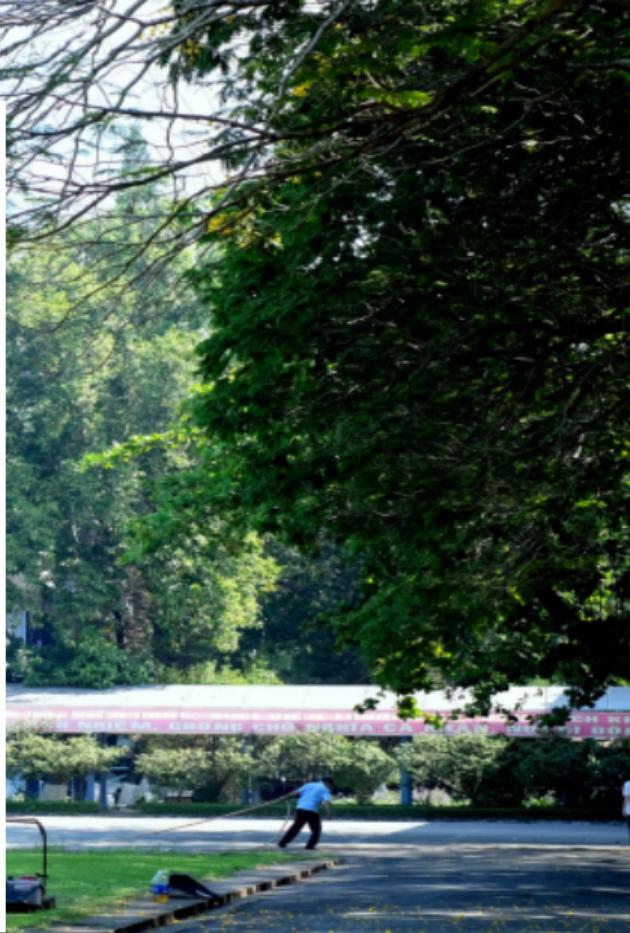


Bảo vệ đồ án môn học

Tạo màu cho ảnh xám (đen trắng)
sử dụng mạng đối nghịch tạo sinh

GVHD: PGS. TS. Hà Hoàng Kha (hhkha@hcmut.edu.vn)
SVTH: Nguyễn Thành Trung - 1814515

Trường Đại học Bách Khoa - ĐHQG TP.HCM



Tổng quan

Một chủ đề nóng hổi và thú vị trong ngành
thị giác máy tính.

Mỗi trường hợp là một bài toán nhiều lời
giải.

Tuy nhiên không đơn giản.

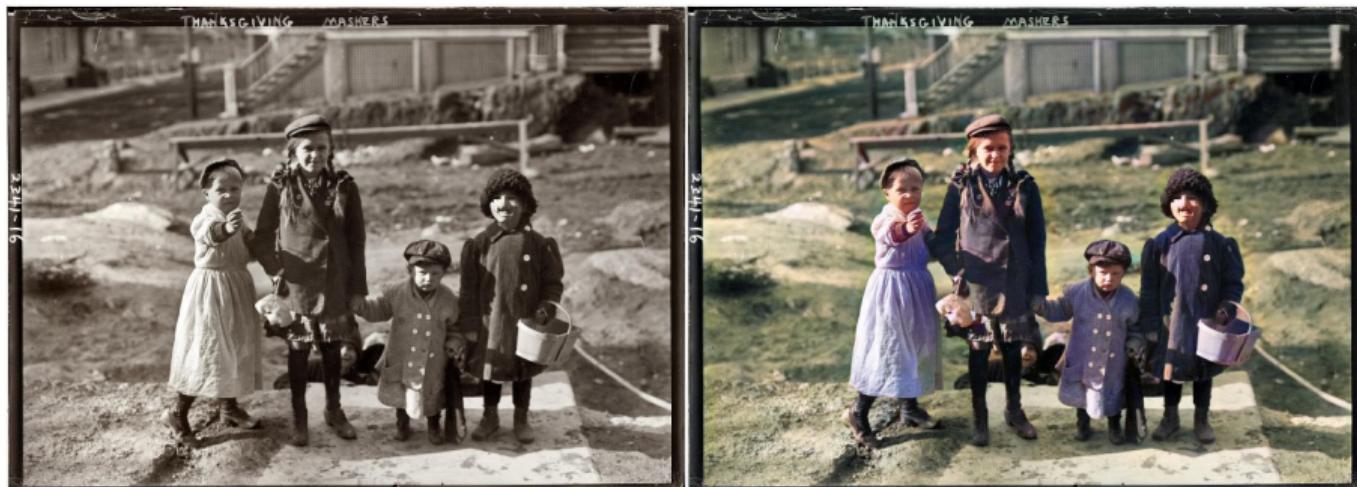


Mục tiêu

Xây dựng và huấn luyện một *mô hình mạng học sâu* để tạo màu cho ảnh xám.

Màu tạo ra không nhất thiết phải giống hoàn toàn so với màu thực.

Độ chân thực, hợp lý về màu sắc là yếu tố được hướng đến.



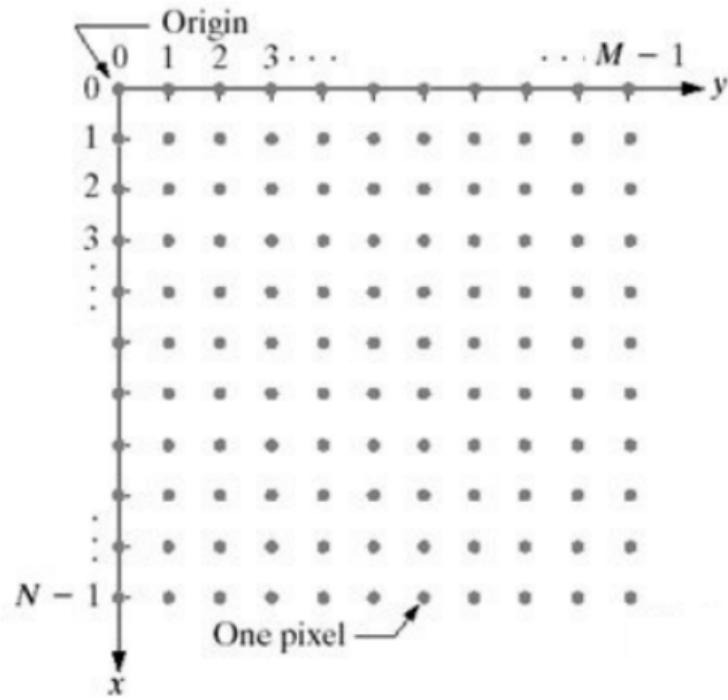
Giới thiệu cơ bản về ảnh số

Tập hợp hữu hạn các điểm ảnh dưới dạng một ma trận 2 chiều:

$$\mathcal{I} \in \mathbf{G}^{W \times H}; \mathbf{G} = \prod_{i=1}^C \mathbf{G}_i.$$

Mỗi điểm ảnh $p_{xy} \in \mathbf{G}$ là một phần tử của ảnh tại toạ độ nguyên (x, y) .

Phân loại: ảnh nhị phân, ảnh xám (đen trắng) và ảnh màu.

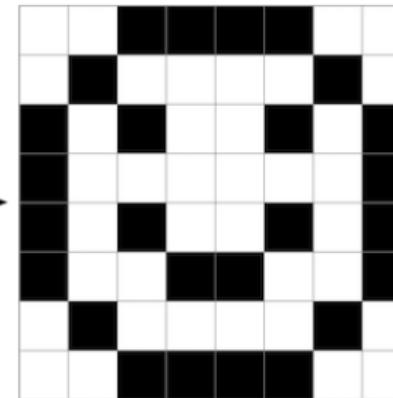


Ảnh nhị phân và ảnh xám

Ảnh nhị phân: $p_{xy} \in G_{\text{nhi phan}} = \{0, 1\}$.

Ảnh xám: $G_{\text{xam}} = [0, 255]$.

1	1	0	0	0	0	1	1
1	0	1	1	1	1	0	1
0	1	0	1	1	0	1	0
0	1	1	1	1	1	1	0
0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0
1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1



256 levels



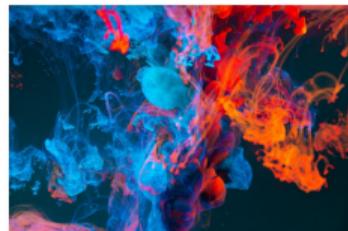
Ảnh màu - không gian màu RGB

Có 3 kênh màu: $G_{RGB} = G_R \times G_G \times G_B$.

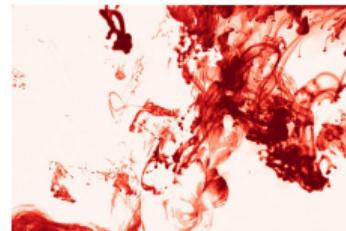
Giá trị các kênh nằm trong khoảng [0, 255].

Tạo ra được khoảng hơn 16 triệu (chính xác là 256^3) màu khác nhau.

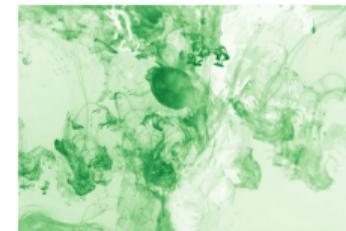
Main Image



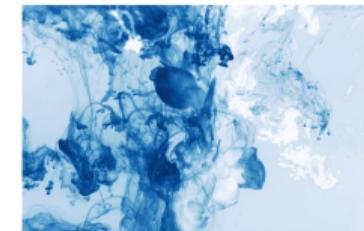
Red Channel



Green Channel



Blue Channel



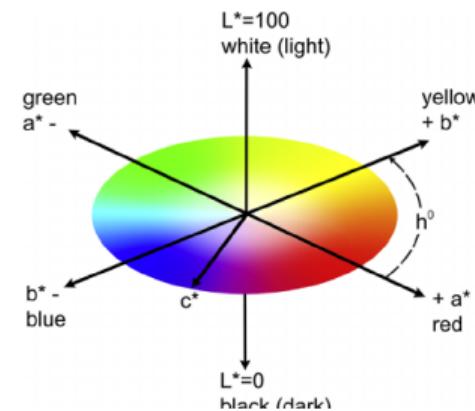
Ảnh màu - không gian màu L*a*b*

Có 3 kênh màu:

$$G_{L^*a^*b^*} = G_{L^*} \times G_{a^*} \times G_{b^*}.$$

$$G_{L^*} = [0, 100].$$

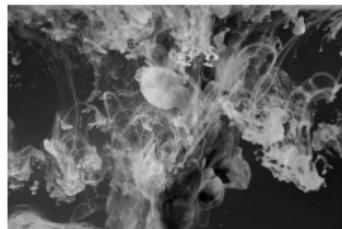
$$G_{a^*} \times G_{b^*} = [-128, 127] \times [-128, 127].$$



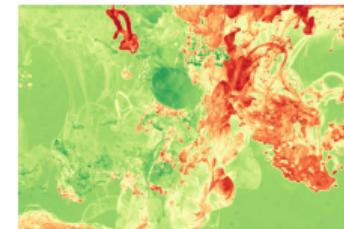
Main Image



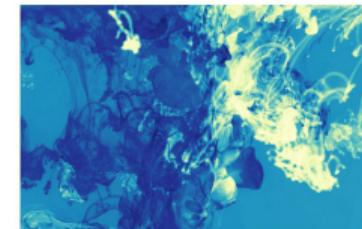
Lightness Channel



*a Channel



*b Channel



Những phương pháp đã được đề xuất cho bài toán tạo màu

Rất nhiều phương pháp đã được đề xuất.

Có thể được chia ra làm 3 loại phương pháp chính:

- Phương pháp *đánh dấu màu một vài điểm ảnh rồi dùng thuật toán loang*.
- Phương pháp *dựa vào mẫu có bối cảnh tương tự*.
- Phương pháp *học sâu*.

Phương pháp đánh dấu màu một vài điểm ảnh rồi dùng thuật toán loang

Vẽ một vài màu cơ bản để từ đó làm nền tảng, định hướng cho mô hình tạo màu.



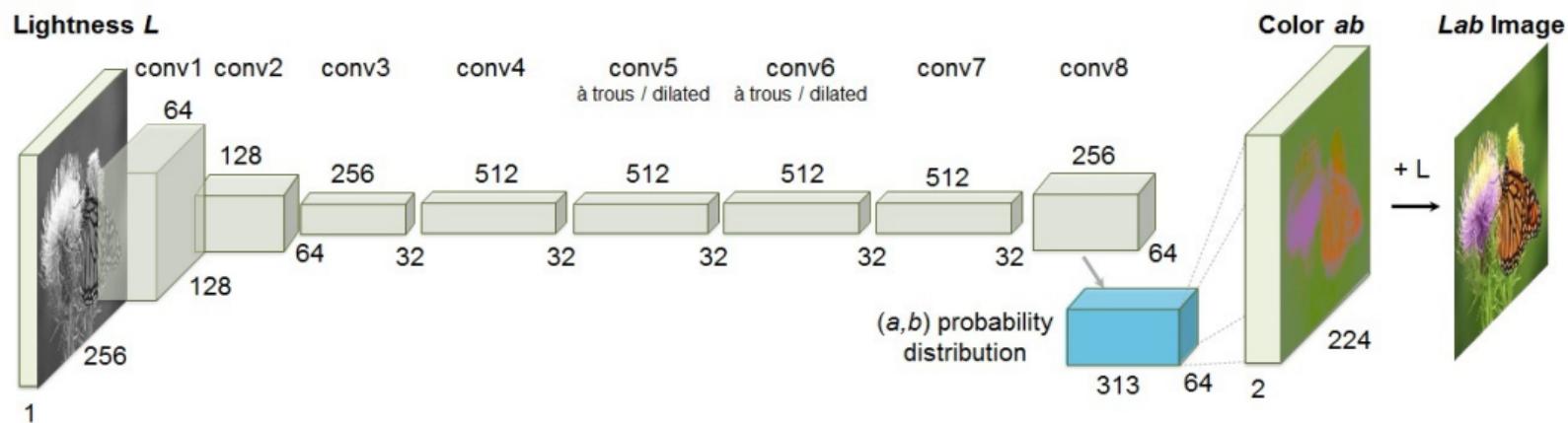
Phương pháp dựa vào mẫu

Chọn ra một tấm ảnh mẫu có bộ cục tương tự rồi dựa vào màu của ảnh mẫu.



Phương pháp học sâu

Bằng cách tận dụng những cặp ảnh (xám, màu), rất nhiều phương pháp dựa vào học sâu đã được ra đời.



Mạng đối nghịch tạo sinh (GAN - Generative Adversarial Network)

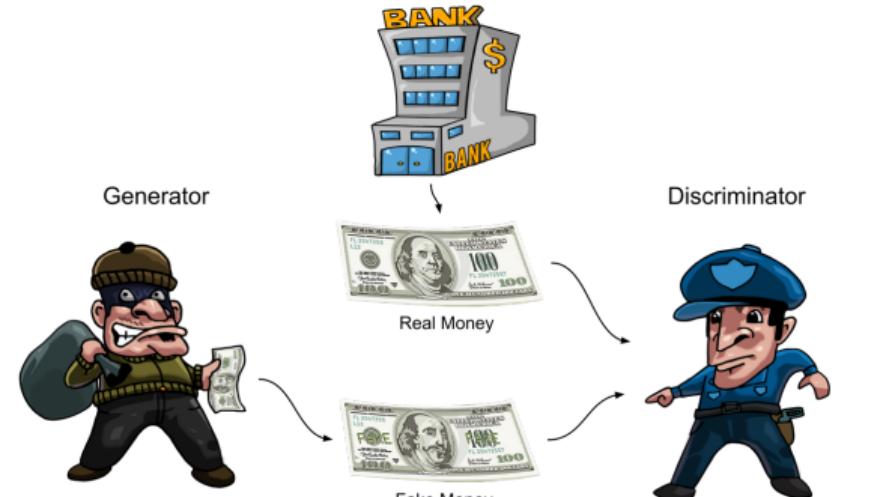
Mạng GAN thuộc nhóm mô hình sinh **dữ liệu mới**.



Khái quát về mạng GAN

- G - **G**enerative (tạo sinh).
- A - **A**dversarial (đối nghịch).
- N - **N**etwork (mạng).

Mục tiêu cuối cùng: người làm tiền giả phải có khả năng làm tiền giả, sao cho cảnh sát không phân biệt được đâu là thật đâu là giả (50/50) để đem tiền giả đi tiêu thụ.



Cơ sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Bài kiểm tra từ hai tập mẫu: liệu tập dữ liệu

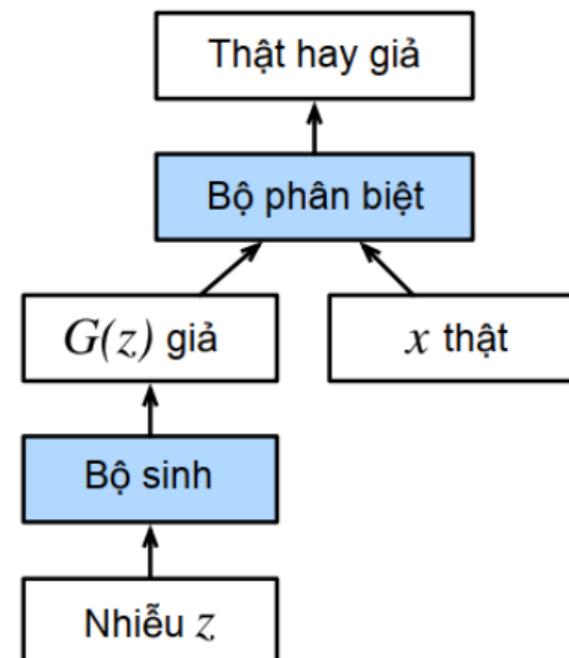
$$X = \{x_1, x_2, \dots, x_n\}$$
 và

$$X' = \{x'_1, x'_2, \dots, x'_n\}$$

có được rút ra từ cùng một phân phối hay không?

GAN sử dụng ý tưởng này theo kiểu có tính *xây dựng*.

Cải thiện bộ sinh để có thể sinh dữ liệu tới khi ra được thứ gì đó giống như dữ liệu thực.



Cơ sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Bộ phân biệt $D(\cdot)$ là **một mạng học sâu phân loại nhị phân**, $D(\mathbf{x} \in \mathbb{R}^n) \in (0, 1)$.

Tầng cuối cùng của bộ phân biệt là hàm Sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$.

Giá trị *càng gần 1* thì bộ phân biệt càng có xu hướng quyết định \mathbf{x} là dữ liệu *thật*.

Giả sử mỗi cặp dữ liệu huấn luyện thứ i có dạng $(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \{0, 1\}$, bộ phân biệt cần được tối ưu sẽ có dạng entropy chéo:

$$D^* = \arg \min_D \left\{ -\frac{1}{N} \sum_{i=1}^N [y_i \log D(\mathbf{x}_i) + (1 - y_i) \log (1 - D(\mathbf{x}_i))] \right\} \quad (1)$$

Cơ sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Bộ sinh $G(\cdot)$ - cũng là **một mạng học sâu**, $G(\mathbf{z} \in \mathbb{R}^d) \in \mathbb{R}^n$.

Nhiều thường được chọn là phân phối chuẩn $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(0, 1)$.

Cho trước một bộ phân biệt D , ta sẽ cập nhật tham số của bộ sinh G nhằm cực đại hóa mất mát entropy chéo như trong (1) khi $y = 0$:

$$\begin{aligned} G^* &= \arg \max_G \left\{ -\frac{1}{N_{\text{nhiều}}} \sum_{i=1}^{N_{\text{nhiều}}} (1 - y_i) \log (1 - D(G(\mathbf{z}_i))) \right\} \\ &= \arg \max_G \left\{ -\frac{1}{N_{\text{nhiều}}} \sum_{i=1}^{N_{\text{nhiều}}} \log (1 - D(G(\mathbf{z}_i))) \right\} \end{aligned} \quad (2)$$

Cơ sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Trong thực tế, công thức (2) không phải là một công thức tốt để tối ưu G .

Khả năng cao là ban đầu ta sẽ có $D(G(\mathbf{z})) \approx 0 \Leftrightarrow (1 - D(G(\mathbf{z}))) \approx 1 \Rightarrow$ dễ bị triệt tiêu đạo hàm.

Sử dụng chiến lược cực tiểu hóa mất mát entropy chéo như trong (1) với $y = 1$:

$$\begin{aligned} G^* &= \arg \min_G \left\{ -\frac{1}{N_{\text{nhiều}}} \sum_{i=1}^{N_{\text{nhiều}}} y_i \log(D(G(\mathbf{z}_i))) \right\} \\ &= \arg \min_G \left\{ -\frac{1}{N_{\text{nhiều}}} \sum_{i=1}^{N_{\text{nhiều}}} \log(D(G(\mathbf{z}_i))) \right\} \end{aligned} \quad (3)$$

Cơ sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Tóm lại, D và G đang chơi trò “cực tiểu hoá cực đại” với một hàm mục tiêu toàn diện như sau:

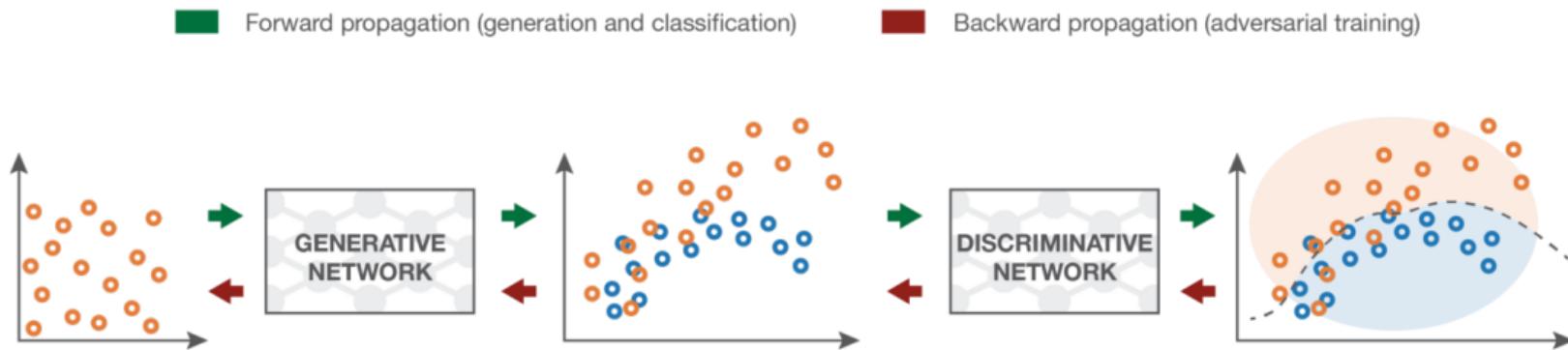
$$\begin{aligned} (D, G)^* &= \arg \min_D \arg \max_G \left\{ -\mathbb{E}_{x \sim \text{dữ liệu thật}} \log D(x) - \mathbb{E}_{z \sim \text{nhiều}} \log (1 - D(G(z))) \right\} \\ \Leftrightarrow (D, G)^* &= \arg \min_G \arg \max_D \left\{ \mathbb{E}_{x \sim \text{dữ liệu thật}} \log D(x) + \mathbb{E}_{z \sim \text{nhiều}} \log (1 - D(G(z))) \right\} \end{aligned} \quad (4)$$

Từ hàm mất mát (4), ta nhận thấy rằng việc huấn luyện bộ phân biệt và bộ sinh là *đối nghịch nhau*.

Về mặt lý thuyết, quá trình dạy học cho GAN kết thúc khi mô hình GAN đạt đến trạng thái cân bằng của hai bộ trong mạng.

Quá trình huấn luyện mạng GAN

Có hai phiên: **lần truyền thuận** và sau đó là **lần truyền ngược**.



Input random variables.

The generative network is trained to **maximise** the final classification error.

The **generated distribution** and the **true distribution** are not compared directly.

The discriminative network is trained to **minimise** the final classification error.

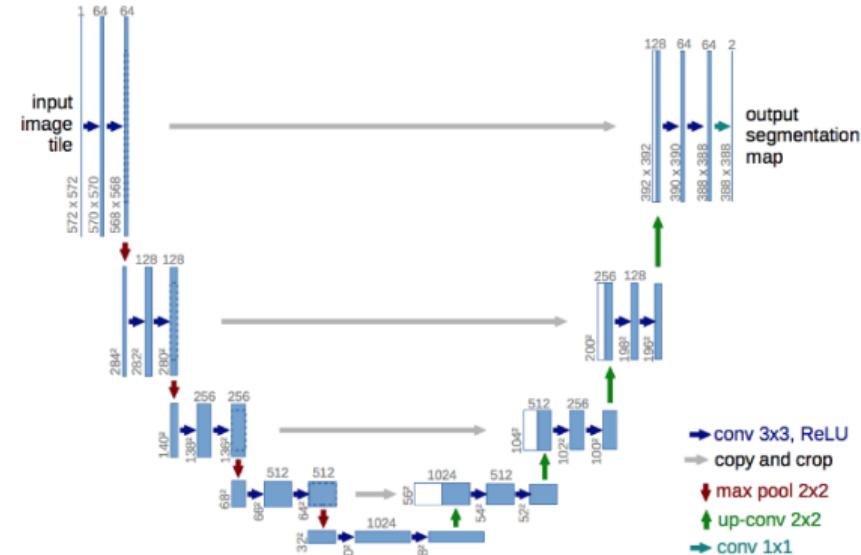
The classification error is the basis metric for the training of both networks.

Mạng tích chập Unet

Unet bao gồm 2 nhánh đối xứng nhau.

Nhánh trái - thu hẹp \Rightarrow CÁI GIÀ,
nhánh phải - mở rộng \Rightarrow Ở ĐÂU.

Đặc trưng riêng trong cấu trúc của Unet đó
là có những *kết nối tắt đối xứng*.

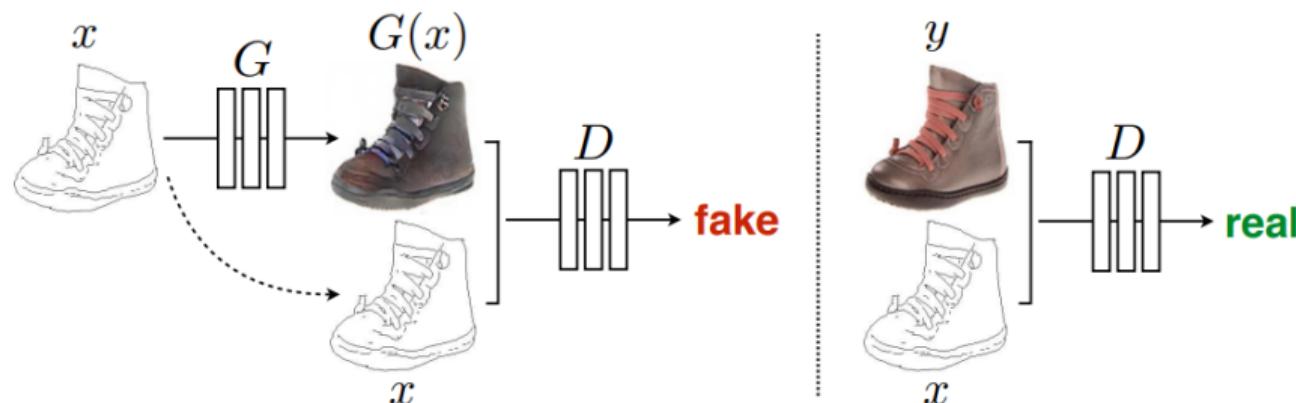


Khái quát mạng GAN trong pix2pix

Mạng GAN trong pix2pix là một **mạng GAN có điều kiện**.

Đầu vào bộ sinh là một bức ảnh nguồn x , biến đổi thành ảnh đích $G(x)$.

Bộ phân biệt giờ đây sẽ nhận vào *một cặp ảnh*.



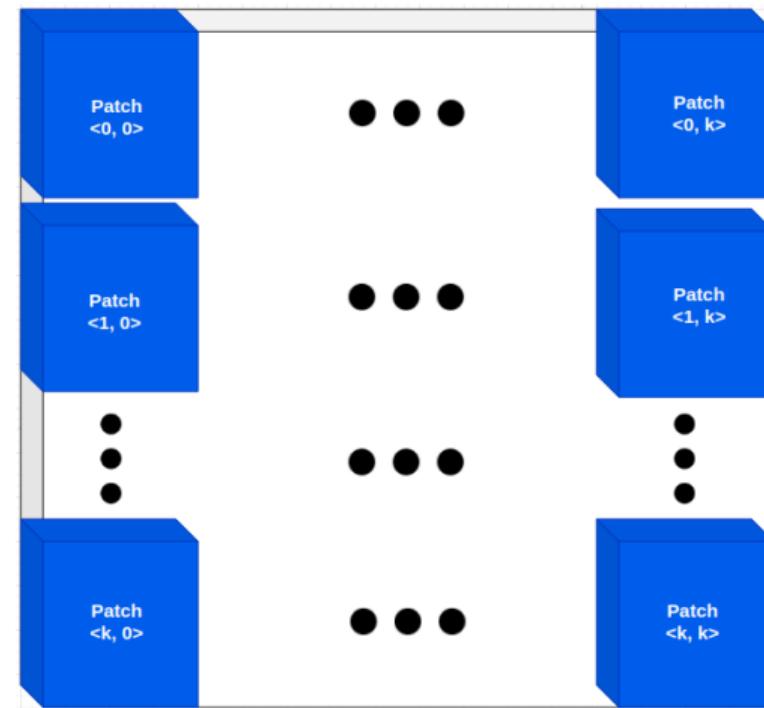
Bộ phân biệt Patch trong pix2pix

Bộ phân biệt Patch sẽ trả về *một ma trận* $\mathbf{P}_D \in \mathbb{R}^{k \times k}$.

Mỗi phần tử của ma trận $[\mathbf{P}_D]_{ij}$ là kết quả phân loại một vùng $N \times N$.

Những vùng $N \times N$ này được gọi là **vùng nhận thức**.

Kích thước Patch *tốt nhất* theo bài báo pix2pix là 70×70 .



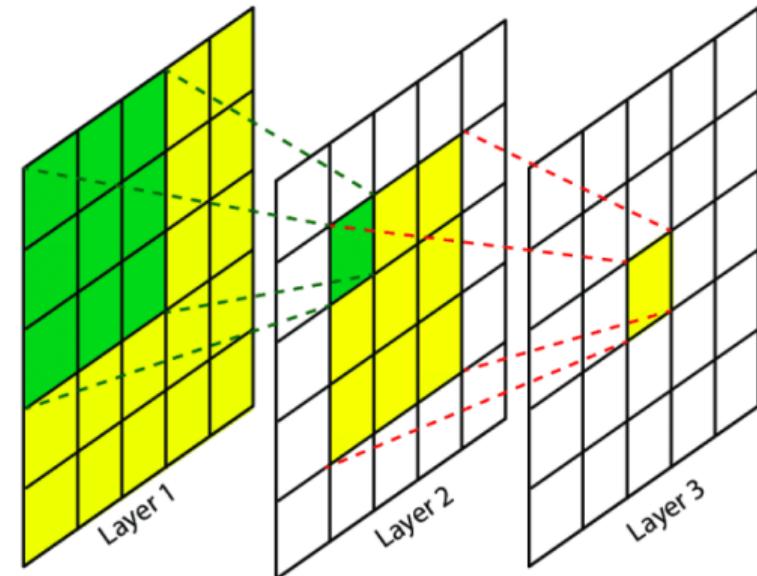
Bộ phân biệt Patch trong pix2pix

Kiến trúc Patch 70×70 : $I \rightarrow 2C64 \rightarrow 2C128 \rightarrow 2C256 \rightarrow 1C512 \rightarrow O$.

Trong đó sCk là một tầng tích chập gồm k bộ lọc và có sải bước là s .

Tầng cuối cùng O sử dụng sải bước là 1 và chỉ sử dụng duy nhất 1 bộ lọc để trả ra ma trận P_D .

Kích thước bộ lọc là 4×4 cùng với kích thước đệm là 1.



Bộ phân biệt Patch trong pix2pix

kích thước vùng nhận thức F tại tầng thứ L sẽ là:

$$F_L = K + S(F_{L+1} - 1) \quad (5)$$

Áp dụng công thức (5), ta sẽ tính được kích thước vùng nhận thức của mỗi giá trị đầu ra của bộ phân biệt Patch 70×70 :

$$F_4 = K + S_4(F_5 - 1) = 4 + 1(1 - 1) = 4$$

$$F_3 = K + S_3(F_4 - 1) = 4 + 1(4 - 1) = 7$$

$$F_2 = K + S_2(F_3 - 1) = 4 + 2(7 - 1) = 16$$

$$F_1 = K + S_1(F_2 - 1) = 4 + 2(16 - 1) = 34$$

$$F_0 = K + S_0(F_1 - 1) = 4 + 2(34 - 1) = 70$$

Hàm mục tiêu (mất mát) pix2pix

Hàm mất mát GAN gần giống với (4):

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y} [\log D(x, y)] + \mathbb{E}_{x,z} [\log (1 - D(x, G(x, z)))] \quad (6)$$

“Xác suất để toàn bộ đầu vào là thật sẽ là trung bình cộng của toàn bộ k^2 Patch”, ta có thể hiểu ngầm giá trị bộ phân biệt Patch thành *một số vô hướng* như sau:

$$D(\cdot) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k (\mathbf{P}_D)_{ij} \quad (7)$$

Tuy nhiên **khi tối ưu**, nên được hiểu là:

$$\log(f(D)) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k \log(f(\mathbf{P}_D)_{ij}) \quad (8)$$

Hàm mục tiêu (mất mát) pix2pix

Không những phải lừa bộ phân biệt, bộ sinh còn phải làm sao tạo ra kết quả **giống với nhãn** \Rightarrow kết hợp với một hàm mất mát thông thường:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z} [\|y - G(x, z)\|_1] \quad (9)$$

Từ (6) và (9), ta có hàm mục tiêu của pix2pix là:

$$G^* = \arg \min_G \max_D \{\mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)\} \quad (10)$$

Hệ số λ là hệ số cân bằng.

Phần nhiễu z có thể được bỏ qua trong một số bài toán với đầu vào x phức tạp.

Kiến trúc pix2pix trong bài toán tạo màu

Cho một ảnh xám đầu vào $\mathcal{I}_{\text{đầu vào}} \in G_{L^*}^{256 \times 256}$, ta cần xây dựng một ánh xạ:

$$\mathcal{G} : G_{L^*}^{256 \times 256} \rightarrow (G_{a^*} \times G_{b^*})^{256 \times 256} \quad (11)$$

$$\widehat{\mathcal{I}} = (\mathcal{I}_{\text{đầu vào}}, \mathcal{G}(\mathcal{I}_{\text{đầu vào}})) \in (G_{L^* a^* b^*})^{256 \times 256} \quad (12)$$

Việc xây dựng ánh xạ sẽ được dẫn đường bởi ánh xạ:

$$\mathcal{D} : (G_{L^* a^* b^*})^{256 \times 256} \rightarrow \mathbb{R}^{30 \times 30} \quad (13)$$

Dựa vào (10) và bỏ nhiễu, ta có **hàm mục tiêu cho bài toán tô màu** là:

$$\begin{aligned} \mathcal{G}^* = \arg \min_{\mathcal{G}} \max_{\mathcal{D}} & \left\{ \mathbb{E}_{\mathcal{I}} [\log \mathcal{D}(\mathcal{I})] + \mathbb{E}_{\mathcal{I}_{\text{đầu vào}}} [\log (1 - \mathcal{D}(\mathcal{I}_{\text{đầu vào}}, \mathcal{G}(\mathcal{I}_{\text{đầu vào}})))] \right. \\ & \left. + \lambda \underbrace{\mathbb{E}_{\mathcal{I}_{\text{đầu vào}}, \mathcal{I}_{\text{đầu ra}}} \|\mathcal{I}_{\text{đầu ra}} - \mathcal{G}(\mathcal{I}_{\text{đầu vào}})\|}_{\mathcal{L}_{L1}(\mathcal{G})} \right\} \quad (14) \end{aligned}$$

Kiến trúc bộ sinh cho bài toán tạo màu

nhánh trái: $I \rightarrow C64 \rightarrow C128 \rightarrow C256 \rightarrow C512 \rightarrow C512 \rightarrow C512 \rightarrow C512$

nhánh phải: $\rightarrow D512 \rightarrow D1024 \rightarrow D1024 \rightarrow D1024 \rightarrow D512 \rightarrow D256 \rightarrow D128 \rightarrow D2$

C_k là một tầng tích chập k bộ lọc.

D_k là một tầng giải chập bằng tích chập chuyển vị với k bộ lọc.

Cả 2 quá trình chập và giải chập đều sử dụng kích thước bộ lọc là 4, sải bước 2 và đệm 1.

Kiến trúc chưa đủ tốt với tập dữ liệu ít \Rightarrow xây dựng Unet từ **xương sống ResNet18**.

Toàn bộ xương sống ResNet18 sẽ *chỉ dùng để xây dựng phần mã hoá, phần giải mã và kết nối tắt* sẽ được khởi tạo tự động nhờ thư viện.

Triển khai mô hình

Sử dụng ngôn ngữ lập trình **Python** với sự hỗ trợ chính từ thư viện **Pytorch**.

Toàn bộ mã nguồn, cũng như những tệp liên quan có thể tìm thấy trong thư mục đồ án tại:
https://github.com/dee-ex/EE3151_SEM202_PROJECT

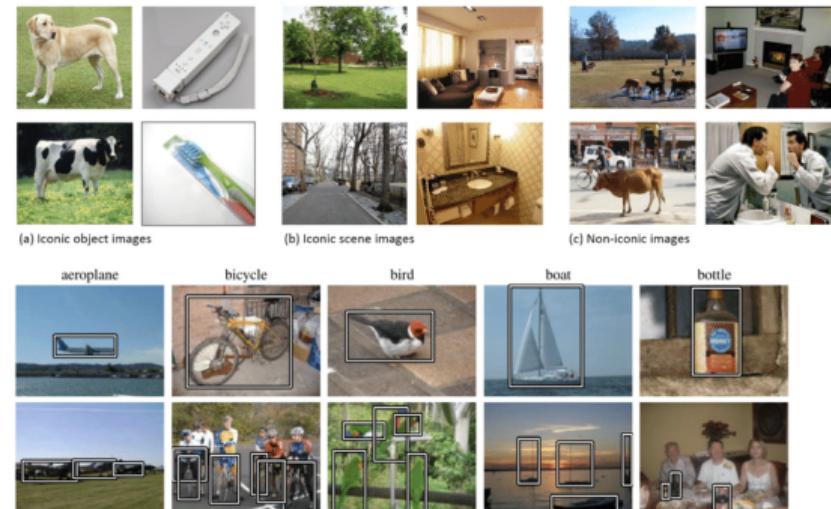


Tập dữ liệu được sử dụng; chuẩn hoá và làm giàu dữ liệu

Sử dụng 10,000 tấm ảnh từ tập dữ liệu **COCO** (tổng cộng gần 19,000 tấm ảnh).

Chuẩn hoá 3 kênh dữ liệu về một khoảng $[-1, 1]$.

Sử dụng *lật đối xứng theo trực tung* để làm giàu dữ liệu cho việc huấn luyện.



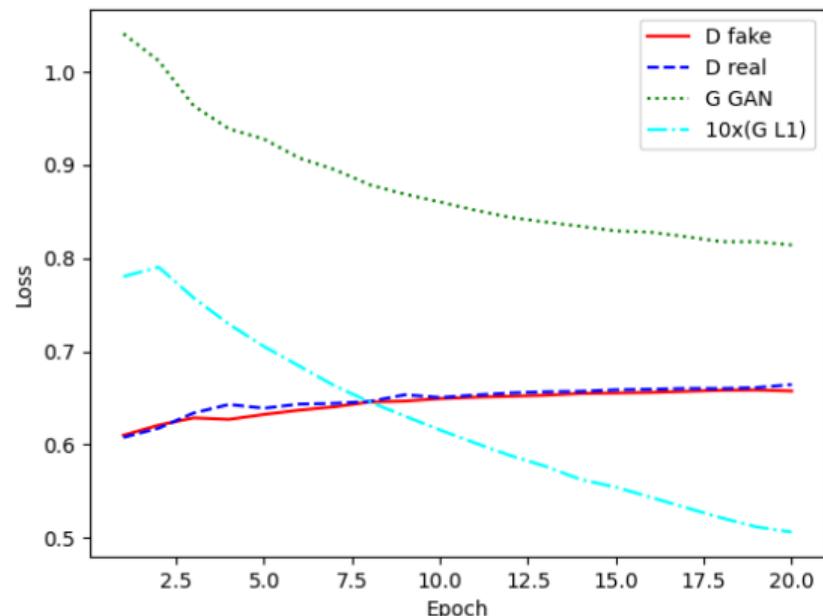
Huấn luyện mô hình

Tiền huấn luyện bộ sinh với hàm mất mát chuẩn 1 $\mathcal{L}_{L1}(\mathcal{G})$ (14) qua 20 epoch.

Kích thước batch: 16, Thuật toán tối ưu: Adam, tốc độ học: $\alpha = 10^{-4}$, mô men: $\beta = (0.9, 0.999)$.

Huấn luyện đối nghịch với bộ sinh vừa được tiền huấn luyện theo (14) với $\lambda = 100$.

Kích thước batch: 16, Thuật toán tối ưu: Adam, tốc độ học: $\alpha = 2 \cdot 10^{-4}$, mô men: $\beta = (0.5, 0.999)$.



Thử nghiệm và đánh giá mô hình

Kết quả dự đoán của mô hình GAN với *bộ sinh đơn giản* sau epoch thứ 34 trên tập dữ liệu COCO (không nằm trong tập huấn luyện):

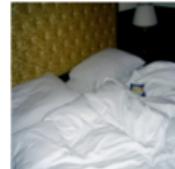
input



prediction



ground-truth



Thử nghiệm và đánh giá mô hình

Kết quả dự đoán của mô hình GAN với *bộ sinh có xương sống ResNet18* sau epoch thứ 15 trên tập dữ liệu COCO (không nằm trong tập huấn luyện):

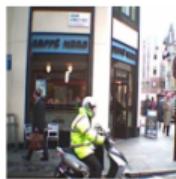
input



prediction



ground-truth



Thử nghiệm và đánh giá mô hình

Mô hình hoàn toàn *bết tắc* trước logo kỷ niệm 60 năm của trường ĐH Bách Khoa TP.HCM:



input



prediction



ground-truth

Thử nghiệm và đánh giá mô hình

Trường hợp đặc biệt khi mô hình làm việc *tệ* với ảnh có đặc trưng đơn giản:

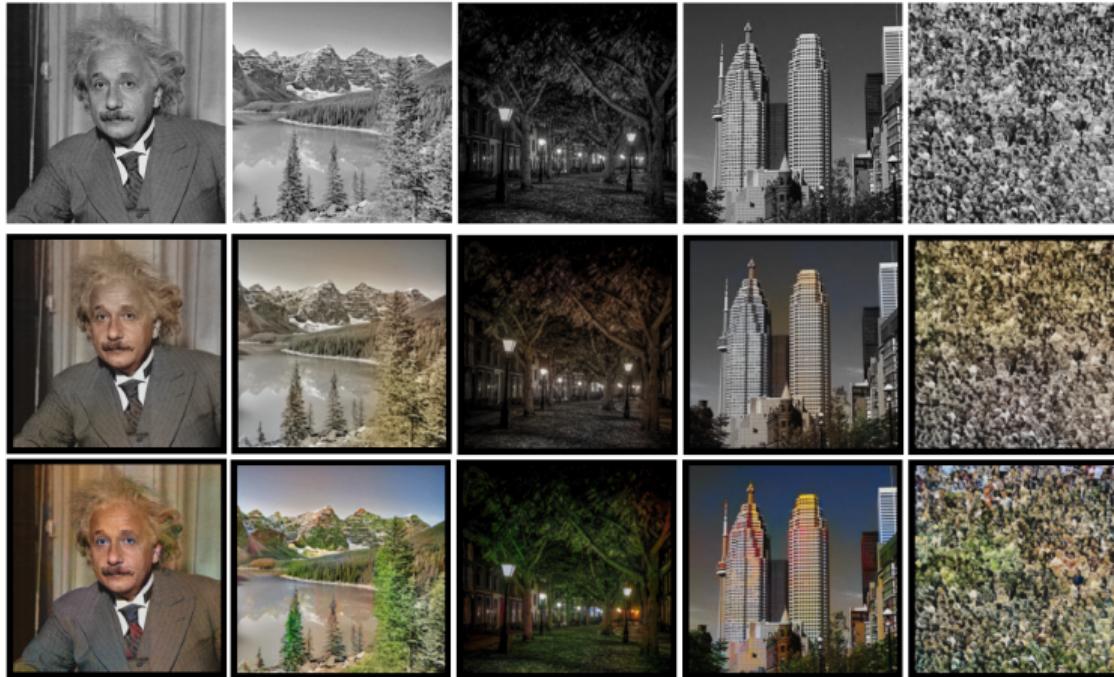


input



prediction

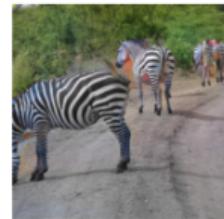
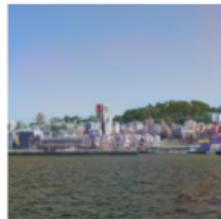
So sánh bộ sinh có xương sống ResNet18 trước và sau khi huấn luyện đối nghịch



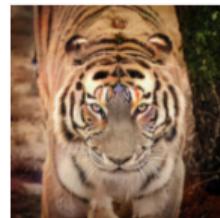
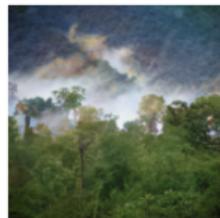
Khảo sát đánh giá định tính chất lượng mô hình



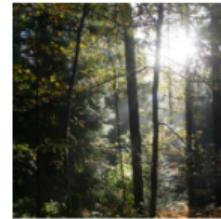
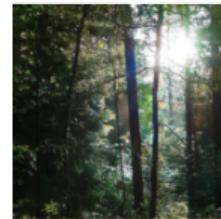
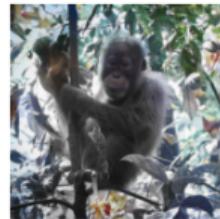
Khảo sát đánh giá định tính chất lượng mô hình



Khảo sát đánh giá định tính chất lượng mô hình



Khảo sát đánh giá định tính chất lượng mô hình



Khảo sát đánh giá định tính chất lượng mô hình

Thang điểm đánh giá từ **1 (rất giả)** đến **5 (rất thật)**.

	Ảnh 01	Ảnh 02	Ảnh 03	Ảnh 04	Ảnh 05
Điểm trung bình	3.65	2.53	3.87	2.88	4.01
	Ảnh 06	Ảnh 07	Ảnh 08	Ảnh 09	Ảnh 10
Điểm trung bình	2.97	3.96	3.86	3.94	3.19
	Ảnh 11	Ảnh 12	Ảnh 13	Ảnh 14	Ảnh 15
Điểm trung bình	3.89	3.56	3.40	3.52	2.75
	Ảnh 16	Ảnh 17	Ảnh 18	Ảnh 19	Ảnh 20
Điểm trung bình	2.91	2.88	3.15	3.89	2.76

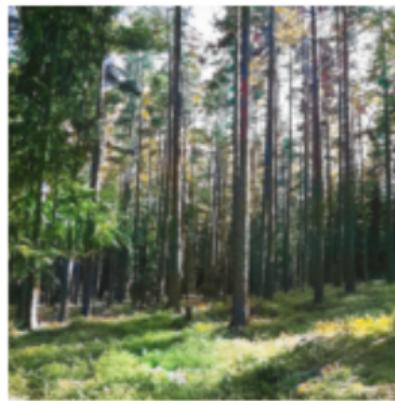
Điểm trung bình của 20 ảnh là: **3.38/5.0.**

Khảo sát đánh giá định tính chất lượng mô hình

Hạng 3 kết quả tốt nhất (ảnh 05, ảnh 07, ảnh 09):



4.01



3.96



3.94

Khảo sát đánh giá định tính chất lượng mô hình

Hạng 3 kết quả tệ nhất (ảnh 02, ảnh 15, ảnh 20):



2.53



2.75



2.76

Cải thiện mô hình với tập dữ liệu lớn hơn

Tập dữ liệu được sử dụng gồm 18,000 ảnh với cách chuẩn hoá và làm giàu dữ liệu *tương tự*.

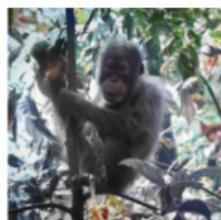
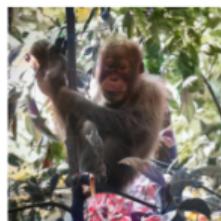
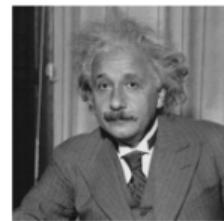
Kiến trúc khởi tạo bộ sinh và bộ phân biệt hoàn toàn *giống* với *mô hình ban đầu*.

Không tiến hành tiền huấn luyện bộ sinh.

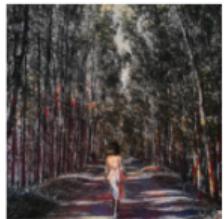
Không sử dụng hàm matsu mít chuẩn 1 trong quá trình huấn luyện đối nghịch.

Kích thước batch: 16, Thuật toán tối ưu: Adam, tốc độ học: $\alpha = 2 \cdot 10^{-4}$, mô men: $\beta = (0.5, 0.999)$.

So sánh kết quả sau khi huấn luyện lại với tập dữ liệu lớn hơn



So sánh kết quả sau khi huấn luyện lại với tập dữ liệu lớn hơn



Đánh giá định lượng hai mô hình trước và sau khi cải thiện thông qua giá trị MSE và giá trị RMSE

Khảo sát 2 ảnh kênh màu (ảnh không tính kênh mức xám).

$\mathcal{I}_{\text{đầu ra}}, \mathcal{I}_{\text{đầu ra } \mathcal{G}_1}, \mathcal{I}_{\text{đầu ra } \mathcal{G}_2}$ đã được chuẩn hóa theo tập $[-110, 110] \times [-110, 110]$.

$$\text{MSE} = \frac{1}{2 \times 256^2} \left\| \widehat{\mathcal{I}}_{\text{đầu ra}} - \mathcal{I}_{\text{đầu ra}} \right\|_F^2 = \frac{1}{2 \times 256^2} \sum_{k=1}^2 \sum_{i,j=1}^{256} \left\{ \left(\widehat{\mathcal{I}}_{\text{đầu ra}} \right)_{ij}^{(k)} - \left(\mathcal{I}_{\text{đầu ra}} \right)_{ij}^{(k)} \right\}^2 \quad (15)$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{2 \times 256^2} \sum_{k=1}^2 \sum_{i,j=1}^{256} \left\{ \left(\widehat{\mathcal{I}}_{\text{đầu ra}} \right)_{ij}^{(k)} - \left(\mathcal{I}_{\text{đầu ra}} \right)_{ij}^{(k)} \right\}^2} \quad (16)$$

Đánh giá định lượng hai mô hình trước và sau khi cải thiện thông qua giá trị MSE và giá trị RMSE

Gọi $\mathcal{G}_1, \mathcal{G}_2$ lần lượt là mô hình được huấn luyện với 10,000 và 18,000 ảnh.

	Ảnh 02 (2.53)	Ảnh 07 (3.96)	Ảnh 09 (3.94)	Ảnh 15 (2.75)
MSE \mathcal{G}_1	374.2613	312.994	240.9714	628.1626
MSE \mathcal{G}_2	409.466	202.8153	150.6662	442.2453
RMSE \mathcal{G}_1	19.3458	17.6916	15.5233	25.0632
RMSE \mathcal{G}_2	20.2353	14.2413	12.2746	21.0296

Mỗi điểm màu chênh khoảng $100\% \times \frac{10}{2 \times 110} = 4.55\%$ đến $100\% \times \frac{20}{2 \times 110} = 9.09\%$ so với khoảng giá trị.

Ứng dụng web tạo màu đơn giản bằng VueJS kết hợp API tạo màu Django

The screenshot shows a web browser window with the following details:

- Title Bar:** Bách Khoa TP.HCM - Tạo màu
- Address Bar:** 192.168.1.3:8080
- Content Area:**
 - Header:** BÁCH KHOA TP.HCM - TẠO MÀU CHO ẢNH XÁM [\[Github\]](#)
 - Logo:** Bách Khoa TP.HCM logo
 - Buttons:** CHỌN ẢNH (Select Image) and TẠO MÀU (Create Color)
 - Image Preview:** Two side-by-side grayscale images of a smiling man (Elon Musk).

Tổng kết

Kết quả: Thành công xây dựng mô hình, đánh giá mô hình ở mức trung bình khá \Rightarrow Đã hoàn thành được yêu cầu đề tài đề ra.

Hạn chế hiện có: Mô hình vẫn còn tô lem ở một số mẫu lạ, nhiều chi tiết hoặc một số trường hợp đặc biệt.

Hướng cải thiện mô hình:

- Thêm nhiều tập dữ liệu hơn.
- Sử dụng một xương sống tốt hơn: ResNet34, ResNet50, Xception, ...
- Chuyên biệt hóa mô hình.

Hướng phát triển cho đề tài:

- Có thể áp dụng cho video (phim lịch sử, tài liệu chiến tranh,...).
- Kiến trúc mô hình có thể xử lý những bài toán khác.

Cảm ơn các thầy đã lắng nghe!

