Deep Learning for Image Analysis

Mehyar MLAWEH

🏫 Master BDIA - Dauphine Tunis
PhD student - Université PSL

🧑‍💻 Generative AI Engineer - Wealthy Technology

Resources: https://mehyarmlaweh.github.io/mehyar-mlaweh/teaching

Module Overview

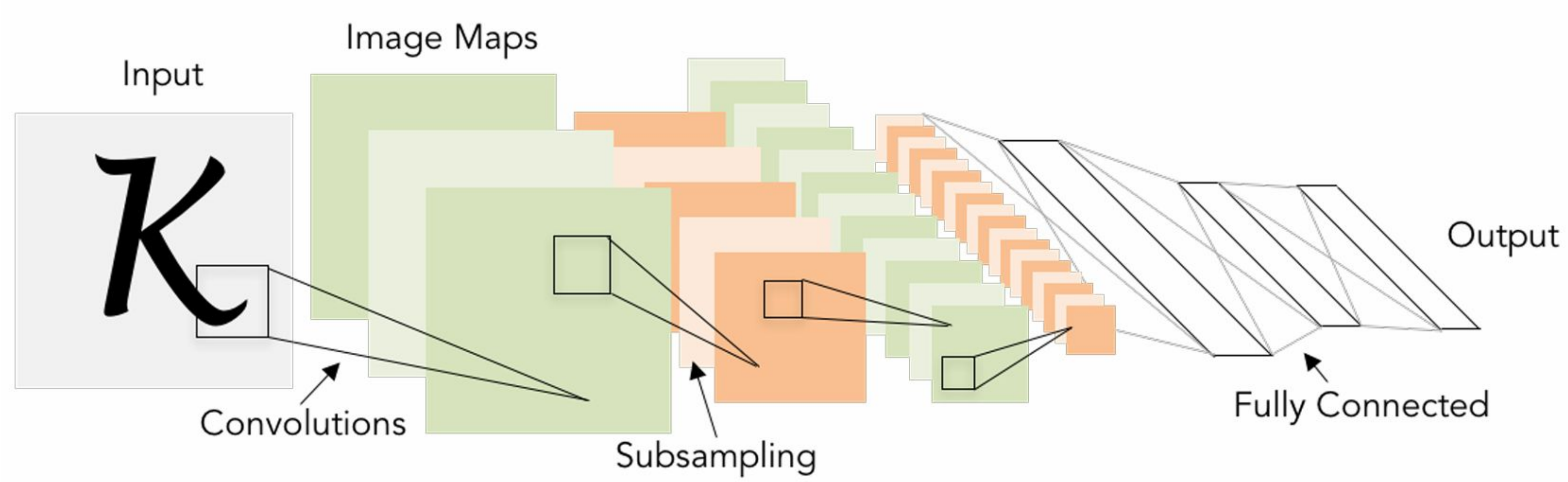
Dauphine | PSL
UNIVERSITÉ PARIS

1. Computer Vision Foundations
2. CNNs
3. Transfer Learning
4. Vision Transformers
5. Object Detection
6. Segment Anything Model (SAM)
7. Final Project

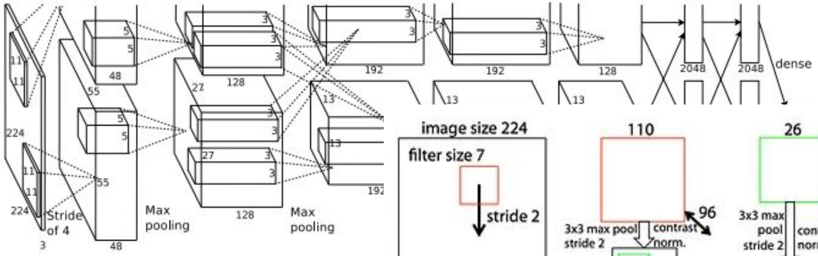
Lecture 3 :
Transfer Learning

Last Time

Problem : What is the right way to combine all these components ?



Last Time


224
Stride of 4
Max pooling
Max pooling
192
192
128
2048
2048
dense

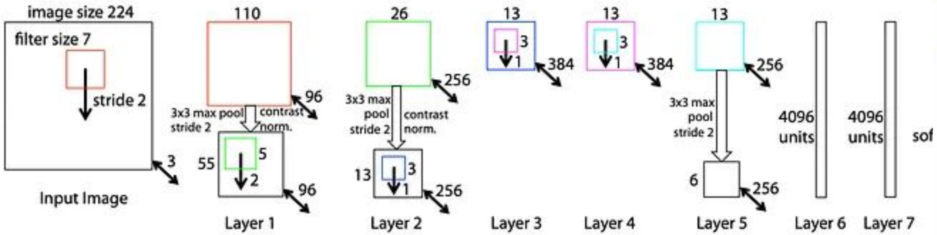
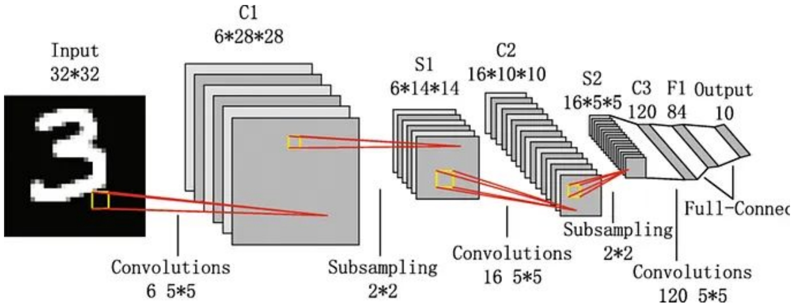
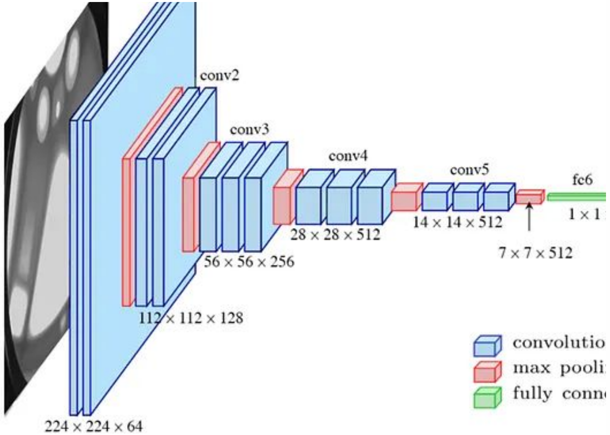
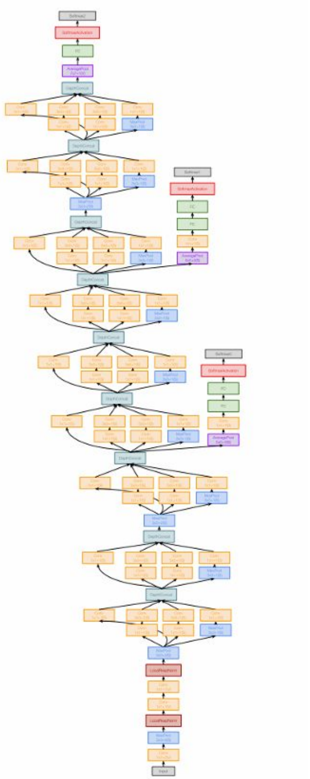


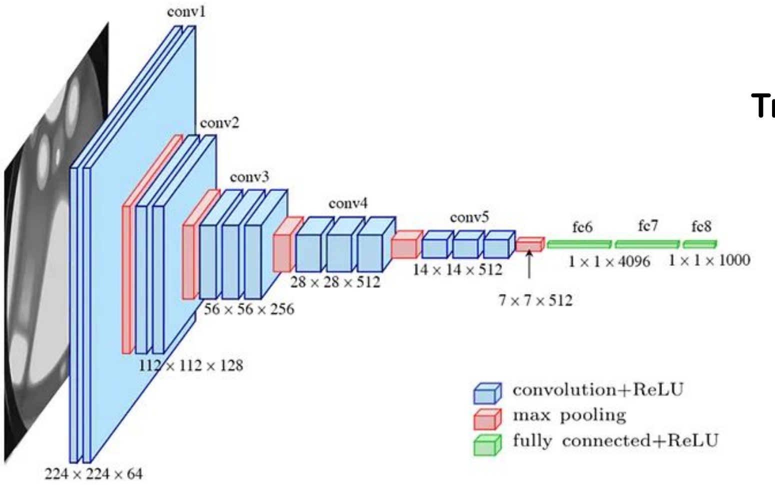
image size 224
filter size 7
stride 2
Input Image
3x3 max pool stride 2
contrast norm.
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
4096 units
4096 units
Layer 6
Layer 7


Input
32*32
C1
6*28*28
S1
6*14*14
C2
16*10*10
S2
16*5*5
C3
120
F1
84
Output
10
Convolutions
6 5*5
Subsampling
2*2
Convolutions
16 5*5
Subsampling
2*2
Convolutions
120 5*5


conv2
conv3
conv4
conv5
fc6
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1



Question



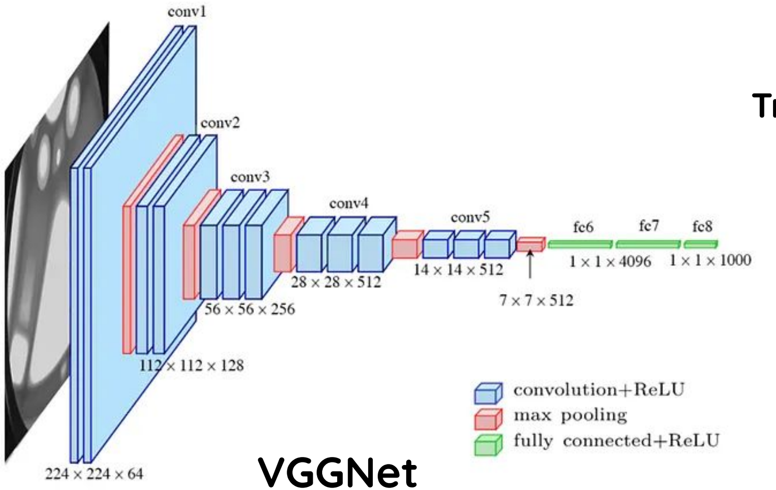
VGGNet

Trained on

ImageNet 1000 object classes

(e.g., dog, airplane, keyboard)

Question



VGGNet

Trained on

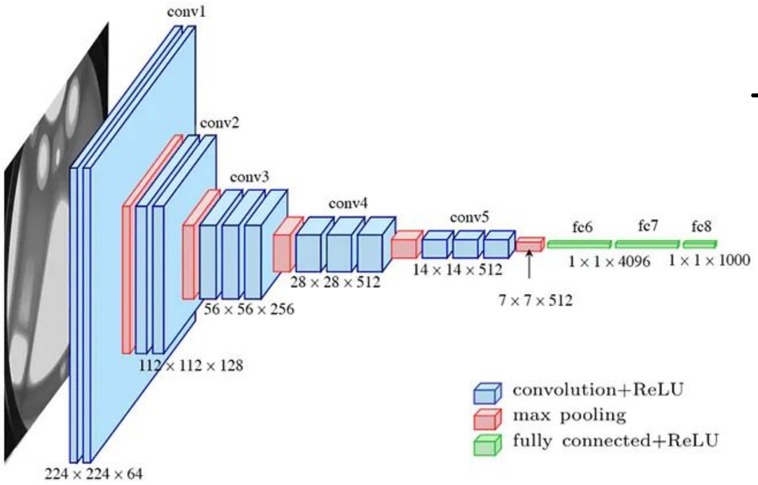
ImageNet 1000 object classes

(e.g., dog, airplane, keyboard)

A CNN trained on ImageNet as already learned:

- Visual primitives (edges, textures)
- Generic representations of the world

Question

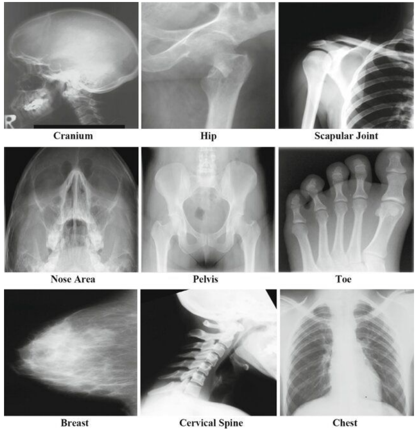


VGGNet

Trained on

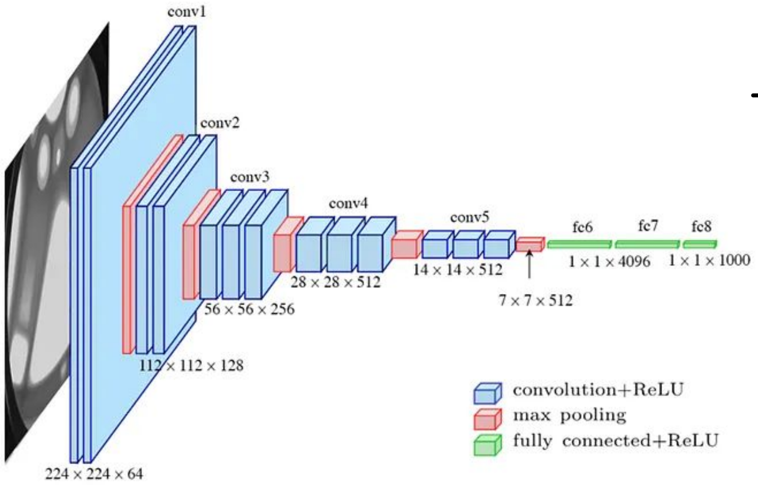
ImageNet 1000 object classes

(e.g., dog, airplane, keyboard)



X-ray images of body parts

Question

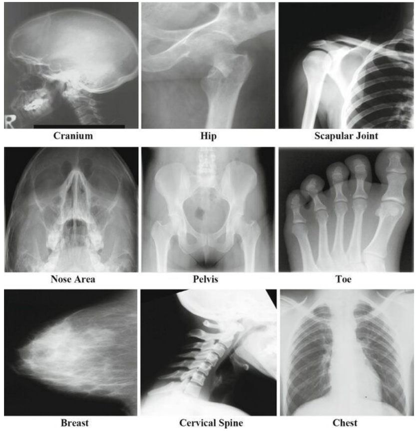


VGGNet

Trained on

ImageNet 1000 object classes

(e.g., dog, airplane, keyboard)



X-ray images of body parts

Do we really need to train a new model from scratch?

Why would we reuse?

What could we reuse?

Human Analogy

You want to learn how to drive a truck

Human Analogy

Transfer knowledge

Learning to drive a car first

You want to learn how to drive a truck

Reality Check

1.2M	2–3 wks	$10K+
training images	training time	
for ImageNet	4× NVIDIA P100	

Reality Check

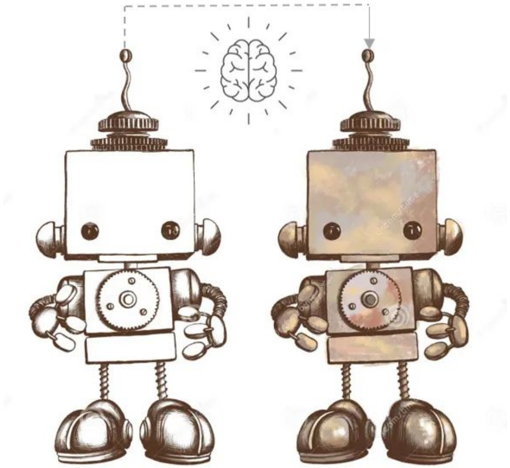
1.2M	2–3 wks	$10K+
training images for ImageNet	training time 4× NVIDIA P100	

Training from scratch is almost never the right choice for most projects.

Transfer Learning - Core Idea

"What has been learned in one setting is exploited to improve generalization in another setting."

— Goodfellow, Bengio & Courville, Deep Learning (2016)



Transfer Learning - Core Idea

Without Transfer Learning

- Collect thousands of labeled images
- Train all layers from random weights
- Weeks of GPU time
- High overfitting risk on small datasets

With Transfer Learning

- Start from a pretrained model
- Adapt only the task-specific parts
- Hours to a few days of training

Why Does It Work?

Network Depth	What is Learned?	Transferability
Early Layers	Edges, colours	✅ Always transfer
Middle Layers	Textures, patterns, shapes	✅ Usually transfer
Deep Layers	Object parts, semantic regions	⚠️ Partly task-specific
Final Layers	Class scores (ImageNet : 1000 categories)	❌ Replace this layer

Transfer Learning Strategies

There are 3 main ways to use a pretrained model:

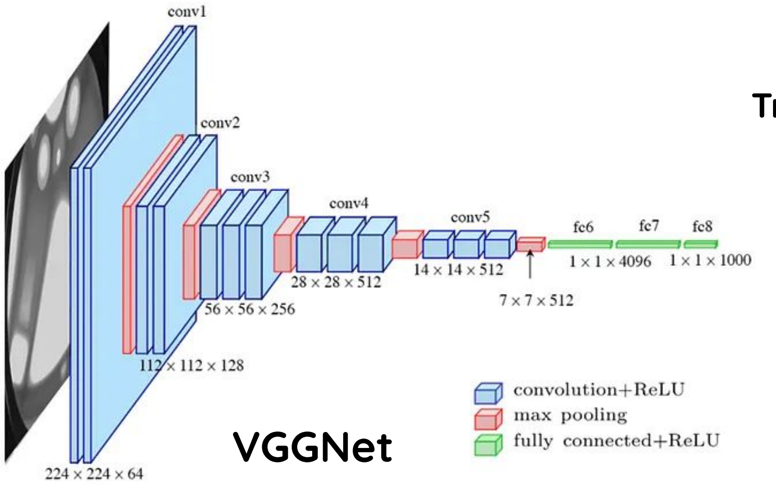
1. Feature Extraction

Transfer Learning Strategies

There are 3 main ways to use a pretrained model:

1. Feature Extraction
2. Fine-Tuning some layers
3. Full Training (rare baseline)

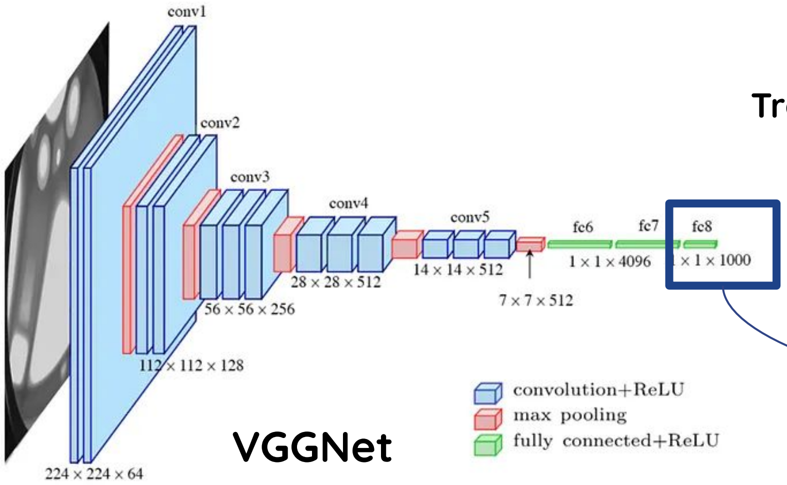
Feature Extraction



VGGNet

Trained on

Feature Extraction

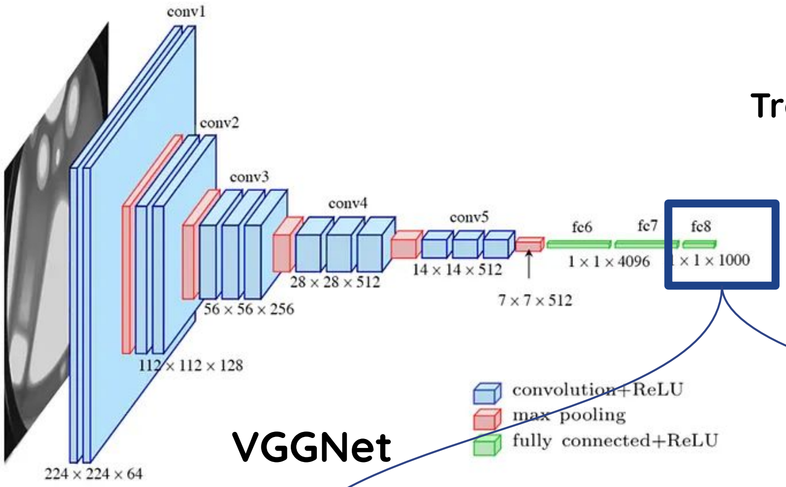


VGGNet

Trained on

Output Layer : 1000 Objects

Feature Extraction

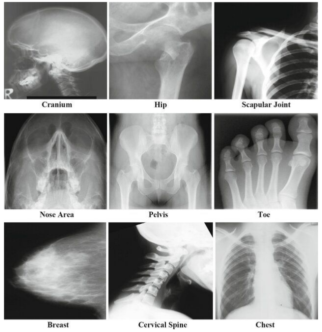


VGGNet

Trained on

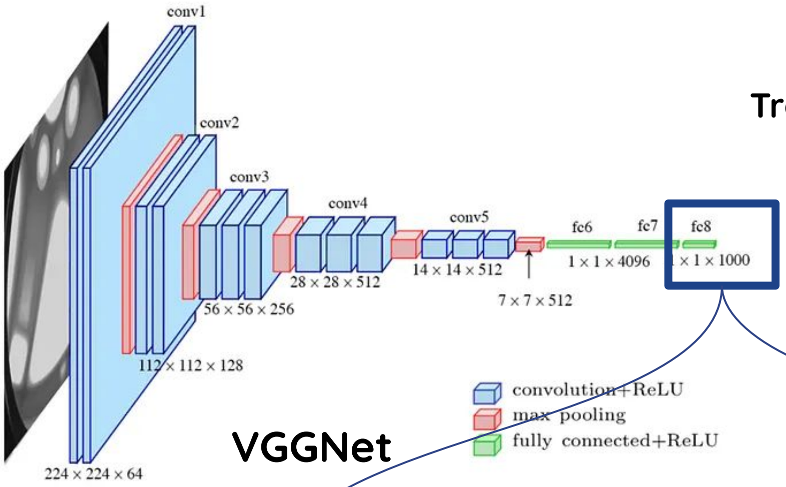
Output Layer : 1000 Objects

Replace with new layer to classify the body parts images (new task)



X-ray images of body parts (100 class)

Feature Extraction



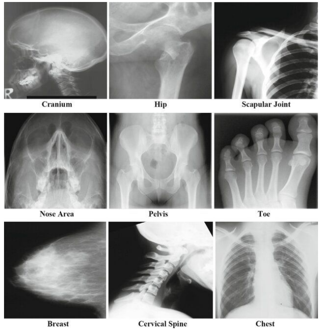
VGGNet

Trained on

The new output layer Layer : 100 Objects

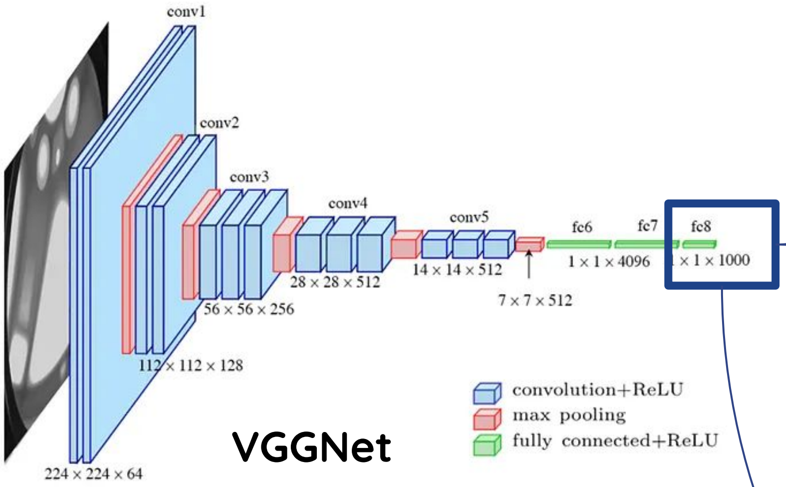
Replace with new layer to classify

the body parts images (new task)



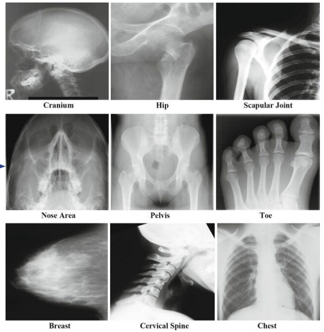
X-ray images of body parts (100 class)

Feature Extraction



VGGNet

Trained on



New task : X-ray images of body parts (100 class)

The new output layer Layer : 100 Objects

What do we **train** in this model?

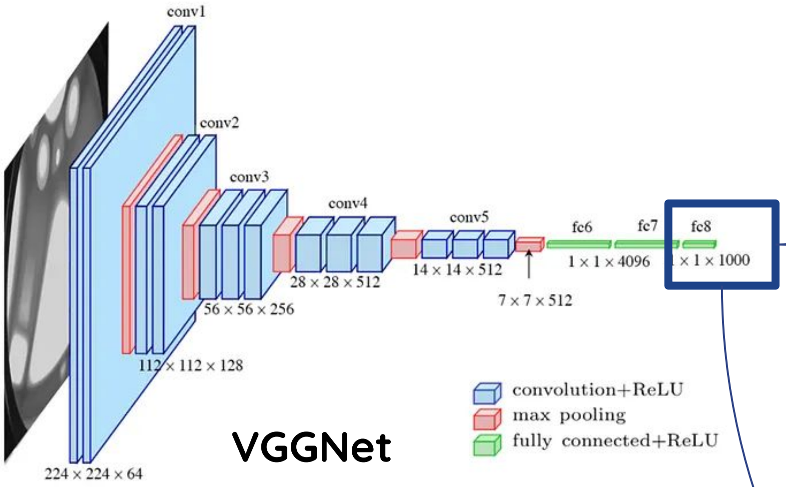
A. Only fc8

B. fc6, fc7, fc8

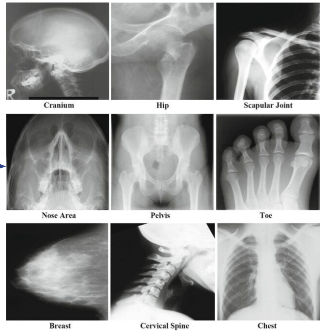
C. The entire network

Feature Extraction

Dauphine | PSL
UNIVERSITÉ PARIS



Trained on 



New task : X-ray images of body parts (100 class)

What do we **train** in this model?

A. Only fc8

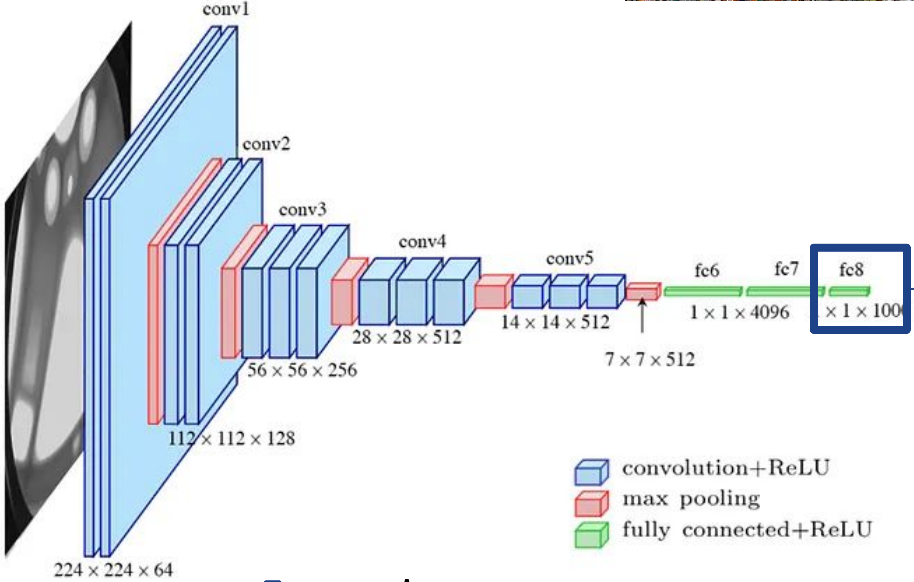
B. fc6, fc7, fc8 ❄️

C. The entire network ❄️

The new output layer Layer : 100 Objects

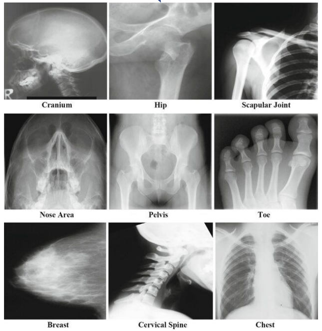
Feature Extraction

VggNet already trained on



We train the vggNet on

the new dataset



Frozen layers :

- Conv 1 -> 5
- Fc 6,7

New task : X-ray images of body parts (100 class)

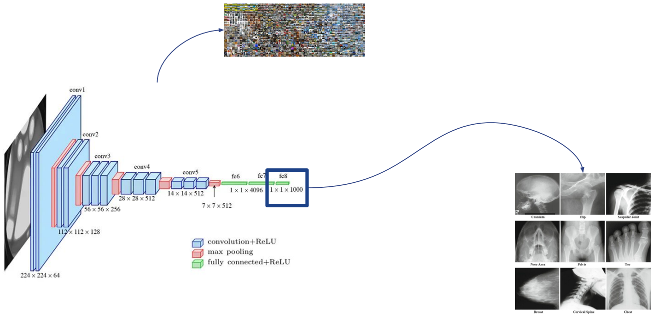
Feature Extraction

In few words :

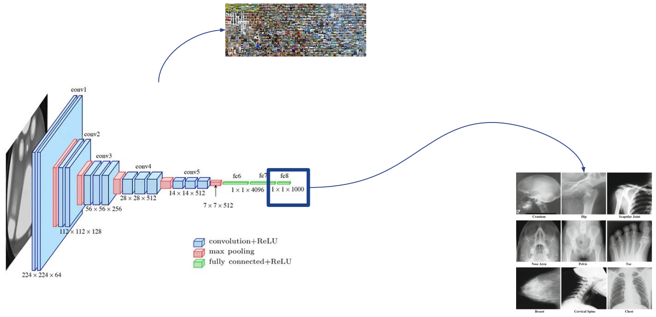
1. Take a pretrained model (e.g., ResNet, VGG)
2. Remove the final classification layer
3. Add a new layer for your task
4. Freeze the rest of the network

→ Only the last layer learns

→ Everything else stays unchanged



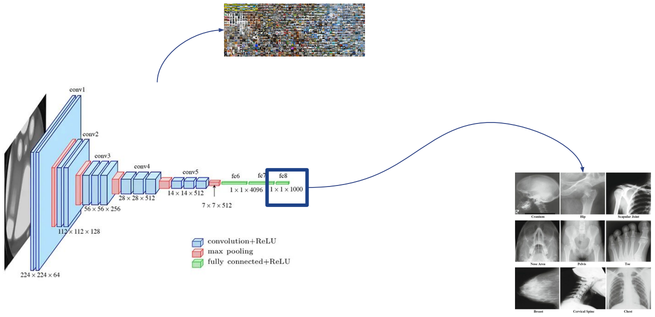
Feature Extraction



When should we use Feature Extraction :

- Small dataset
- The pretrained model is very large
- The domain is similar to the pre-training domain (e.g. ImageNet)

Feature Extraction

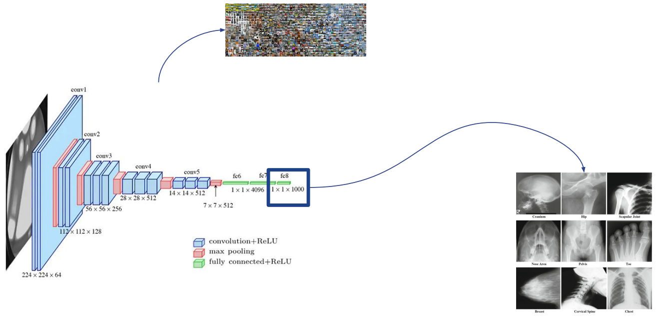


When should we use Feature Extraction :

- Small dataset
- The pretrained model is very large
- The domain is similar to the pre-training domain (e.g. ImageNet)

Limitation:

- Domain Mismatch : Medical X-rays → features learned from natural photos don't generalize well at lower layers.



When should we use Feature Extraction :

- Small dataset
- The pretrained model is very large
- The domain is similar to the pre-training domain (e.g. ImageNet)

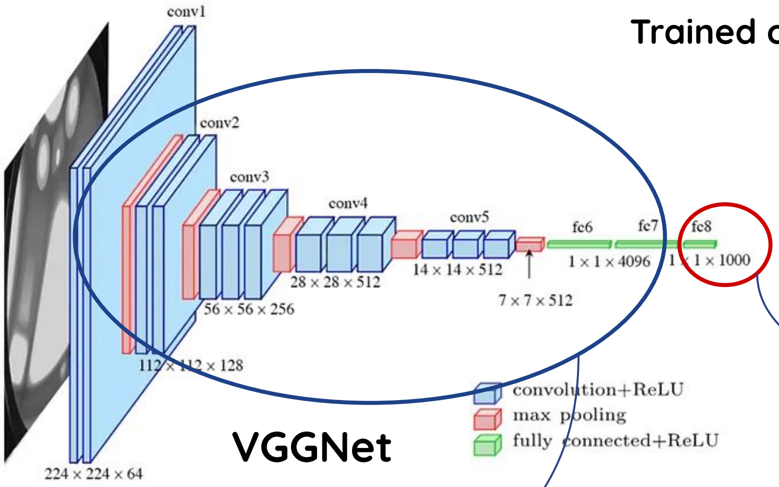
Limitation:

- Domain Mismatch : Medical X-rays → features learned from natural photos don't generalize well at lower layers.

Can we slightly adapt the model instead of freezing everything ?

Fine-Tuning

Trained on



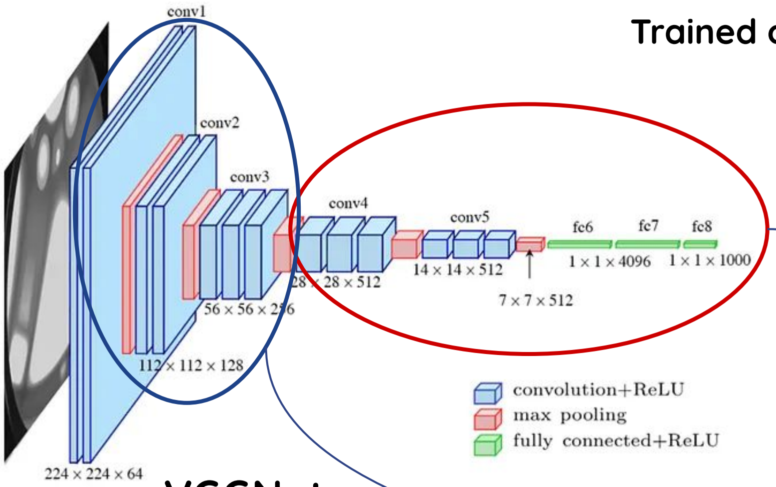
VGGNet

Freeze

Train

Fine-Tuning

Trained on



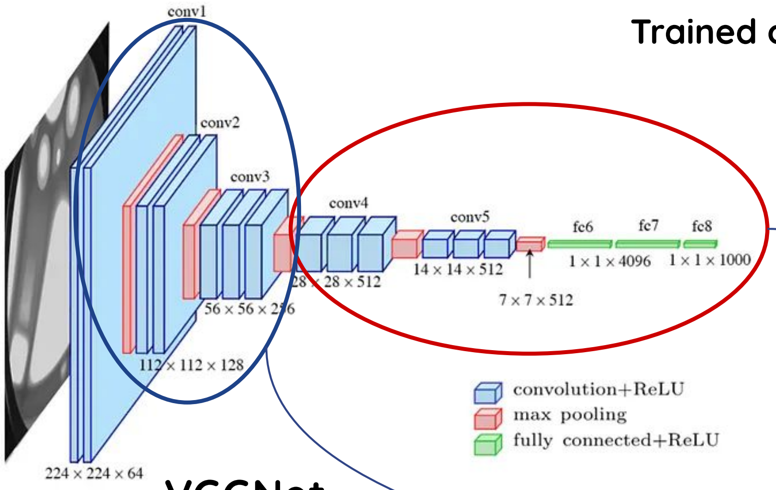
VGGNet

Train

Freeze

Fine-Tuning

Trained on



VGGNet

Train

Freeze

Early layers = generic → keep frozen
Later layers = task-specific → fine-tune

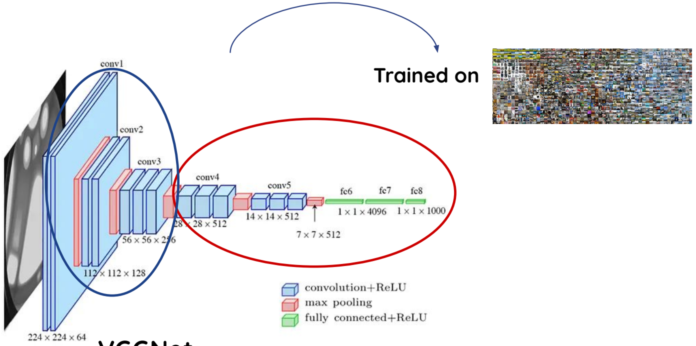
Fine-Tuning

How Much Should We Fine-Tune?

We don’t always fine-tune the same number of layers

It depends on:

- Dataset size
- Similarity to ImageNet



VGGNet

Fine-Tuning

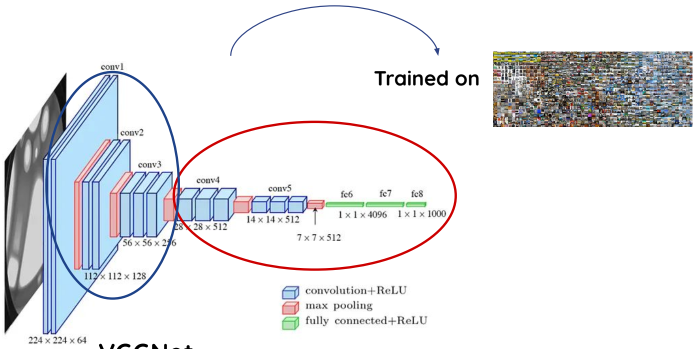
How Much Should We Fine-Tune?

We don't always fine-tune the same number of layers

It depends on:

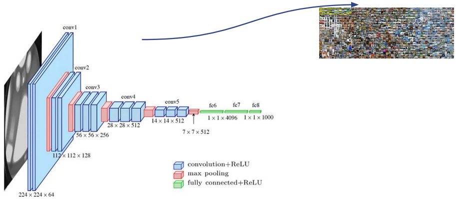
- Dataset size
- Similarity to ImageNet

The more different the task → the more layers we fine-tune



VGGNet

Training Strategy



Phase 1

❄️ Early Layers

❄️ Mid Layers

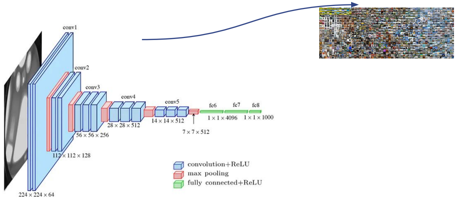
❄️ Late Layers

Classifier head
Lr = 1e-3

Freeze backbone

Train head only

Fine-Tuning



VGGNet

Training Strategy

Phase 1

- ❄️ Early Layers
- ❄️ Mid Layers
- ❄️ Late Layers
- Classifier head
  Lr = 1e-3

Freeze backbone

Train head only

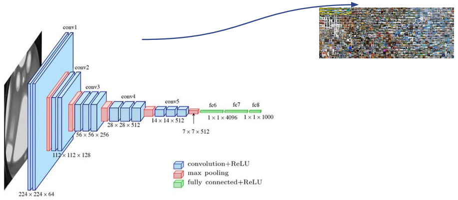
Phase 2

- ❄️ Early Layers
- ❄️ Mid Layers
- Late Layers
  Lr = 1e-4
- Classifier head
  Lr = 1e-3

Unfreeze last block

Fine-Tuning

Training Strategy



VGGNet

Phase 1

- Early Layers
- Mid Layers
- Late Layers
- Classifier head
  Lr = 1e-3

Freeze backbone

Train head only

Phase 2

- Early Layers
- Mid Layers
- Late Layers
  Lr = 1e-4
- Classifier head
  Lr = 1e-3

Unfreeze last block

Phase 3

- Early Layers
  Lr = 1e-5
- Mid Layers
  Lr = 1e-4
- Late Layers
  Lr = 1e-4
- Classifier head
  Lr = 1e-3

Full backbone unfrozen

Optional — if data allows

Fine-Tuning

Training Strategy

Phase 1

- ❄️ Early Layers
- ❄️ Mid Layers
- ❄️ Late Layers
- Classifier head
  Lr = 1e-3

Freeze backbone
Train head only

Phase 2

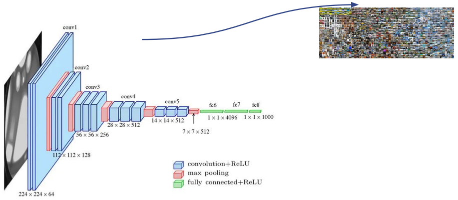
- ❄️ Early Layers
- ❄️ Mid Layers
- Late Layers
  Lr = 1e-4
- Classifier head
  Lr = 1e-3

Unfreeze last block

Phase 3

- Early Layers
  Lr = 1e-5
- Mid Layers
  Lr = 1e-4
- Late Layers
  Lr = 1e-4
- Classifier head
  Lr = 1e-3

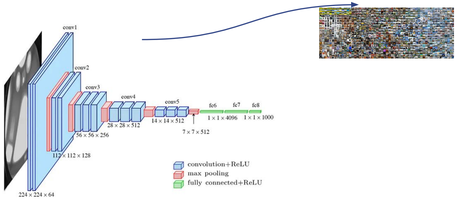
Full backbone unfrozen
Optional — if data allows



VGGNet

LR decreases as you go deeper, this is what prevents **catastrophic forgetting** while still allowing adaptation.

Fine-Tuning vs Feature extraction



	Similar to ImageNet	Very Different
Large Dataset	**Fine-tune entire network** Standard approach. Use a moderate LR. Low overfitting risk due to data volume.	**Fine-tune entire network** More epochs, heavier augmentation. Early layers will also adapt significantly.
Small Dataset	**Feature extraction** Features transfer well. Keep the head small, add dropout. Avoid overfitting.	**Hardest case** Start with feature extraction. Unfreeze last block only, with heavy regularisation.

Conclusion

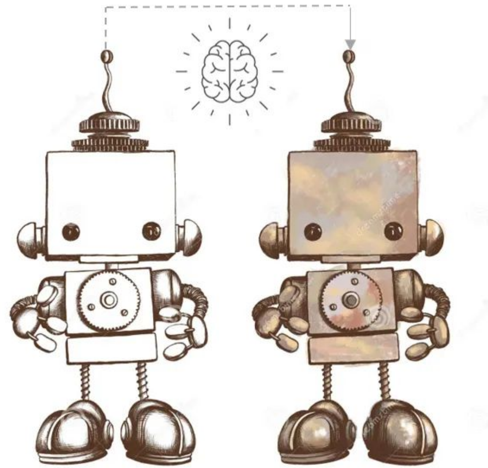
Transfer learning = reuse + adapt knowledge

Two approaches:

- Feature Extraction (freeze most layers)
- Fine-Tuning (adapt some layers)

Strategy depends on:

- Data size
- Task similarity



Next Time ?

Vision Transformers

References I

Books

• Computer Vision: Models, Learning, and Inference (2012). https://amzn.to/2rxrdOF
• Programming Computer Vision with Python (2012). https://amzn.to/2QKTaAL
• Multiple View Geometry in Computer Vision (2004). https://amzn.to/2LfHLE8
• Computer Vision: Algorithms and Applications (2010). https://amzn.to/2LcIt4J
• Computer Vision: A Modern Approach (2002). https://amzn.to/2rv7AqJ
• Deep Learning for Computer Vision : Image Classification, Object Detection and Face Recognition in Python(2019). https://machinelearningmastery.com/deep-learning-for-computer-vision/
• Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow — Aurélien Géron (2019) https://amzn.to/3uZbN6Z

Articles & Online Resources

• Krizhevsky, Sutskever, Hinton (2012) — ImageNet Classification with Deep CNNs (AlexNet — deep learning breakthrough)
• Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis, 2017
• "Computer vision," Wikipedia. https://en.wikipedia.org/wiki/Computer_vision
• "Machine vision," Wikipedia. https://en.wikipedia.org/wiki/Machine_vision
• "Digital image processing," Wikipedia. https://en.wikipedia.org/wiki/Digital_image_processing
• "Training Convolutional Neural Networks: What Is Machine Learning" Analog Devices. https://www.analog.com/en/resources/analog-dialogue/articles/training-convolutional-neural-networks-what-is-machine-learning-part-2.html

References II

- “Computer vision,” Wikipedia. https://en.wikipedia.org/wiki/Computer_vision
- https://medium.com/data-reply-it-datatech
- https://huggingface.co/blog/RDTvlokip/quand-l-ia-apprend-de-l-experience-comme-toi
- Zhuang et al., “A Comprehensive Survey on Transfer Learning,” IEEE, 2020.