

**- Nhập môn học máy -**

Lớp: CQ2018/21 – Học kì II/ 2020-2021

**Giảng viên:** Bùi Tiến Lên

---

# Seminar

## Neural Style Transfer

Nhóm 26: 18120555 – Đặng Hữu Thắng  
18120561 – Quách Hải Thanh

18120584 – Phạm Đình Thực  
18120579 – Đặng Minh Thọ

---

TP.Hồ Chí Minh, ngày 14/7/2021

# NỘI DUNG



01

**Động lực nghiên cứu  
khoa học**



02

**Phát biểu bài toán**



03

**Kiến thức nền tảng**



04

**Bài báo**

Tuesday  
14.07.2021

---

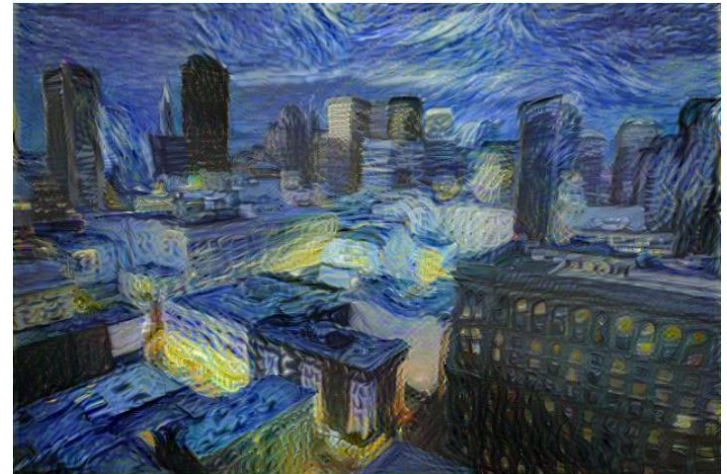
# ĐỘNG LỰC NGHIÊN CỨU

---

# Động lực nghiên cứu khoa học

## ▲ Đặt vấn đề

- Làm thế nào để vẽ lại tranh dựa trên những phong cách đã có?
- Thiết kế những bộ lọc sử dụng cho việc chụp ảnh, video call.



# Phát biểu bài toán

Neural Style Transfer là gì?

Neural Style Transfer (NST) là phương pháp tối ưu cho phép kết hợp giữa hai ảnh (ảnh content và ảnh style) để tạo một bức ảnh giữ lại những đặc trưng gần nhất với hai ảnh đầu vào.

Source image → Style



Target image → Content



Our result



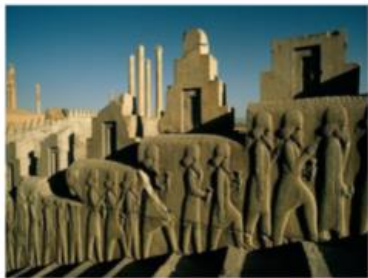
# Phát biểu bài toán

## Tác vụ:

Input: Hình ảnh nội dung và phong cách sẽ được tạo với nhau.

Output: Một hình ảnh duy nhất thể hiện nội dung sau khi được chuyển đổi phong cách.

**content image**



Ancient city of Persepolis

**style image**



The Starry Night (Van Gogh)

**generated image**



Persepolis  
in Van Gogh style

Tuesday  
14.07.2021

---

# KIẾN THỨC NỀN TẢNG

---

# Convolutional Neural Network

## Convolutional Neural Network (CNN) là gì?

Convolutional Neural Network một trong những mô hình Deep learning được ứng dụng trong ngành thị giác máy tính, phân loại ảnh, xử lý ngôn ngữ tự nhiên,...

Sau năm 2012, mô hình đã có sự tiến bộ vượt bậc nhờ sức mạnh của GPU.



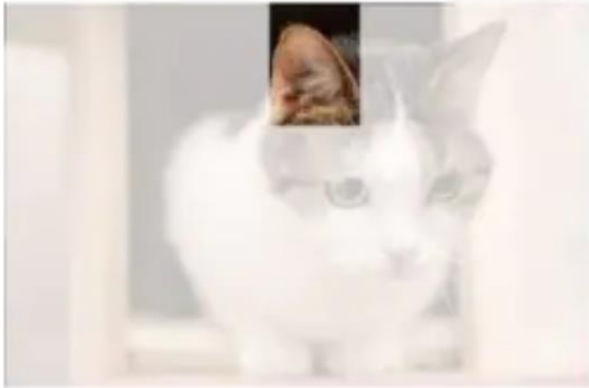


# Convolutional Neural Network

## Đặc trưng của CNN

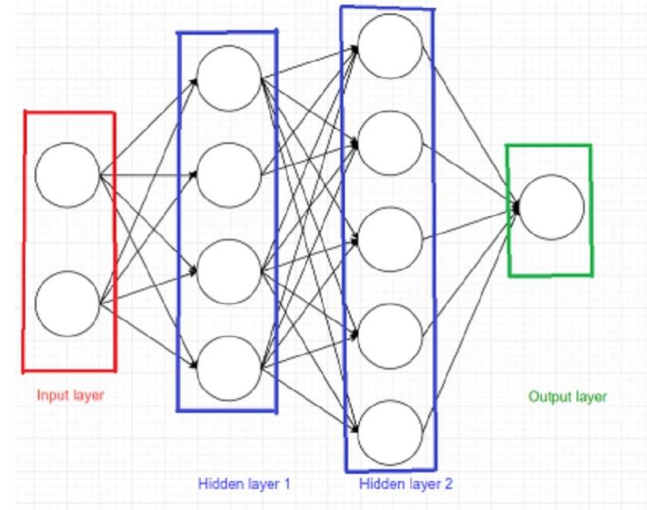
### 1. Feature

CNN so sánh hình ảnh theo từng hình ảnh con để tìm ra sự tương đồng giữa các feature



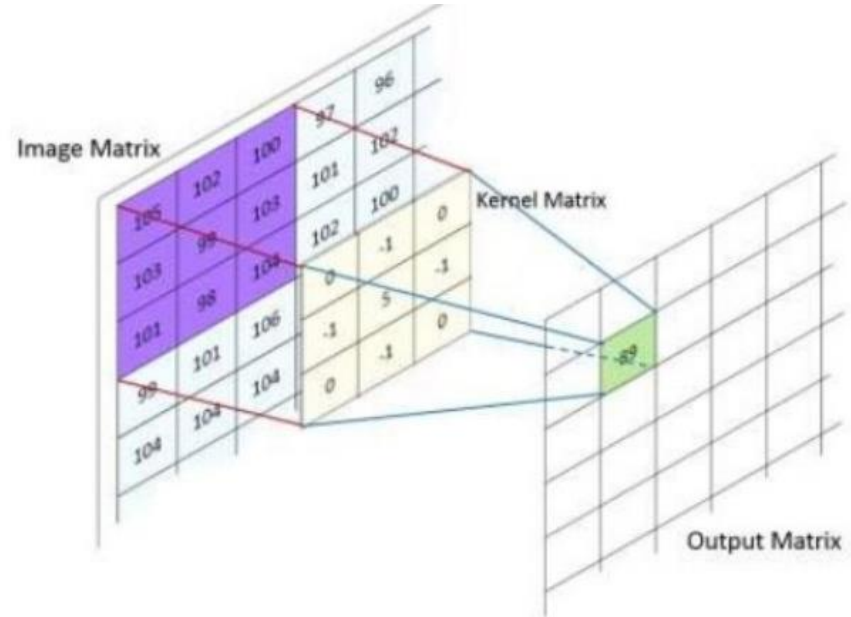
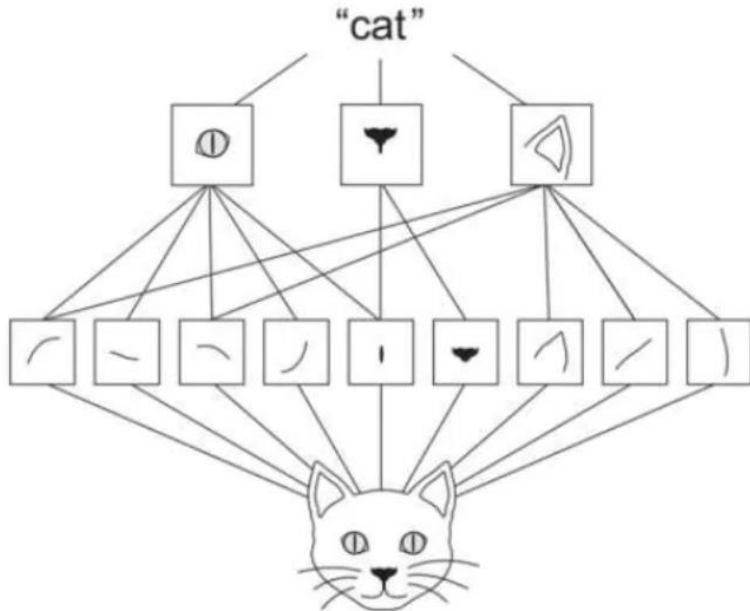
### 2. Convolutional

Phép này dùng để thử các feature ở các vị trí khác nhau để tìm ra vị trí khớp với ảnh. Sử dụng bộ lọc để thu được kết quả cao hơn



# Convolutional Neural Network

## Đặc trưng của CNN



# Convolutional Neural Network

## Các lớp cơ bản của CNN

### 1. Convolutional layer

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

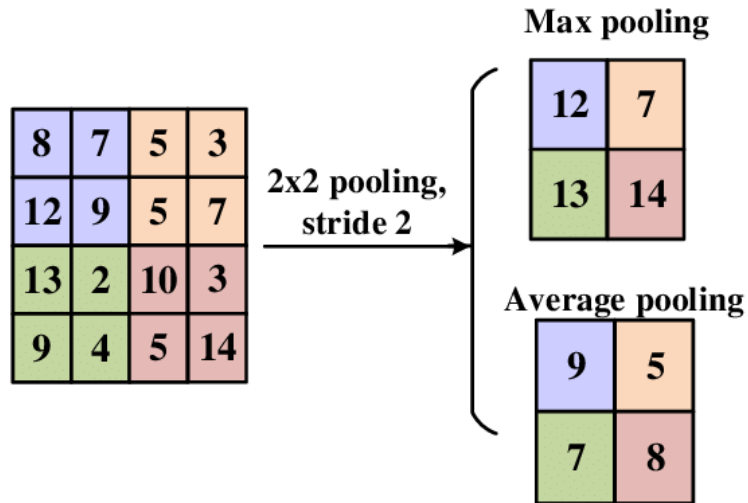
	0	1	0	
	1	-4	1	
	0	1	0	



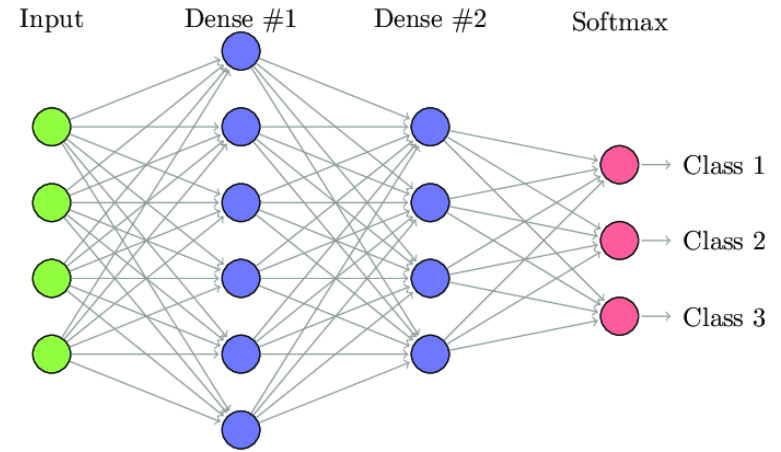
# Convolutional Neural Network

## Các lớp cơ bản của CNN

### 2. Pooling layer



### 3. Fully connected layer



**Tuesday**  
14.07.2021

---

# BÀI BÁO (PAPER)

---

# Giới thiệu về bài báo

**Tên bài báo:** A Neural Algorithm of Artistic Style.

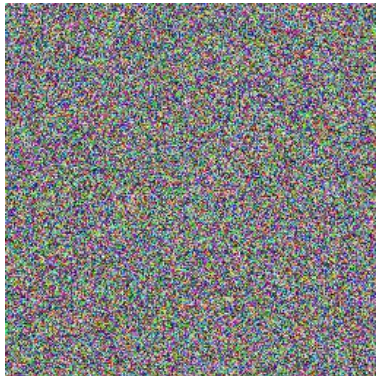
**Nhóm tác giả:** Leon A. Gatys, Alexander S. Ecker, Matthias Bethge  
từ đại học Tübingen, Đức

- Thuật toán mà paper đề ra về cơ bản là thuật toán chuyển đổi đặc trưng (texture transfer algorithm) sử dụng phương pháp tổng hợp đặc trưng dựa trên feature representation của CNN
- Ảnh mới được tạo ra sẽ thỏa điều kiện khớp với cả 2 ảnh content và style (mang đặc trưng), và chúng ta có thể điều chỉnh ảnh nào sẽ ảnh hưởng nhiều hơn đến kết quả
- Loss function:

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

- 3 phần:
  - Content representation
  - Style representation
  - Style transfer

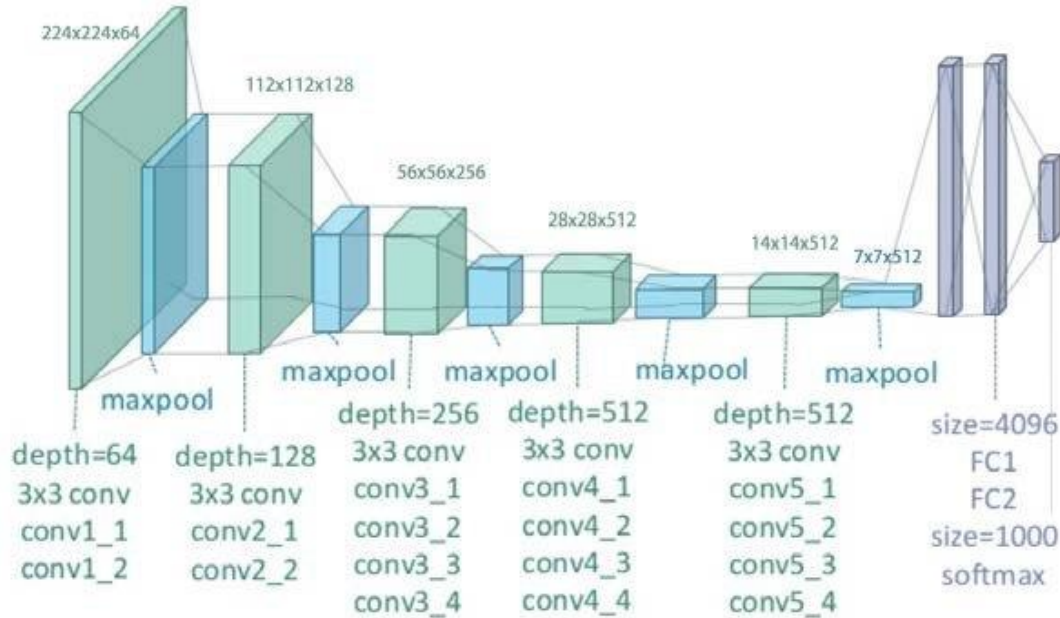
# Giới thiệu về bài báo



*White noise image*

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

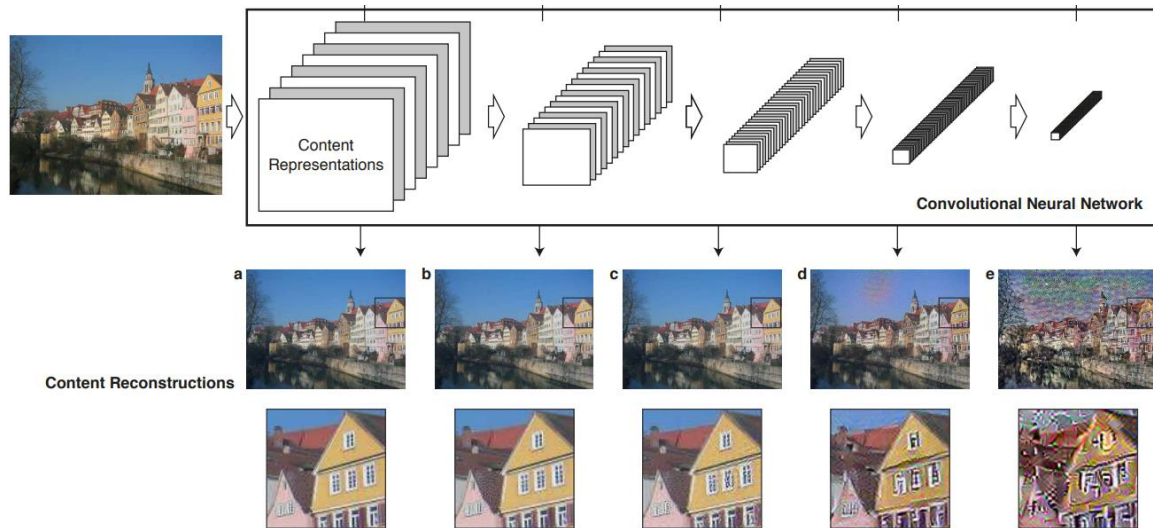
# MODEL



- **VGG-19 Network** dùng trong nhận diện vật thể (object recognition), đã được train, lược bỏ đi các *Fully connected layer*
- Mỗi convolutional layer trong mạng sẽ có 1 bộ các kernel (filter) khác nhau, với độ phức tạp sẽ tăng dần theo độ sâu của mạng



# CONTENT REPRESENTATION



- Từ đó, chúng ta có thể lưu output của mỗi convolution layer bằng ma trận  $F^l \in R^{N_l \times M_l}$ , trong đó  $F_{ij}^l$  là output tại vị trí  $j$  sau khi được đưa qua filter thứ  $i^{th}$  ở layer  $l$

— Cho content image qua convolutional layer, ta sẽ được các feature map. Filter khác nhau sẽ cho ra feature map

— Một convolutional layer  $l$  với  $N_l$  filter khác nhau, sau khi cho input qua ta sẽ được  $N_l$  bộ feature maps khác nhau, mỗi bộ có kích thước  $M_l$  - kích thước của input

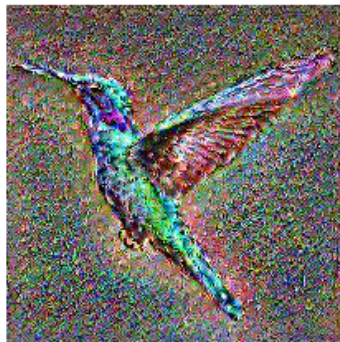
# CONTENT REPRESENTATION



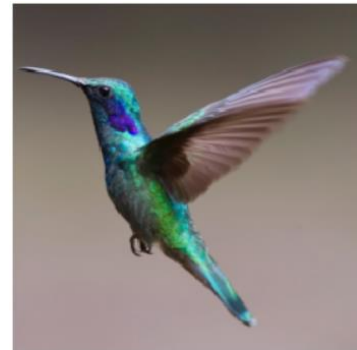
*conv1\_1*



*conv3\_1*



*conv4\_1*



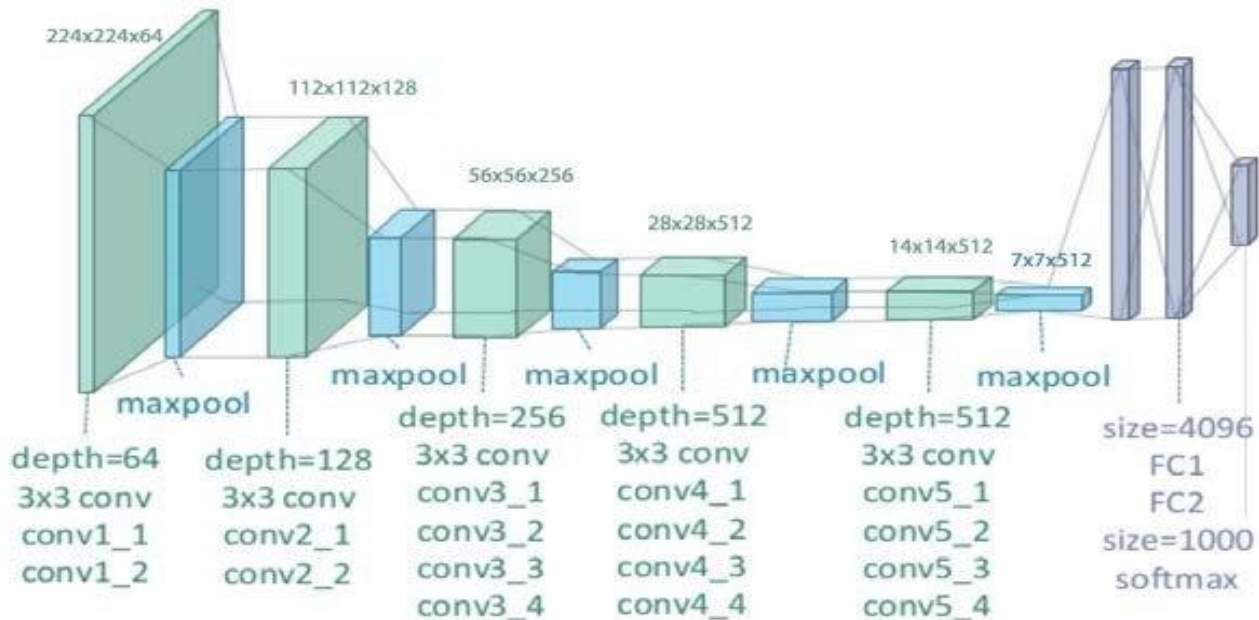
*input*

- Gọi  $\vec{p}$  và  $\vec{x}$  lần lượt là ảnh gốc và ảnh cần visualize ở layer  $l$ , tương ứng  $P^l$  và  $F^l$  lần lượt là ma trận lưu output của 2 ảnh sau khi qua layer  $l$
- Loss function:

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

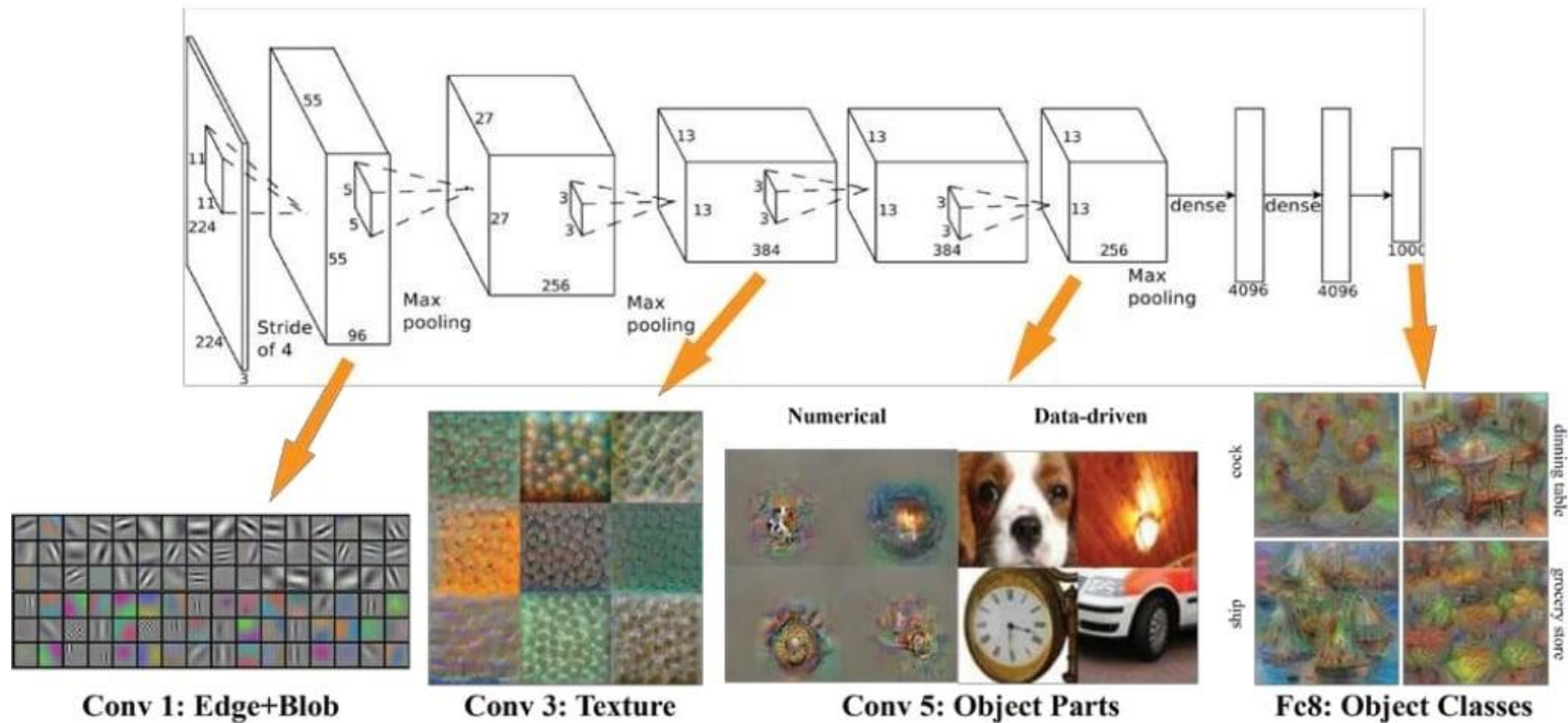
- Dùng Gradient Descent để cực tiểu hóa hàm loss

# CONTENT REPRESENTATION



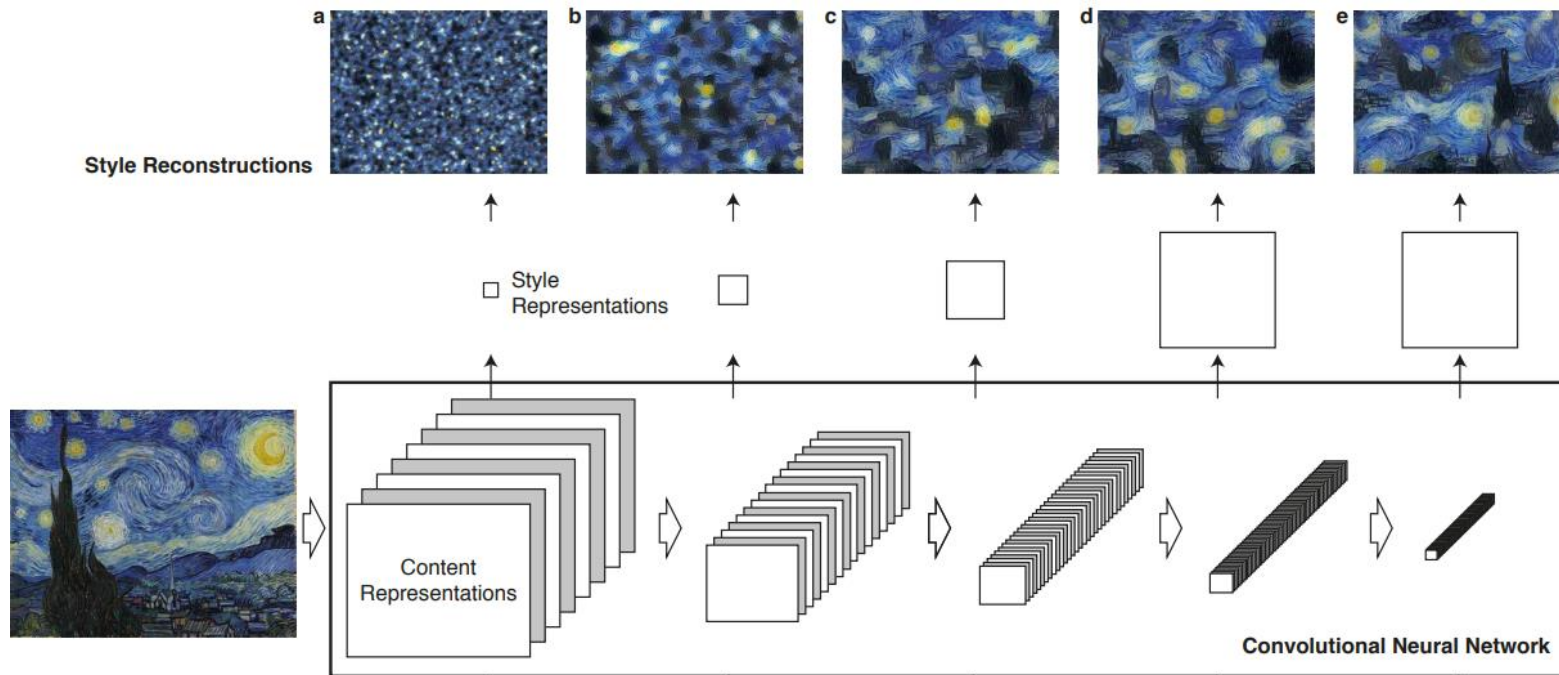
- Càng về các layer sau, các actual-content của bức ảnh sẽ được giữ lại càng nhiều và paper sử dụng output của các layer này làm *content representation*

# CONTENT REPRESENTATION



- Trong các ví dụ được trình bày, paper thường dùng output của layer [conv4\\_2](#)

# STYLE REPRESENTATION



- Gần tương tự như content representation nhưng sử dụng Gram matrix làm style representation
- Dùng feature map ở các layer thấp: *conv1\_1*, *conv2\_1*, *conv3\_1*, *conv4\_1*, *conv5\_1* với trọng số  $w = 0.2$

# GRAM MATRIX

- **Gram matrix** ở layer  $l$ :  $G^l \in R^{N_l \times N_l}$  là ma trận chứa kết quả của các phép **inner product** giữa các feature map đã được vector hóa ở layer  $l$

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l.$$

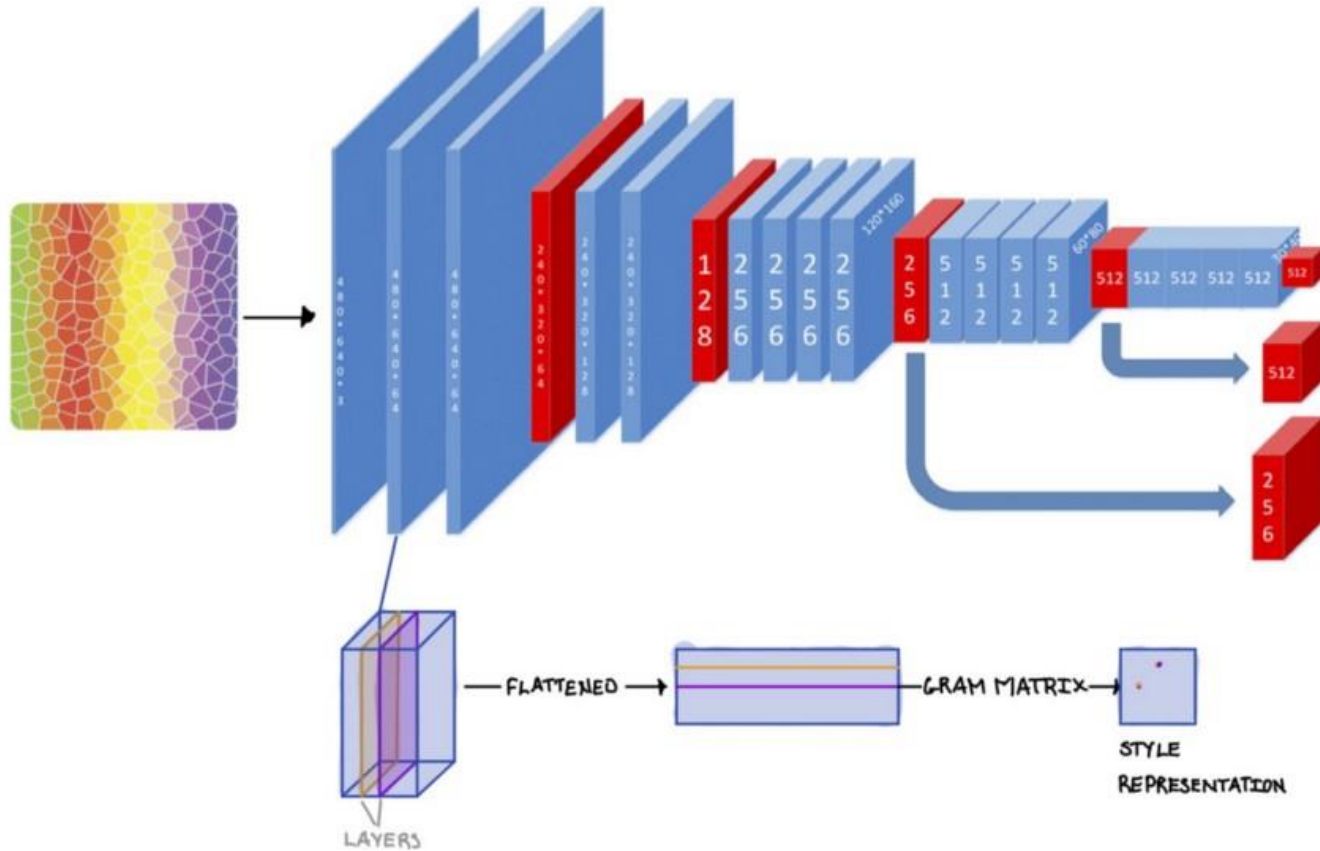
- **Inner product** giữa 2 vector  $\vec{a}(a_1, a_2, a_3, \dots, a_n)$  và  $\vec{b}(b_1, b_2, b_3, \dots, b_n)$

$$a.b = a_1 b_1 + a_2 b_2 + a_3 b_3 + \dots + a_n b_n = \sum_{j=1}^n a_j b_j$$

- Kết quả **inner product** giữa 2 vector càng lớn  $\rightarrow$  2 vector này càng giống nhau



# GRAM MATRIX

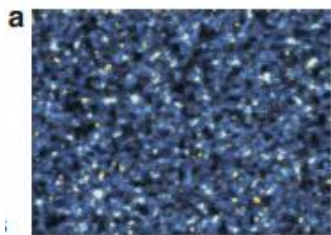


# STYLE REPRESENTATION

- Gọi  $\vec{a}$  và  $\vec{x}$  lần lượt là ảnh gốc và ảnh cần visualize ở layer  $l$ , tương ứng  $A^l$  và  $G^l$  lần lượt là **Gram matrix** tương ứng của 2 ảnh sau khi qua layer  $l$
- Loss function:

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l,$$

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$



(conv)1\_1



1\_1 + 2\_1



1\_1 + 2\_1 + 3\_1



1\_1 + 2\_1 + 3\_1  
+ 4\_1



1\_1 + 2\_1 + 3\_1  
+ 4\_1 + 5\_1



# STYLE TRANSFER

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

- $\alpha$  và  $\beta$  là 2 trọng số có thể điều chỉnh được, tùy vào mục đích
- Dùng L-BFGS thay cho Gradient descent



B



$$\frac{\alpha}{\beta} = 10^{-3}$$



# STYLE TRANSFER



$$\frac{\alpha}{\beta} = 10^{-3}$$



$$\frac{\alpha}{\beta} = 8 \times 10^{-4}$$

# TỔNG KẾT



Newyork by night  
*Style image*



London daytime  
*Content image*



# TỔNG KẾT



Output

**Tuesday**  
14.07.2021

---

DEMO

---

# TÀI LIỆU THAM KHẢO

- [Gram matrix - Wikipedia](#)
- [Neural Networks Intuitions: 2. Dot product, Gram Matrix and Neural Style Transfer | by Raghul Asokan | Towards Data Science](#)
- [An Intuitive Understanding to Neural Style Transfer | by Eddie Huang | Towards Data Science](#)
- <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/#Convolution>
- <https://nttuan8.com/bai-6-convolutional-neural-network/>
- <https://www.slideshare.net/keshan88/neural-style-transfer?fbclid=IwAR3qdV3E2he0slSFSAKLFBtH4JRICZCvNFLU1Rv3eRSpx61-p2UHKmU6pyY>
- [https://web.cs.ucdavis.edu/~yjee/teaching/ecs289g-fall2016/wenjian.pdf?fbclid=IwAR3ita1BhpemGR\\_9gFla3kKOLd9t5jTIIX9iwICAs\\_QcZcQVCmHChXh560Q](https://web.cs.ucdavis.edu/~yjee/teaching/ecs289g-fall2016/wenjian.pdf?fbclid=IwAR3ita1BhpemGR_9gFla3kKOLd9t5jTIIX9iwICAs_QcZcQVCmHChXh560Q)
- <https://www.quora.com/What-is-the-VGG-19-neural-network>
- <https://towardsdatascience.com/convolutional-neural-network-feature-map-and-filter-visualization-f75012a5a49c>
- <https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/>

Tuesday  
14.07.2021

---

CẢM ƠN THẦY  
ĐÃ LẮNG NGHE!

---