

Deep Learning in practical application

Nguyen Hong Thinh



Các kiến thức nền tảng

- *0.1 Làm thế nào để sử dụng deep-learning cho một bài toán cụ thể?*

Các bước thực hiện để có 1 model

- Xác định rõ yêu cầu bài toán: đầu vào là gì, đầu ra là gì?

Đầu vào/đầu ra

↗ Đầu vào:

- ↗ -Toàn bộ ảnh (full-frame)+ID
- ↗ -Ảnh crop (face) or box-coordinator (object detection...)
- ↗ Đầu ra: ID(label) cho từng đối tượng; EMBvector; box (vị trí object)

↗ VD: Emotion Recognition:

- ↗ Input: Face crop; Emotion_id
- ↗ output: Emotion-id

- Xác định lượng dữ liệu sẵn có
 - =>Cấu trúc mạng: độ sâu của mạng (số lớp, số conv mỗi lớp...), **hàm loss...**
 - =>Design: (**tool**) Tensorflow, Keras,Caffe..

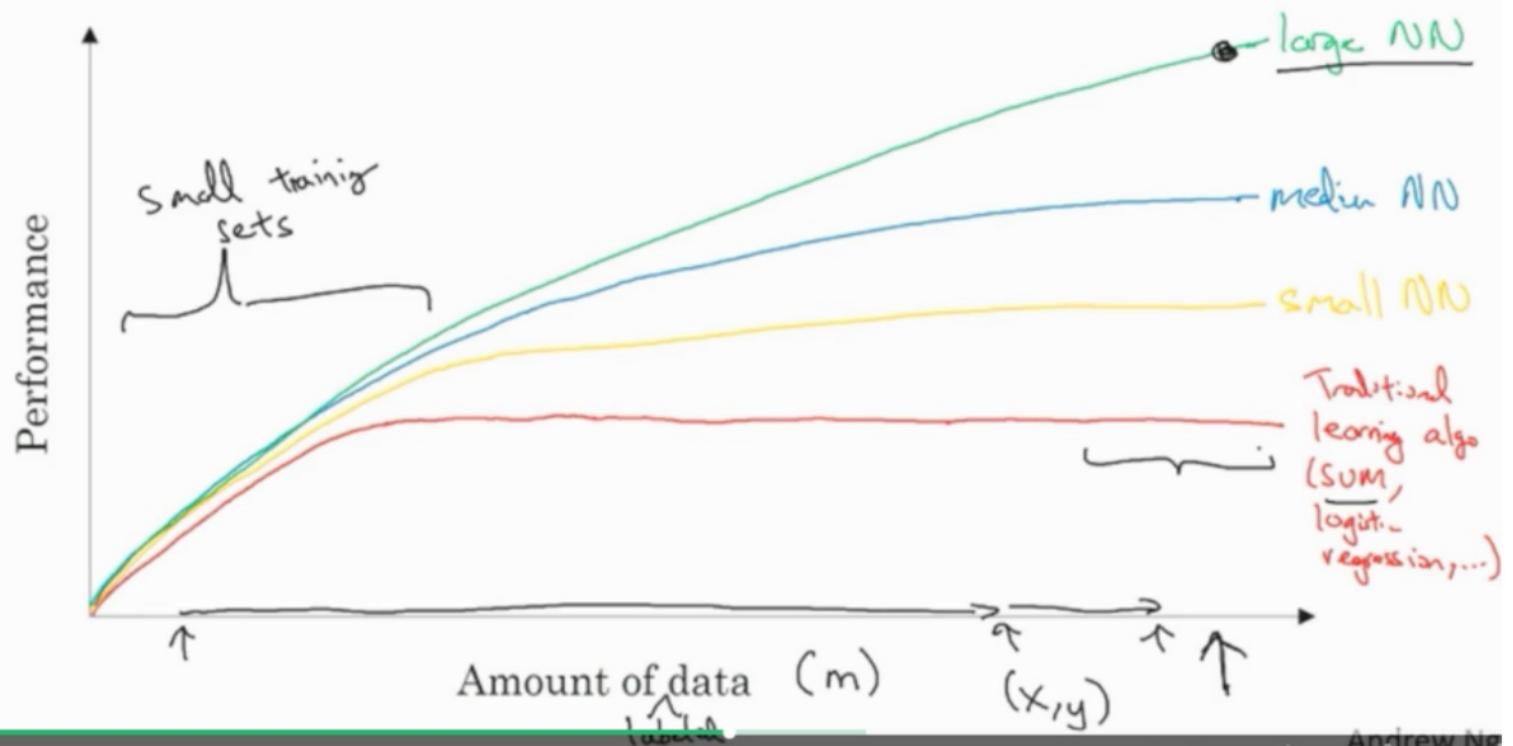
- ↗ Training/testing:
 - ↗ -Cấu hình phần cứng
 - ↗ -Lượng epoch training
- ⇒ **Save model, và sử dụng cho dữ liệu thực**

Dữ liệu

- ↗ Chuẩn hoá dữ liệu (size, crop image (re-shape), normalized, posed, light...)
- ↗ Tuỳ vào lượng dữ liệu=> kích thước (độ sâu) của mạng
- ↗ Nếu dữ liệu ít quá???
 - ↗ Data augmentation
 - ↗ Data synthesis
 - ↗ Transfer Learning // Fine tuning

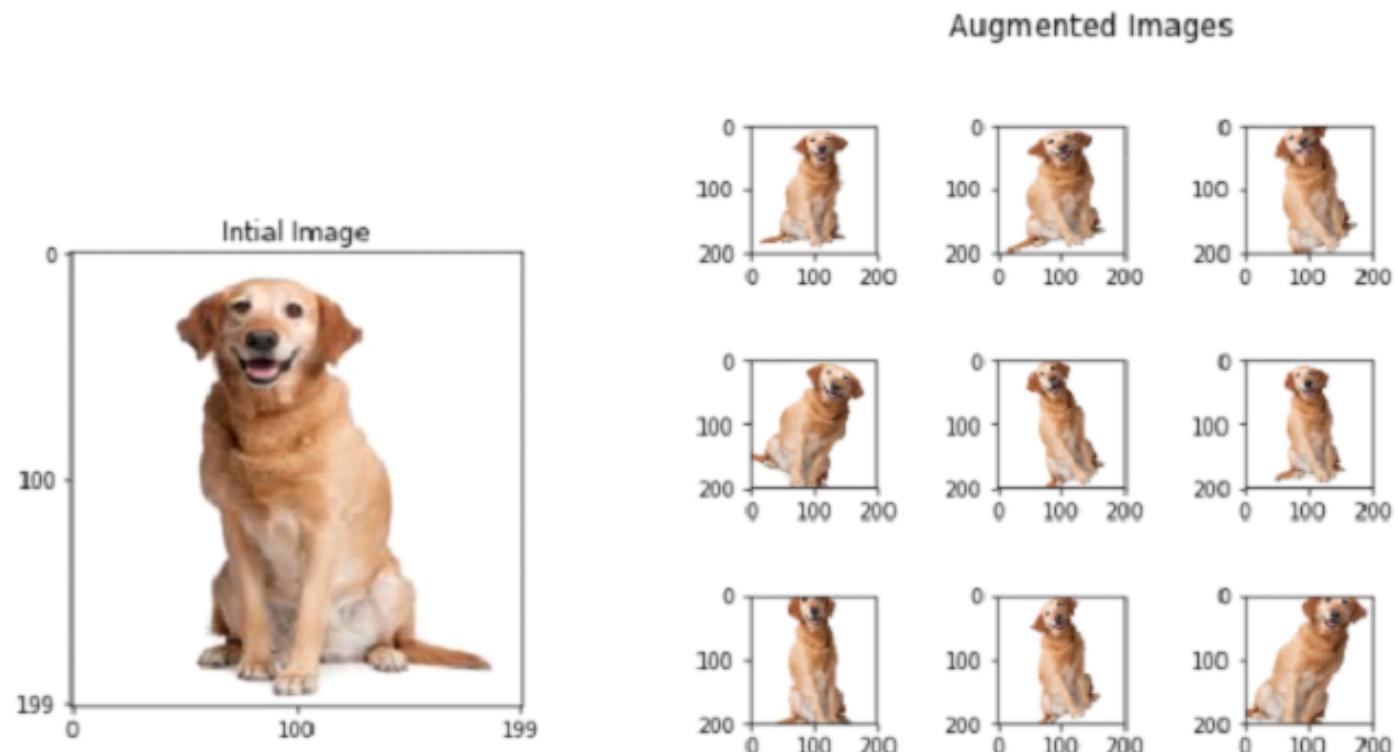
Kích thước dữ liệu và kích thước mạng

Scale drives deep learning progress



Dữ liệu

- ↗ **Data augmentation:** Ảnh có đặc điểm là chứa thông tin theo không gian=> xoay ảnh, lật ảnh, crop ảnh (dịch trái/phải/trên dưới) hoặc thêm nhiễu ta vẫn có thể “nhìn ra” nội dung ảnh



Keras: data augmentation :

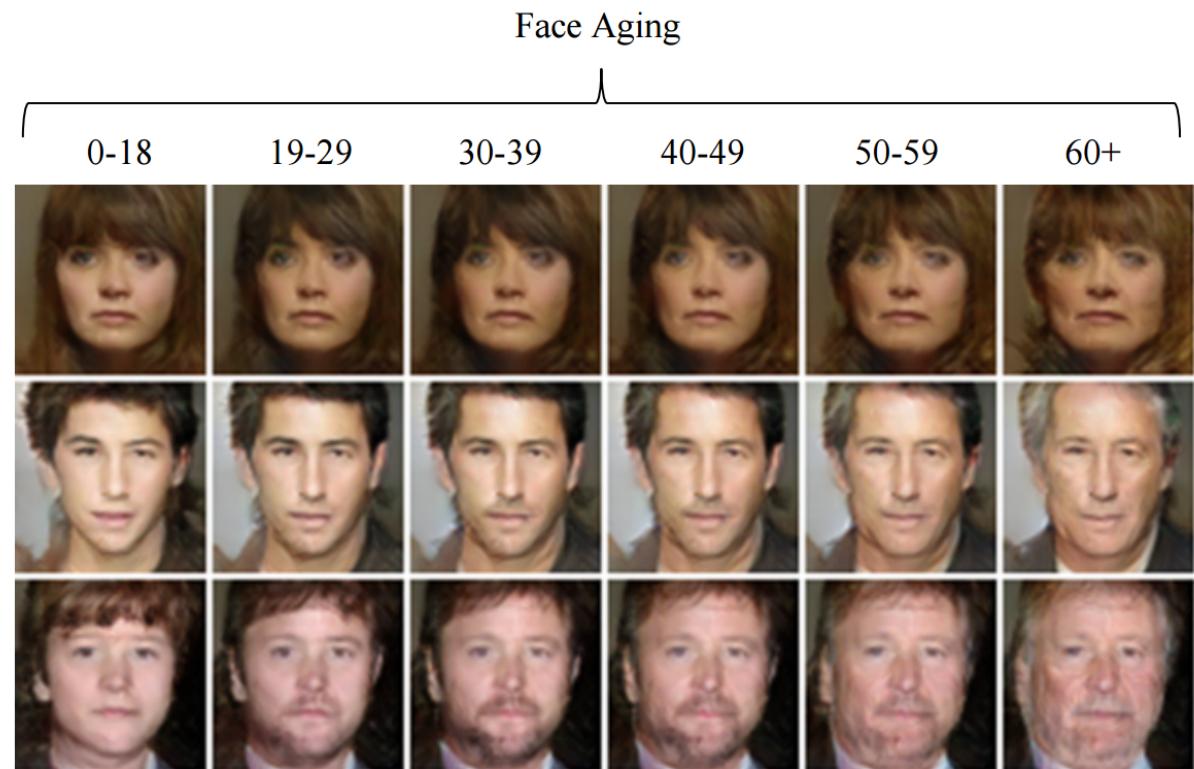
- ↗ Accepting a batch of images used for training.
- ↗ Taking this batch and applying a **series of random transformations** to each image in the batch (including random rotation, resizing, shearing, etc.).
- ↗ Replacing the original batch with the new, randomly transformed batch.
- ↗ Training the CNN on this randomly transformed batch (i.e., the original data itself is not used for training)

```
datagen = ImageDataGenerator(  
    featurewise_center=True,  
    featurewise_std_normalization=True,  
    rotation_range=20,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    horizontal_flip=True)
```

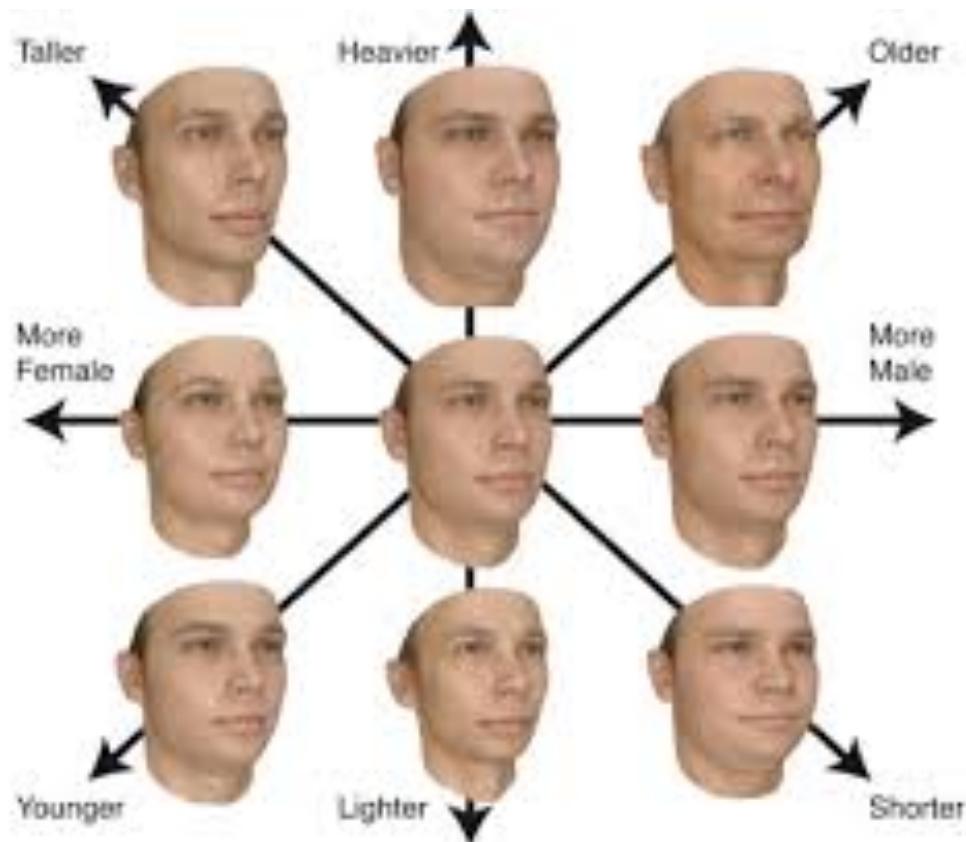
Dữ liệu

- ↗ **Data synthesis:**
- ↗ Ảnh giả/ ảnh tổng hợp,
- ↗ ảnh dựng lại từ tập ảnh có sẵn, có **cùng đặc điểm** như tập ảnh ban đầu
- ↗ **+GAN:**
 - ↗ Sử dụng các cấu trúc GAN để tạo dựng dữ liệu
 - ↗ Tuỳ vào mục đích cụ thể ta có thể tự training GAN hoặc dùng mạng có sẵn
- ↗ **+Dùng 3D model** (sử dụng cho ảnh mặt)

VD về sử dụng GAN-face aging



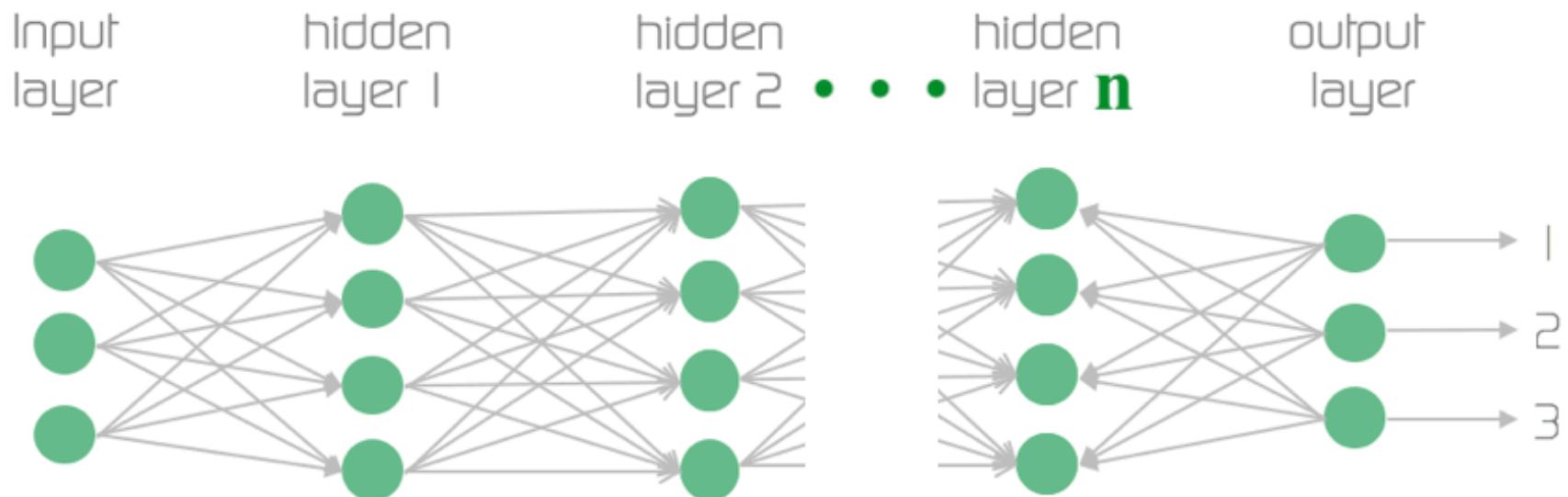
Tạo ảnh giả từ mô hình 3D face



- Như vậy, từ một dataset nhỏ, có thể sử dụng các kỹ thuật để tạo ra tập dữ liệu lớn.
- **Ưu điểm:** Tiết kiệm thời gian gán nhãn (label) cho dữ liệu
- **Nhược điểm:**
 - +Mất tính đa dạng (vì dữ liệu mới tạo ra có đặc trưng giống với dữ liệu ban đầu)
 - +Gây sai số (ảnh GAN, 3D)

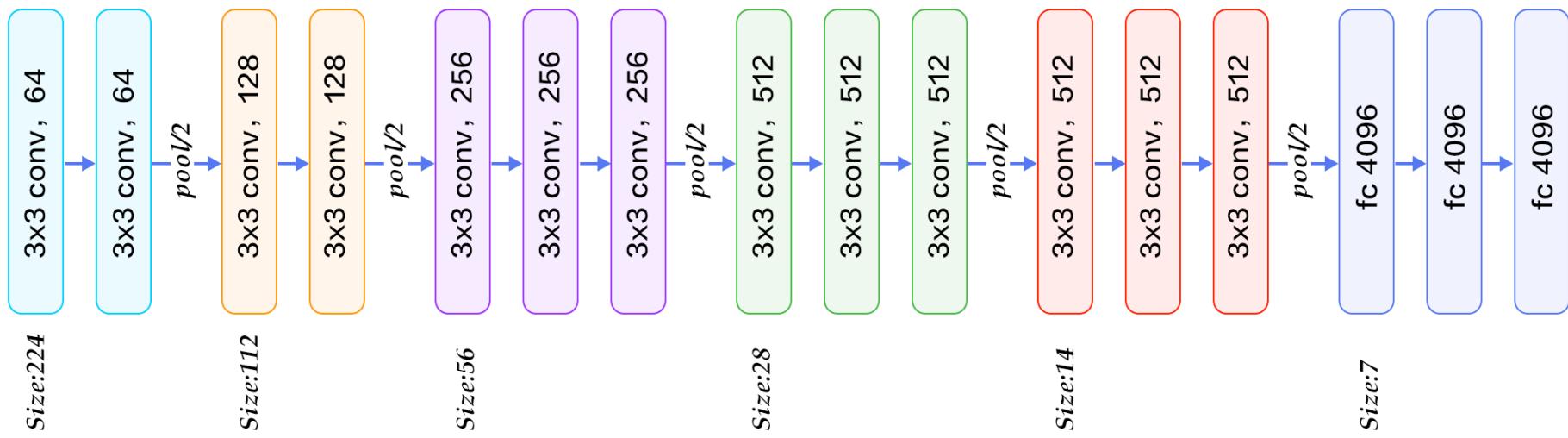
Thiết kế một mạng deeplearning

➤ Cấu trúc chung của một mạng CNN:



↗ Một của layer???

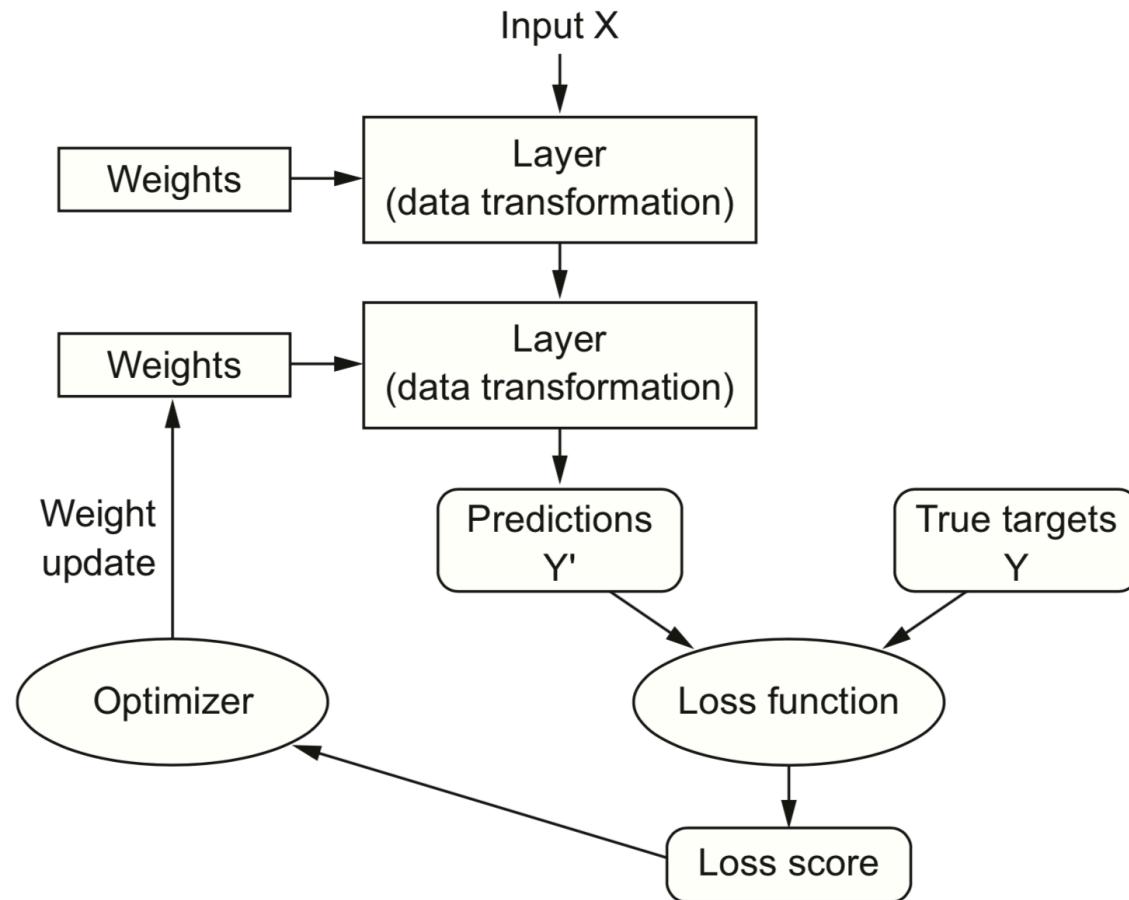
- ↗ Conv, pooling, fully-connected
- ↗ Drop-out



Training và Testing

- ↗ Quá trình training và testing luôn được tiến hành song song để kiểm tra độ hội tụ của mạng
- ↗ Thông thường data-training được tự động tách làm 2 phần, 1 phần training và 1 phần để test
- ↗ Training = back-propagations

Back-propagation



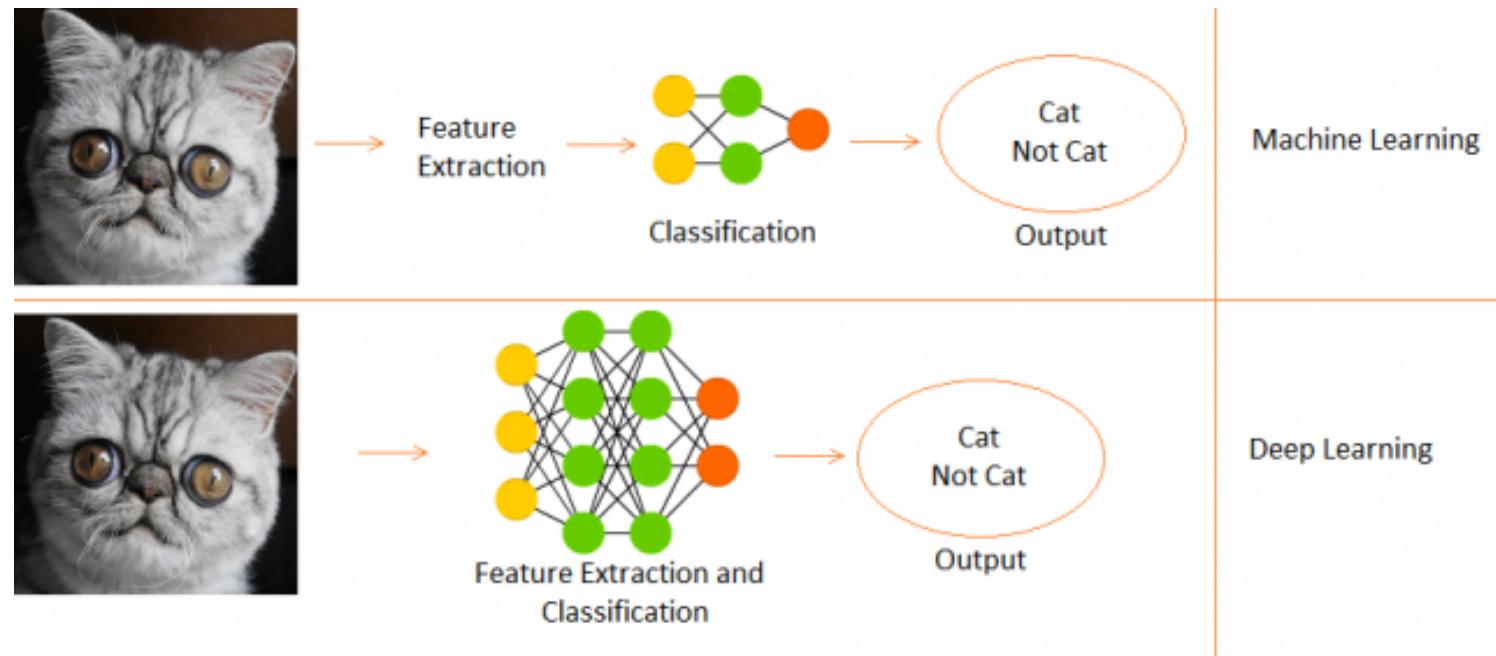
Training dữ liệu

- ↗ Quá trình training từ đầu gọi là “**training from scratch**”
- ↗ Quá trình này rất **tốn thời gian, tài nguyên (phần cứng, dữ liệu...)**
- ↗ Có cách nào rút ngắn thời gian tính toán, tài nguyên, tận dụng kiến thức có sẵn?

Thiết kế mạng và training Model design



Meaning of deep-learning

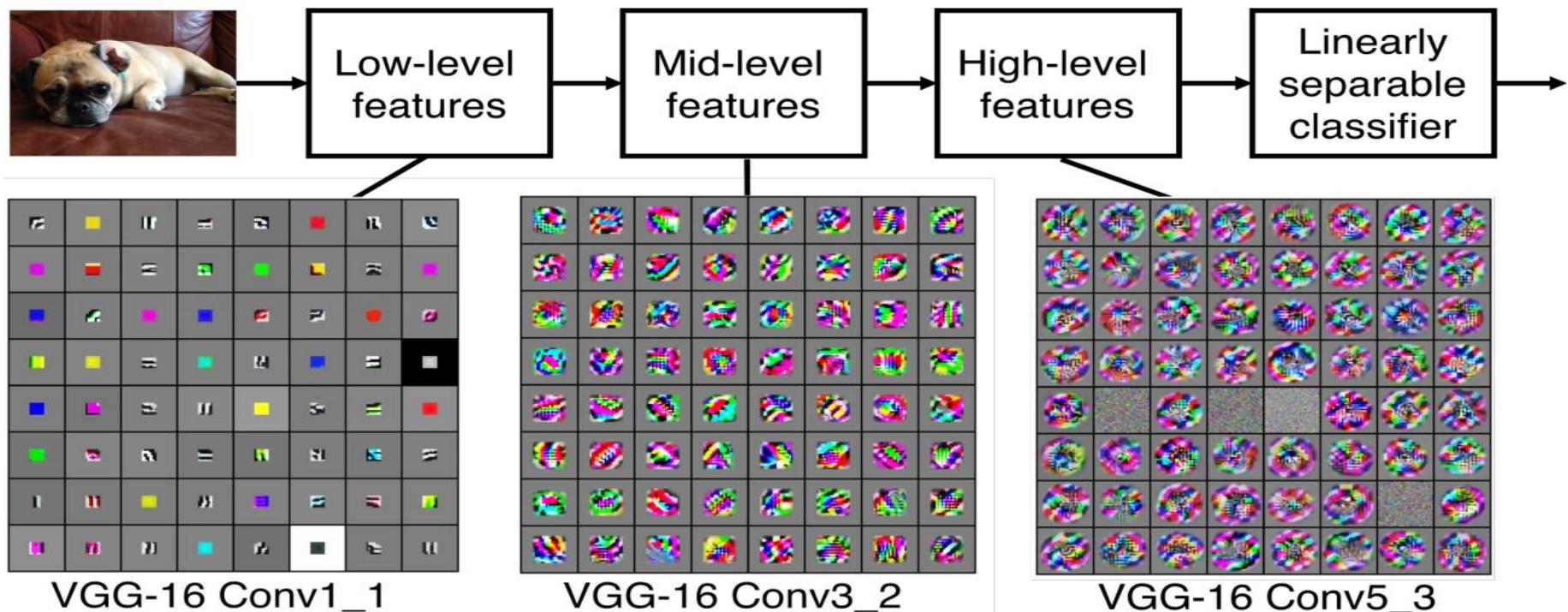


Nhiệm vụ của từng layers?

- khi mạng hội tụ, người ta thực hiện visualized dữ liệu khi đi qua từng layers, để xem tác dụng của từng layers đối với dữ liệu cụ thể

VGG-16

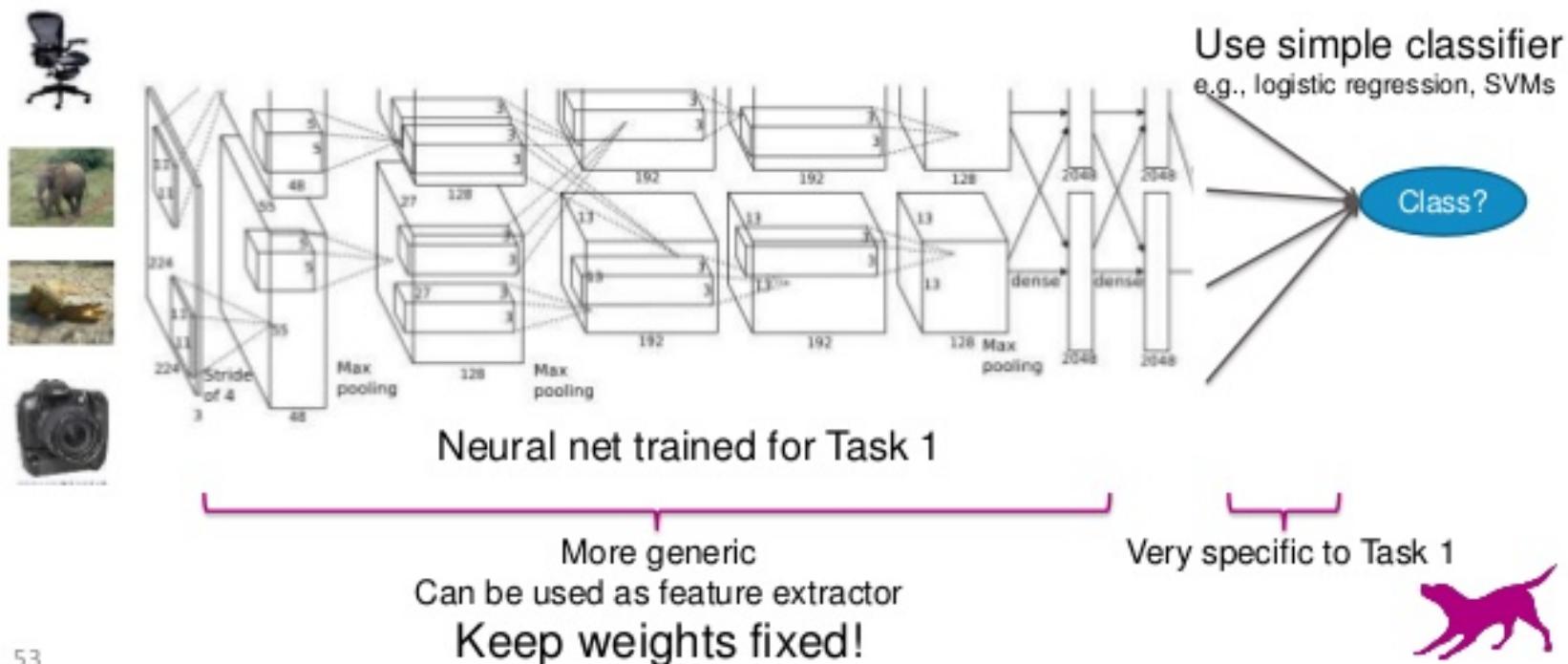
Resnet :
conv filters ở các lowlevel + midlevel: giống với VGG 16



- Với các dữ liệu tương đồng thì quá trình extract features (low level và mid-level) *thường giống nhau*
- =>  Fine-tuning hoặc transfer learning

Transfer learning in more detail...

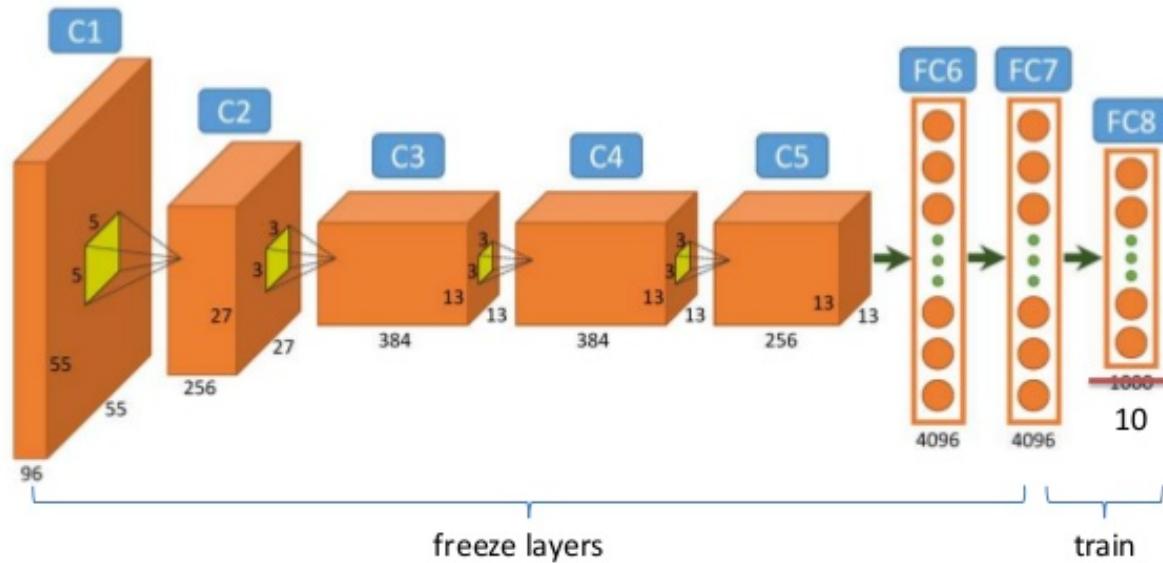
For Task 2, learn only end part



Quá trình fine tuned?

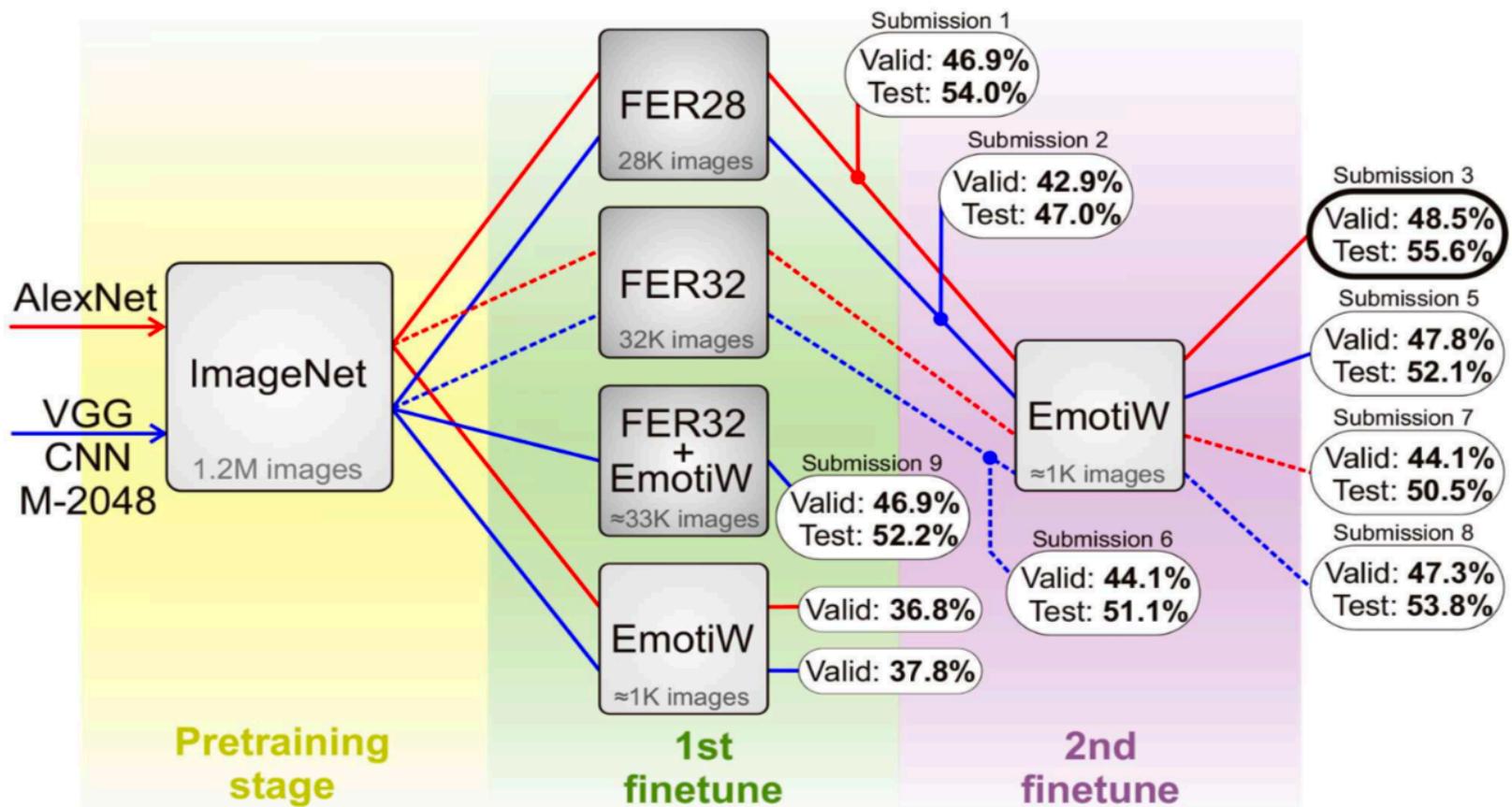
Caffe

Fine-tuning Pretrained Network



VD: bài toán nhận diện cảm xúc FER?

- Nên pre-trained on large Face Recognition dataset sau đó fine-tuned với FER dataset
 - Combine Facenet/Resnet101(dlib) + FER2013....
<https://github.com/d-acharya/CovPoolFER.git>
- Nên pre-trained nhiều giai đoạn: giai đoạn 1, như trên, giai đoạn 2, trên dataset mà mình muốn áp dụng (target FER dataset)



Sử dụng khi nào?

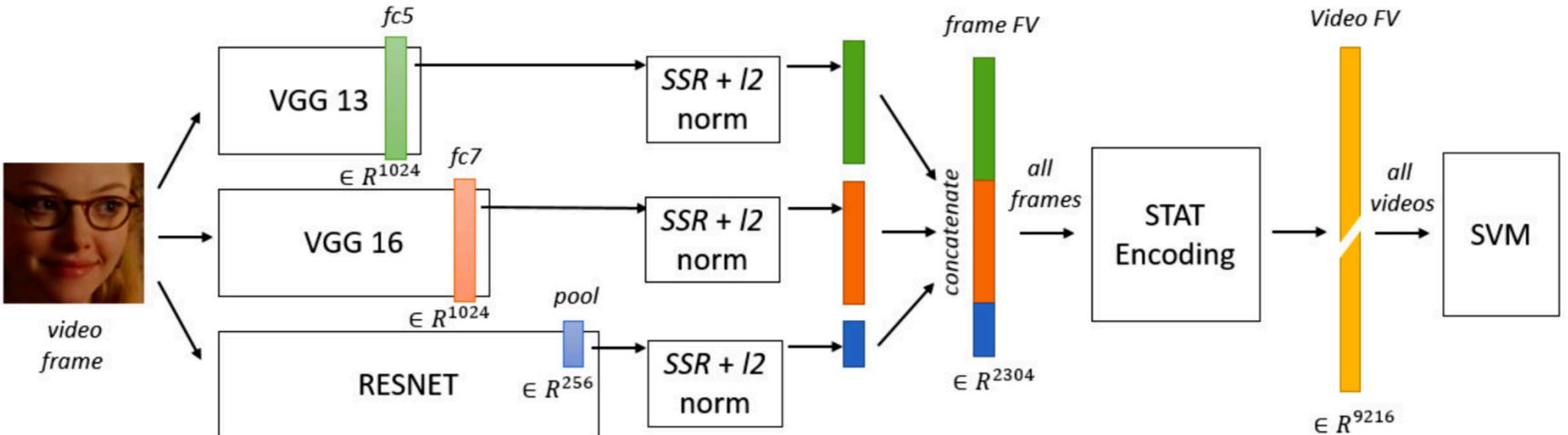
- ↗ Khi dữ liệu ta có quá ít, khó training một mạng CNN lớn
- ↗ Khi dữ liệu ta có đặc thù, nhưng có không quá khác biệt với dữ liệu đã dùng để training cho một mạng lớn có sẵn
- ↗ Khi muốn tận dụng mạng có sẵn với dữ liệu riêng của chúng ta

➤ Có cách nào khác để tận dụng các mạng đang có không?

 =>Combine networks

Tại các lớp trích xuất đặc trưng features levels

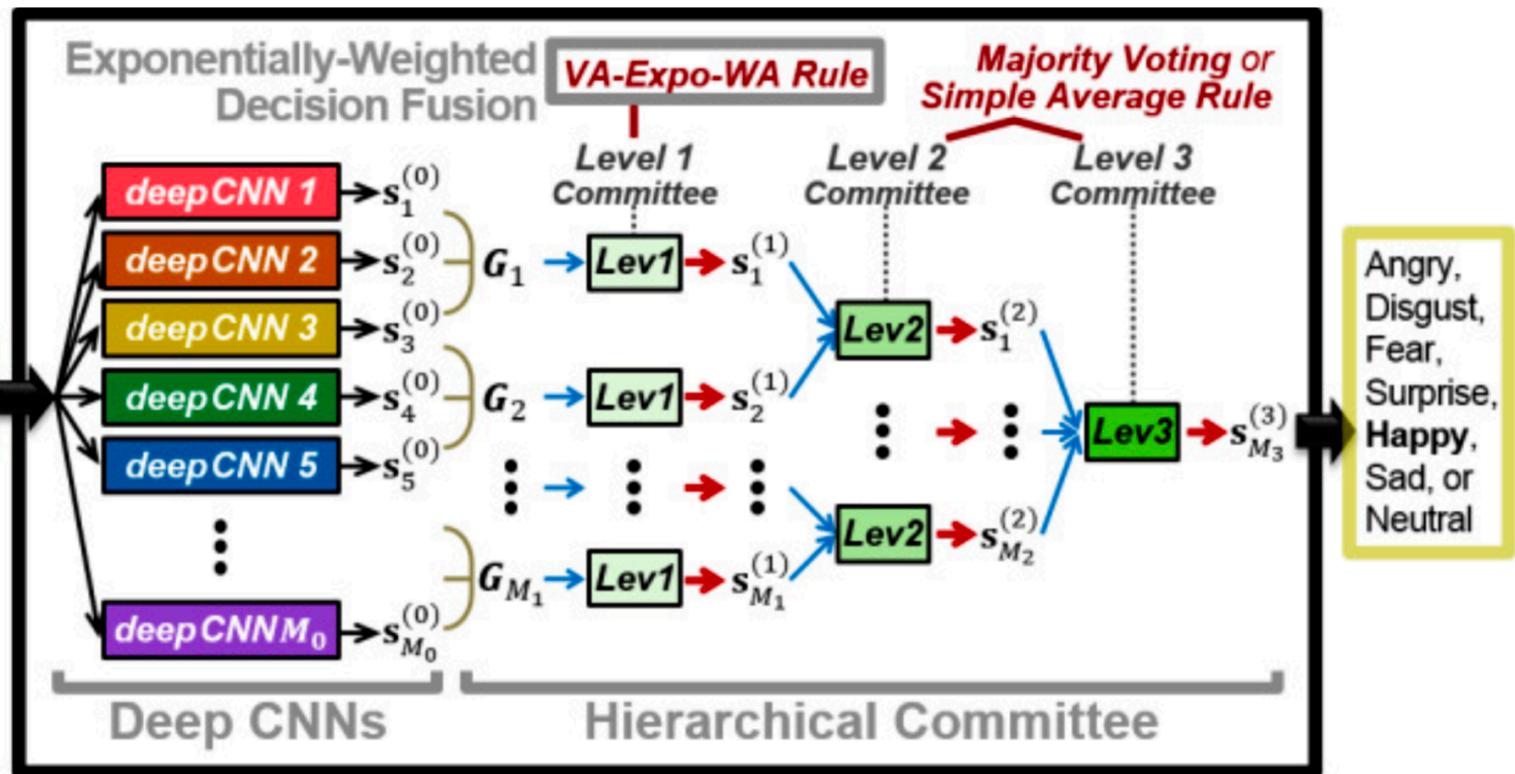
- ↗ -Mỗi mạng CNN: ngay trước khi đưa ra quyết định, sẽ là các features vectors
- ↗ -Nếu sử dụng **N** mạng khác nhau, với cùng một ảnh đầu vào; ta được **N** vectors đầu ra khác nhau
- ↗ -Có thể kết hợp **N** vectors này thành vector dài (**cascade vector**) hoặc có trọng số (**weighted-cascade vector**) và dùng nó cho phép phân loại cuối cùng



(a) Feature-level ensemble in [88]. Three different features ($fc5$ of VGG13 + $fc7$ of VGG16 + $pool$ of Resnet) after normalization are concatenated to create a single feature vector (FV) that describes the input frame.

Tại lớp quyết định decision level

- ↗ Với mỗi **một đầu vào**, đi qua 1 mạng CNN, sẽ được gán một **label xác định** hoặc **xác suất** gán label đấy
- ↗ Với một đầu vào, nhưng cho qua **N** mạng CNN khác nhau, sẽ có **N** label khác nhau (hoặc ma trận xác suất tương ứng)
- ↗ Label cuối cùng được quyết định khi kết hợp **N** mạng được xác định bằng cách:
 - ↗ +Vote trên **N** nhãn nhận được
 - ↗ +Tổng **N** xác suất nhận được
 - ↗ +Max **N** xác suất nhận được



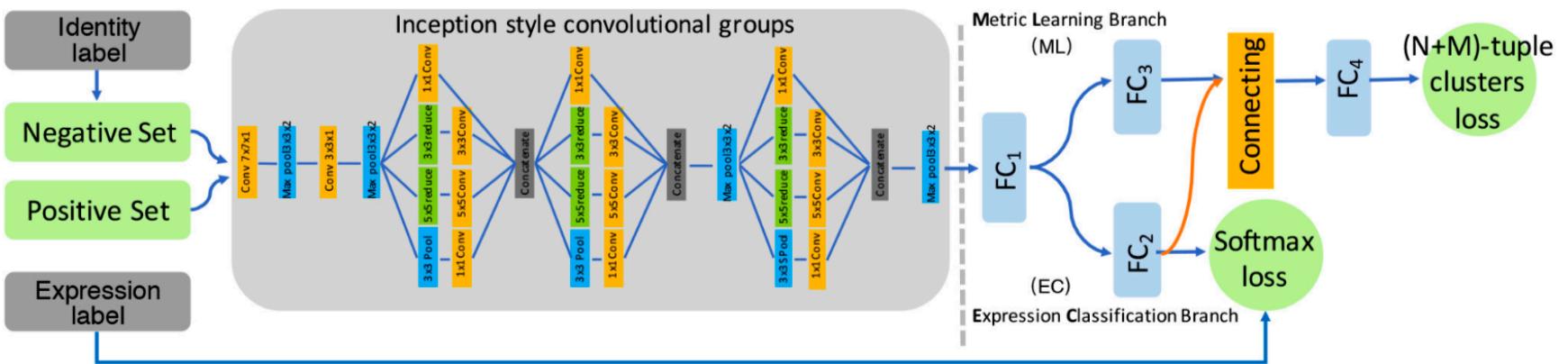
(b) Decision-level ensemble in [76]. A 3-level hierarchical committee architecture with hybrid decision-level fusions was proposed to obtain sufficient decision diversity.

Các mạng như thế nào nên kết hợp?

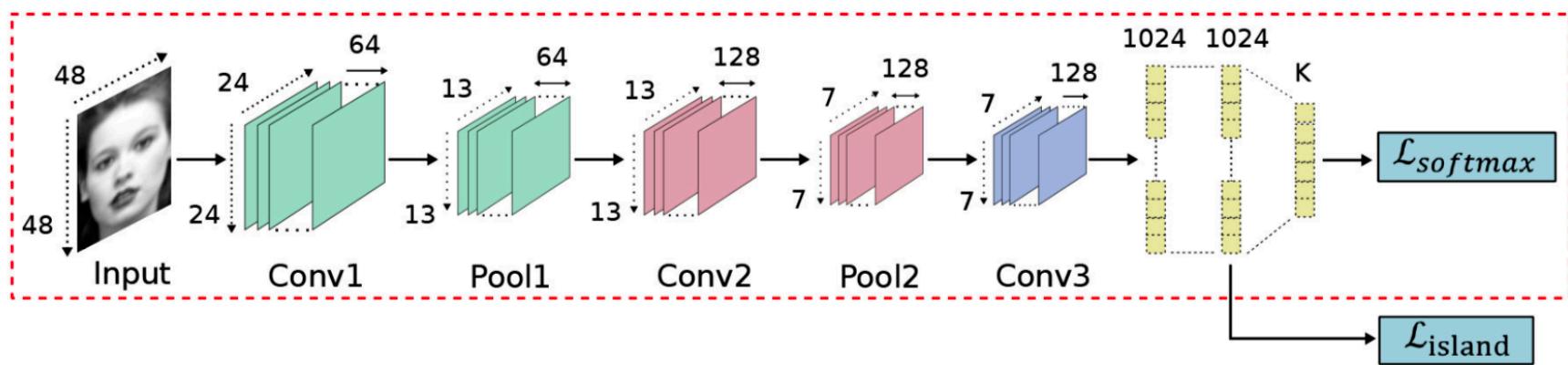
- ↗ Các mạng kết hợp phải được trained **trên các dữ liệu khác nhau**, để đảm bảo tính đa dạng của features extraction
- ↗ Các mạng kết hợp có thể **thực hiện các nhiệm vụ khác nhau**

Lựa chọn hàm Loss function

- ↗ Nhiều hàm loss đã được thử nghiệm: cross-entropy, softmax, covariance
- ↗ Đề xuất sử dụng nhiều hàm loss khác nhau, đồng thời để tối ưu theo các cách khác nhau



(c) (N+M)-tuple clusters loss layer in [77]. During training, the identity-aware hard-negative mining and online positive mining schemes are used to decrease the inter-identity variation in the same expression class.



(b) Island loss layer in [140]. The island loss calculated at the feature extraction layer and the softmax loss calculated at the decision layer are combined to supervise the CNN training.

