ liệu được tạo ra.

❖ **Ví dụ ứng dụng**

- Tô màu ảnh: Generator nhận một ảnh thang độ xám và tạo ra ảnh màu.
- Discriminator kiểm tra xem ảnh màu này có khớp với ảnh thang độ xám đầu vào hay không.
- Sinh dữ liệu theo nhãn: Tạo hình ảnh động vật tương ứng với nhãn “mèo” hoặc “chó”

***Kiến trúc GAN được sử dụng***

❖ **Generator**



Generator trong đồ án nhận một ảnh thang độ xám (1 kênh) làm đầu vào và tạo ra 2 kênh màu (\*a và \*b trong không gian màu Lab). Các kênh này được kết hợp lại thành ảnh màu hoàn chỉnh. Kiến trúc Generator bao gồm:

- Các lớp tích chập (Convolutional Layers) để trích xuất đặc trưng.
- Các lớp dropout để bổ sung nhiễu ngẫu nhiên, giúp đa dạng hóa đầu ra.

#### ❖ Discriminator

Discriminator nhận 3 kênh (kênh đầu vào và 2 kênh từ Generator), sau đó phân loại xem ảnh này là thật hay giả. Kiến trúc Discriminator gồm:

- Các lớp tích chập để học đặc trưng phức tạp.
- Một lớp đầu ra (Sigmoid) để đưa ra xác suất

#### ❖ Quá trình huấn luyện

- Huấn luyện Discriminator: Nhận ảnh thật và ảnh giả, học cách phân biệt.
- Huấn luyện Generator: Dùng kết quả từ Discriminator để cải thiện khả năng sinh ảnh giả.
- Lặp lại quy trình đến khi mô hình hội tụ.

### Các hàm mất mát (Loss Function)

Trong mô hình GAN và cụ thể hơn là Conditional GAN (cGAN), các hàm mất mát đóng vai trò rất quan trọng, không chỉ giúp điều hướng quá trình huấn luyện mà còn ảnh hưởng trực tiếp đến chất lượng đầu ra của mô hình. Đồ án sử dụng hai loại hàm mất mát chính: Hàm mất mát GAN (cGAN Loss) và L1 Loss.

#### *Hàm mất mát GAN (cGAN Loss)*

Hàm mất mát GAN được xây dựng dựa trên lý thuyết trò chơi Minimax, trong đó Generator (bộ sinh) và Discriminator (bộ phân biệt) cạnh tranh với nhau. Discriminator cố gắng phân biệt giữa dữ liệu "thật" (từ tập dữ liệu gốc) và dữ liệu "giả" (tạo bởi Generator), trong khi Generator cố gắng đánh lừa Discriminator.

#### ❖ Công thức hàm mất mát cGAN

Với Conditional GAN, hàm mất mát của Generator và Discriminator được định nghĩa như sau:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,z} [\log(1 - D(x, G(x, z)))]$$

*Hình 12. Công thức hàm mất mát cGAN*

Trong đó:

- x: Dữ liệu đầu vào (ví dụ: ảnh thang độ xám).

- $y$ : Dữ liệu thật (ví dụ: màu sắc thực tế của ảnh).
- $z$ : Nhiễu đầu vào (noise).
- $G(x,z)$ : Kết quả dự đoán của Generator.
- $D(x,y)$ : Khả năng Discriminator đánh giá dữ liệu thật.
- $D(x,G(x,z))$ : Khả năng Discriminator đánh giá dữ liệu giả.

#### ❖ Ý nghĩa

- Thành phần thứ nhất: Discriminator học cách phân biệt dữ liệu thật ( $x,y$ ).
- Thành phần thứ hai: Generator học cách tạo ra dữ liệu giả ( $G(x,z)$ ) sao cho Discriminator không thể phân biệt được.
- Hàm mất mát này đảm bảo rằng Generator và Discriminator cạnh tranh và học tập lẫn nhau, dẫn đến việc cải thiện hiệu suất của cả hai.

#### **Hàm mất mát L1 (L1 Loss)**

Hàm mất mát L1, hay còn gọi là lỗi trung bình tuyệt đối (Mean Absolute Error), được sử dụng để đo lường sự khác biệt giữa đầu ra dự đoán và dữ liệu thực tế. Trong bài toán tô màu ảnh, L1 Loss tính toán sai số giữa màu được dự đoán và màu thực tế.

#### ❖ Công thức hàm mất mát L1

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1]$$

*Hình 13. Công thức hàm mất mát L1*

Trong đó:

- $y$ : Màu thực tế của ảnh.
- $G(x,z)$ : Kết quả màu do Generator dự đoán.

22

- $\|\cdot\|_1$ : Giá trị tuyệt đối của sự khác biệt giữa dự đoán và thực tế

#### ❖ Ý nghĩa

- L1 Loss giúp giảm thiểu khoảng cách giữa dữ liệu thực tế và dự đoán.
- Khuyến khích Generator tạo ra dữ liệu giống với dữ liệu thật về mặt định lượng.

#### ❖ Lý do chọn L1 Loss thay vì L2 Loss

L2 Loss (Mean Squared Error) có xu hướng làm mờ kết quả, khiến ảnh đầu ra trở nên "xám" hoặc "nhòe."

L1 Loss giảm hiệu ứng làm mờ này và giữ được chi tiết trong ảnh tốt hơn.

#### **Hàm mất mát kết hợp (Combined Loss)**

Trong đề tài này, để đạt được kết quả tốt nhất, em kết hợp cGAN Loss với L1 Loss thành một hàm mất mát tổng hợp. Hàm mất mát này được biểu diễn như sau:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

Hình 14. Công thức hàm mất mát kết hợp

Trong đó:

- $\mathcal{L}_{cGAN}(G, D)$  : Hàm mất mát của cGAN.
- $\mathcal{L}_{L1}(G)$ : Hàm mất mát L1.
- $\lambda$ : Hệ số cân bằng, điều chỉnh mức độ đóng góp của L1 Loss vào tổng thể.

#### Ý nghĩa:

- cGAN Loss: Đảm bảo rằng ảnh đầu ra của Generator có tính "thật" để đánh lừa Discriminator.
- L1 Loss: Đảm bảo rằng ảnh đầu ra giống với dữ liệu thật về mặt định lượng, giảm thiểu sai lệch màu sắc.
- Kết hợp cả hai giúp cải thiện chất lượng ảnh đầu ra, cả về mặt trực quan và định lượng

#### Vai trò của hệ số $\lambda$

Hệ số  $\lambda$  đóng vai trò quan trọng trong việc cân bằng giữa hai hàm mất mát.<sup>23</sup>

- Nếu  $\lambda$  quá lớn, mô hình sẽ ưu tiên L1 Loss, dẫn đến kết quả an toàn nhưng thiếu sáng tạo.
- Nếu  $\lambda$  quá nhỏ, mô hình sẽ ưu tiên cGAN Loss, có thể dẫn đến kết quả trông không thực tế.

#### Kết luận

Việc sử dụng hàm mất mát kết hợp giúp cải thiện hiệu suất của mô hình cGAN trong bài toán tô màu ảnh. Sự kết hợp giữa cGAN Loss và L1 Loss không chỉ đảm bảo tính chân thực của ảnh đầu ra mà còn duy trì sự chính xác so với dữ liệu gốc.

### Thu thập dữ liệu

#### Bộ dữ liệu

Bộ dữ liệu COCO (Common Objects in Context) Sample là một phần của tập dữ liệu COCO, một bộ dữ liệu nổi tiếng trong lĩnh vực học máy và thị giác máy tính được cung cấp bởi thư viện fastai.

Bộ dữ liệu gồm các hình ảnh tự nhiên với ngữ cảnh đa dạng, nơi các đối tượng xuất hiện trong nhiều kịch bản thực tế khác nhau.

#### Thành phần chính của bộ dữ liệu

Số lượng ảnh: Khoảng 21,837 hình ảnh.

Chú thích (Annotations): Chứa các chú thích bounding box để định vị các đối tượng trong ảnh.

- Đối tượng (Categories): Gồm nhiều đối tượng khác nhau, trong đó có 6 đối tượng được đánh dấu gồm (chair, couch, tv, remote, book, vase)

### Định dạng

- Ảnh: Định dạng JPEG, với độ phân giải đa dạng.
- Annotations: Lưu trữ dưới dạng JSON, chứa thông tin như:
  - image\_id: ID của ảnh.
  - category\_id: Lớp của đối tượng.
  - bbox: Bounding box với định dạng [x, y, width, height].

### Phân tích và trực quan hóa dữ liệu

#### ❖ Thống kê kích thước các đối tượng được đánh dấu

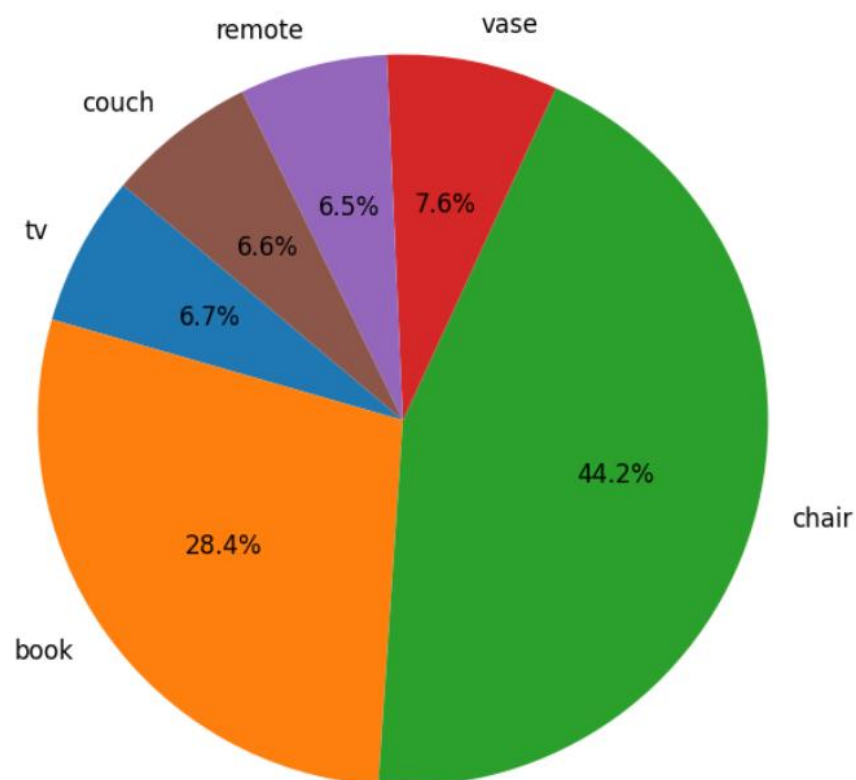
	Class	Object count	Avg height (px)	Max height (px)	Min height (px)	Avg width (px)	Max width (px)
0	book	24715	52.166509	632.00	1.53	54.227240	640.0
1	chair	38491	87.054532	640.00	0.82	81.960915	640.0
2	couch	5779	192.099839	640.00	5.71	265.761244	640.0
3	remote	5703	47.349181	619.86	1.05	55.107198	640.0
4	tv	5805	123.669742	628.39	4.66	137.725525	640.0
5	vase	6613	103.991205	631.37	2.04	66.936654	640.0

Hình 15. Bảng thống kê kích thước các đối tượng được đánh dấu

→ Nhận xét: Kích thước các vật thể đa dạng, dao động trong một biên độ lớn.

#### ❖ Phân bố của các đối tượng được đánh dấu

Percentage of Object Categories in the Dataset

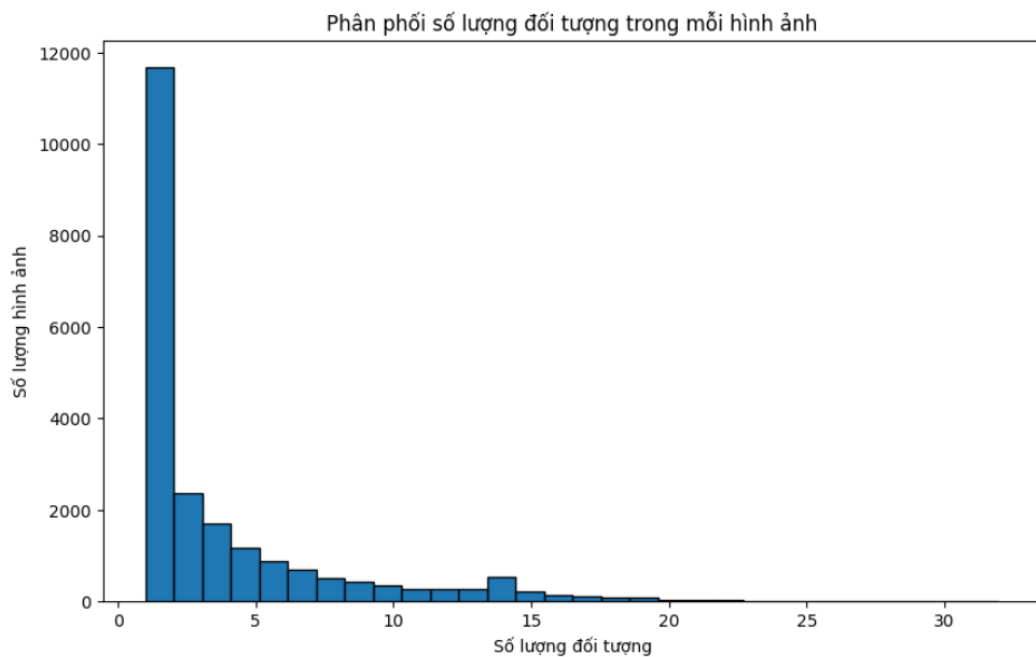


Hình 16. Phân bố các đối tượng được đánh dấu

Nhận xét về phân bố các đối tượng được đánh dấu:

- Chair là đối tượng chiếm tỷ lệ lớn nhất với 44.2% tổng số các đối tượng được đánh dấu.
- Các đối tượng ngoài chair và book còn có tv, couch, remote, vase với tỉ lệ nhỏ hơn rất nhiều ( $< 8\%$ )

#### ❖ Số lượng các đối tượng trong mỗi ảnh



Hình 17. Số lượng các đối tượng trong mỗi ảnh

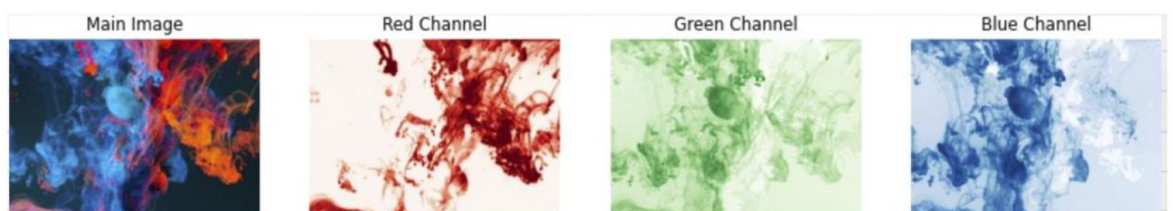
Nhận xét số lượng các đối tượng trong mỗi ảnh:

- Đa số các hình ảnh trong tập dữ liệu có số lượng đối tượng nằm trong khoảng từ 0 đến 5. Đây là nhóm chiếm số lượng hình ảnh lớn nhất.
- Số lượng hình ảnh giảm dần một cách đáng kể khi số lượng đối tượng tăng lên (từ trên 5 đối tượng).

#### Xử lý dữ liệu

##### ❖ Chuyển đổi không gian màu của ảnh

- Không gian RGB (Red - Green - Blue)

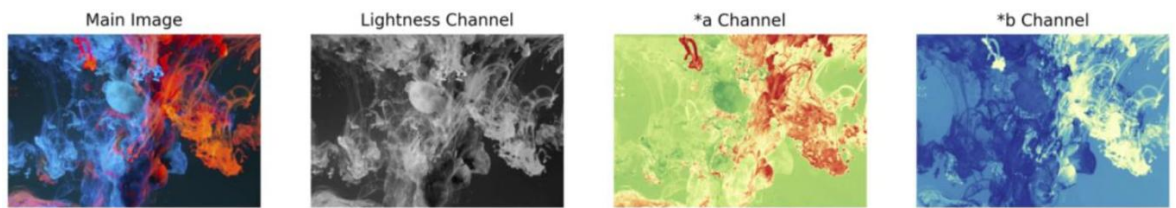


Hình 18. Không gian RGB (Red - Green - Blue)

RGB (Red, Green, Blue) là mô hình màu phổ biến nhất để biểu diễn hình ảnh kỹ thuật số. Mỗi pixel trong hình ảnh được biểu diễn bằng sự kết hợp của ba

giá trị màu sắc: đỏ (Red), xanh lá (Green), và xanh dương (Blue). Bằng cách thay đổi cường độ của ba kênh màu này, có thể tạo ra hàng triệu màu khác nhau

➤ Không gian  $L^*a^*b$



Hình 19. Không gian  $L^*a^*b$

$L^*a^*b$ : Là không gian màu được thiết kế để gần với cách mà mắt người cảm nhận màu sắc. Nó bao gồm ba thành phần:

- $L^*$ : Độ sáng (Lightness), biểu thị độ sáng tối của màu.
- $a^*$ : Biểu thị trục màu từ xanh lá đến đỏ.
- $b^*$ : Biểu thị trục màu từ xanh dương đến vàng.

→ Nhận xét: Đối với yêu cầu bài toán, tô màu cho hình ảnh trắng đen thì, việc chuyển đổi không gian màu của ảnh từ RGB sang  $L^*a^*b$  sẽ giúp đơn giản hoá bài toán. Lúc này chúng ta chỉ cần dự đoán 2 kênh màu là  $a$ ,  $b$  thay vì 2 kênh màu RGB. Điều này giảm số lượng thông tin cần dự đoán, giúp mô hình ổn định hơn và dễ dàng hơn trong việc dự đoán màu sắc.

❖ Chuẩn bị dữ liệu trước khi huấn luyện mô hình

- Resize hình ảnh: Điều chỉnh kích thước tất cả các hình ảnh về một kích thước chuẩn (256x256 pixel) để đồng nhất dữ liệu đầu vào cho mô hình.
- Augmentation (Tăng cường dữ liệu): Áp dụng các kỹ thuật tăng cường dữ liệu như lật ngang ngẫu nhiên để mở rộng tập huấn luyện và giúp mô hình học được các biến thể khác nhau của hình ảnh.
- Chuyển từ RGB sang  $L^*a^*b$ : Chuyển đổi từng hình ảnh từ không gian màu RGB sang không gian màu  $L^*a^*b$  rồi thực hiện chuẩn hóa. Kênh  $L$  (grayscale) được sử dụng làm đầu vào cho mô hình. Kênh  $a$  và  $b$  được sử dụng làm nhãn để mô hình học cách dự đoán màu sắc từ độ sáng.

→ Kết quả thu được:

- Đầu vào mô hình:
  - Kênh  $L$  (grayscale) của hình ảnh, sau khi được chuẩn hóa.
  - Dạng: Tensor với kích thước (Batch\_size, 1, 256, 256).
- Nhãn (Label):
  - Kênh  $a$  và  $b$  (thông tin màu sắc), sau khi được chuẩn hóa.
  - Dạng: Tensor với kích thước (Batch\_size, 2, 256, 256).



❖ Phân chia dữ liệu

Tập train chiếm 95% mẫu dữ liệu, còn lại là tập validate.

✚ **Huấn luyện mô hình Resnet18\_Unet**

**Mục đích:** được làm bộ sinh (Generator) của mô hình GAN. Lúc đó Bộ phân biệt (Discriminator) trong GAN để đánh giá chất lượng hình ảnh sinh ra từ Res18\_Unet

**1. Chuẩn bị dữ liệu**

Tập dữ liệu COCO Sample được tải về và chia thành hai tập:

- Train: 95% số ảnh
- Validation: 5% số ảnh

```
coco_path = untar_data(URLs.COCO_SAMPLE)
coco_path = str(coco_path) + "/train_sample"

# Lấy danh sách tệp ảnh
image_files = glob.glob(coco_path + "/*.jpg")
num_samples = len(image_files)
print(f"Total number of images available: {num_samples}")

# Chia tập dữ liệu thành train/val
paths_subset = np.array(image_files)
rand_idxs = np.random.permutation(num_samples)
train_idxs = rand_idxs[:int(num_samples * 0.95)]
val_idxs = rand_idxs[int(num_samples * 0.95):]
train_paths = paths_subset[train_idxs]
val_paths = paths_subset[val_idxs]

print(f"Number of training images: {len(train_paths)}")
print(f"Number of validation images: {len(val_paths)}")
```

100.00% [3245883392/3245877008 02:19<00:00]

Total number of images available: 21837  
Number of training images: 20745  
Number of validation images: 1092

Hình 20. Tải dữ liệu và chia dữ liệu

- `untar_data(URLs.COCO_SAMPLE)` Tải tập dữ liệu COCO Sample và giải nén.
- `glob.glob(coco_path + "/*.jpg")` Tìm tất cả các tệp ảnh có định dạng .jpg trong thư mục dữ liệu.
- `np.random.permutation(num_samples)` Xáo trộn danh sách các ảnh để chia tập dữ liệu ngẫu nhiên.

Dữ liệu được xử lý và biến đổi sang không gian màu Lab, trong đó:

- Kênh L (độ sáng) được chuẩn hóa trong khoảng  $[-1, 1]$ .
- Hai kênh a, b (thông tin màu sắc) được chuẩn hóa trong khoảng  $[-1, 1]$ .

```

class ColorizationDataset(Dataset):
    def __init__(self, paths, split='train'):
        if split == 'train':
            self.transforms = transforms.Compose([
                transforms.Resize((SIZE, SIZE), Image.BICUBIC),
                transforms.RandomHorizontalFlip(),
            ])
        else:
            self.transforms = transforms.Resize((SIZE, SIZE), Image.BICUBIC)

        self.split = split
        self.paths = paths

    def __getitem__(self, idx):
        img = Image.open(self.paths[idx]).convert("RGB")
        img = self.transforms(img)
        img = np.array(img)
        img_lab = rgb2lab(img).astype("float32") # Chuyển RGB sang L*a*b
        img_lab = transforms.ToTensor()(img_lab)
        L = img_lab[[0], ...] / 50. - 1. # Chuẩn hóa kênh L
        ab = img_lab[[1, 2], ...] / 110. # Chuẩn hóa kênh ab
        return {'L': L, 'ab': ab}

    def __len__(self):
        return len(self.paths)

```

Hình 21. Xử lý dữ liệu

## 2. Xây dựng mô hình

- Bộ trích xuất đặc trưng: Sử dụng ResNet18 pre-trained để lấy đặc trưng từ ảnh đầu vào.
- Bộ giải mã (Decoder): Sử dụng Dynamic U-Net để tái tạo lại màu sắc.
- Hàm loss:
  - L1 Loss: Tính toán sự khác biệt giữa ảnh gốc và ảnh dự đoán.
  - GAN Loss: Giúp cải thiện độ chân thực của màu sắc.

```

def build_res_unet(n_input=1, n_output=2, size=256):
    resnet18_model = resnet18(pretrained=True)
    body = create_body(resnet18_model, n_in=n_input, cut=-2)
    net_G = DynamicUnet(body, n_output, (size, size)).to(device)
    return net_G

```

Hình 22. Xây dựng mô hình

## 3. Theo dõi quá trình huấn luyện:

Lớp AverageMeter giúp theo dõi giá trị loss trong quá trình huấn luyện:

- **reset():** Đặt lại các giá trị thống kê.
- **update(val, count=1):** Cập nhật giá trị loss mới và tính trung bình.
- **create\_loss\_meters():** Tạo dictionary chứa các bộ đo loss khác nhau.



```
# Hàm theo dõi loss
class AverageMeter:
    def __init__(self):
        self.reset()

    def reset(self):
        self.count, self.avg, self.sum = [0.] * 3

    def update(self, val, count=1):
        self.count += count
        self.sum += count * val
        self.avg = self.sum / self.count

def create_loss_meters():
    return {
        'loss_D_fake': AverageMeter(),
        'loss_D_real': AverageMeter(),
        'loss_D': AverageMeter(),
        'loss_G_GAN': AverageMeter(),
        'loss_G_L1': AverageMeter(),
        'loss_G': AverageMeter()
    }
```

Hình 23. Theo dõi loss.

#### 4. Hàm huấn luyện

`pretrain_generator` chịu trách nhiệm huấn luyện mô hình:

- Chạy qua từng epoch.
- Tính toán loss và cập nhật trọng số bằng thuật toán tối ưu.
- Lưu checkpoint khi tìm thấy epoch có loss tốt nhất.

```
def pretrain_generator(net_G, train_dl, opt, criterion, start_epoch, end_epochs):
    best_loss = float('inf')
    best_checkpoint_path = None

    for epoch in range(start_epoch, end_epochs):
        loss_meter = AverageMeter()
        for data in tqdm(train_dl):
            L, ab = data['L'].to(device), data['ab'].to(device)
            preds = net_G(L)
            loss = criterion(preds, ab)
            opt.zero_grad()
            loss.backward()
            opt.step()
            loss_meter.update(loss.item(), L.size(0))

        print(f"Epoch {epoch}/{end_epochs - 1} - L1 Loss: {loss_meter.avg:.5f}")

        if loss_meter.avg < best_loss:
            best_loss = loss_meter.avg
            best_epoch = epoch
            checkpoint_path = os.path.join(checkpoint_dir, f"best_checkpoint_{best_epoch}.pt")
            torch.save({
                'epoch': best_epoch,
                'model_state_dict': net_G.state_dict(),
                'optimizer_state_dict': opt.state_dict(),
                'loss': best_loss,
            }, checkpoint_path)
            print(f"New best checkpoint saved at {checkpoint_path}")
```

Hình 24. Hàm huấn luyện.

#### 5. Huấn luyện mô hình

```
net_G = build_res_unet(n_input=1, n_output=2, size=SIZE)
opt = optim.Adam(net_G.parameters(), lr=1e-4)
criterion = nn.L1Loss()
pretrain_generator(net_G, train_dl, opt, criterion, start_epoch=start_epoch, end_epochs=100)
```

Hình 24. Huấn luyện mô hình

- `net_G`: Mô hình tô màu ảnh.
- `train_dl`: Bộ dữ liệu huấn luyện.
- `opt`: Bộ tối ưu hóa Adam.
- `criterion`: Hàm mất mát L1.
- `start_epoch=start_epoch`: Epoch bắt đầu huấn luyện.
- `end_epochs=100`: Tổng số epoch để huấn luyện.

## 🌟 Huấn luyện mô hình GAN

Mô hình tô màu ảnh thang xám bằng cách dự đoán các kênh màu. Nó bao gồm hai mạng chính:

- Generator (`net_G`): Dự đoán hai kênh màu ab từ kênh L của ảnh trong không gian màu Lab.
- Discriminator (`net_D`): Đánh giá xem ảnh màu được tạo ra có giống ảnh thật không.

Mô hình sử dụng một U-Net kết hợp ResNet-18 làm bộ tạo ảnh (Generator) và một PatchGAN làm bộ phân biệt ảnh (Discriminator).

### 1. Xây dựng Generator

```
def build_res_unet(n_input=1, n_output=2, size=256):
    device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
    resnet18_model = resnet18(pretrained=True) # Khởi tạo model resnet18
    body = create_body(resnet18_model, n_in=n_input, cut=-2)
    net_G = DynamicUnet(body, n_output, (size, size)).to(device)
    return net_G
```

Hình 25. Generator

- Dùng ResNet18 làm encoder, trích xuất đặc trưng ảnh.
- DynamicUnet là U-Net, giúp tái tạo ảnh màu.

### 2. Xây dựng Discriminator – PatchGAN

```
class PatchDiscriminator(nn.Module):
    def __init__(self, input_c, num_filters=64, n_down=3):
        super().__init__()
        model = [self.get_layers(input_c, num_filters, norm=False)]
        model += [self.get_layers(num_filters * 2 ** i, num_filters * 2 ** (i + 1), s=1 if i == (n_down-1) else 2)
                  for i in range(n_down)] # the 'if' statement is taking care of not using
                                          # stride of 2 for the last block in this loop
        model += [self.get_layers(num_filters * 2 ** n_down, 1, s=1, norm=False, act=False)] # Make sure to not use normalization or
                                                                                             # activation for the last layer of the model
        self.model = nn.Sequential(*model)

    def get_layers(self, ni, nf, k=4, s=2, p=1, norm=True, act=True): # when needing to make some repetitive blocks of layers,
        layers = [nn.Conv2d(ni, nf, k, s, p, bias=not norm)] # it's always helpful to make a separate method for that purpose
        if norm: layers += [nn.BatchNorm2d(nf)]
        if act: layers += [nn.LeakyReLU(0.2, True)]
        return nn.Sequential(*layers)

    def forward(self, x):
        return self.model(x)
```

Hình 26. Discriminator

Dùng kiến trúc PatchGAN để phân biệt ảnh thật/giả theo từng vùng nhỏ của ảnh.

### 3. Hàm mất mát GAN

```
class GANLoss(nn.Module):
    def __init__(self, gan_mode='vanilla', real_label=1.0, fake_label=0.0):
        super().__init__()
        self.register_buffer('real_label', torch.tensor(real_label))
        self.register_buffer('fake_label', torch.tensor(fake_label))
        if gan_mode == 'vanilla':
            self.loss = nn.BCEWithLogitsLoss()
        elif gan_mode == 'lsgan':
            self.loss = nn.MSELoss()

        def get_labels(self, preds, target_is_real):
            if target_is_real:
                labels = self.real_label
            else:
                labels = self.fake_label
            return labels.expand_as(preds)

        def __call__(self, preds, target_is_real):
            labels = self.get_labels(preds, target_is_real)
            loss = self.loss(preds, labels)
            return loss
```

Hình 27. Hàm mất mát GAN

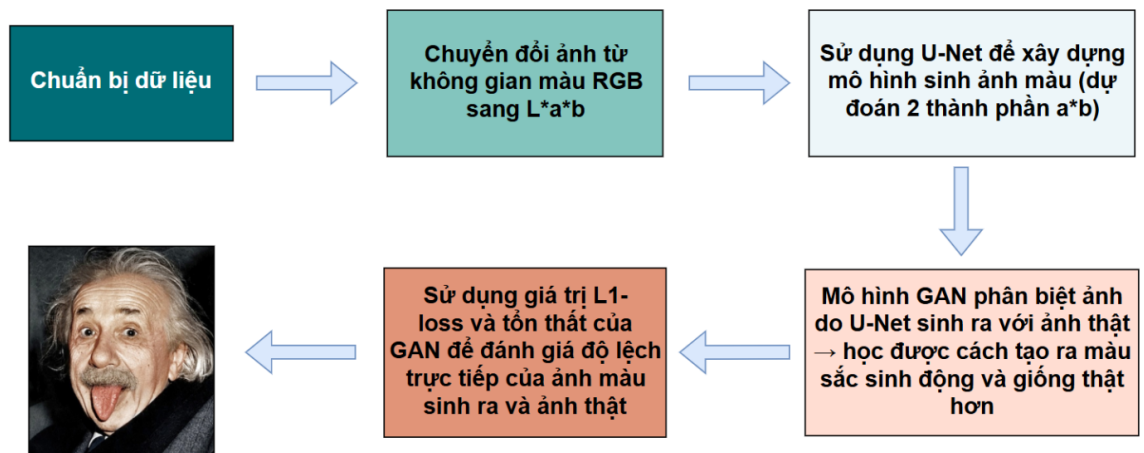
### 4. Mô hình GAN

```
class MainModel(nn.Module):
    def __init__(self, net_G, lr_G=2e-4, lr_D=2e-4,
                 beta1=0.5, beta2=0.999, lambda_L1=100.):
        super().__init__()

        self.device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
        self.lambda_L1 = lambda_L1
        self.net_G = net_G.to(self.device)
        self.net_D = init_model(PatchDiscriminator(input_c=3, n_down=3, num_filters=64), self.device)
        self.GANcriterion = GANLoss(gan_mode='vanilla').to(self.device)
        self.L1criterion = nn.L1Loss()
        self.opt_G = optim.Adam(self.net_G.parameters(), lr=lr_G, betas=(beta1, beta2))
        self.opt_D = optim.Adam(self.net_D.parameters(), lr=lr_D, betas=(beta1, beta2))
```

Hình 28. Mô hình GAN

### ✚ Xây dựng hệ thống:



Hình 29. Sơ đồ tổng quát hệ thống Image Colorization

## 2.2. KẾT QUẢ CÔNG VIỆC

### 2.2.1. Huấn luyện mô hình Resnet18\_Unet

Huấn luyện: 78 epochs (13min/epoch)

Kết quả:

- Epoch 78
- Loss: 0.4264

```
100% ██████████ 1297/1297 [12:37<00:00, 1.94it/s]
Epoch 73/99 - L1 Loss: 0.04409
100% ██████████ 1297/1297 [12:36<00:00, 1.94it/s]
Epoch 74/99 - L1 Loss: 0.04438
100% ██████████ 1297/1297 [12:36<00:00, 1.93it/s]
Epoch 75/99 - L1 Loss: 0.04431
100% ██████████ 1297/1297 [12:36<00:00, 1.95it/s]
Epoch 76/99 - L1 Loss: 0.04354
New best checkpoint saved at /content/drive/MyDrive/Colab Notebooks/ImageColori
Previous best checkpoint /content/drive/MyDrive/Colab Notebooks/ImageColorizati
100% ██████████ 1297/1297 [12:37<00:00, 1.94it/s]
Epoch 77/99 - L1 Loss: 0.04306
New best checkpoint saved at /content/drive/MyDrive/Colab Notebooks/ImageColori
Previous best checkpoint /content/drive/MyDrive/Colab Notebooks/ImageColorizati
100% ██████████ 1297/1297 [12:37<00:00, 1.94it/s]
Epoch 78/99 - L1 Loss: 0.04264
New best checkpoint saved at /content/drive/MyDrive/Colab Notebooks/ImageColori
```

*Hình 30. Kết quả huấn luyện mô hình Resnet19\_Unet ở những epoch cuối*

#### Nhận xét:

- Xây dựng thành công mô hình Resnet18\_Unet
- Hàm loss hội tụ ổn định

### 2.2.2. Huấn luyện mô hình GAN

Huấn luyện 65 epochs

Loss D: 0.6301

Loss G: 0,954

#### Nhận xét:

Loss D đang có xu hướng và Loss G có xu hướng tăng cho thấy hệ thống đang cân bằng giữa việc Discriminator phân biệt tốt dần và Generator đang học hỏi để đạt được chất lượng ảnh cao hơn.

## 2.3. KẾT CHƯƠng

Ở trên em đã đúc kết lại quá trình thực tập của em tại công ty để hoàn thành việc xây dựng các mô hình liên quan đến đề tài được chọn. Bên cạnh những lý thuyết cơ

bản, em còn học được rất nhiều về quy trình làm việc ở môi trường chuyên nghiệp của công ty, để từ đó là nền tảng để em hoàn thành việc xây dựng và huấn luyện mô hình của mình.

Trong quá trình nghiên cứu, em đã biết cách kết hợp nhiều mô hình Deep Learning lại với nhau để tạo ra được mô hình mang hiệu suất cao nhất. Bên cạnh đó là việc kết hợp các hàm mất mát để tối ưu quá trình huấn luyện. Đây là những kiến thức giúp ích rất nhiều để giúp đề tài này thành công.

Ở chương này trình bày về quy trình nghiên cứu, thu thập dữ liệu, thiết kế mô hình và huấn luyện hệ thống tô màu ảnh. Mô hình cGAN với ResNet18 + U-Net đã được triển khai thành công và đạt kết quả tốt trong việc phục hồi màu sắc cho ảnh grayscale. Trong chương tiếp theo, em sẽ tập trung vào việc đánh giá chi tiết hiệu suất mô hình và các hướng cải tiến tiếp theo để nâng cao chất lượng hình ảnh tô màu.

## CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG

### 3.1. KẾT QUẢ TRIỂN KHAI

#### ❖ Đối với mô hình Resnet18\_Unet

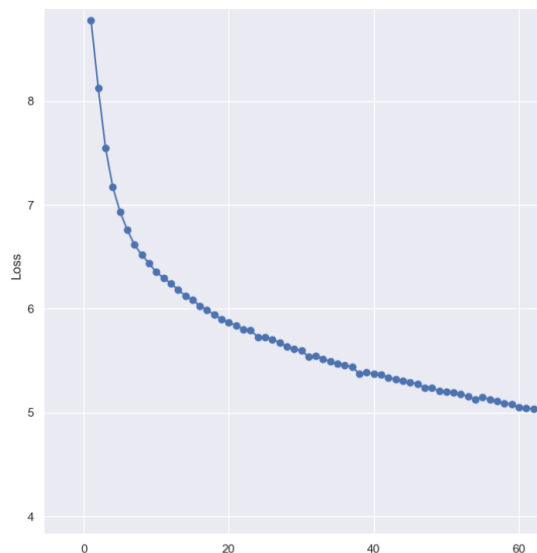


Hình 31. Training Loss\_L1 Res18-Unet

#### Nhận xét:

- Giảm loss ổn định: Mô hình học tốt qua từng epoch, loss giảm đều đặn.
- Loss thấp: Giá trị cuối khoảng 0.05 (L1 loss), chứng tỏ dự đoán khá chính xác.
- Tối ưu tốt: Đường loss mượt, không dao động, thể hiện việc tối ưu được thiết lập phù hợp

#### ❖ Đối với mô hình GAN



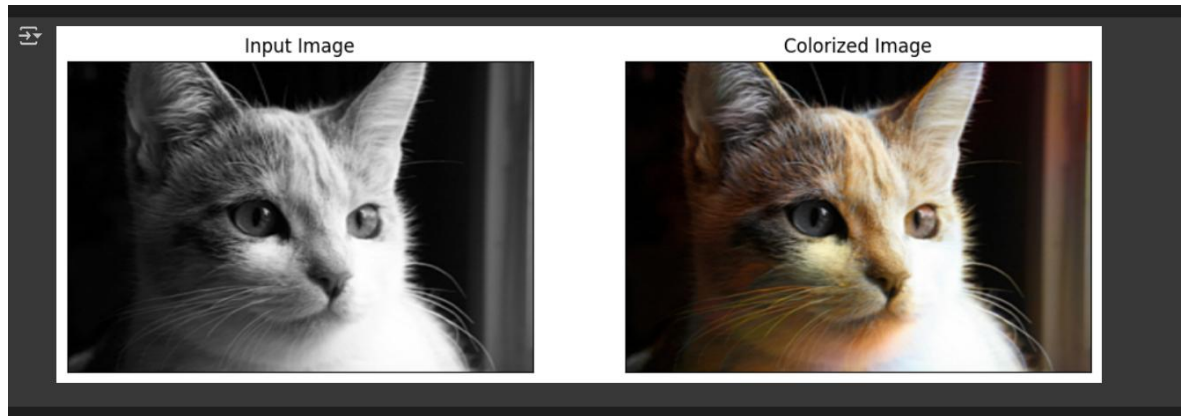
Hình 32. Training Loss GAN

#### Nhận xét:

- Giảm loss ổn định trước checkpoint: loss giảm đều và ổn định, cho thấy quá trình học và tối ưu hóa của mô hình hoạt động hiệu quả.

- Tổng thể: Quá trình huấn luyện của GAN với Res18-UNet làm bộ sinh hoạt động hiệu quả, loss giảm ổn định và mô hình dần đạt được sự tối ưu.

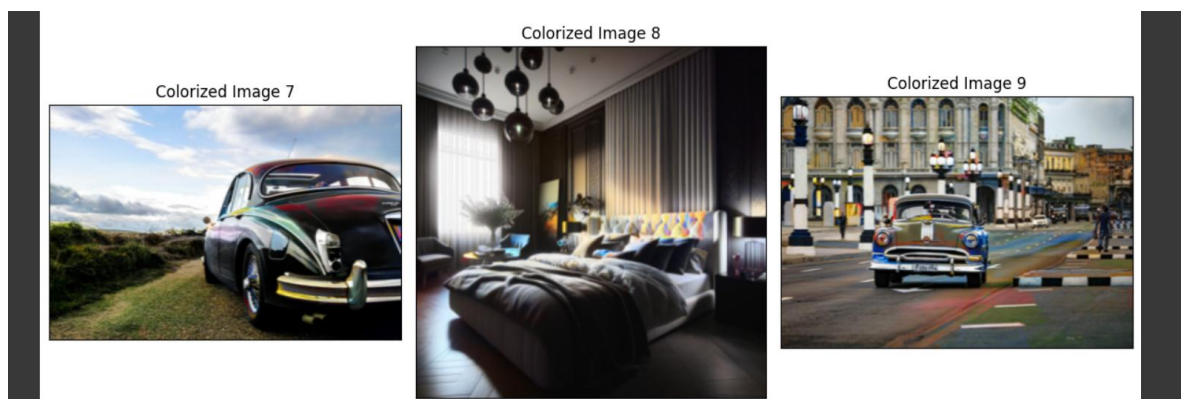
❖ **Kết quả mô hình:**



*Hình 33. Kết quả khi train mô hình*

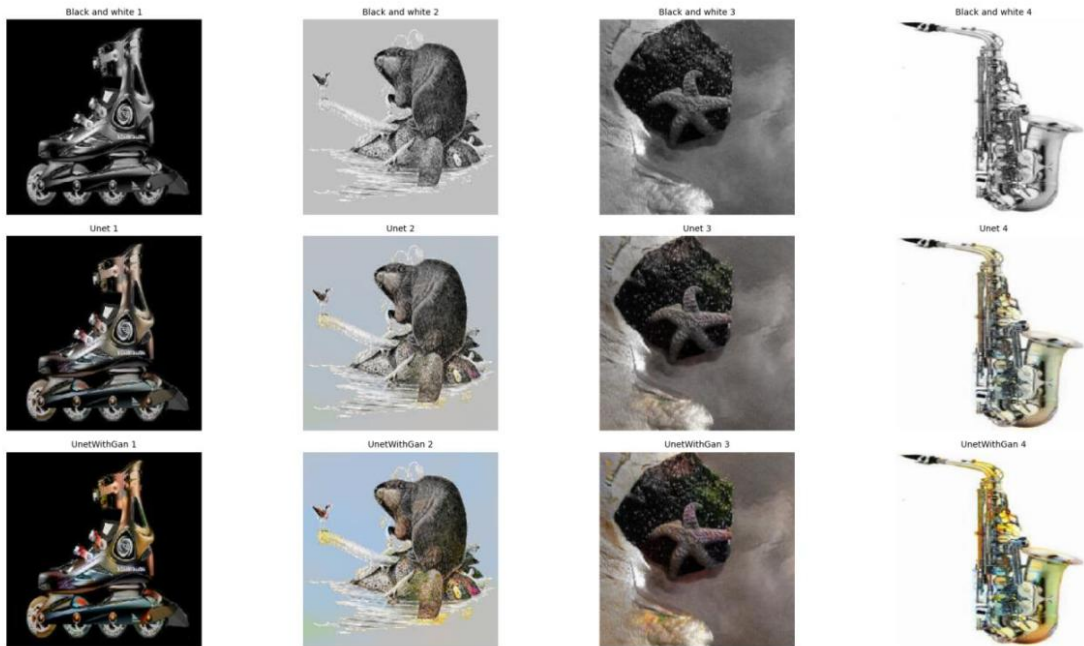


*Hình 34. Đưa vào các input*



*Hình 35. Kết quả trả về*





Hình 36. So sánh kết quả trả về của mô hình Unet không/ có sử dụng GAN

### Đánh giá:

**Chỉ số colorfulness** là một thước đo định lượng để biểu thị mức độ cảm nhận về cường độ và sự đa dạng của màu sắc trong một hình ảnh hoặc cảnh quan. Trong bối cảnh khoa học, nó được định nghĩa dựa trên các tính chất của màu sắc, bao gồm sự bão hòa màu (saturation) và phân bố độ sáng (lightness) trong không gian màu.

Cụ thể, chỉ số colorfulness được tính dựa trên sự khác biệt giữa các kênh màu và phân bố tổng thể của các giá trị màu trong không gian màu, chẳng hạn như không gian RGB, Lab, hoặc YU.

Một cách phổ biến để tính chỉ số colorfulness trong không gian RGB như sau:

#### 1. Tính sự khác biệt màu sắc giữa các kênh:

Trong không gian màu RGB (Red, Green, Blue), colorfulness được tính toán dựa trên sự khác biệt giữa các thành phần màu:

- Chuyển đổi không gian màu từ RGB sang hai thành phần:
  - $RG = R - G$  (chênh lệch giữa Red và Green)
  - $YB = 0.5 \times (R + G) - B$  (chênh lệch giữa Yellow và Blue)

#### 2. Tính độ lệch chuẩn (standard deviation) của các sự khác biệt:

- Độ phân tán:
  - Tính độ lệch chuẩn của RG ( $\sigma_{RG}$ ) và YB ( $\sigma_{YB}$ ).
  - Độ lệch chuẩn đo sự lan tỏa của giá trị pixel trong từng kênh.
- Độ lớn trung bình:
 

Tính giá trị trung bình tuyệt đối của RG ( $\mu_{RG}$ ) và YB ( $\mu_{YB}$ ).

### Công thức:



Dựa trên các tham số đã tính, chỉ số colorfulness được ước lượng theo công thức:

$$C = \sqrt{\sigma_{RG}^2 + \sigma_{YB}^2} + 0.3 \times \sqrt{\mu_{RG}^2 + \mu_{YB}^2}$$

Trong đó:

- $\sqrt{\sigma_{RG}^2 + \sigma_{YB}^2}$ : Mức độ "lan tỏa" của màu sắc.
- $0.3 \times \sqrt{\mu_{RG}^2 + \mu_{YB}^2}$ : Mức độ "trung bình" của màu sắc, với trọng số 0.3 để cân bằng.

Hình 37. Công thức tính chỉ số colorfulness

Kết quả **C** biểu thị chỉ số colorfulness, với giá trị càng cao nghĩa là hình ảnh càng nhiều màu sắc và có tính chất rực rỡ hơn.

→ **Nhận xét:**

- Nhóm ảnh gốc: Chỉ số colorfulness cao nhất, phân bố rộng và đa dạng màu sắc với nhiều ngoại lệ. Trung vị lớn hơn hẳn so với hai nhóm còn lại.
- Nhóm Unet and GAN Images: Chỉ số colorfulness thấp hơn ảnh gốc nhưng cao hơn Unet đơn thuần. Màu sắc ổn định hơn, trung vị cao hơn, cho thấy GAN giúp tạo ảnh màu sống động hơn.
- Nhóm Unet Images: Chỉ số colorfulness thấp nhất, trung vị nhỏ. Kết quả đồng đều nhưng thiếu sự đa dạng và độ rực rỡ màu sắc.

→ **Kết luận:** Ảnh gốc có màu sắc đa dạng và rực rỡ nhất. Unet kết hợp GAN cải thiện đáng kể màu sắc, trong khi Unet đơn thuần còn nhiều hạn chế.

### 3.2. KẾT CHƯƠng

Như vậy, hệ thống tô màu ảnh đã được triển khai thành công. Hệ thống có khả năng tô màu với màu ảnh rực rỡ, gần giống nhất với ảnh gốc. Bên cạnh đó, việc đánh giá hệ thống với chỉ số colorfulness cho thấy tiềm năng của mô hình nghiên cứu so với các mô hình gốc khác. Đây là kết quả của việc kết hợp ưu điểm của nhiều mô hình và áp dụng thành công các hàm mất mát để tối ưu mô hình.

Với những kết quả đạt được, hệ thống đã bước đầu thành công trong việc nghiên cứu, tạo tiền đề cho việc triển khai hệ thống lớn hơn đáp ứng về mặt tế.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. KẾT QUẢ ĐẠT ĐƯỢC

Trong thời gian thực tập tại Trung tâm VNPT IT Khu Vực 3, em đã có cơ hội áp dụng những kiến thức đã học vào thực tiễn, nâng cao kỹ năng chuyên môn và tiếp cận với các công nghệ tiên tiến trong lĩnh vực trí tuệ nhân tạo. Đề tài "Image Colorization" đã được triển khai thành công với các kết quả sau:

- **Về mặt lý thuyết:**
  - Tìm hiểu sâu về các mô hình học sâu như U-Net, ResNet-18 và GAN.
  - Hiểu rõ nguyên lý hoạt động của Conditional GAN (cGAN) và cách kết hợp với các mô hình khác để nâng cao chất lượng tô màu ảnh.
  - Nghiên cứu các phương pháp đánh giá chất lượng ảnh tô màu bằng chỉ số colorfulness và so sánh hiệu suất giữa các mô hình.
- **Về mặt thực tiễn ứng dụng:**
  - Xây dựng và triển khai thành công hệ thống tô màu ảnh sử dụng mô hình cGAN kết hợp ResNet-18 và U-Net.
  - Huấn luyện mô hình trên tập dữ liệu COCO và đạt được kết quả tốt với loss ổn định, mô hình hội tụ nhanh.
  - Ứng dụng các kỹ thuật tiền xử lý dữ liệu, chuyển đổi không gian màu và tối ưu hóa mô hình để cải thiện hiệu suất.
  - Đánh giá mô hình thông qua các tiêu chí khách quan, chứng minh hiệu quả của phương pháp được đề xuất.

### **Những điểm phù hợp giữa chương trình đào tạo và thực tế:**

- Kiến thức nền tảng về Machine Learning và Deep Learning giúp em tiếp cận và hiểu nhanh các mô hình sử dụng trong bài toán.
- Các môn học liên quan như Trí tuệ nhân tạo, Xử lý ảnh, Học sâu đóng vai trò quan trọng trong việc triển khai thực tế.

### **Những điểm chưa phù hợp:**

- Thiếu các bài tập thực hành chuyên sâu về việc triển khai mô hình trên GPU và tối ưu hóa phần cứng.

### **Đề xuất cải thiện:**

- Bổ sung các môn học chuyên sâu về tối ưu hóa mô hình AI, triển khai thực tế trên các nền tảng như Google Colab.
- Tăng cường thực hành với các bài toán thực tế, giúp sinh viên tiếp cận nhanh hơn với yêu cầu của doanh nghiệp.

## 2. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Trong quá trình thực tập, em đã học hỏi được nhiều kiến thức bổ ích về cách làm việc nhóm, kỹ năng nghiên cứu, và cách triển khai một mô hình AI từ giai đoạn tiền xử lý dữ liệu đến đánh giá kết quả. Đây là những kỹ năng quan trọng giúp em chuẩn bị tốt hơn cho công việc sau này.

Một số định hướng phát triển của đề tài trong tương lai:

- Cải thiện chất lượng ảnh tô màu: Tiếp tục nghiên cứu và thử nghiệm các biến thể mới của GAN như StyleGAN, Pix2PixHD để nâng cao chất lượng ảnh đầu ra.
- Mở rộng tập dữ liệu: Thu thập thêm các tập dữ liệu ảnh chất lượng cao hơn, đa dạng hơn để giúp mô hình tổng quát hóa tốt hơn.
- Tối ưu hóa hiệu suất mô hình: Áp dụng các kỹ thuật tối ưu hóa như để giảm thời gian suy luận và tiết kiệm tài nguyên tính toán.
- Ứng dụng vào thực tế: Triển khai mô hình vào các hệ thống phục chế ảnh cũ, cải thiện chất lượng ảnh y tế, hỗ trợ chỉnh sửa ảnh trong nhiếp ảnh chuyên nghiệp.

Kết thúc quá trình thực tập, em nhận thấy bản thân đã trưởng thành hơn về mặt chuyên môn và kỹ năng làm việc. Em mong muốn tiếp tục nghiên cứu và phát triển trong lĩnh vực AI, đặc biệt là các ứng dụng liên quan đến xử lý ảnh và thị giác máy tính. Đồng thời, em hy vọng có cơ hội áp dụng những kiến thức đã học vào các dự án thực tế trong tương lai.

## TÀI LIỆU THAM KHẢO

- [1]. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *Mingxing Tan, Quoc V.le*, <https://arxiv.org/abs/1905.11946v5>, 28/05/2019
- [2]. EfficientDet: Scalable and Efficient Object Detection, *Mingxing Tan, Quoc V.le, Rouming Pang*, <https://arxiv.org/abs/1911.09070v7>, 20/11/2019
- [3]. You Only Look Once: Unified, Real Time Object Detecton, *Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi*, <https://arxiv.org/abs/1506.02640>, 08/06/2015
- [4]. Ayush Gupta, "A Comprehensive Guide on Optimizers in Deep Learning", Analytics Vidhya. Xem tại: <https://www.analyticsvidhya.com/blog/2021/10/acomprehensive-guide-on-deep-learning-optimizers/> , 12/11/2024]
- [5]. statio, "Model Evaluation Metrics trong Machine Learning là gì? Các chỉ số đánh giá mô hình và cách sử dụng chúng hiệu quả", statio. Xem tại: <https://statio.vn/blog/model-evaluation-metrics-trong-machine-learning-la-gi-cac-chiso-anh-gia-mo-hinh-va-cach-su-dung-chung-hieu-qua>, 10/11/2024