VIP Cheatsheet: Mạng nơ ron tích chập

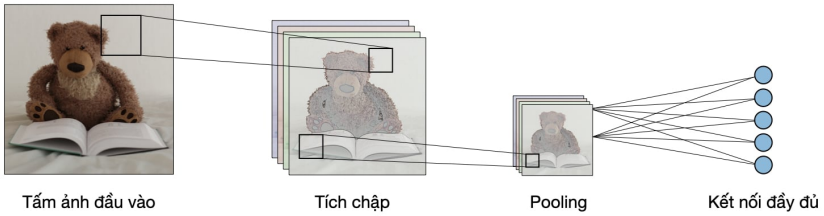
Afshine Amidi và Shervine Amidi

Ngày 17 tháng 5 năm 2020

Dịch bởi Phạm Hồng Vinh và Đàm Minh Tiến

Tổng quan

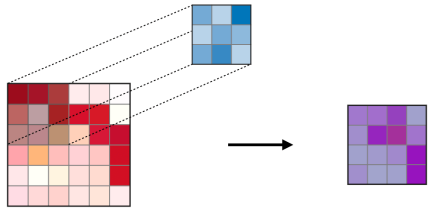
❒ **Kiến trúc truyền thống của một mạng CNN** – Mạng neural tích chập (Convolutional neural networks), còn được biết đến với tên CNNs, là một dạng mạng neural được cấu thành bởi các tầng sau:



Tầng tích chập và tầng pooling có thể được hiệu chỉnh theo các siêu tham số (hyperparameters) được mô tả ở những phần tiếp theo.

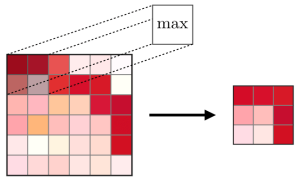
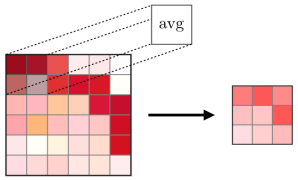
Các kiểu tầng

❒ **Tầng tích chập (CONV)** – Tầng tích chập (CONV) sử dụng các bộ lọc để thực hiện phép tích chập khi đưa chúng đi qua đầu vào I theo các chiều của nó. Các siêu tham số của các bộ lọc này bao gồm kích thước bộ lọc F và độ trượt (stride) S. Kết quả đầu ra O được gọi là feature map hay activation map.

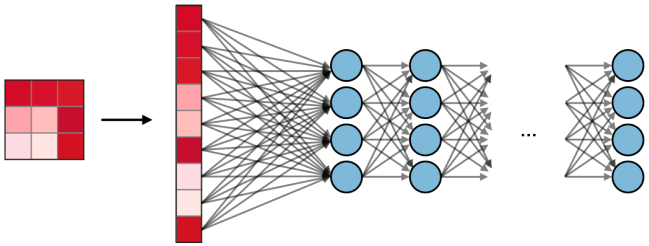


Lưu ý: Bước tích chập cũng có thể được khái quát hóa cả với trường hợp một chiều (1D) và ba chiều (3D).

❒ **Pooling (POOL)** – Tầng pooling (POOL) là một phép downsampling, thường được sử dụng sau tầng tích chập, giúp tăng tính bất biến không gian. Cụ thể, max pooling và average pooling là những dạng pooling đặc biệt, mà tương ứng là trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.

	Max pooling	**Average pooling**
Kiểu	Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng	Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng
Minh họa		
Nhận xét	- Bảo toàn các đặc trưng đã phát hiện - Được sử dụng thường xuyên	- Giảm kích thước feature map - Được sử dụng trong mạng LeNet

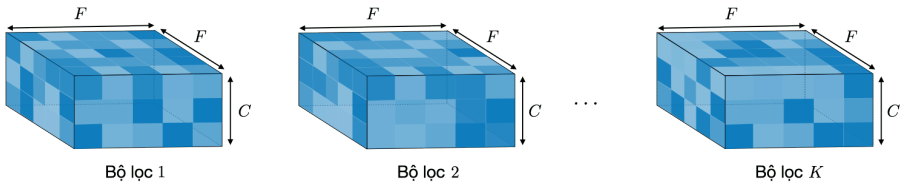
❒ **Fully Connected (FC)** – Tầng kết nối đầy đủ (FC) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả neuron. Trong mô hình mạng CNNs, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp.



Các siêu tham số của bộ lọc

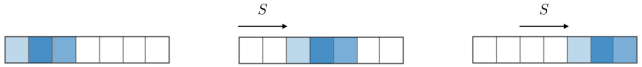
Tầng tích chập chứa các bộ lọc mà rất quan trọng cho ta khi biết ý nghĩa đằng sau các siêu tham số của chúng.

❒ **Các chiều của một bộ lọc** – Một bộ lọc kích thước $F \times F$ áp dụng lên đầu vào chứa C kênh (channels) thì có kích thước tổng kể là $F \times F \times C$ thực hiện phép tích chập trên đầu vào kích thước $I \times I \times C$ và cho ra một feature map (hay còn gọi là activation map) có kích thước $O \times O \times 1$.

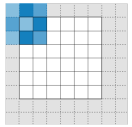


Lưu ý: Việc áp dụng K bộ lọc có kích thước $F \times F$ cho ra một feature map có kích thước $O \times O \times K$.

❒ **Stride** – Đối với phép tích chập hoặc phép pooling, độ trượt S ký hiệu số pixel mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính.



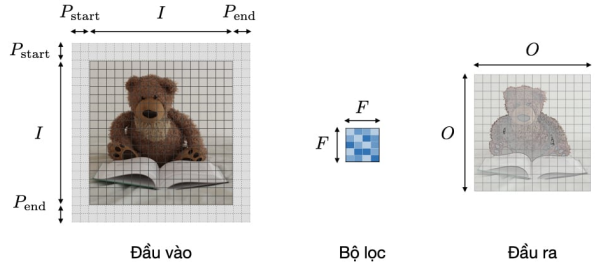
❒ **Zero-padding** – Zero-padding là tên gọi của quá trình thêm P số không vào các biên của đầu vào. Giá trị này có thể được lựa chọn thủ công hoặc một cách tự động bằng một trong ba những phương pháp mô tả bên dưới:

	Valid	**Same**	**Full**
Giá trị	$P = 0$	$P_{\text{start}} = \left\lfloor \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rfloor$ $P_{\text{end}} = \left\lceil \frac{S\lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rceil$	$P_{\text{start}} \in [\![0, F-1]\!]$ $P_{\text{end}} = F - 1$
Minh họa			
Mục đích	- Không sử dụng padding - Bỏ phép tích chập cuối nếu số chiều không khớp	- Sử dụng padding để làm cho feature map có kích thước $\left\lceil \frac{I}{S} \right\rceil$ - Kích thước đầu ra thuận lợi về mặt toán học - Còn được gọi là 'half' padding	- Padding tối đa sao cho các phép tích chập có thể được sử dụng tại các rìa của đầu vào - Bộ lọc 'thấy' được đầu vào từ đầu đến cuối

Điều chỉnh siêu tham số

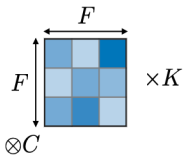
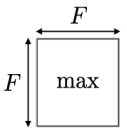
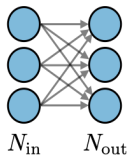
❒ **Tính tương thích của tham số trong tầng tích chập** – Bằng cách ký hiệu I là độ dài kích thước đầu vào, F là độ dài của bộ lọc, P là số lượng zero padding, S là độ trượt, ta có thể tính được độ dài O của feature map theo một chiều bằng công thức:

$$\boxed{O = \frac{I - F + P_{\text{start}} + P_{\text{end}}}{S} + 1}$$



Lưu ý: Trong một số trường hợp, $P_{start} = P_{end} \triangleq P$, ta có thể thay thế $P_{start} + P_{end}$ bằng $2P$ trong công thức trên.

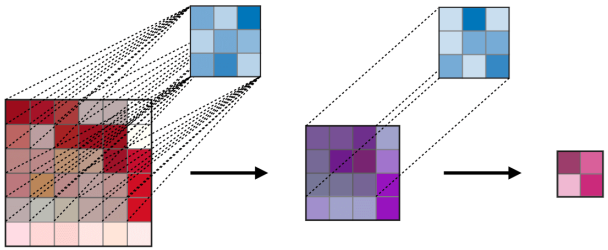
❒ **Hiểu về độ phức tạp của mô hình** – Để đánh giá độ phức tạp của một mô hình, cách hữu hiệu là xác định số tham số mà mô hình đó sẽ có. Trong một tầng của mạng neural tích chập, nó sẽ được tính toán như sau:

	CONV	**POOL**	**FC**
Minh họa			
Kích thước đầu vào	$I \times I \times C$	$I \times I \times C$	N_{in}
Kích thước đầu ra	$O \times O \times K$	$O \times O \times C$	N_{out}
Số lượng tham số	$(F \times F \times C + 1) \cdot K$	0	$(N_{\text{in}} + 1) \times N_{\text{out}}$
Lưu ý	- Một tham số bias với mỗi bộ lọc - Trong đa số trường hợp, $S < F$ - Một lựa chọn phổ biến cho K là $2C$	- Phép pooling được áp dụng lên từng kênh (channel-wise) - Trong đa số trường hợp, $S = F$	- Đầu vào được làm phẳng - Mỗi neuron có một tham số bias - Số neuron trong một tầng FC phụ thuộc vào ràng buộc kết cấu

❒ **Trường thụ cảm** – Trường thụ cảm (receptive field) tại tầng k là vùng được ký hiệu $R_k \times R_k$ của đầu vào mà những pixel của activation map thứ k có thể "nhìn thấy". Bằng cách gọi F_j là kích thước bộ lọc của tầng j và S_i là giá trị độ trượt của tầng i và để thuận tiện, ta mặc định $S_0 = 1$, trường thụ cảm của tầng k được tính toán bằng công thức:

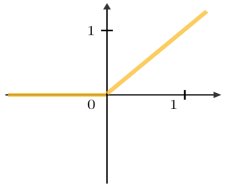
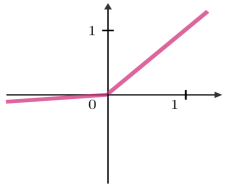
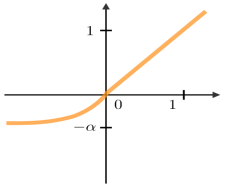
$$\boxed{R_k = 1 + \sum_{j=1}^{k} (F_j - 1) \prod_{i=0}^{j-1} S_i}$$

Trong ví dụ bên dưới, ta có $F_1 = F_2 = 3$ và $S_1 = S_2 = 1$, nên cho ra được $R_2 = 1+2 \cdot 1+2 \cdot 1 = 5$..



Các hàm kích hoạt thường gặp

❒ **Rectified Linear Unit** – Tầng rectified linear unit (ReLU) là một hàm kích hoạt g được sử dụng trên tất cả các thành phần. Mục đích của nó là tăng tính phi tuyến tính cho mạng. Những biến thể khác của ReLU được tổng hợp ở bảng dưới:

ReLU	Leaky ReLU	ELU
$g(z) = \max(0,z)$	$g(z) = \max(\epsilon z,z)$ với $\epsilon \ll 1$	$g(z) = \max(\alpha(e^z - 1),z)$ với $\alpha \ll 1$
		
Độ phức tạp phi tuyến tính có thể thông dịch được về mặt sinh học	Gán vấn đề ReLU chết cho những giá trị âm	Khả vi tại mọi nơi

❒ **Softmax** – Bước softmax có thể được coi là một hàm logistic tổng quát lấy đầu vào là một vector chứa các giá trị $x \in \mathbb{R}^n$ và cho ra là một vector gồm các xác suất $p \in \mathbb{R}^n$ thông qua một hàm softmax ở cuối kiến trúc. Nó được định nghĩa như sau:

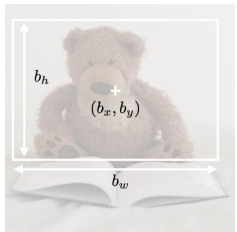
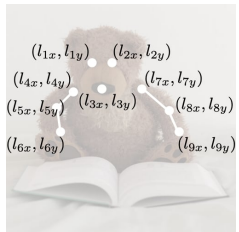
$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{với} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Phát hiện vật thể (object detection)

❒ **Các kiểu mô hình** – Có 3 kiểu thuật toán nhận diện vật thể chính, vì thế mà bản chất của thứ được dự đoán sẽ khác nhau. Chúng được miêu tả ở bảng dưới:

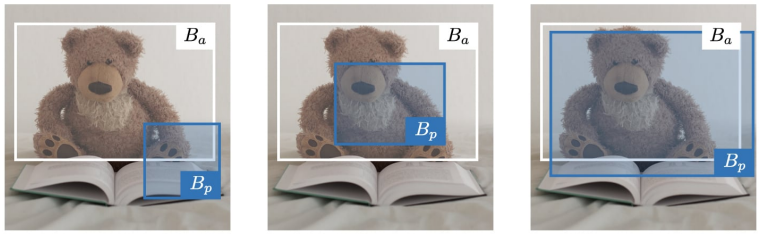
Phân loại hình ảnh	Phân loại cùng với khoanh vùng	Phát hiện
		
- Phân loại một tấm ảnh - Dự đoán xác suất của một vật thể	- Phát hiện một vật thể trong ảnh - Dự đoán xác suất của vật thể và định vị nó	- Phát hiện nhiều vật thể trong cùng một tấm ảnh - Dự đoán xác suất của các vật thể và định vị chúng
CNN cổ điển	YOLO đơn giản hóa, R-CNN	YOLO, R-CNN

❒ **Phát hiện** – Trong bối cảnh phát hiện (detection) vật thể, những phương pháp khác nhau được áp dụng tùy thuộc vào liệu chúng ta chỉ muốn định vị vật thể hay phát hiện được những hình dạng phức tạp hơn trong tấm ảnh. Hai phương pháp chính được tổng hợp ở bảng dưới:

Phát hiện hộp giới hạn (bounding box)	Phát hiện landmark
Phát hiện phần trong ảnh mà có sự xuất hiện của vật thể	- Phát hiện hình dạng và đặc điểm của một đối tượng (vd: mắt) - Nhiều hạt
	
Hộp có tọa độ trung tâm (b_x,b_y), chiều cao b_h và chiều rộng b_w	Các điểm tương quan (l_{1x},l_{1y}), ..., (l_{nx},l_{ny})

❒ **Intersection over Union** – Tỉ lệ vùng giao trên vùng hợp, còn được biết đến là IoU, là một hàm định lượng vị trí B_p của hộp giới hạn dự đoán được định vị đúng như thế nào so với hộp giới hạn thực tế B_a. Nó được định nghĩa:

$$\text{IoU}(B_p,B_a) = \frac{B_p \cap B_a}{B_p \cup B_a}$$



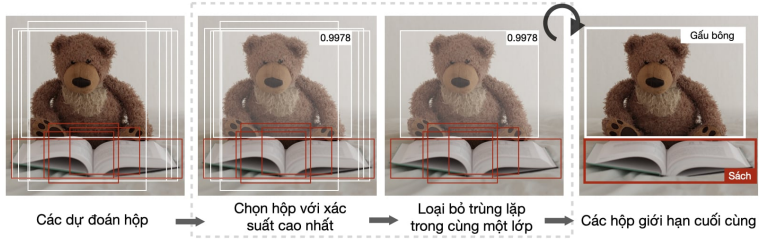
$\text{IoU}(B_p, B_a) = 0.1$ $\text{IoU}(B_p, B_a) = 0.5$ $\text{IoU}(B_p, B_a) = 0.9$

Lưu ý: ta luôn có $IoU \in [0,1]$. Để thuận tiện, một hộp giới hạn B_p được cho là khá tốt nếu $IoU(B_p,B_a) \geqslant 0.5$.

❒ **Anchor boxes** – Hộp mỏ neo là một kỹ thuật được dùng để dự đoán những hộp giới hạn nằm chồng lên nhau. Trong thực nghiệm, mạng được phép dự đoán nhiều hơn một hộp cùng một lúc, trong đó mỗi dự đoán được giới hạn theo một tập những tính chất hình học cho trước. Ví dụ, dự đoán đầu tiên có khả năng là một hộp hình chữ nhật có hình dạng cho trước, trong khi dự đoán thứ hai sẽ là một hộp hình chữ nhật nữa với hình dạng hình học khác.

❒ **Non-max suppression** – Kỹ thuật non-max suppression hướng tới việc loại bỏ những hộp giới hạn bị trùng chồng lên nhau của cùng một đối tượng bằng cách chọn chiếc hộp có tính đặc trưng nhất. Sau khi loại bỏ tất cả các hộp có xác suất dự đoán nhỏ hơn 0.6, những bước tiếp theo được lặp lại khi vẫn còn tồn tại những hộp khác.

- Bước 1: Chọn chiếc hộp có xác suất dự đoán lớn nhất.
- Bước 2: Loại bỏ những hộp có IoU $\geqslant 0.5$ với hộp đã chọn.



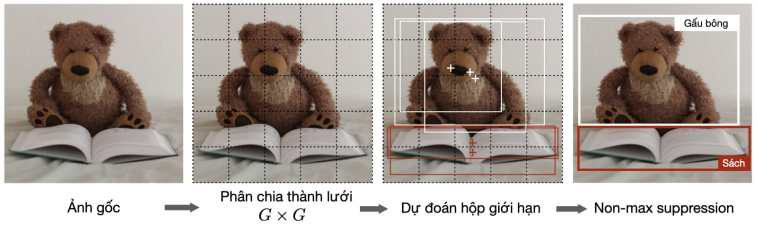
❒ **YOLO** – You Only Look Once (YOLO) là một thuật toán phát hiện vật thể thực hiện những bước sau:

- Bước 1: Phân chia tấm ảnh đầu vào thành một lưới $G \times G$.
- Bước 2: Với mỗi lưới, chạy một mạng CNN dự đoán y có dạng sau:

$$y = \Big[\underbrace{p_c, b_x, b_y, b_h, b_w, c_1, c_2, \ldots, c_p}_{\text{lặp lại } k \text{ lần}}, \ldots\Big]^T \in \mathbb{R}^{G \times G \times k \times (5+p)}$$

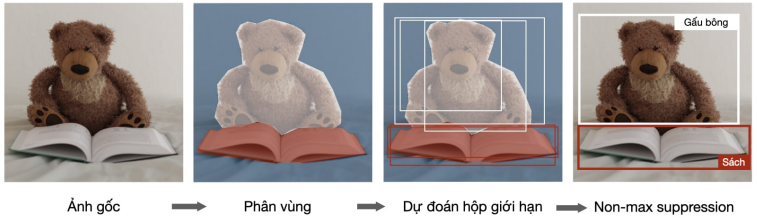
với p_c là xác suất dự đoán được một vật thể, b_x,b_y,b_h,b_w là những thuộc tính của hộp giới hạn được dự đoán, c_1,...,c_p là biểu diễn one-hot của việc lớp nào trong p các lớp được dự đoán, và k là số lượng các hộp mỏ neo.

- Bước 3: Chạy thuật toán non-max suppression để loại bỏ bất kỳ hộp giới hạn có khả năng bị trùng lặp.



Lưu ý: khi $p_c = 0$, thì mạng không phát hiện bất kỳ vật thể nào. Trong trường hợp đó, Các dự đoán liên quan $b_x, \ldots, c_p$ sẽ bị lờ đi.

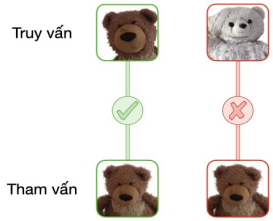
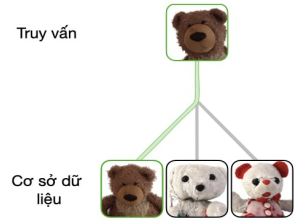
❒ **R-CNN** – Region with Convolutional Neural Networks (R-CNN) là một thuật toán phát hiện vật thể mà đầu tiên phân chia ảnh thành các vùng để tìm các hộp giới hạn có khả năng liên quan cao rồi chạy một thuật toán phát hiện để tìm những thứ có khả năng cao là vật thể trong những hộp giới hạn đó.



Lưu ý: mặc dù thuật toán gốc có chi phí tính toán cao và chậm, những kiến trúc mới đã có thể cho phép thuật toán này chạy nhanh hơn, như là Fast R-CNN và Faster R-CNN.

Xác nhận khuôn mặt và nhận diện khuôn mặt

❒ **Các kiểu mô hình** – Hai kiểu mô hình chính được tổng hợp trong bảng dưới:

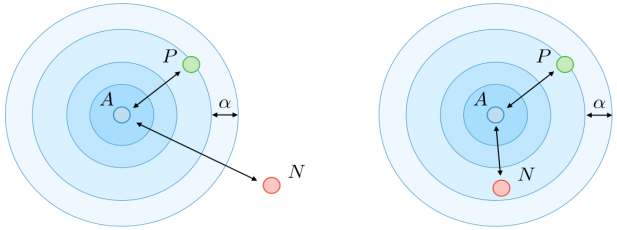
Xác nhận khuôn mặt	Nhận diện khuôn mặt
- Có đúng người không? - Tra cứu một-một	- Đây có phải là 1 trong K người trong cơ sở dữ liệu không? - Tra cứu một với tất cả
	

❒ **One Shot Learning** – One Shot Learning là một thuật toán xác minh khuôn mặt sử dụng một tập huấn luyện hạn chế để học một hàm similarity nhằm ước lượng sự khác nhau giữa hai tấm hình. Hàm này được áp dụng cho hai tấm ảnh thường được ký hiệu $d(\text{image 1}, \text{image 2})$.

❒ **Siamese Network** – Siamese Networks hướng tới việc học cách mã hóa tấm ảnh để rồi định lượng sự khác nhau giữa hai tấm ảnh. Với một tấm ảnh đầu vào $x^{(i)}$, đầu ra được mã hóa thường được ký hiệu là $f(x^{(i)})$.

❒ **Triplet loss** – Triplet loss ℓ là một hàm mất mát được tính toán dựa trên biểu diễn nhúng của bộ ba hình ảnh A (mỏ neo), P (dương tính) và N(âm tính). Ảnh mỏ neo và ảnh dương tính đều thuộc một lớp, trong khi đó ảnh âm tính thuộc về một lớp khác. Bằng các gọi $\alpha \in \mathbb{R}^+$ là tham số margin, hàm mất mát này được định nghĩa như sau:

$$\boxed{\ell(A,P,N) = \max\left(d(A,P) - d(A,N) + \alpha, 0\right)}$$



$\ell(A,P,N) = 0$ $\ell(A,P,N) > 0$

Neural style transfer

❒ **Ý tưởng** – Mục tiêu của neural style transfer là tạo ra một ảnh G dựa trên một nội dung C và một phong cách S.

Nội dung C Phong cách S Ảnh tạo được G

❒ **Tầng kích hoạt** – Trong một tầng l cho trước, tầng kích hoạt được ký hiệu $a^{[l]}$ và có các chiều là $n_H \times n_w \times n_c$

❒ **Hàm mất mát nội dung** – Hàm mất mát nội dung $J_{\text{content}}(C,G)$ được sử dụng để xác định nội dung của ảnh được tạo G khác biệt với nội dung gốc trong ảnh C. Nó được định nghĩa như dưới đây:

$$\boxed{J_{\text{content}}(C,G) = \frac{1}{2}||a^{[l](C)} - a^{[l](G)}||^2}$$

❒ **Ma trận phong cách** – Ma trận phong cách $G^{[l]}$ của một tầng cho trước l là một ma trận Gram mà mỗi thành phần $G^{[l]}_{kk'}$ của ma trận xác định sự tương quan giữa kênh k và kênh k'. Nó được định nghĩa theo tầng kích hoạt $a^{[l]}$ như sau:

$$\boxed{G^{[l]}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_w^{[l]}} a^{[l]}_{ijk} a^{[l]}_{ijk'}}$$

Lưu ý: ma trận phong cách cho ảnh phong cách và ảnh được tạo được ký hiệu tương ứng là $G^{[l](S)}$ và $G^{[l](G)}$.

❒ **Hàm mất mát phong cách** – Hàm mất mát phong cách $J_{\text{style}}(S,G)$ được sử dụng để xác định sự khác biệt về phong cách giữa ảnh được tạo G và ảnh phong cách S. Nó được định nghĩa như sau:

$$\boxed{J^{[l]}_{\text{style}}(S,G) = \frac{1}{(2n_H n_w n_c)^2}||G^{[l](S)} - G^{[l](G)}||^2_F = \frac{1}{(2n_H n_w n_c)^2} \sum_{k,k'=1}^{n_c} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'}\right)^2}$$

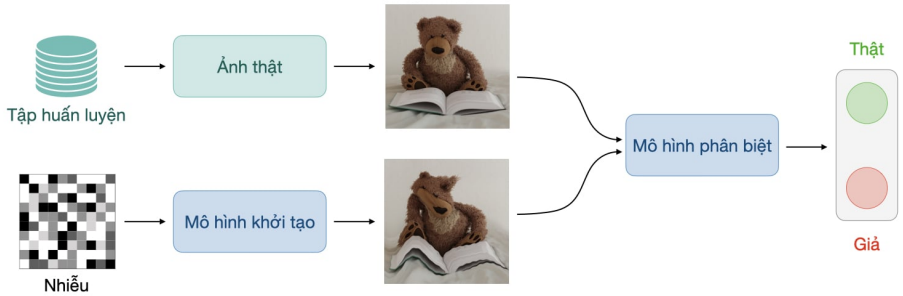
❒ **Hàm mất mát tổng quát** – Hàm mất mát tổng quát được định nghĩa là sự kết hợp của hàm mất mát nội dung và hàm mất mát phong cách, độ quan trọng của chúng được xác định bởi hai tham số α,β, như dưới đây:

$$\boxed{J(G) = \alpha J_{\text{content}}(C,G) + \beta J_{\text{style}}(S,G)}$$

Lưu ý: giá trị của α càng lớn dẫn tới việc mô hình sẽ quan tâm hơn cho nội dung, trong khi đó, giá trị của β càng lớn sẽ khiến nó quan tâm hơn đến phong cách.

Những kiến trúc sử dụng computational tricks

❒ **Generative Adversarial Network** – Generative adversarial networks, hay còn được gọi là GAN, là sự kết hợp giữa mô hình khởi tạo và mô hình phân biệt, khi mà mô hình khởi tạo cố gắng tạo ra hình ảnh đầu ra chân thực nhất, sau đó được đưa vô mô hình phân biệt, mà mục tiêu của nó là phân biệt giữa ảnh được tạo và ảnh thật.



Lưu ý: có nhiều loại GAN khác nhau bao gồm từ văn bản thành ảnh, sinh nhạc và tổ hợp.

❒ **ResNet** – Kiến trúc Residual Network (hay còn gọi là ResNet) sử dụng những khối residual (residual blocks) cùng với một lượng lớn các tầng để giảm lỗi huấn luyện. Những khối residual có những tính chất sau đây:

$$\boxed{a^{[l+2]} = g(a^{[l]} + z^{[l+2]})}$$

❒ **Inception Network** – Kiến trúc này sử dụng những inception module và hướng tới việc thử các tầng tích chập khác nhau để tăng hiệu suất thông qua sự đa dạng của các feature. Cụ thể, kiến trúc này sử dụng thủ thuật tầng tích chập 1×1 để hạn chế gánh nặng tính toán.

⋆ ⋆ ⋆