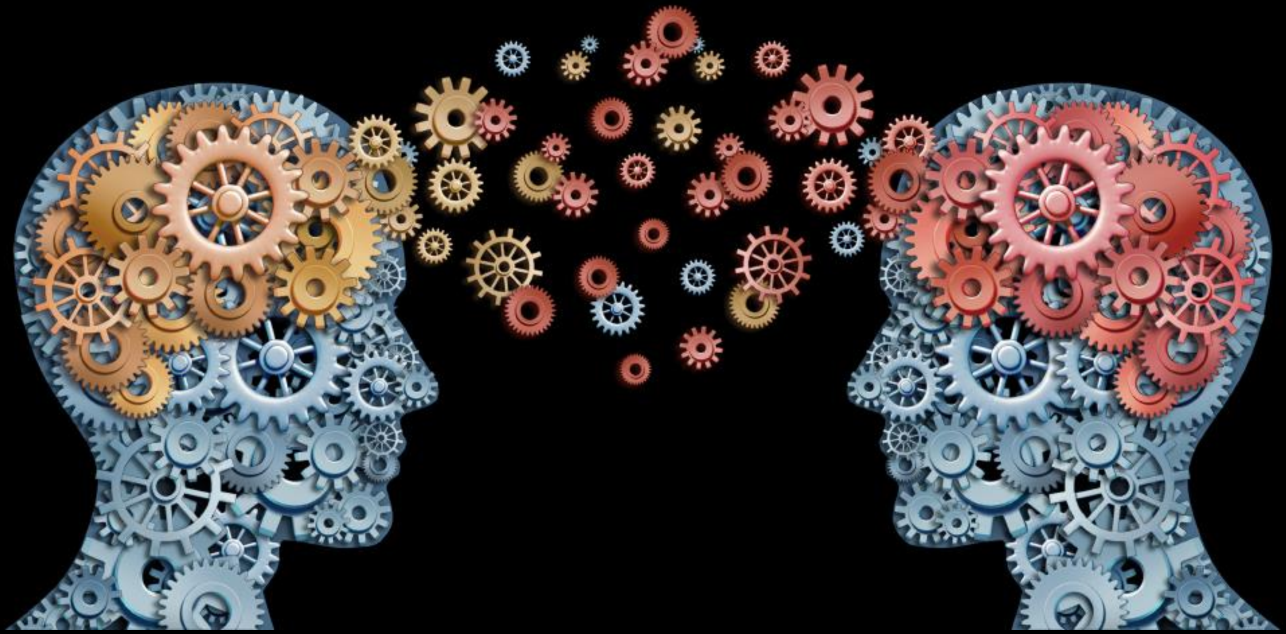
IWANN 2019

Transfer learning tutorial



Dario Garcia Gasulla dario.garcia@bsc.es
Armand Vilalta Arias armand.vilalta@bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Transfer Learning

Your first driving lesson

Your first driving lesson

Imagine learning to drive a car without knowing absolutely nothing about anything

Your first driving lesson

Randomly initialized Deep Neural Network

Your first driving lesson

Any previous learning can be useful

Knowing how to cook is better than knowing nothing at all

Your first driving lesson

Any previous learning can be useful

Knowing how to cook is better than knowing nothing at all

We naturally reuse what we previously learnt to be able to solve a new task.

How transfer learning emerged

Image classification

- **1998 LeNet-5**

  Gradient-based learning applied to document recognition.

  Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner

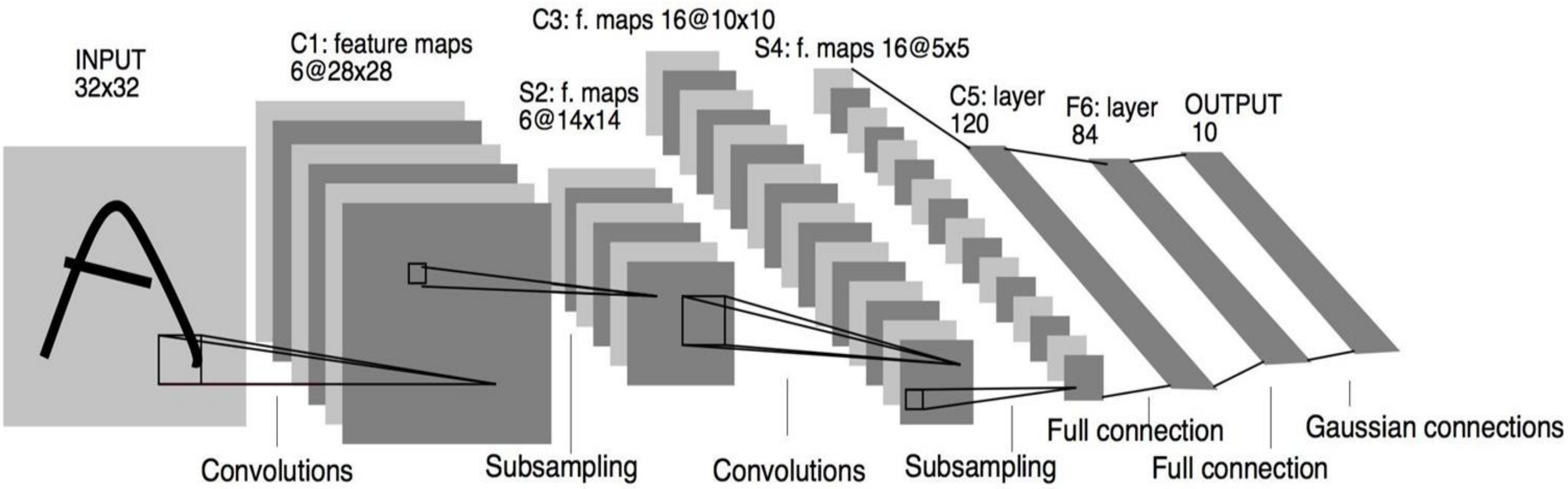
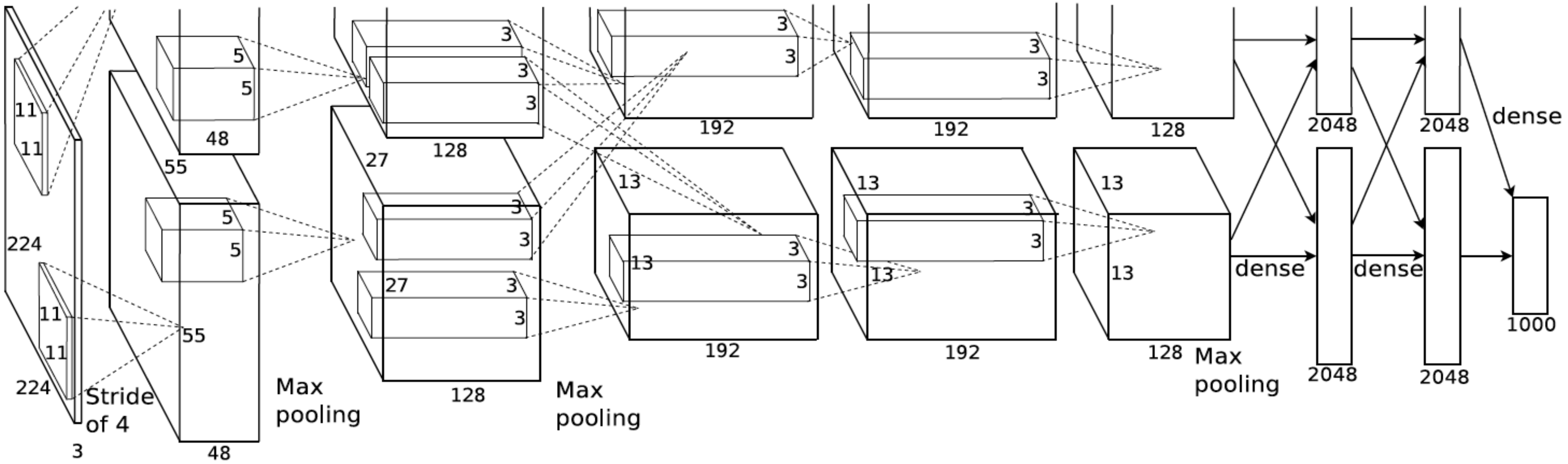


Image classification

- **2012 AlexNet**

  ImageNet Classification with Deep Convolutional Neural Networks

  Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton



1998 LeNet-5

2012 AlexNet

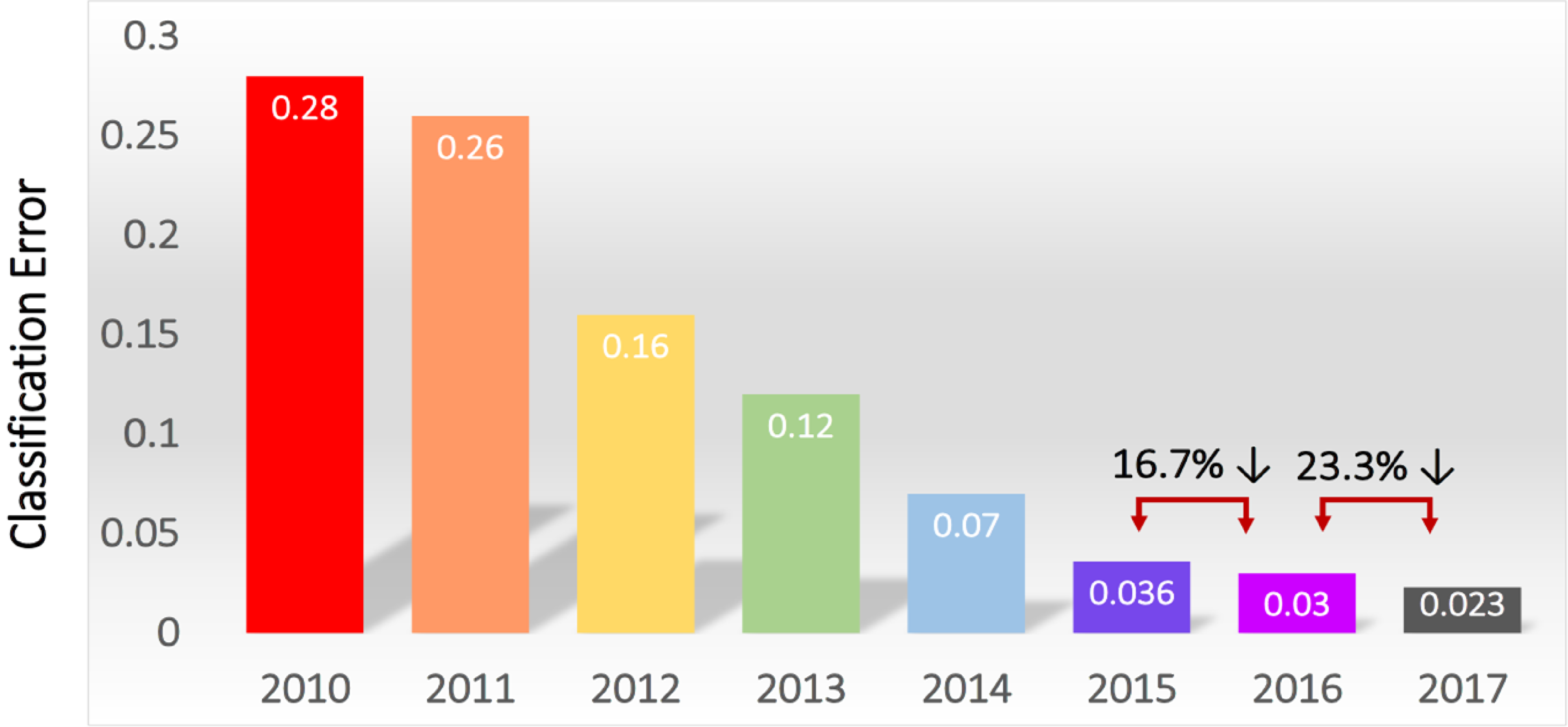
2014 VGG19

2014 GoogLeNet

2015 Inception-V3

2015 ResNet-56

ImageNet classification results



From image-net.org

At what price?

- Data available
  - 1,000 images per class
- Computational cost
  - Specific hardware
  - Energy cost
- Human effort
  - Highly skilled professionals
  - Architecture design
  - Hyper-parameter fine tuning

At what price?

- Data available
  - 1,000 images per class
- Computational cost
  - Specific hardware
  - Energy cost
- Human effort
  - Highly skilled professionals
  - Architecture design
  - Hyper-parameter fine tuning

We can’t do that for every single problem!!

At what price?

- Data available
  - 1,000 images per class
- Computational cost
  - Specific hardware
  - Energy cost
- Human effort
  - Highly skilled professionals
  - Architecture design
  - Hyper-parameter fine tuning

We don’t want to **do that for every single problem!!**

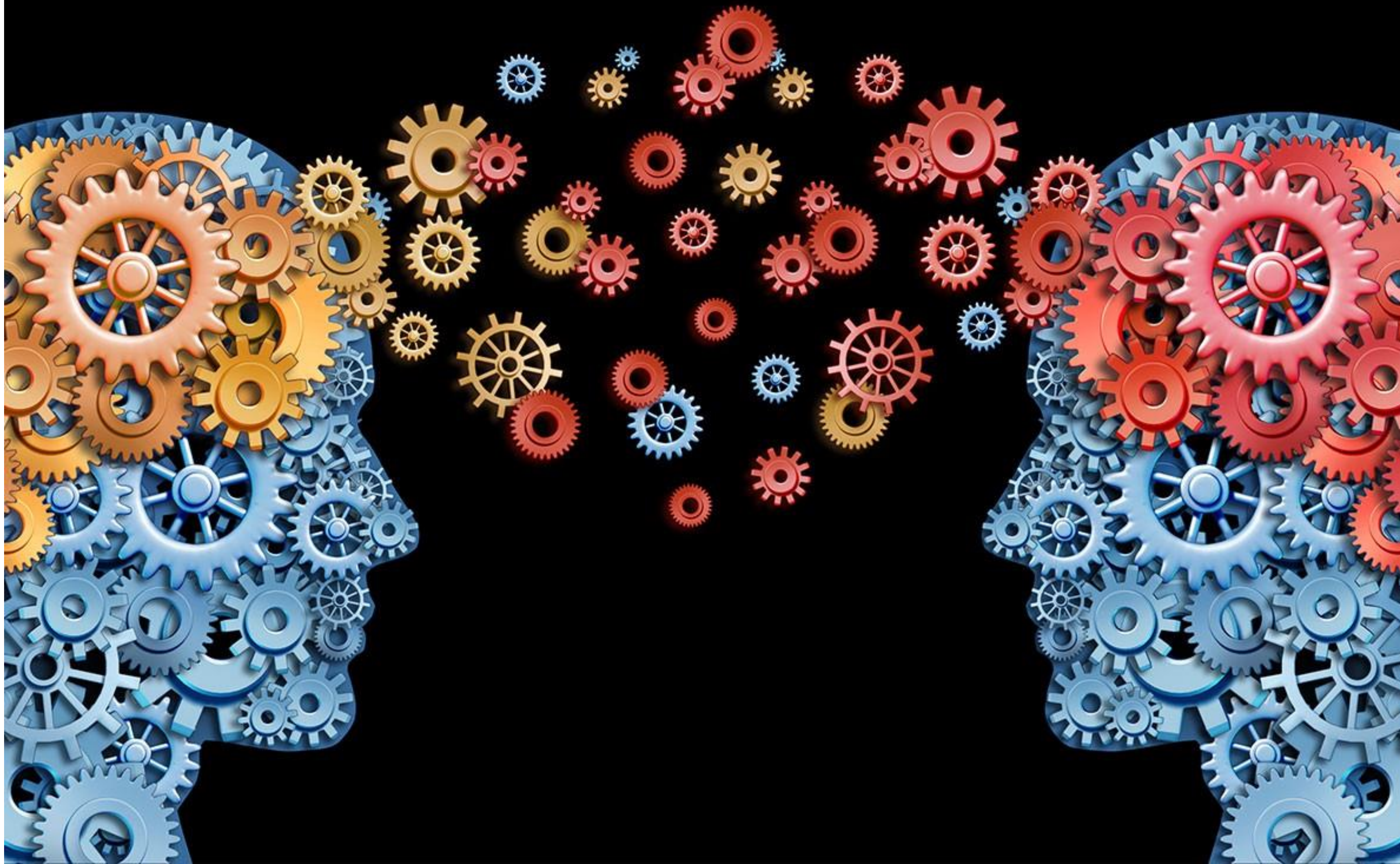
At what price?

- Data available
  - 1,000 images per class
- Computational cost
  - Specific hardware
  - Energy cost
- Human effort
  - Highly skilled professionals
  - Architecture design
  - Hyper-parameter fine tuning

We don’t want to **do that for every single problem!!**

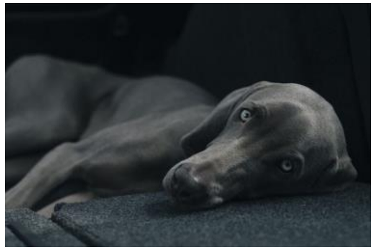
→ Transfer Learning to the rescue

What is transfer learning?



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

What is learning about?





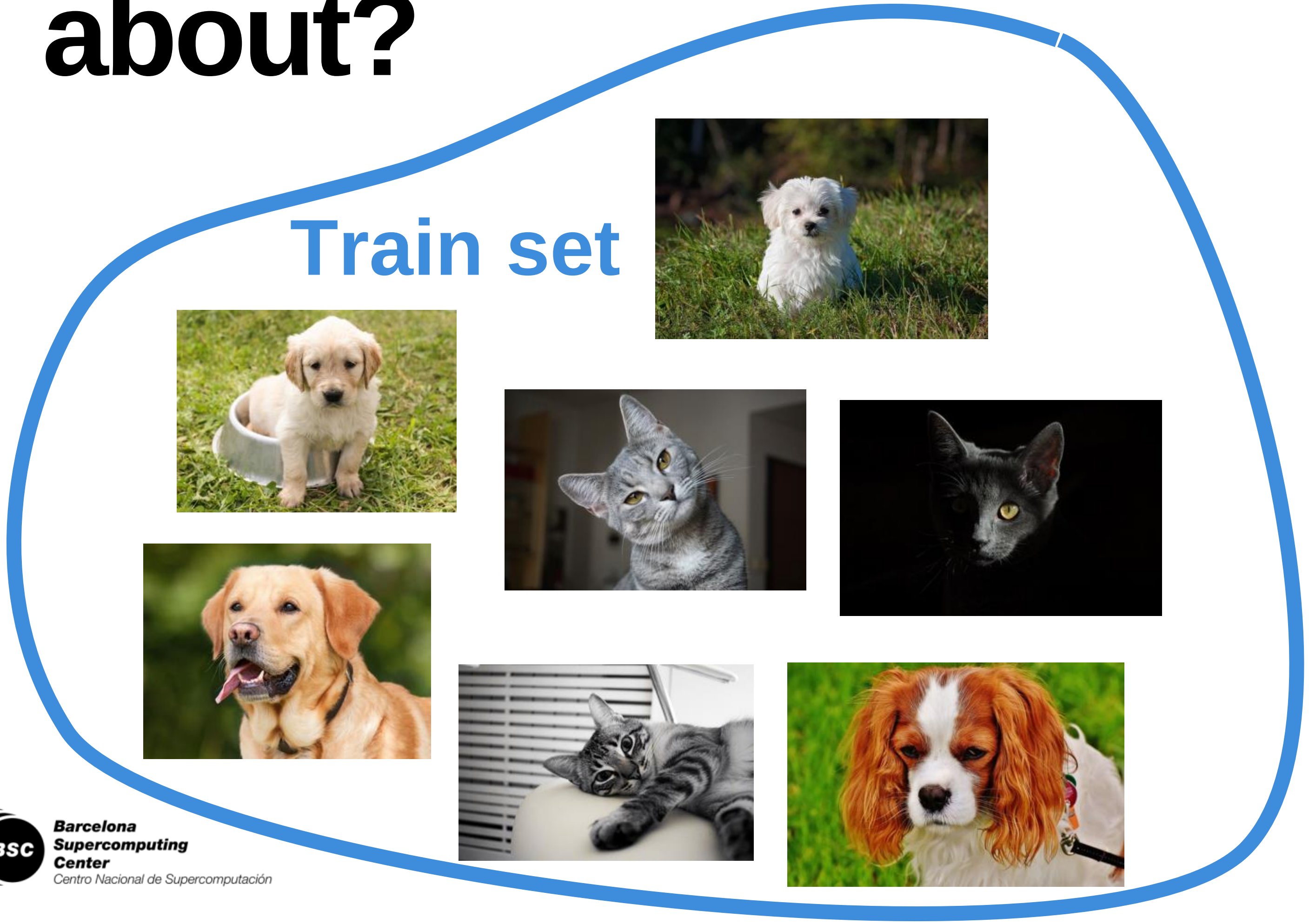


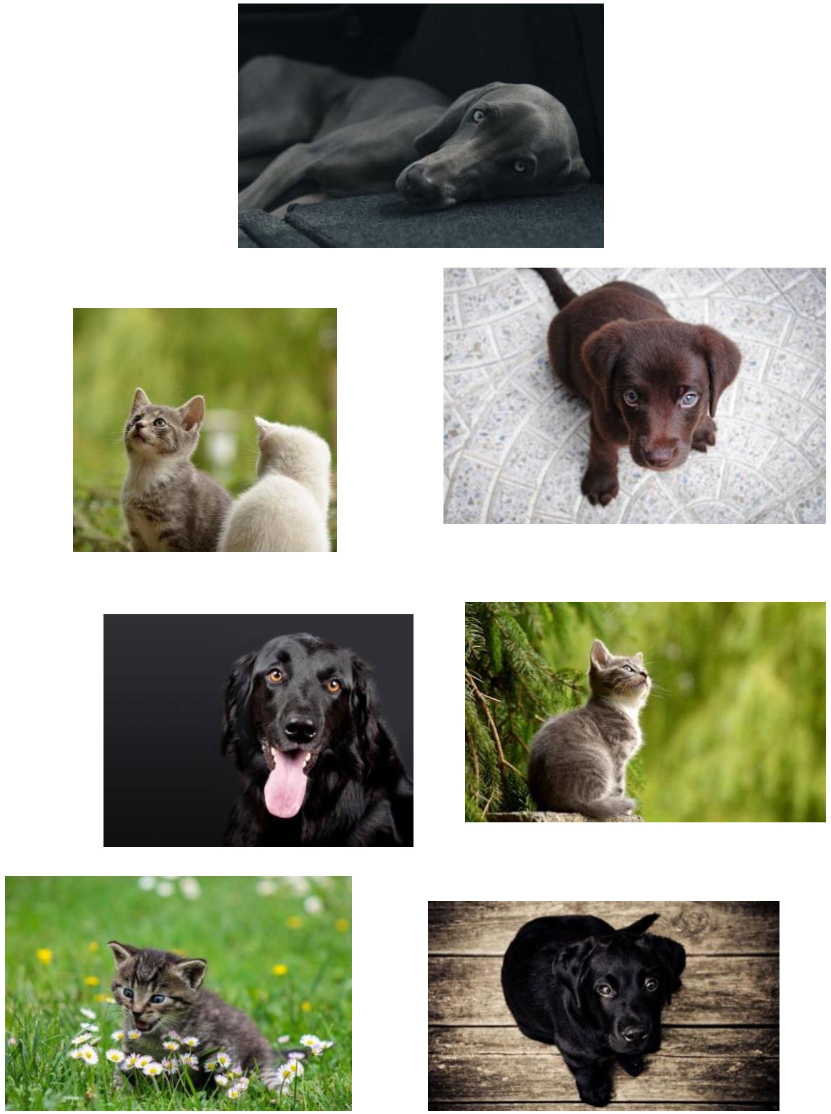






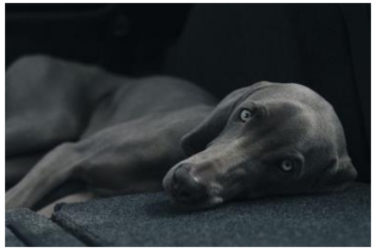

What is learning about?





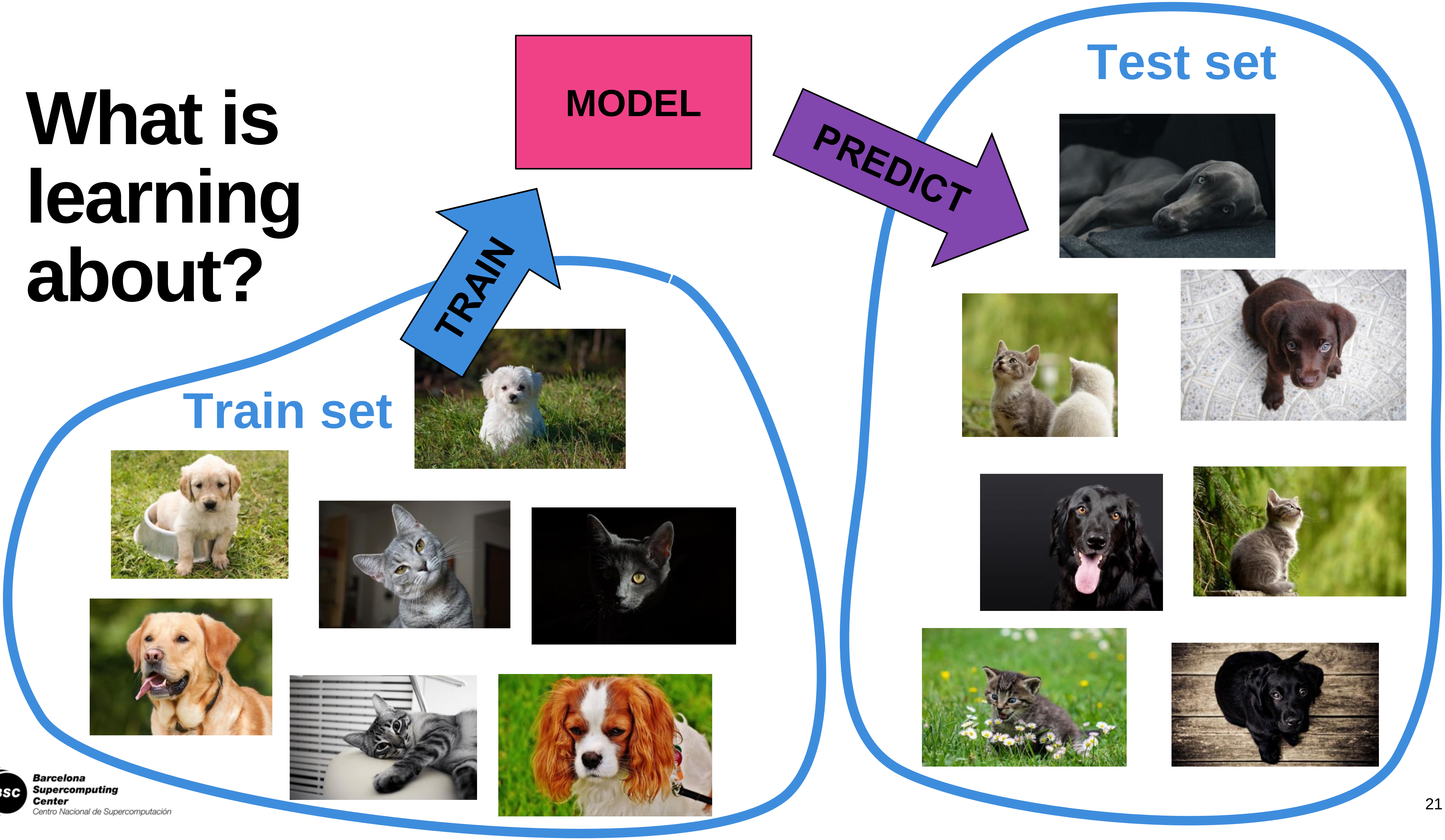
What is learning about?

Test set

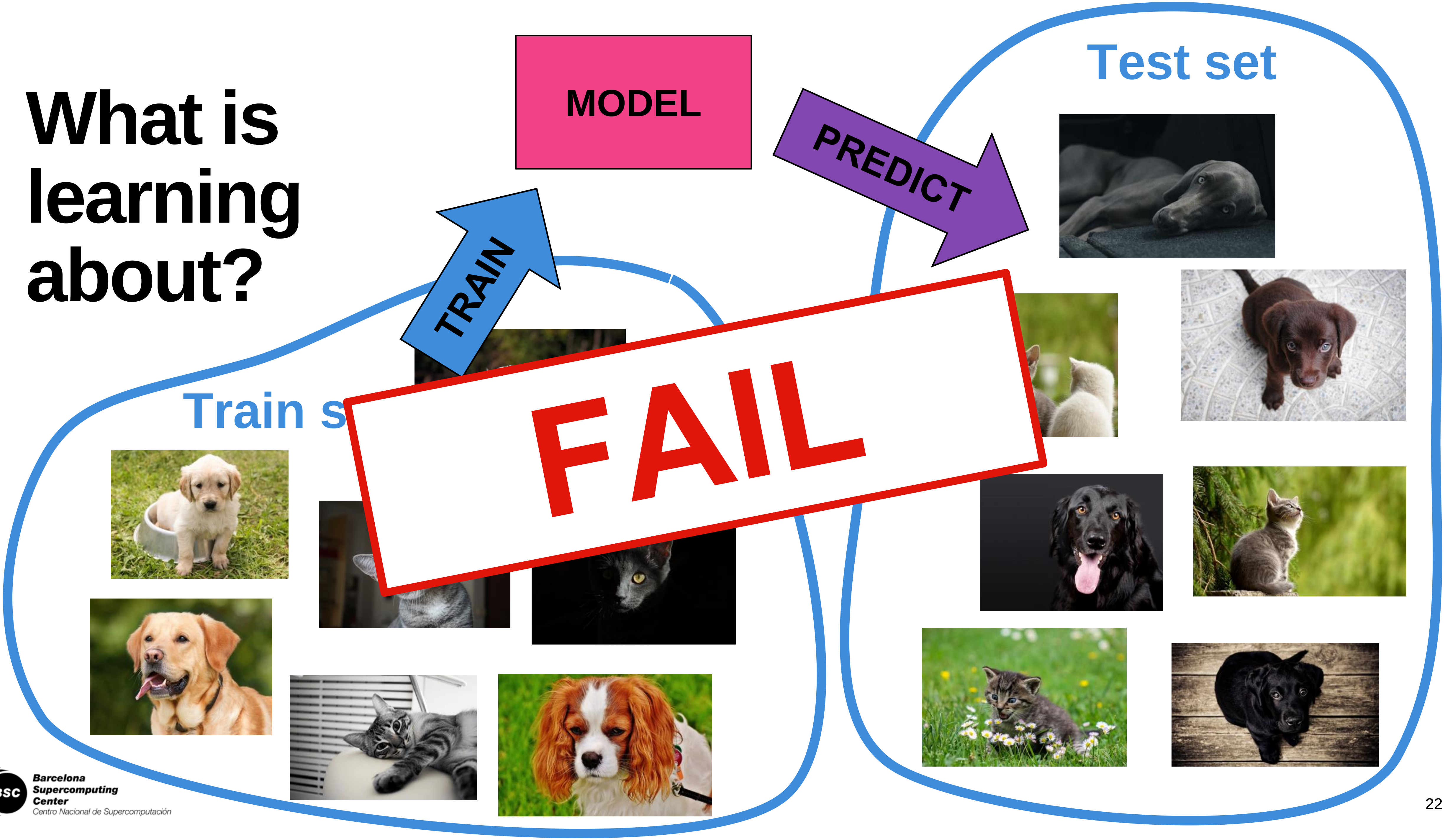


Train set

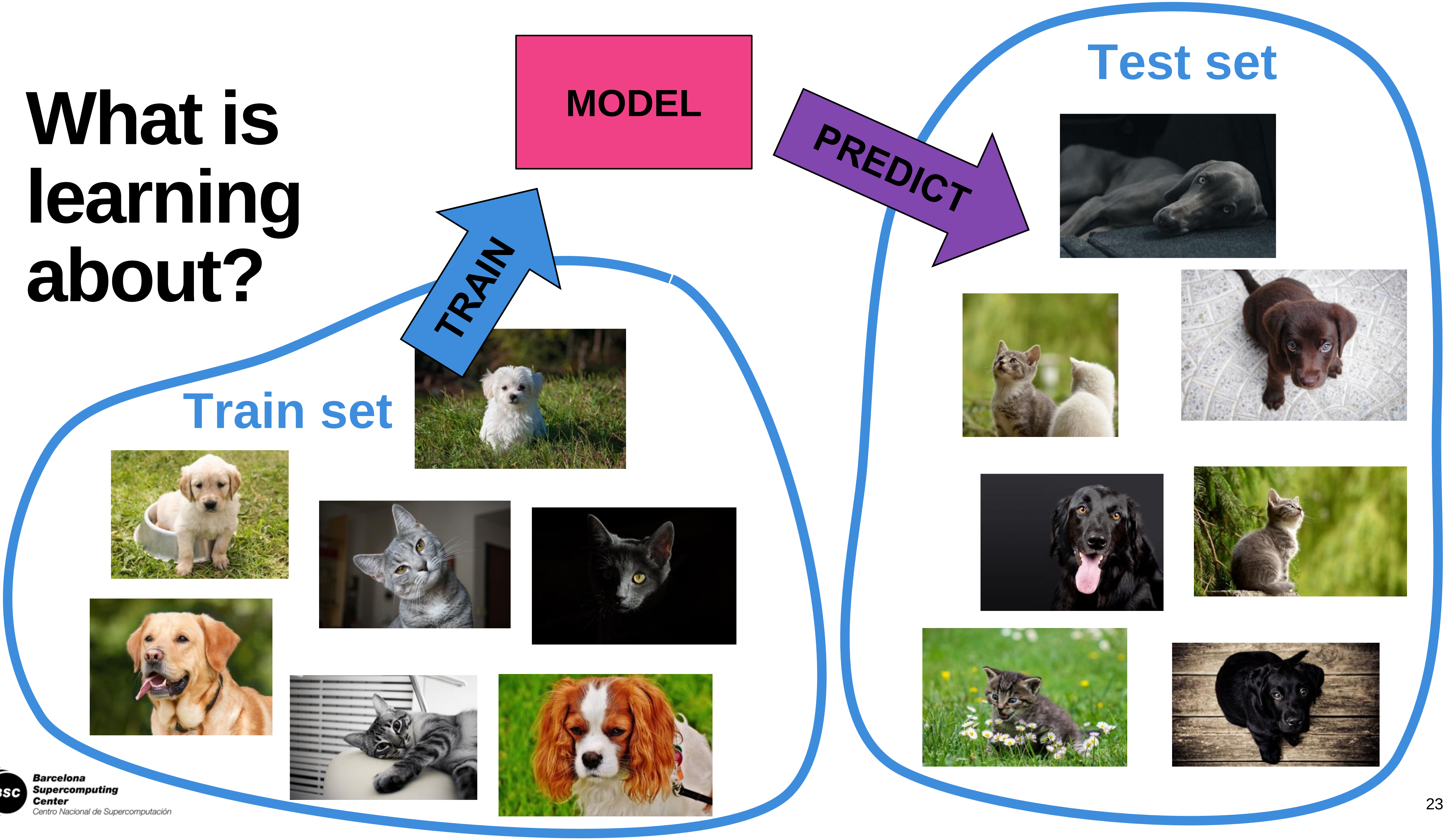
What is learning about?



What is learning about?

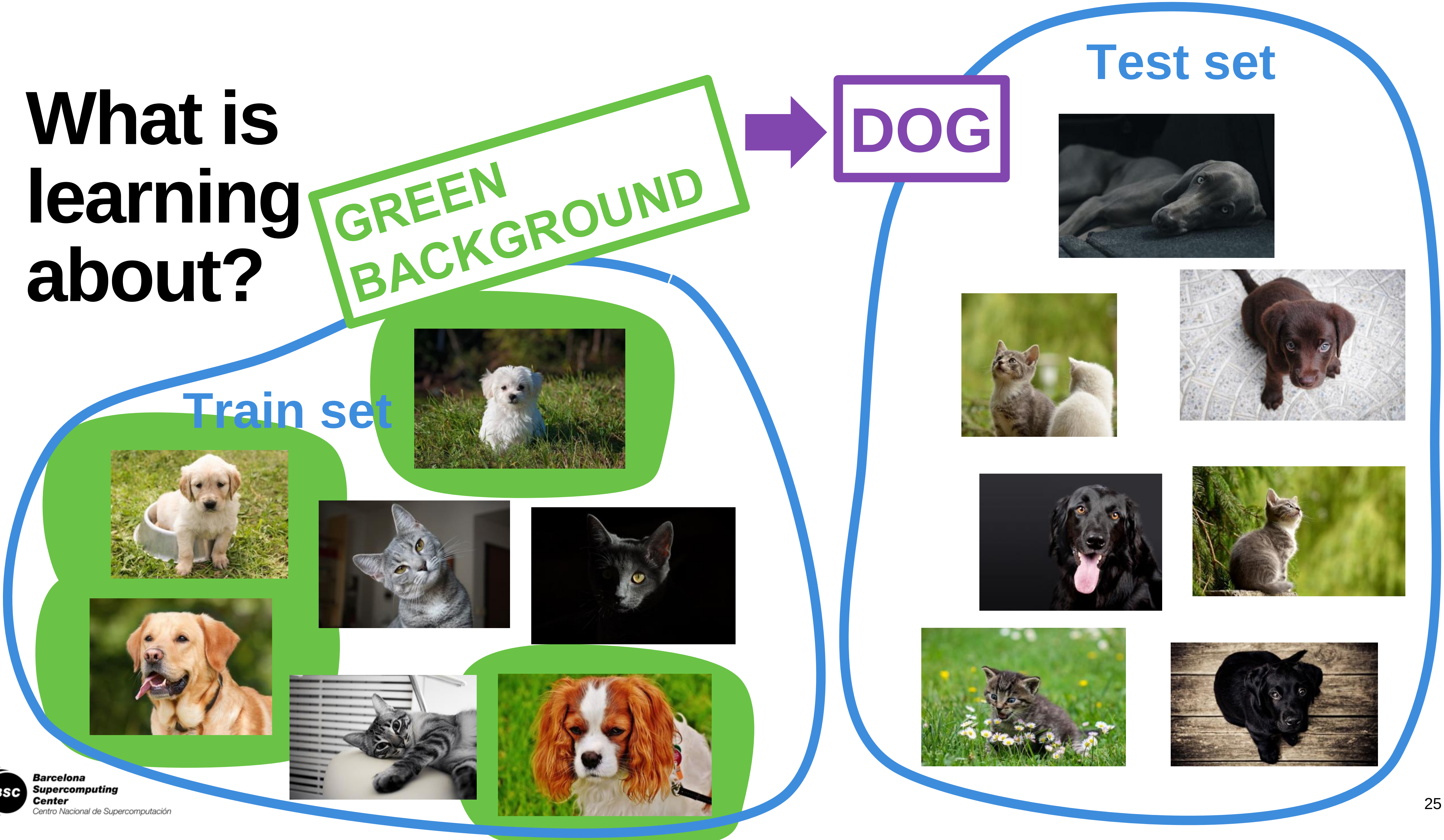


What is learning about?

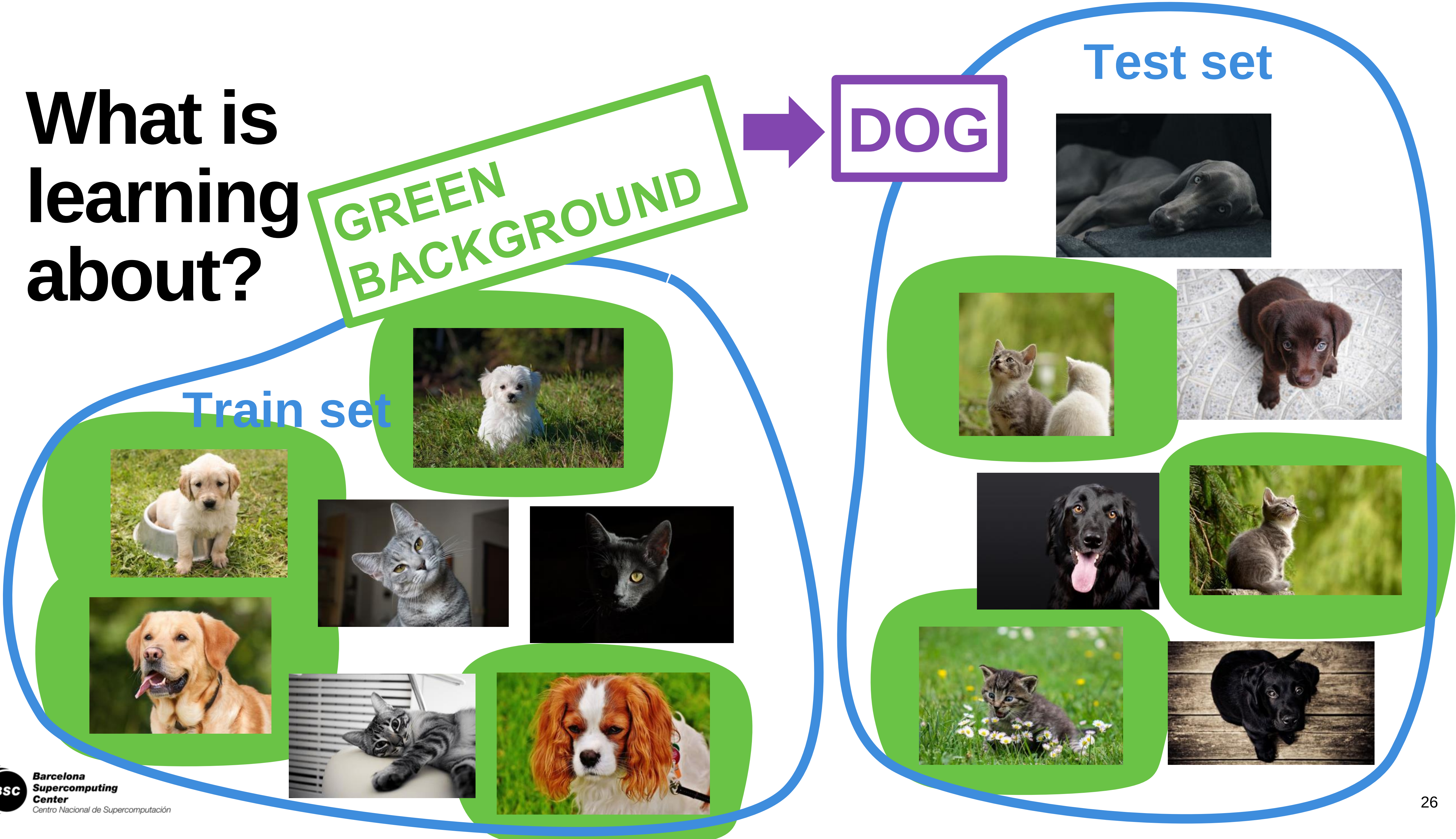


What is learning about?

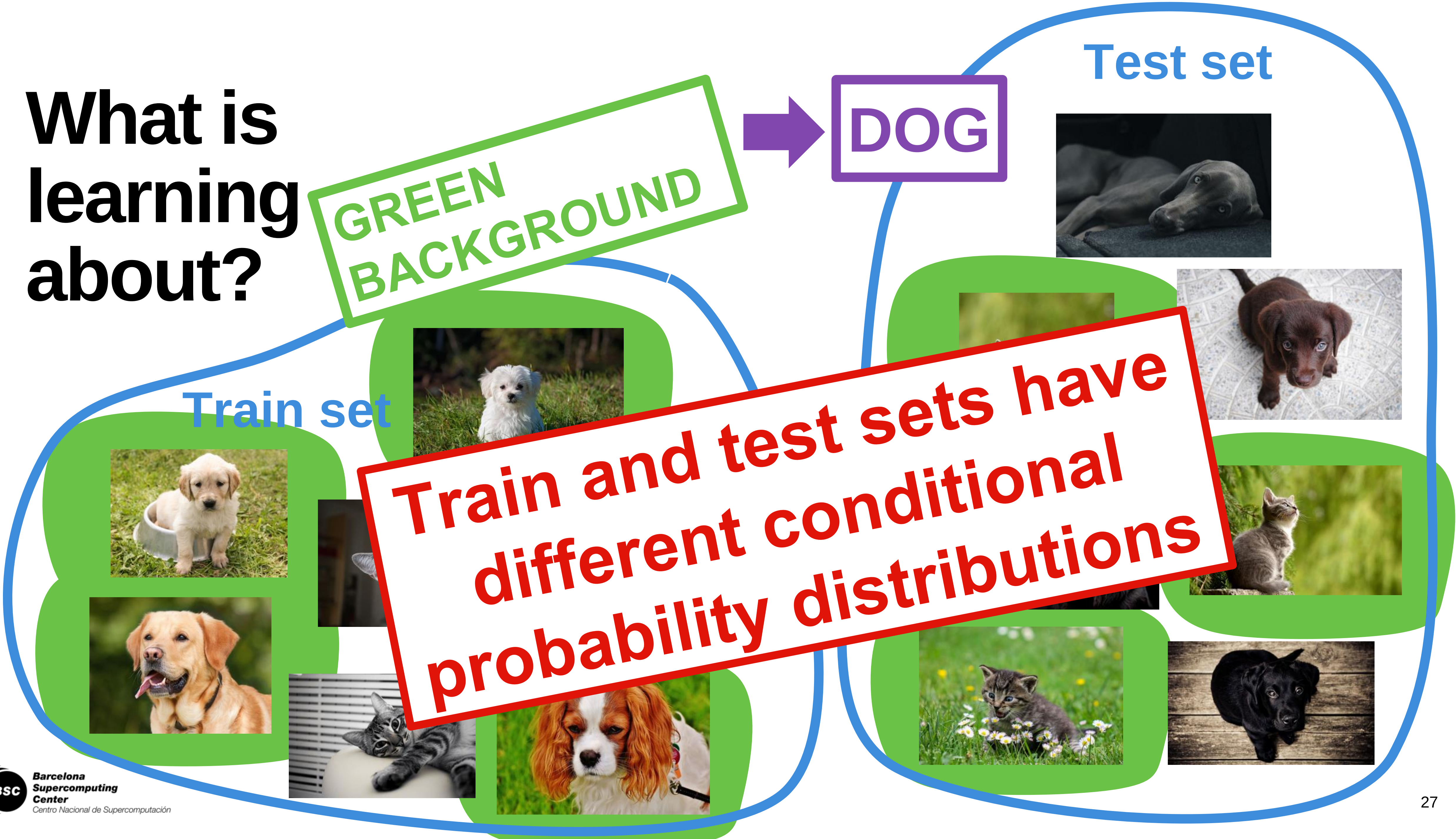
What is learning about?



What is learning about?



What is learning about?



What is learning about?

We must **seek** to have **similar** conditional probability distributions in train and test sets

What is learning about?

THEY WILL NEVER BE **EXACTLY** EQUAL

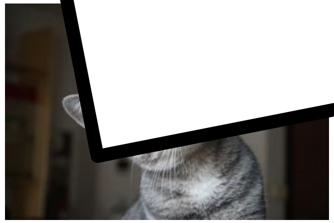
We must **seek** to have **similar** conditional probability distributions in train and test sets

Test set



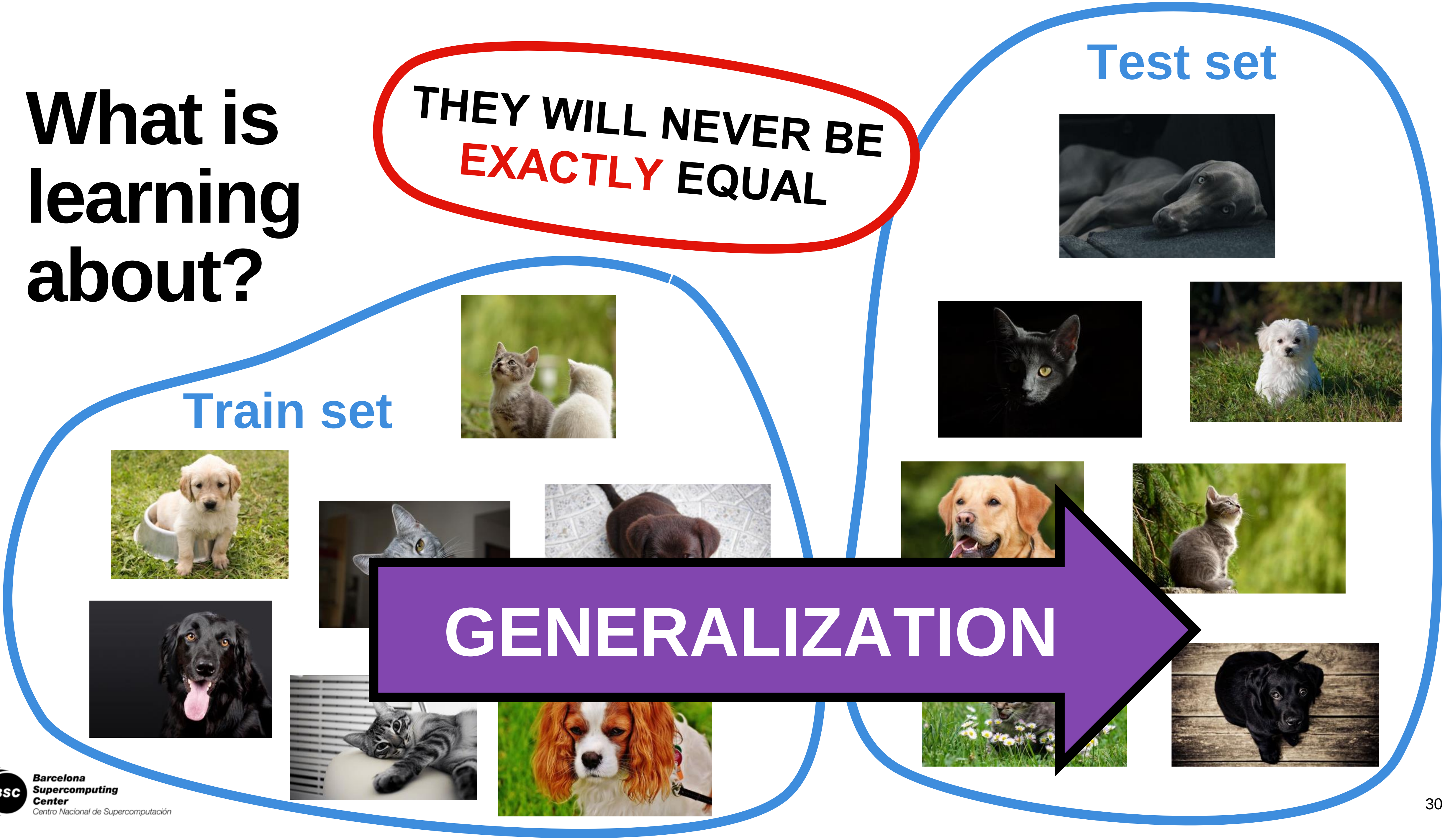



Train s



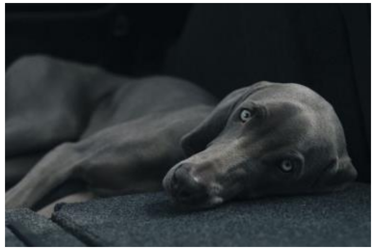
What is learning about?



What is **transfer** learning about?

Train set







Test set







What is **transfer** learning about?

Train set

GENERALIZATION

Test set

What is **transfer** learning about?

Train a **Machine Learning Model** on a **train set** with the hope that what has been learnt will be useful to solve a **different task.**

Train set

GENERALIZATION

Test set

What is transfer learning about?

Train a **Machine Learning Model** on a **train set** with the hope that what has been learnt will be useful to solve a **different task.**

Train set

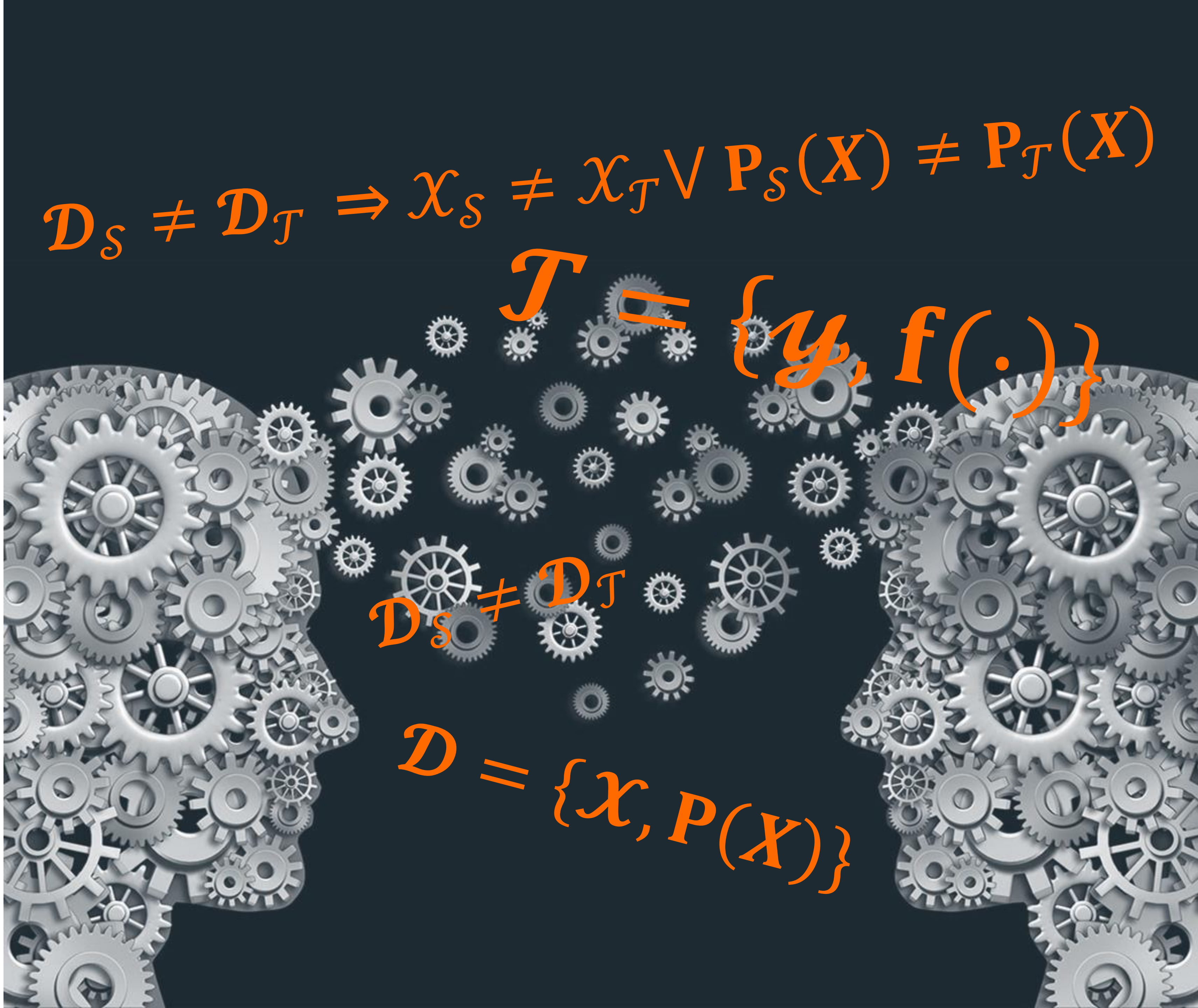
GENERALIZATION

Test set

Train and **test sets** are drawn from a **not so similar** underlying probability distribution.

Formalizing **transfer** learning

Pan, Sinno Jialin, and Qiang Yang.
"**A survey on transfer learning.**"
IEEE Transactions on knowledge and data engineering (2010)



Formalizing transfer learning

Domain:

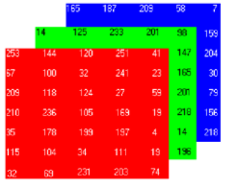
Task:

Formalizing transfer learning

Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$

➔



$\neq$

“The Elgar Concert Hall at the University of Birmingham for our third conference”

➔ Bag of words

➔ Content vector

- A marginal probability distribution $P(X), where\ X = \{x_1, \dots, x_n\} \in \mathcal{X}$

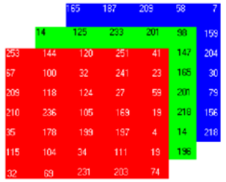
$\neq$

Task:

Formalizing transfer learning

Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$

 ➔  ≠ "The Elgar Concert Hall at the University of Birmingham for our third conference" ➔ Bag of words ➔ Content vector

- A marginal probability distribution $P(X), where\ X = \{x_1, \dots, x_n\} \in \mathcal{X}$

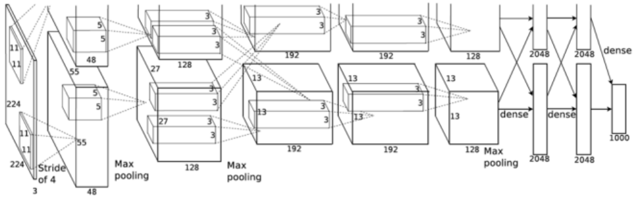
   ≠  

Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

- A label space $\mathcal{Y}$

CAT, DOG ≠ LION, WOLF

- An objective predictive function $f(\cdot) \Leftrightarrow P(y|x)$



Formalizing transfer learning

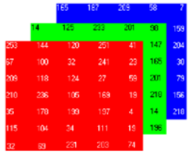
	Source	Target
Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$		
Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$		

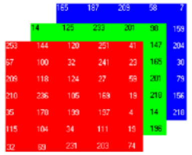
Formalizing transfer learning

Source | **Target**

Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$
  - **The Same (different)**

 ➔ 

 ➔ 

- A marginal probability distribution $P(X)$
  - **Different**
  - **Similar**

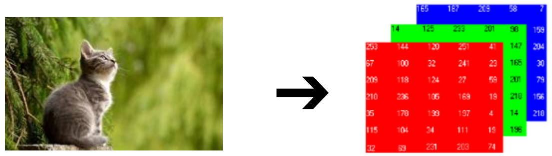
Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

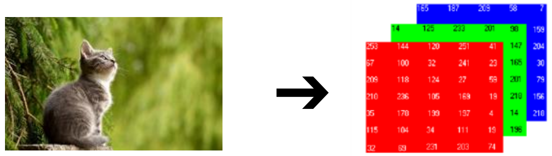
Formalizing transfer learning

Source | **Target**

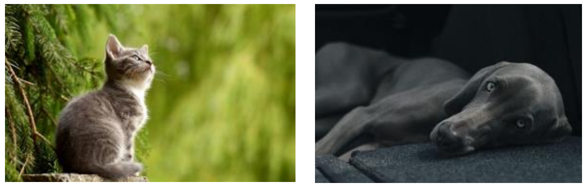
Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$
  - **The Same (different)**





- A marginal probability distribution $P(X)$
  - **Different**
  - **Similar**



Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

- A label space $\mathcal{Y}$
  - **Different**
  - **The same**

Source	Target
{CAT, DOG}	{LION, WOLF}
{FELINE, CANINE}	{FELINE, CANINE}

- An objective predictive function
  - **Different (but similar?)**

$f_S(\cdot)$ $f_T(\cdot)$

What is **transfer** learning about?

Test set

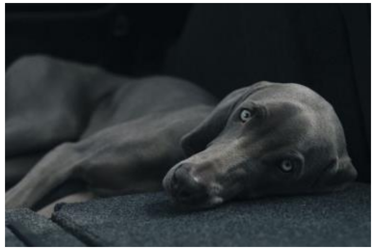






Train set

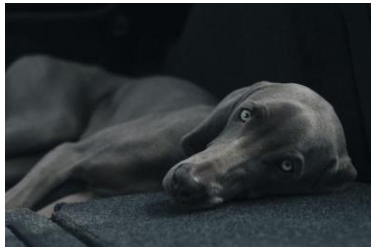






What is **transfer** learning about?

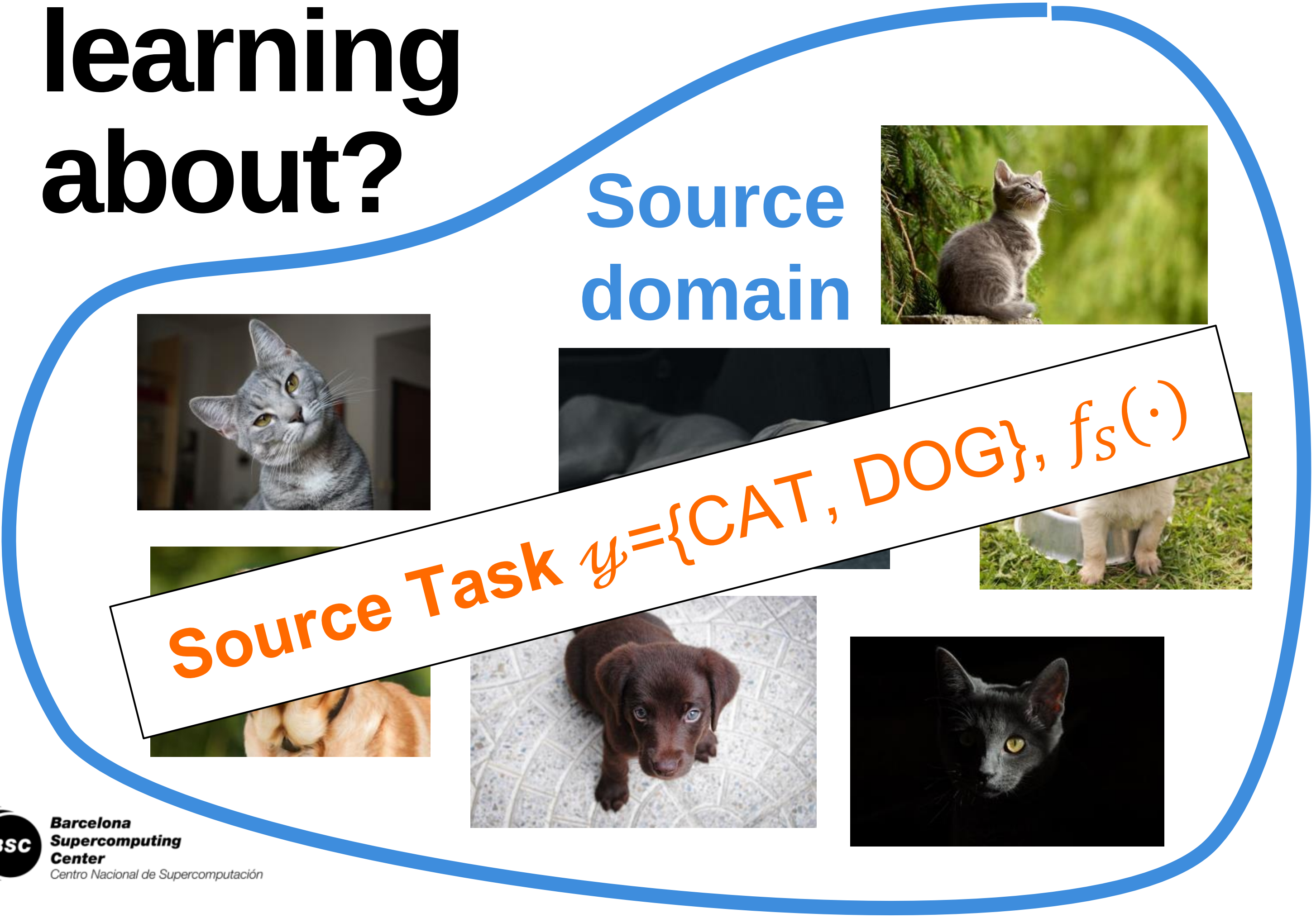
Source domain

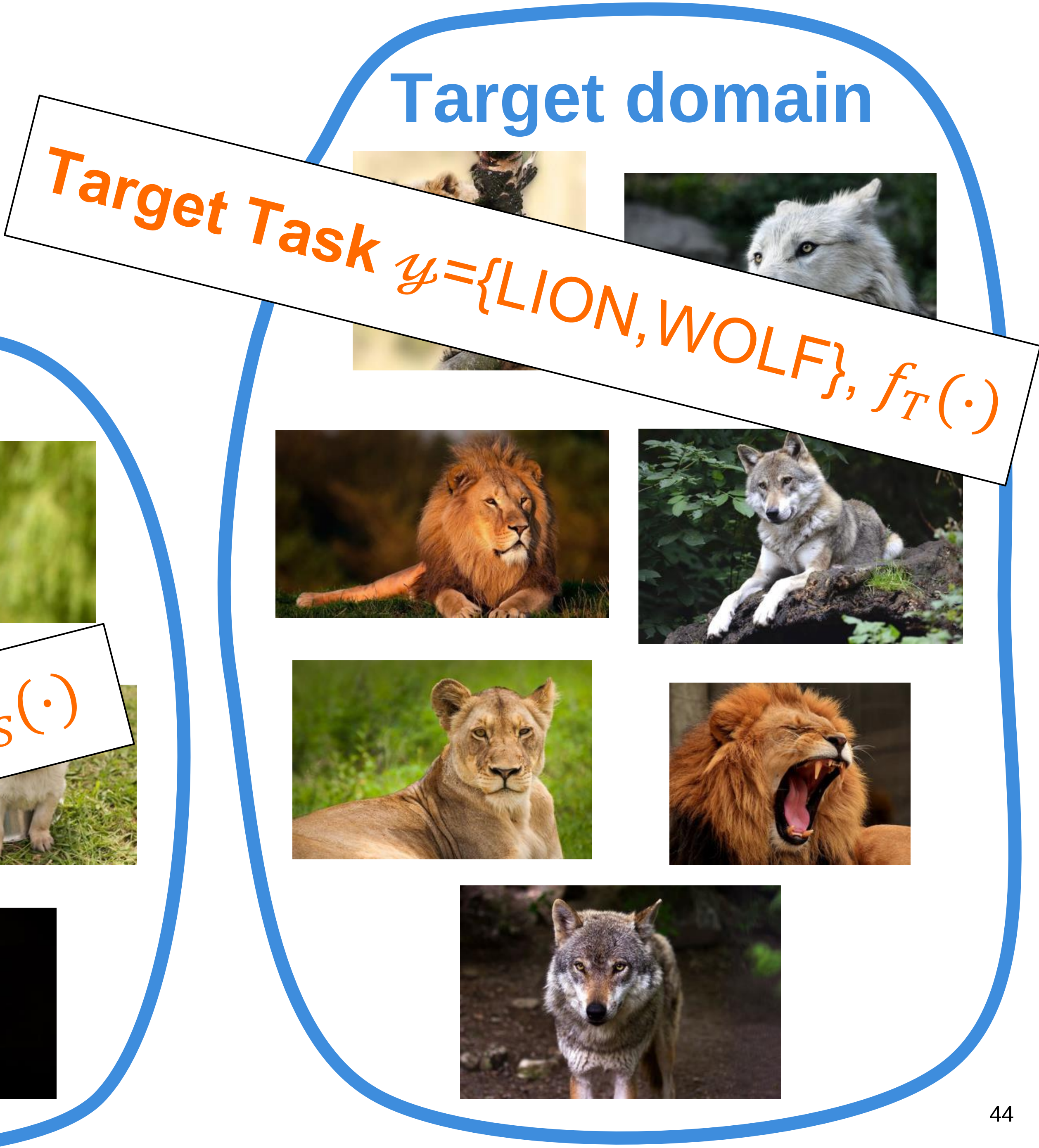


Target domain

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

What is **transfer** learning about?



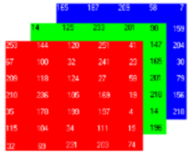


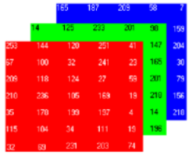
Formalizing transfer learning

Source | **Target**

Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$
  - The Same (different)

 ➔ 

 ➔ 

- A marginal probability distribution $P(X)$
  - Different
  - Similar

Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

- A label space $\mathcal{Y}$
  - Different
  - The same

Source: **{CAT, DOG}** / **{FELINE, CANINE}**

Target: **{LION, WOLF}** / **{FELINE, CANINE}**

- An objective predictive function — Source: $f_S(\cdot)$ — Target: $f_T(\cdot)$
  - Different (but similar?)

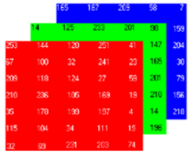
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

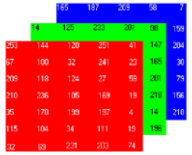
Formalizing transfer learning

Source | Target

Domain: $\mathcal{D} = \{\mathcal{X}, P(X)\}$

- A feature space $\mathcal{X}$
  - **The Same (different)**

 ➔ 

 ➔ 

- A marginal probability distribution $P(X)$
  - **Different**
  - **Similar**

Task: $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$

- A label space $\mathcal{Y}$
  - **Different** — Source: **{CAT, DOG}** — Target: **{LION, WOLF}**
  - **The same** — Source: **{FELINE, CANINE}** — Target: **{FELINE, CANINE}**

- An objective predictive function — Source: $f_S(\cdot)$ — Target: $f_T(\cdot)$
  - **Different (but similar?)**

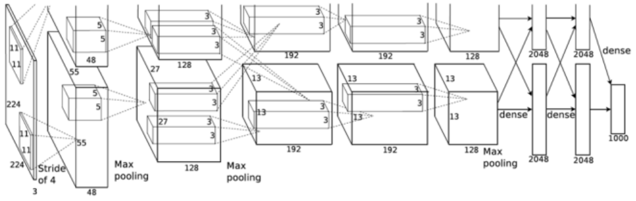
Formalizing transfer learning

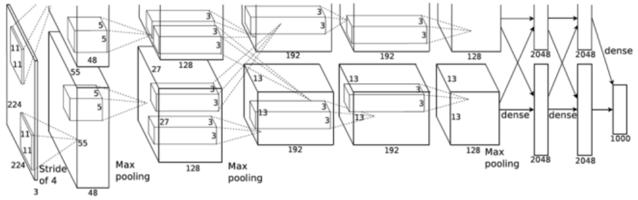
Source

$f_S(\cdot)$

Target

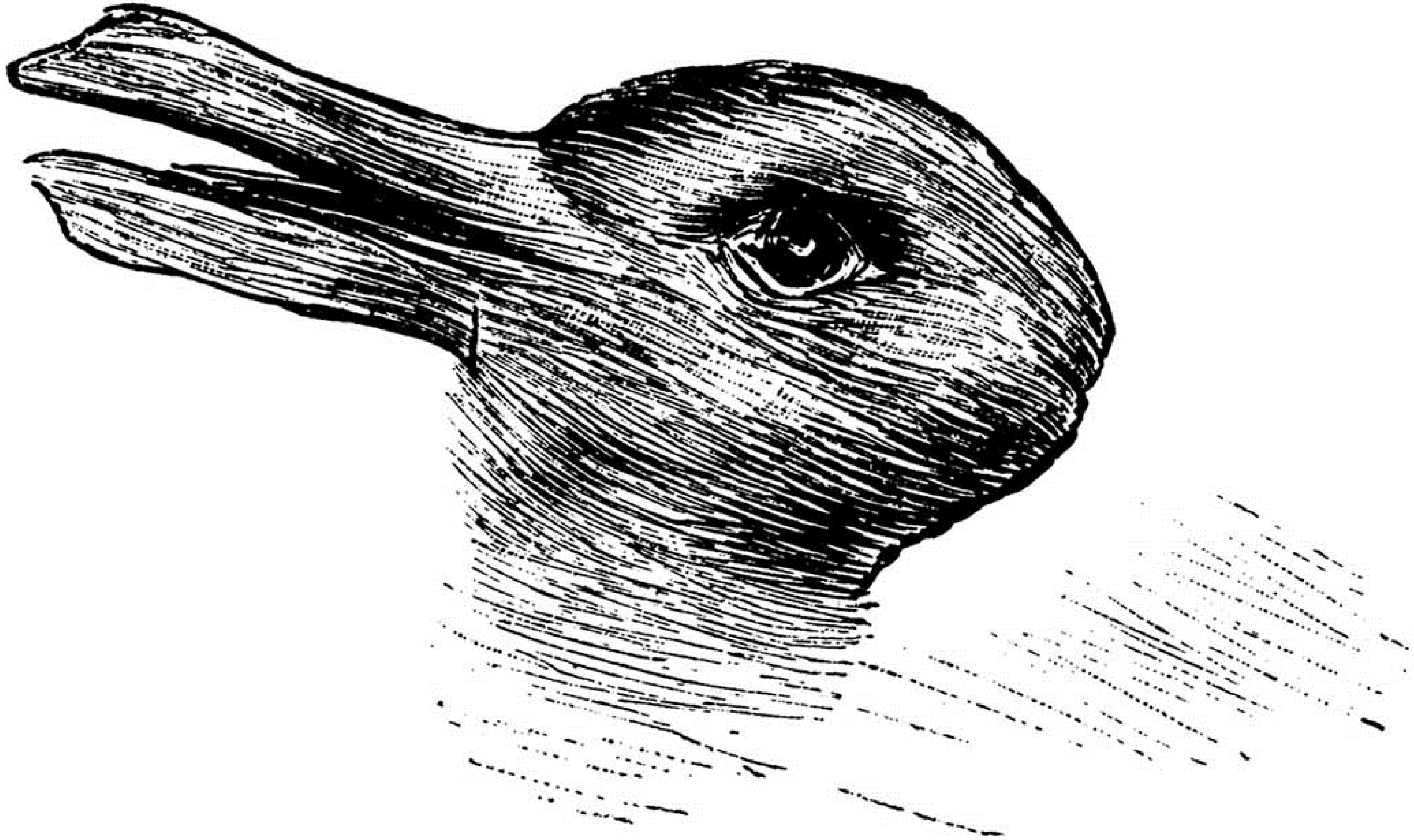
$f_T(\cdot)$

$\hat{f}_S(\cdot) =$ 

$\hat{f}_T(\cdot) =$ 

- Are they similar?
- Can we just use $\hat{f}_S(\cdot)$ to approximate $f_T(\cdot)$?
- Can we reuse part of it?

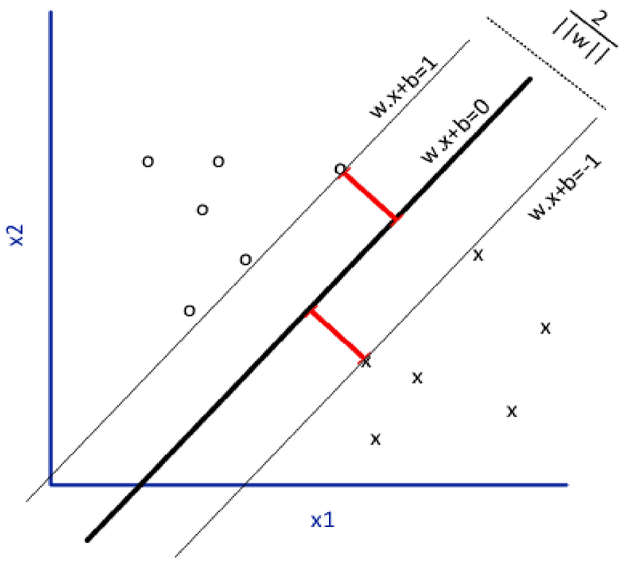
Representation learning



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

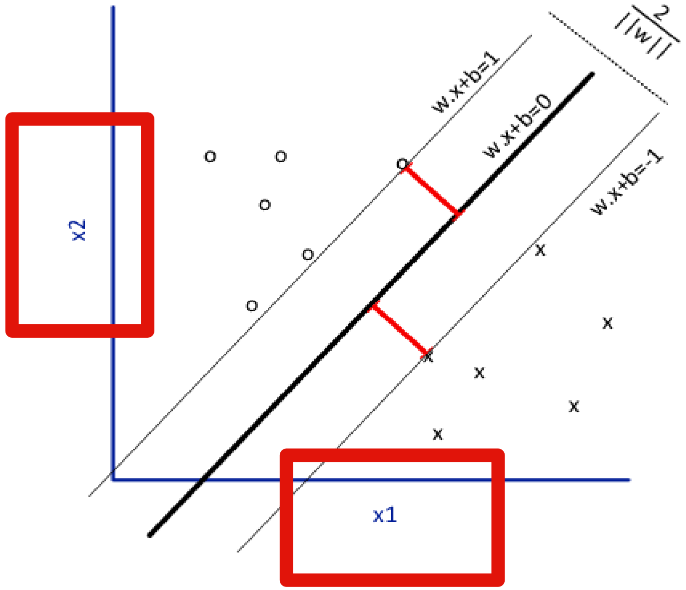
Deep Neural Networks are **representation** learning techniques

- **Support Vector Machine** (SVM) is just a **classifier** (a very good one).
- SVM find the best boundary separating the data instances into different classes in a **given** feature space.



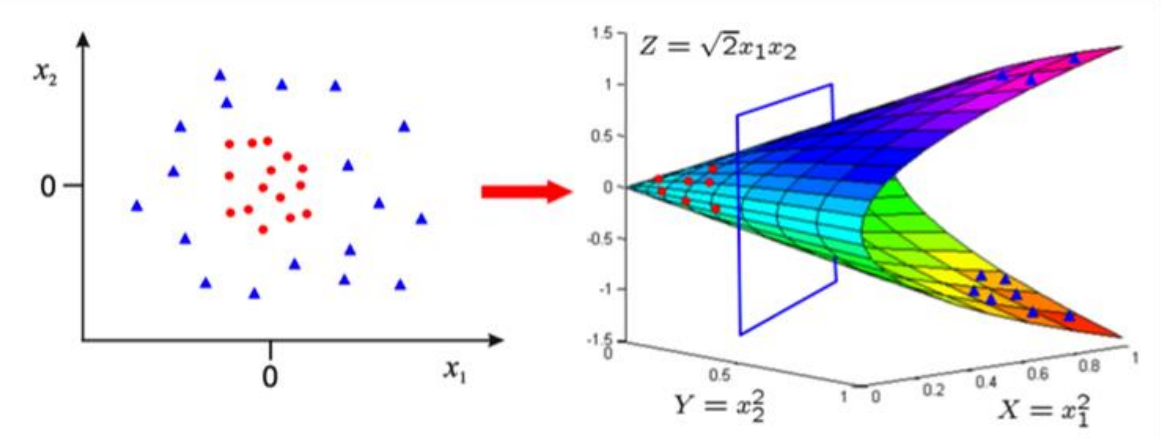
Deep Neural Networks are **representation** learning techniques

- **Support Vector Machine** (SVM) is just a **classifier** (a very good one).
- SVM find the best boundary separating the data instances into different classes in a **given** feature space.

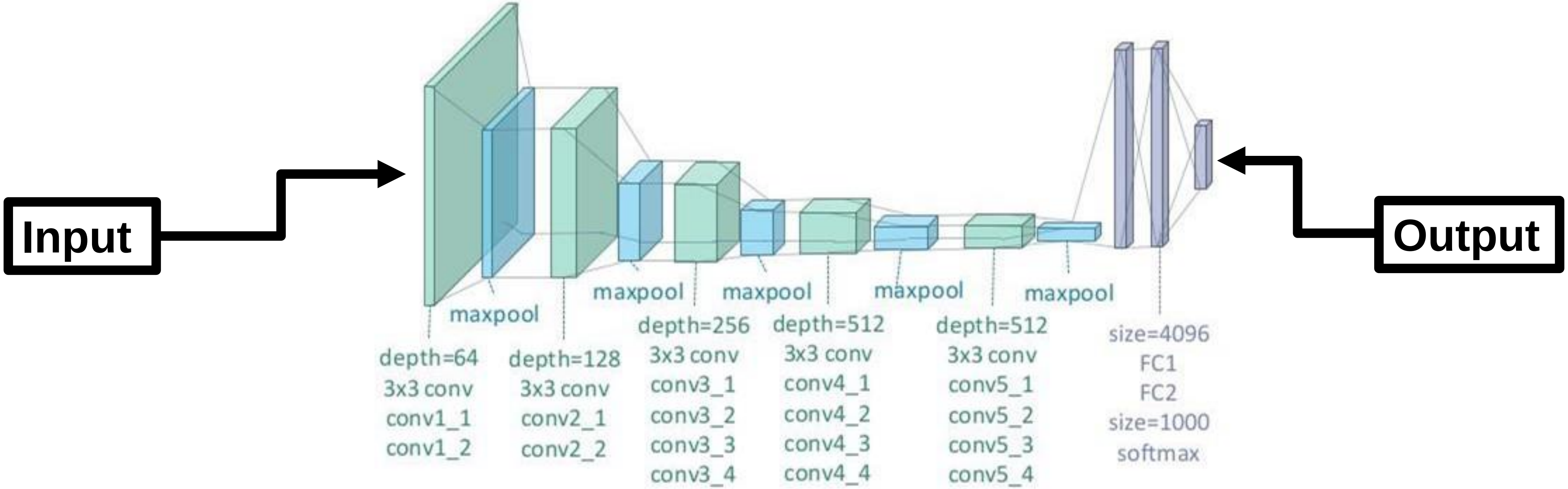


Deep Neural Networks are **representation** learning techniques

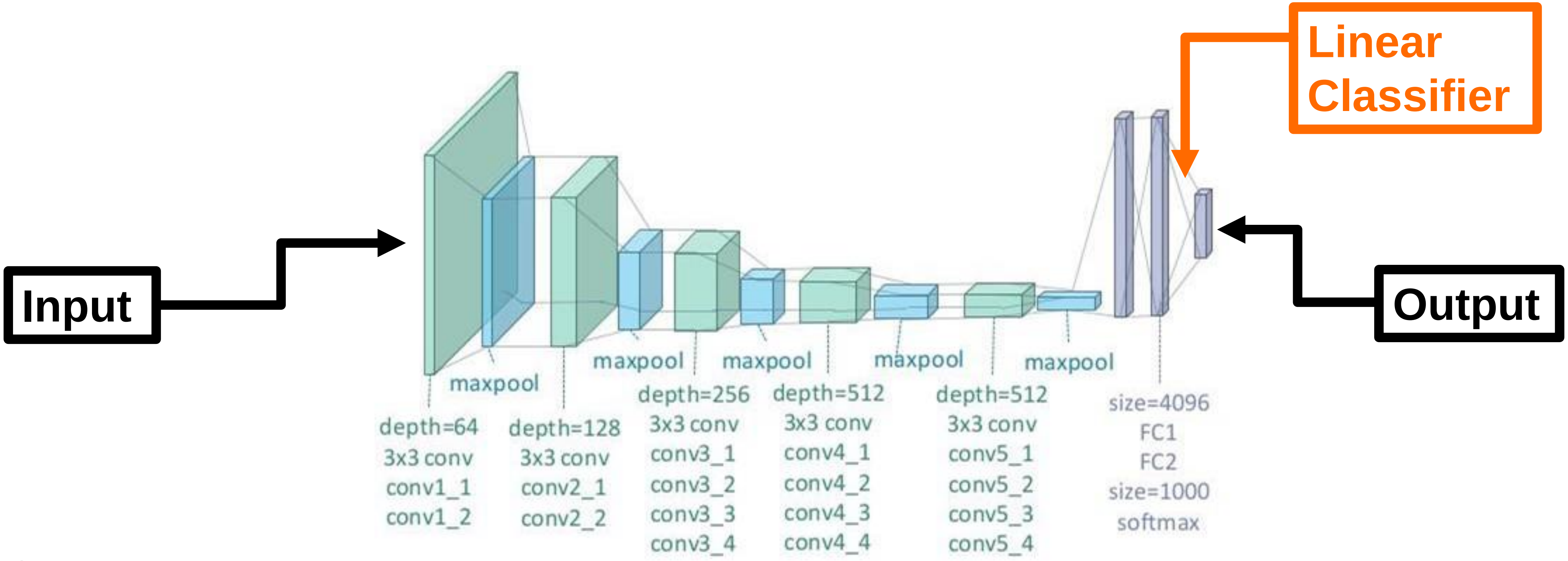
- SVMs using the **kernel trick** can overcome the linear limitation through an **implicit** mapping to a higher dimensional feature space



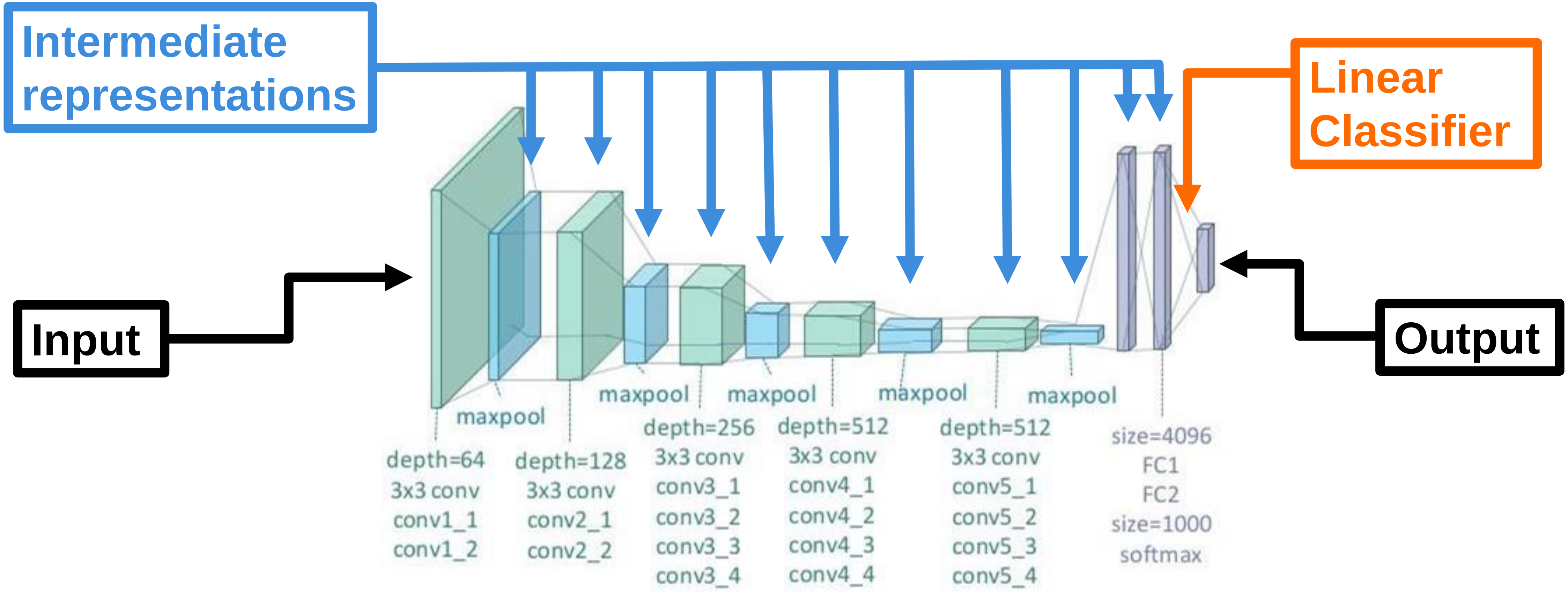
Deep Neural Networks are **representation** learning techniques



Deep Neural Networks are **representation** learning techniques



Deep Neural Networks are **representation** learning techniques



Reusing DNNs knowledge

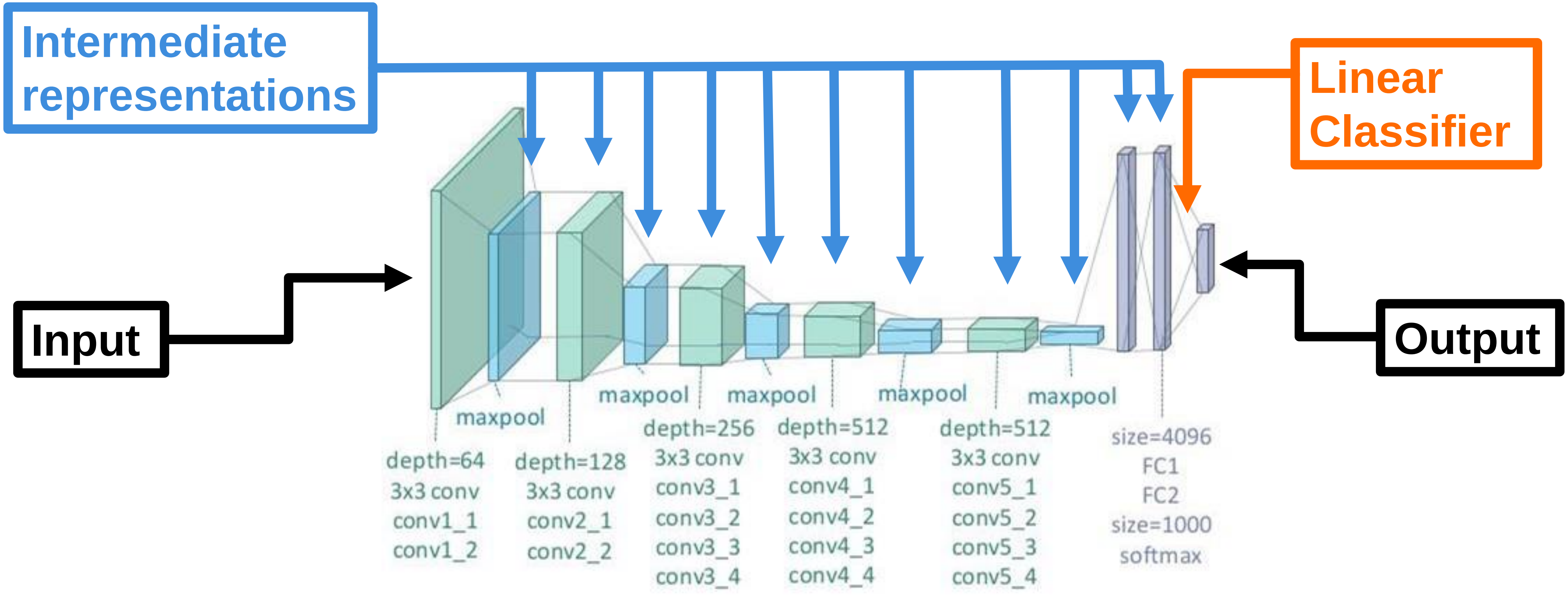
Reusing DNNs knowledge

- Feature Extraction
- Fine-tuning

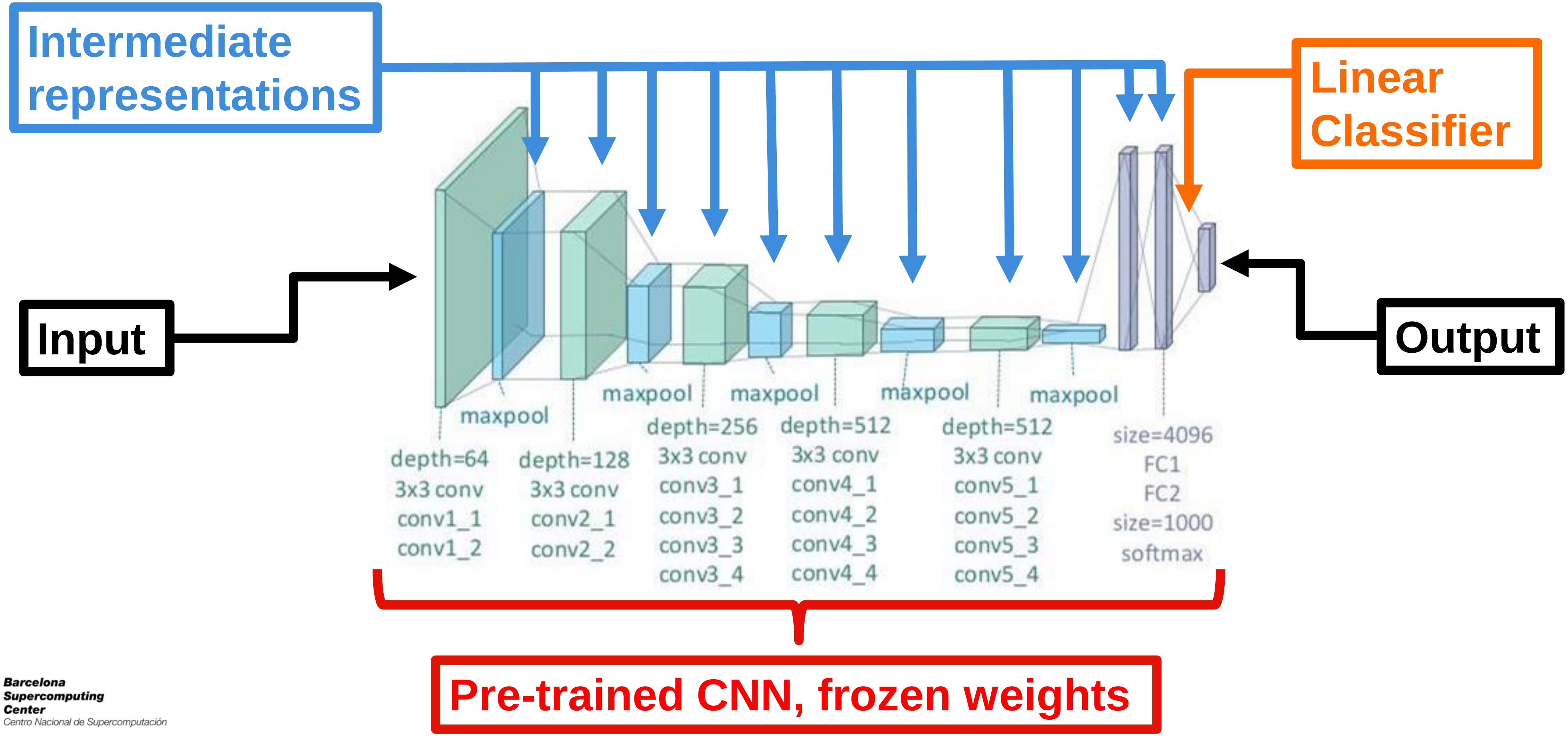
Reusing DNNs knowledge

- **Feature Extraction**
- Fine-tuning

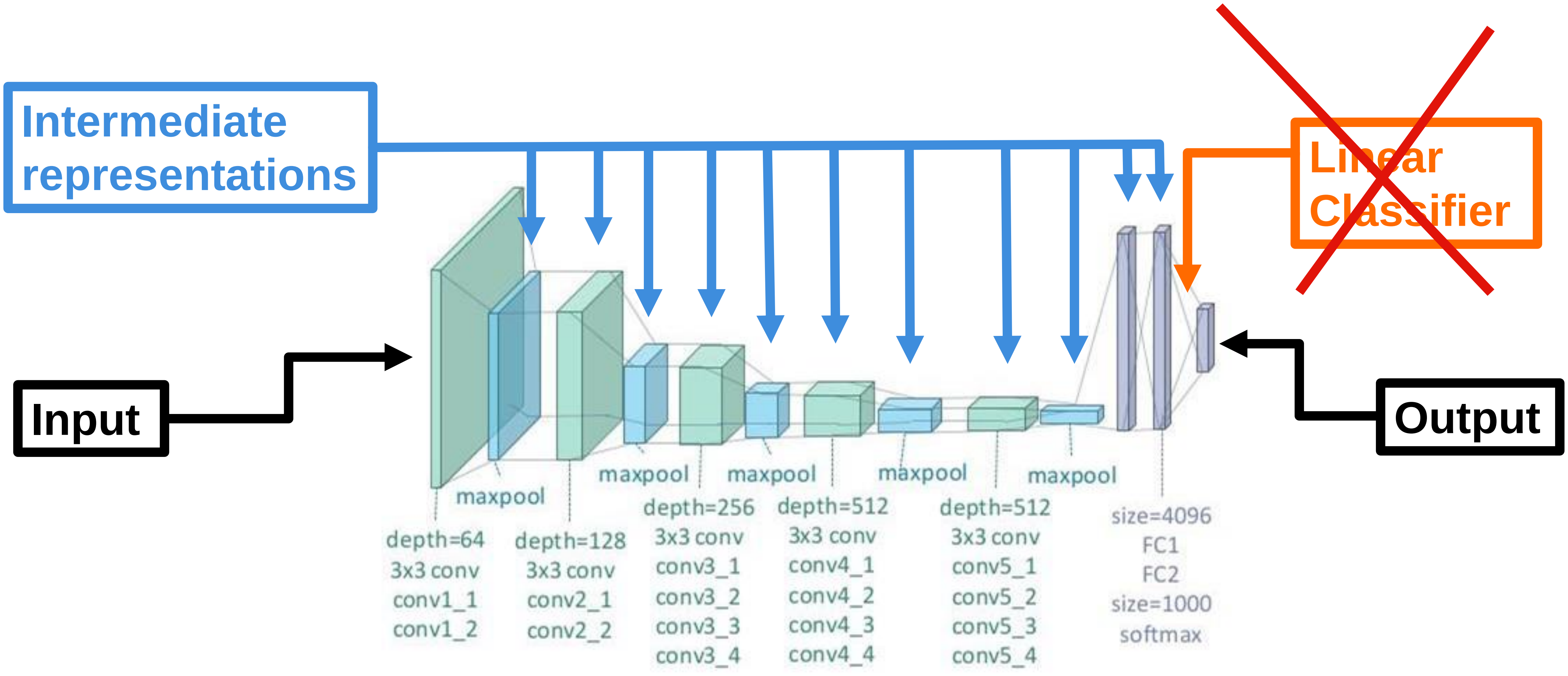
Feature extraction



Feature extraction

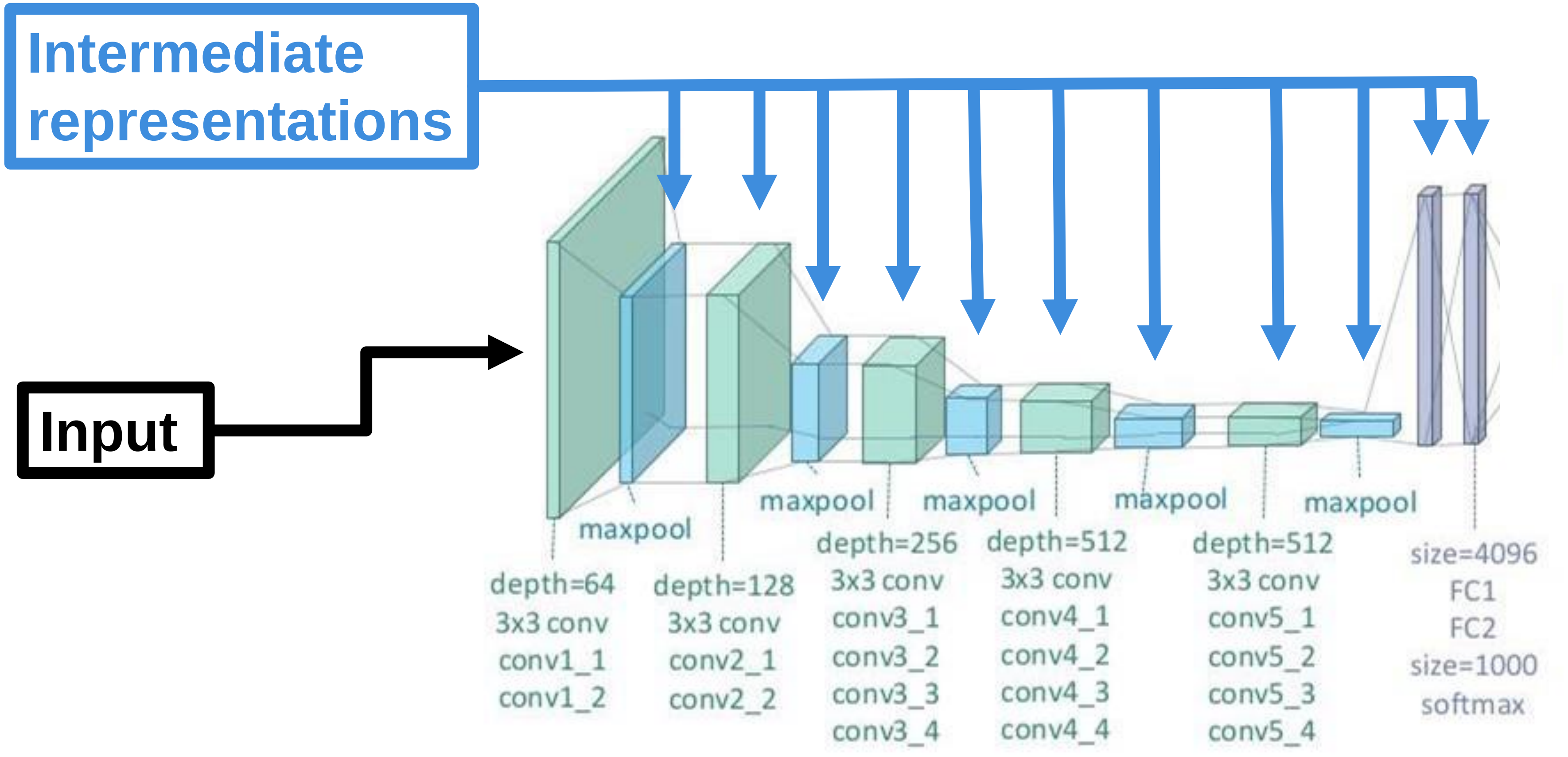


Feature extraction

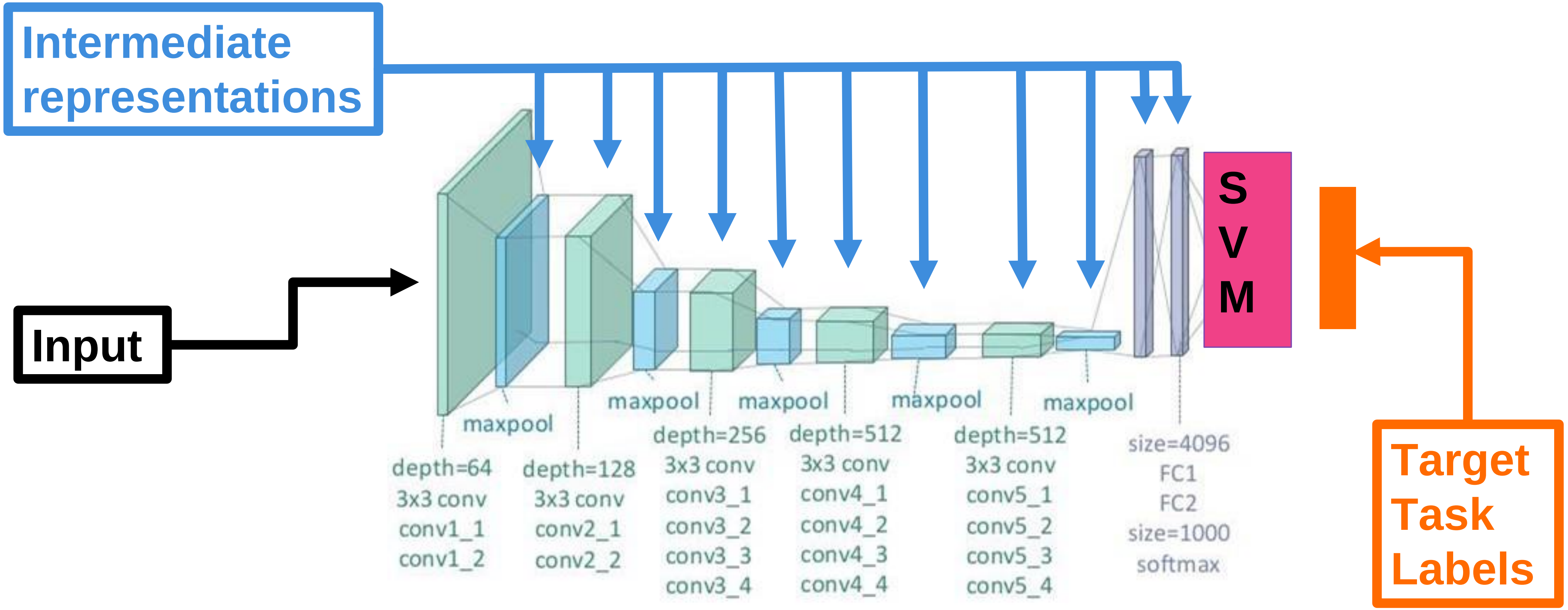


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Feature extraction

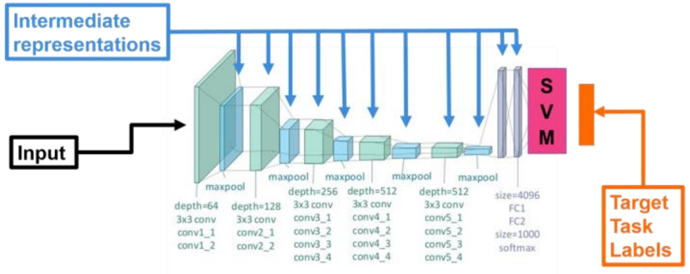


Feature extraction



Simple solutions

- DNN last layer features + SVM
  (Feature extraction)

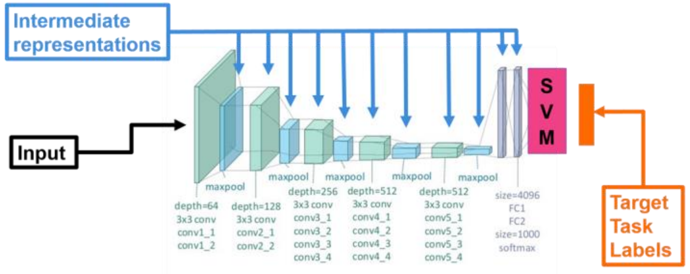


Simple solutions

- DNN last layer features + SVM

  (Feature extraction)

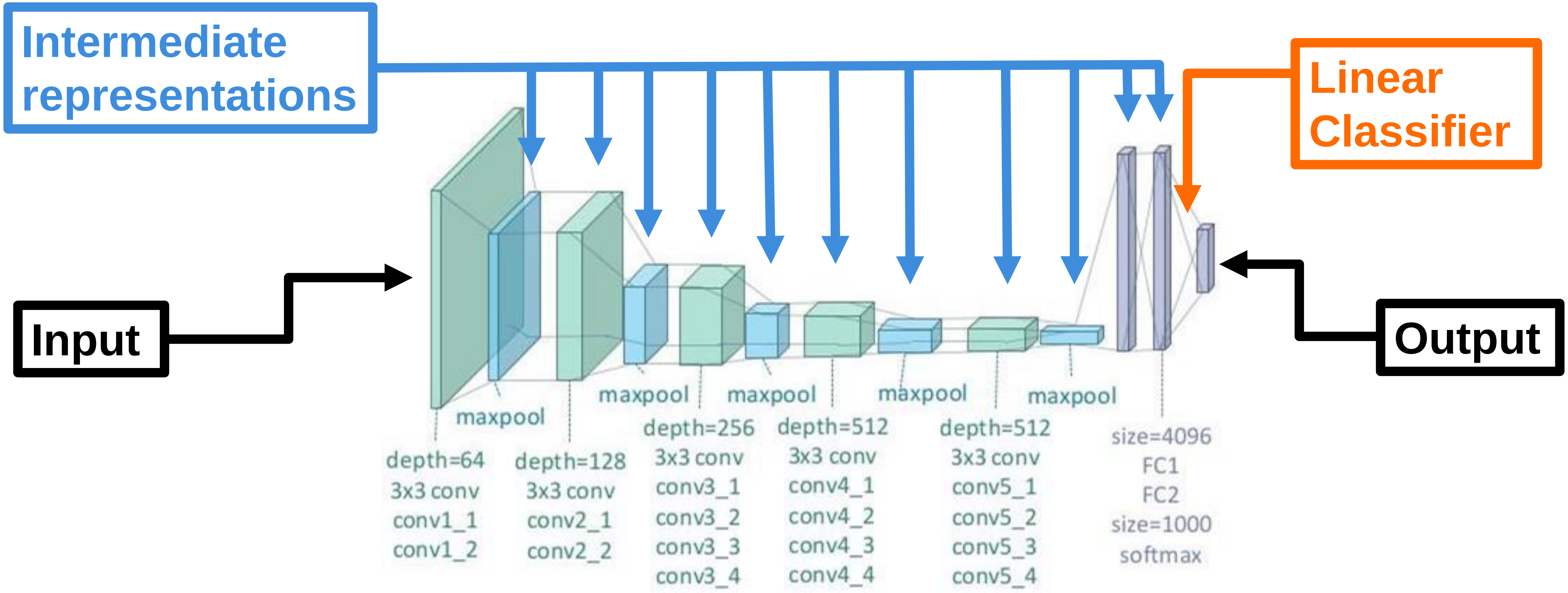
  We need: **Similar task and domain**



Reusing DNNs knowledge

- Feature Extraction
- **Fine-tuning**

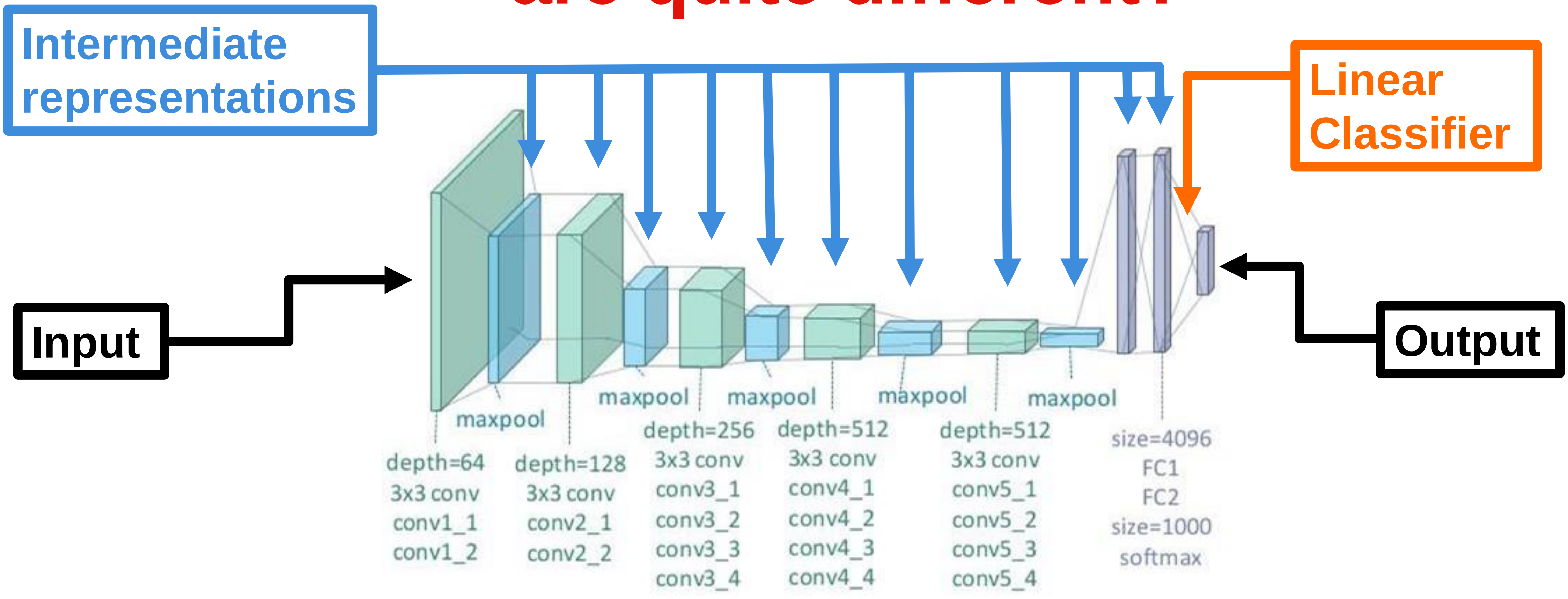
Fine tuning



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fine tuning

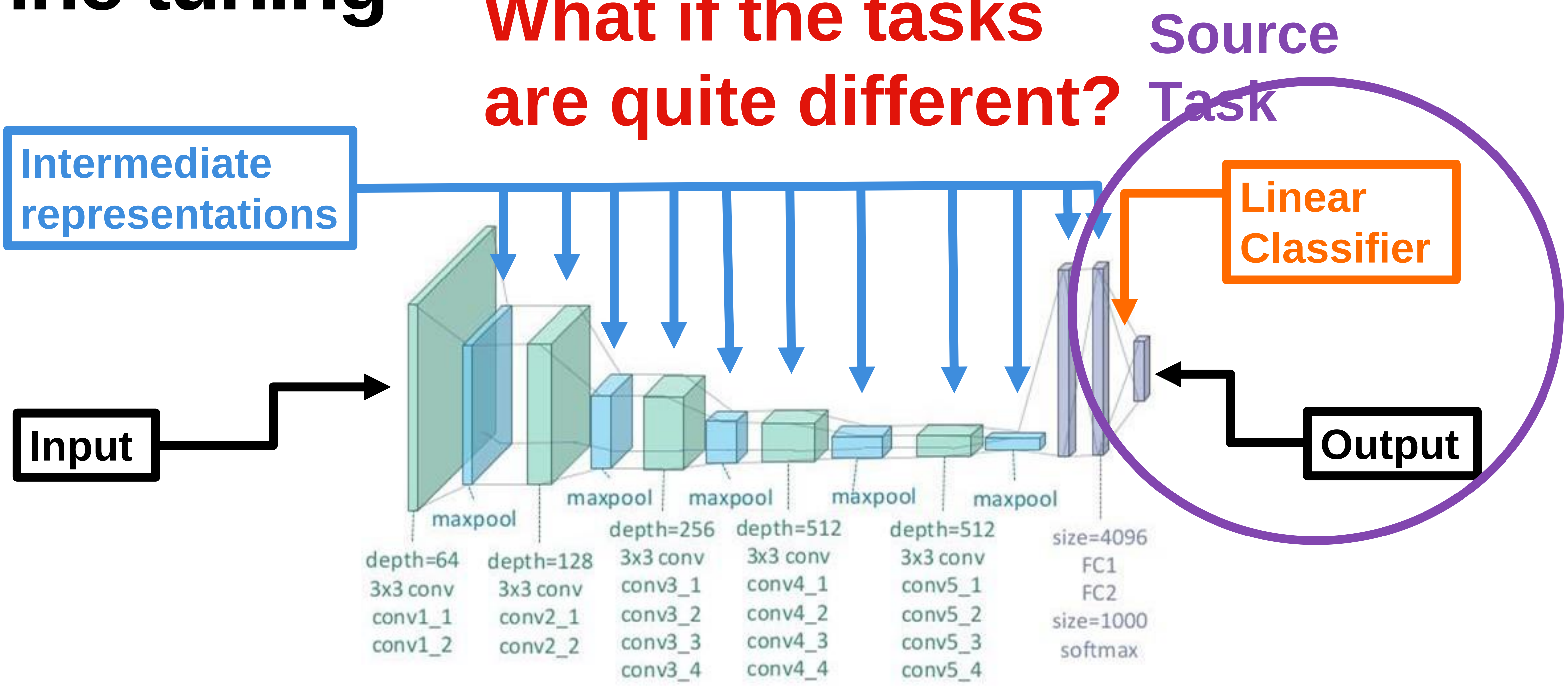
What if the tasks are quite different?



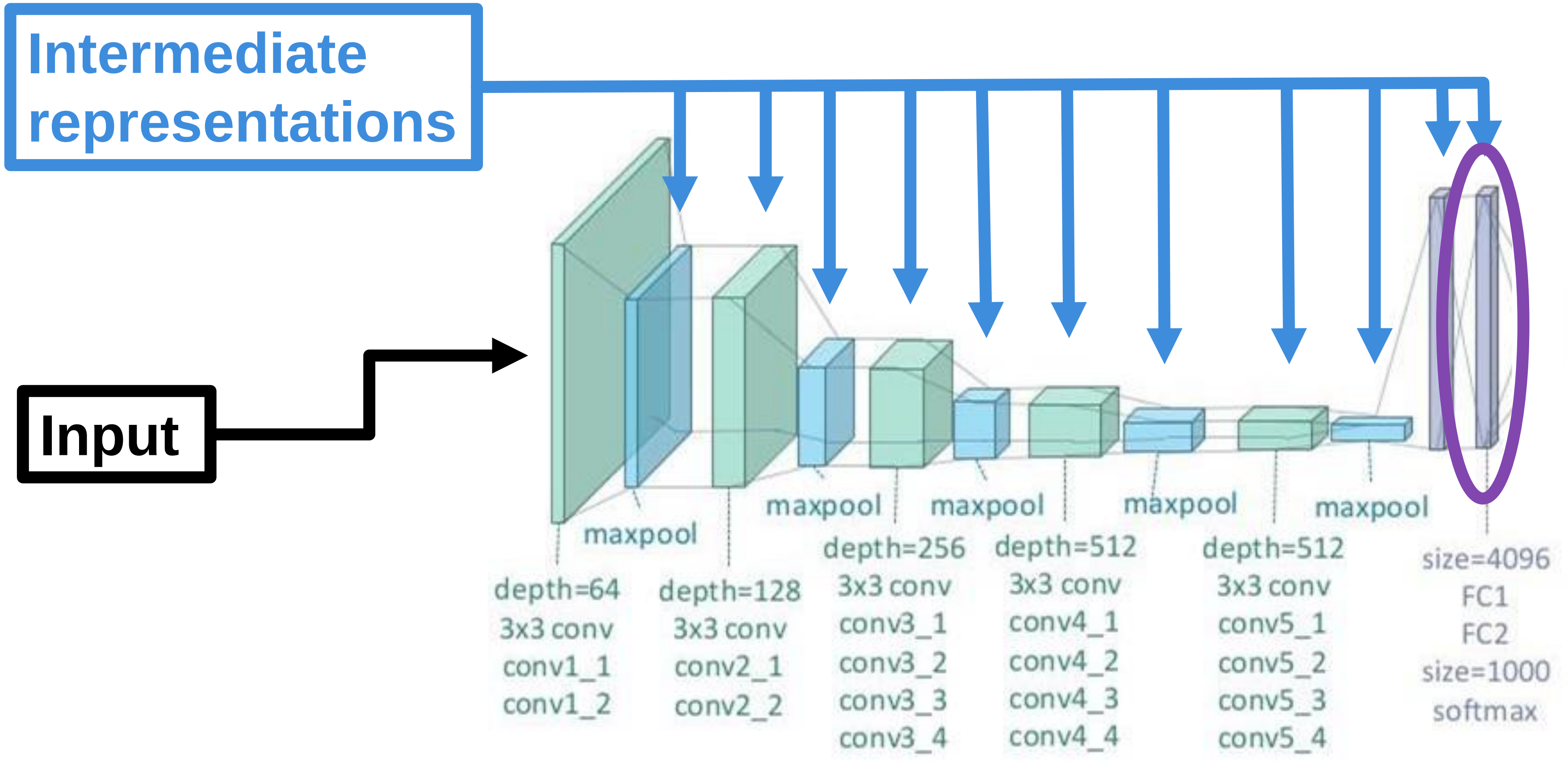
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fine tuning

What if the tasks are quite different?



Fine tuning

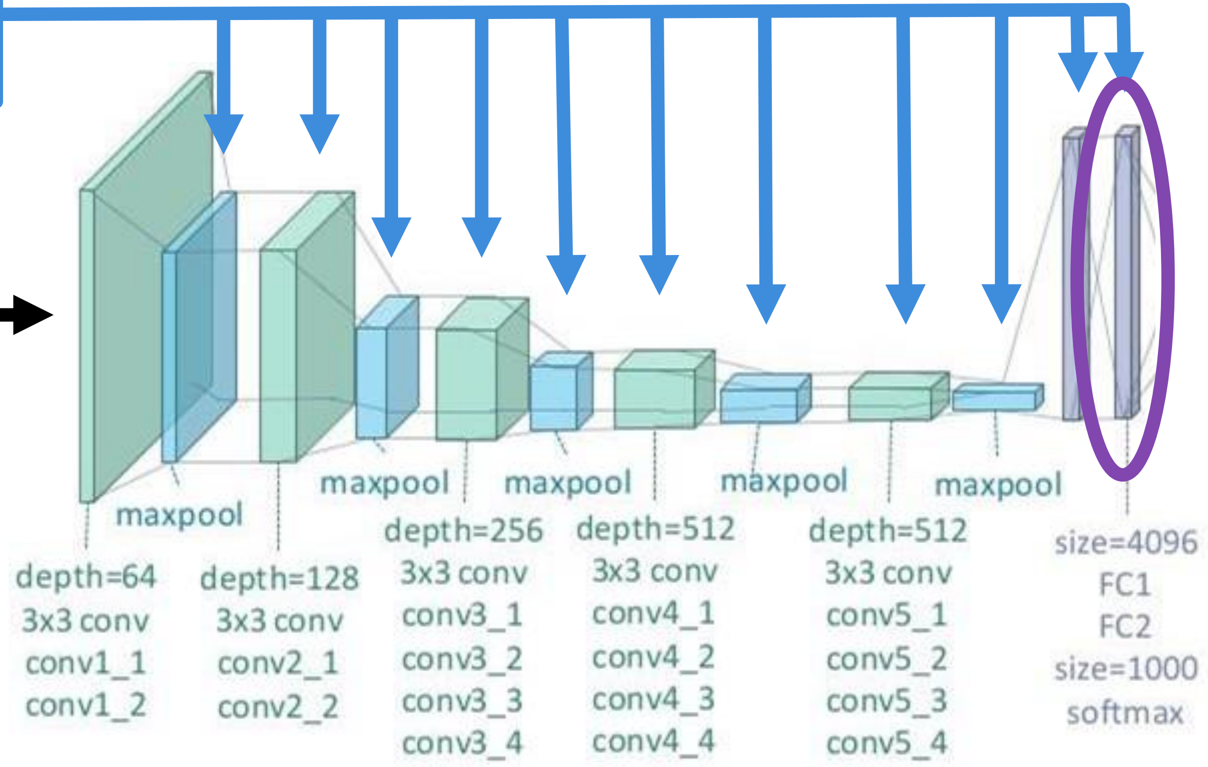


Features learned for the **Source Task**

Fine tuning

Intermediate representations

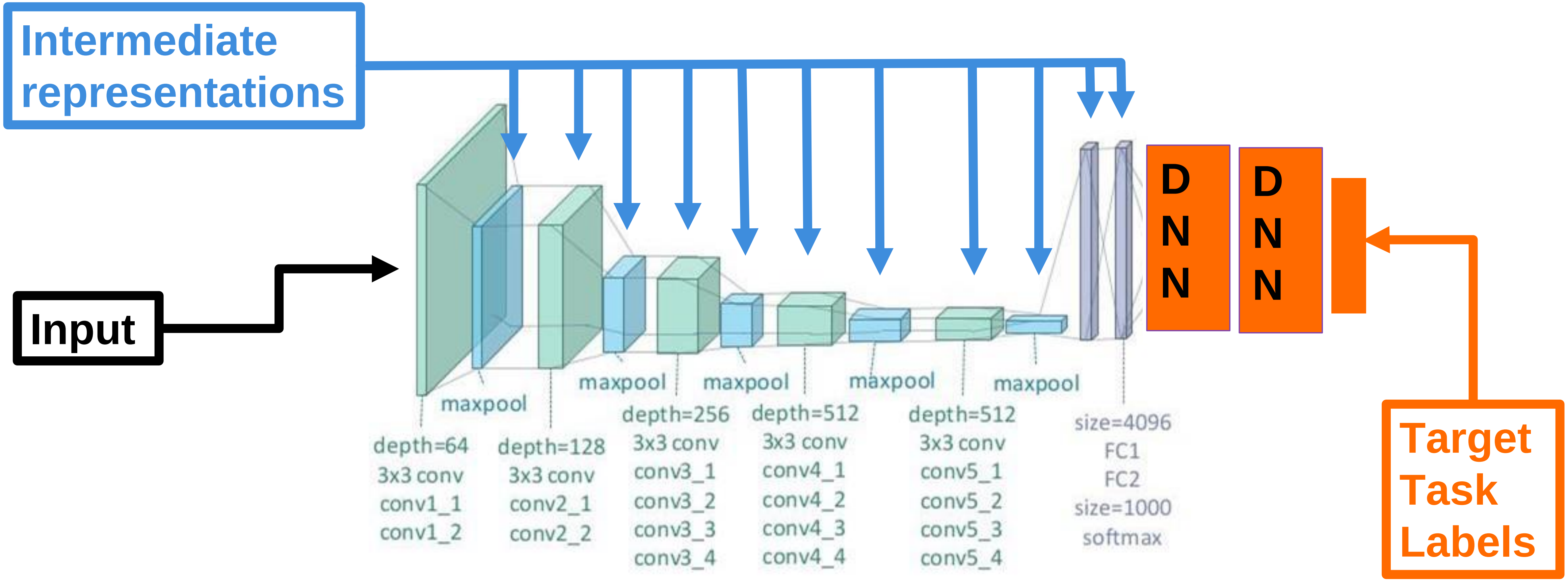
Input



Features learned for the **Source Task**

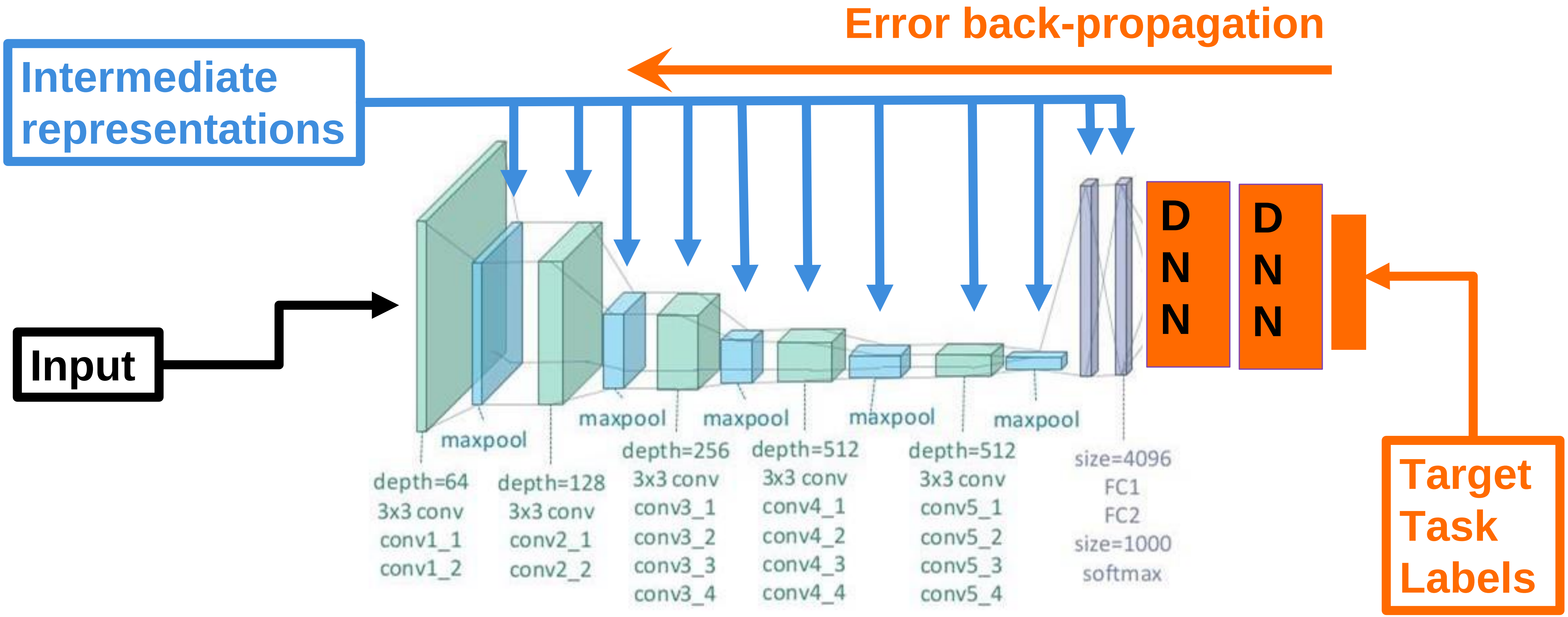
Can we make them better?

Fine tuning



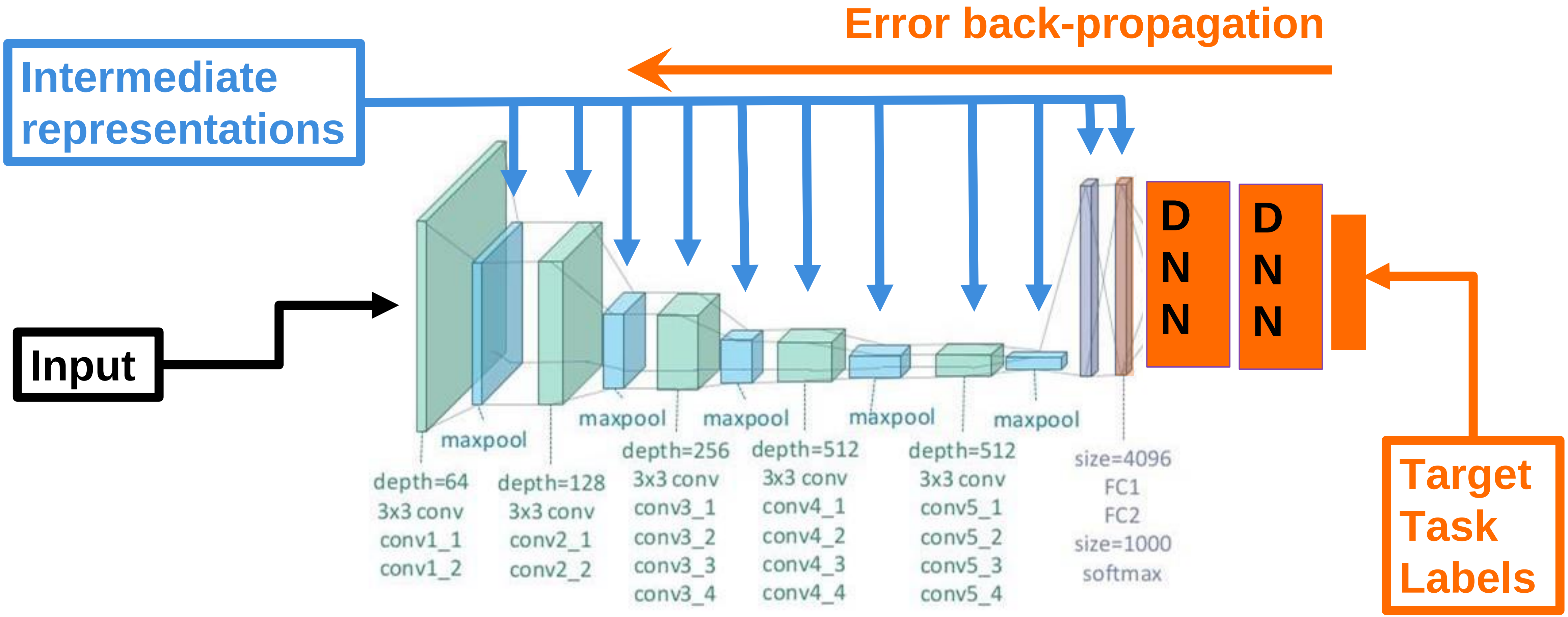
Intermediate representations
Input
maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
conv3_4
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
conv4_4
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
conv5_4
size=4096
FC1
FC2
size=1000
softmax
DNN
DNN
Target Task Labels

Fine tuning



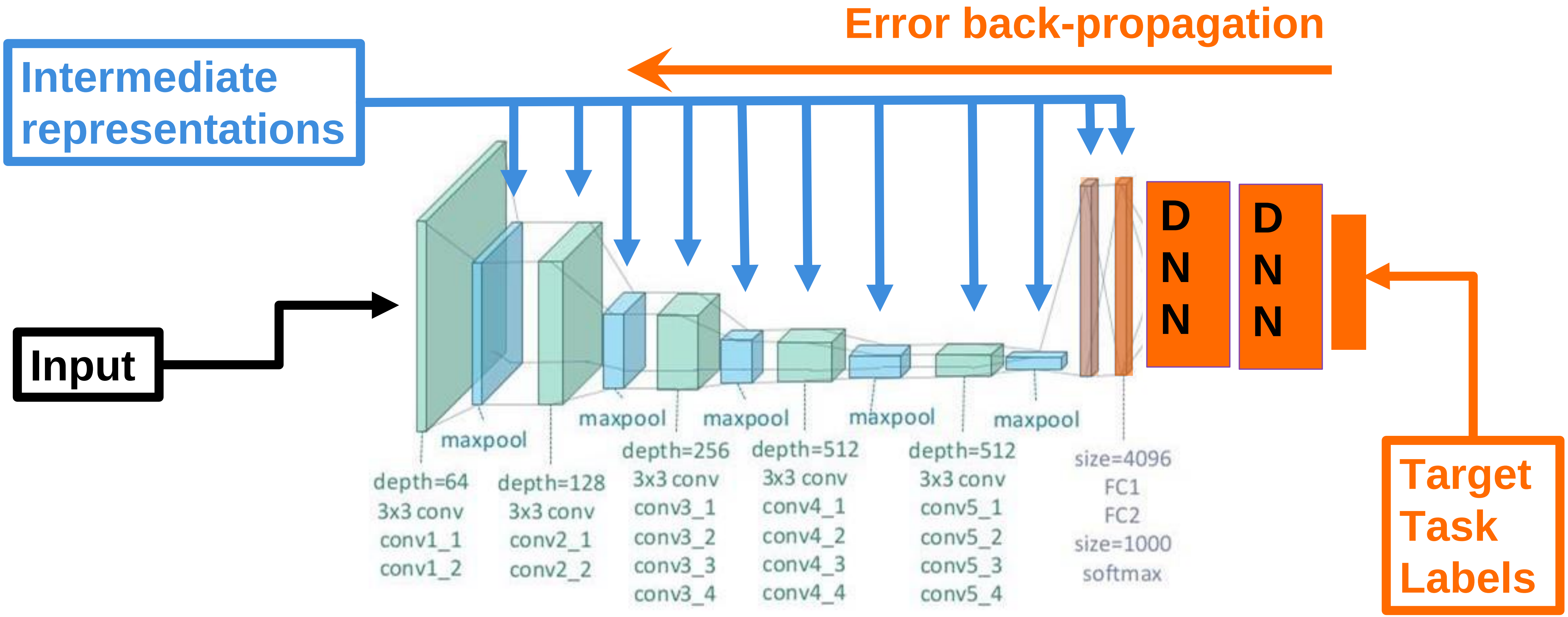
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fine tuning



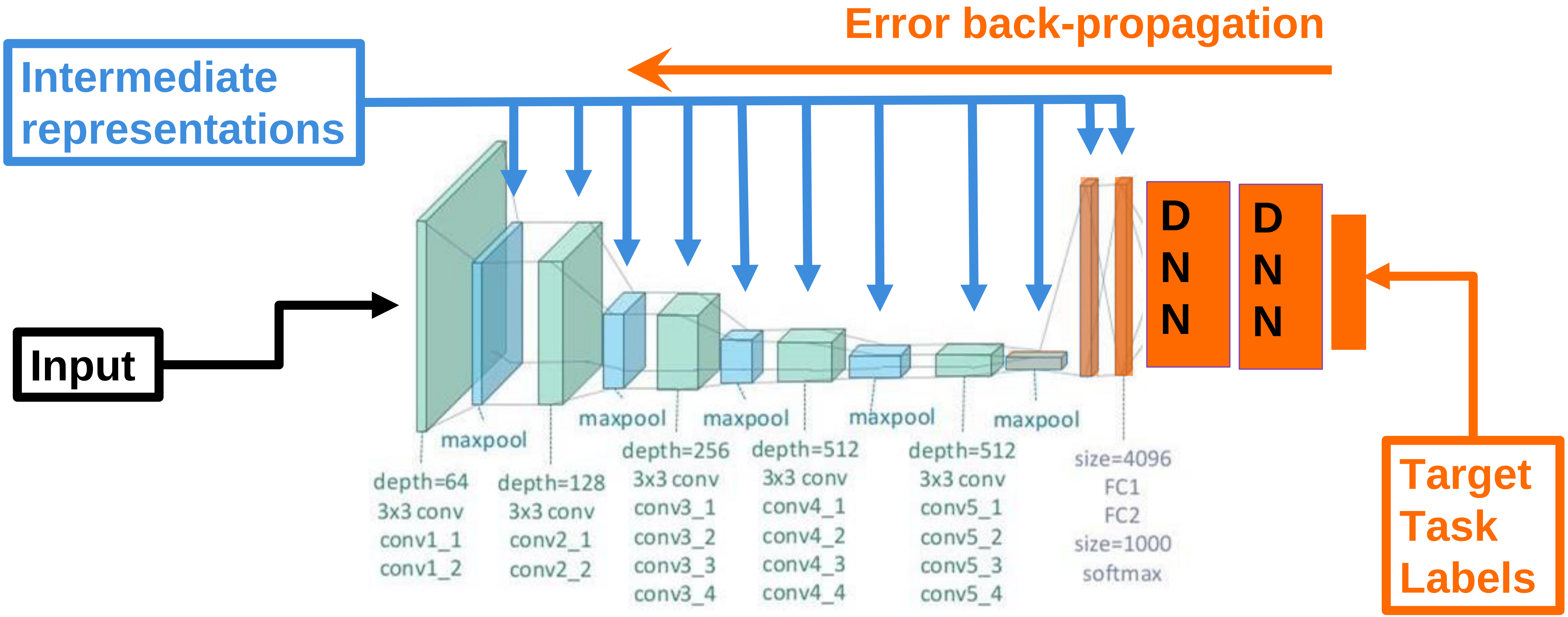
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fine tuning



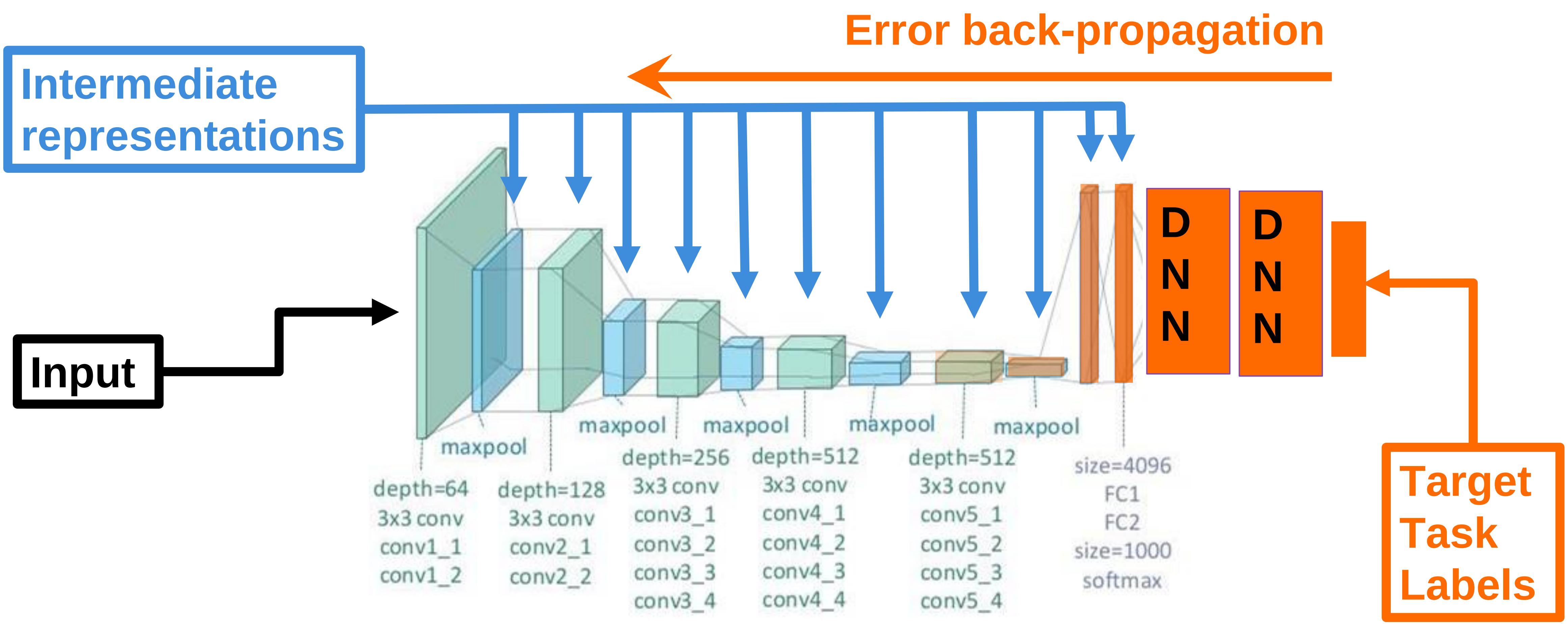
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fine tuning

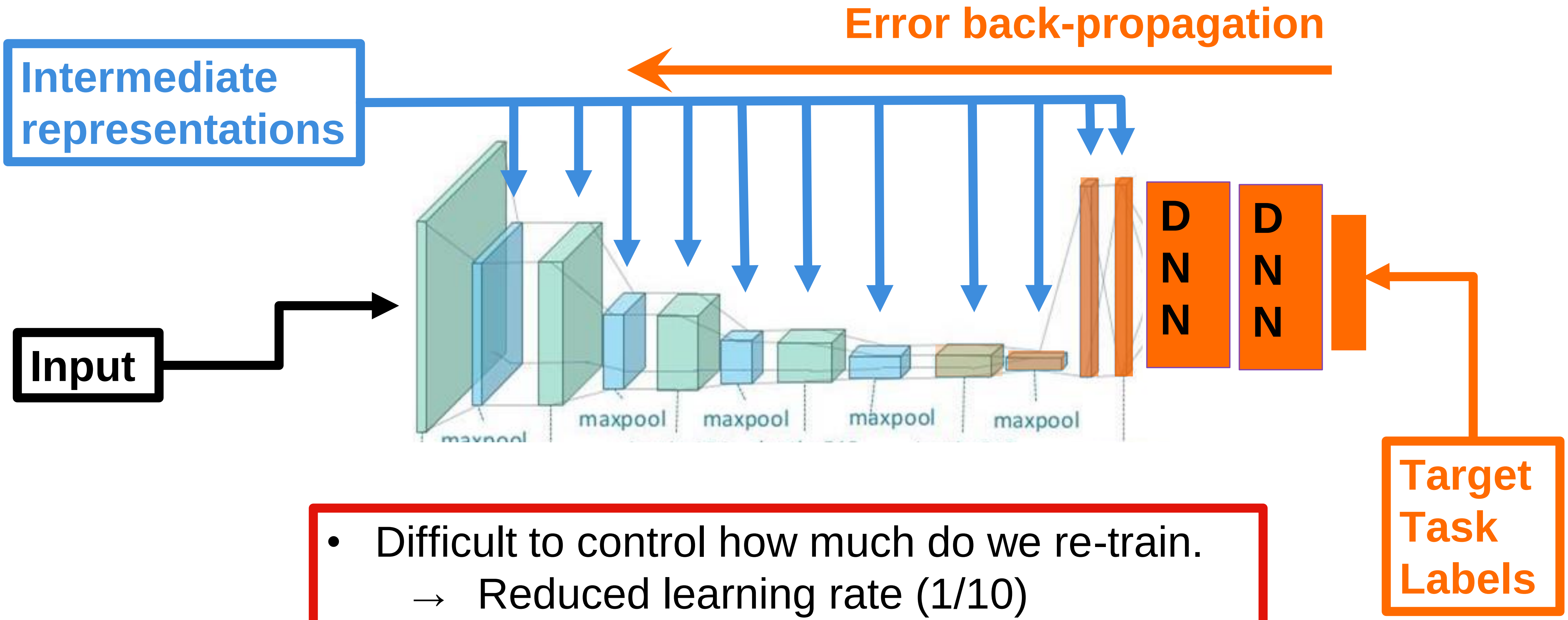


Error back-propagation
Intermediate representations
Input
maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
conv3_4
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
conv4_4
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
conv5_4
size=4096
FC1
FC2
size=1000
softmax
DNN
DNN
Target Task Labels

Fine tuning


Error back-propagation
Intermediate representations
Input
maxpool
maxpool
maxpool
maxpool
maxpool
depth=64
3x3 conv
conv1_1
conv1_2
depth=128
3x3 conv
conv2_1
conv2_2
depth=256
3x3 conv
conv3_1
conv3_2
conv3_3
conv3_4
depth=512
3x3 conv
conv4_1
conv4_2
conv4_3
conv4_4
depth=512
3x3 conv
conv5_1
conv5_2
conv5_3
conv5_4
size=4096
FC1
FC2
size=1000
softmax
DNN
DNN
Target Task Labels

Fine tuning

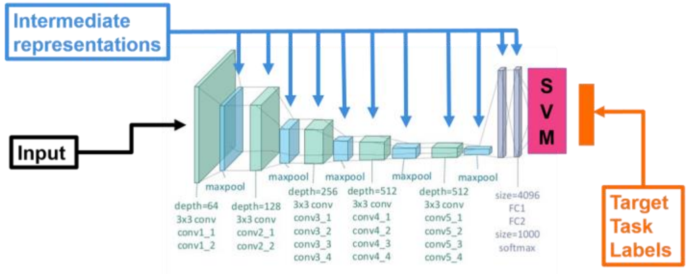


- Difficult to control how much do we re-train.
  - → Reduced learning rate (1/10)
  - → Early stopping
  - → Alternate source/target sampling

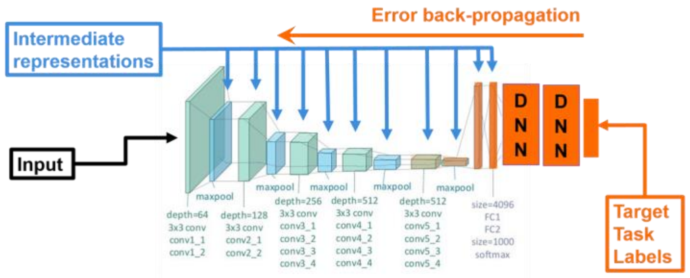
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Simple solutions

- DNN last layer features + SVM
  (Feature extraction)
  We need: **Similar task and domain**



- Add one or several NN layers +
  Fine-tuning pre-trained layers

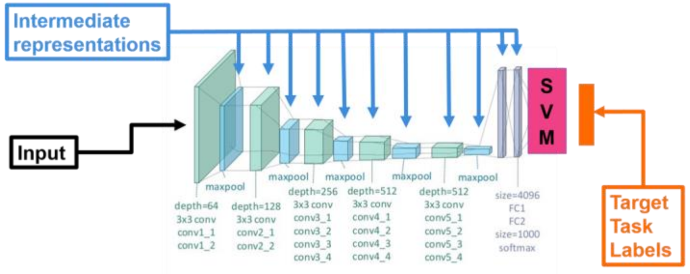


Simple solutions

- DNN last layer features + SVM

    (Feature extraction)

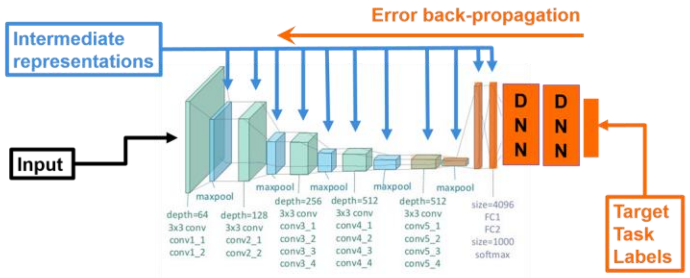
    We need: **Similar task and domain**



- Add one or several NN layers +

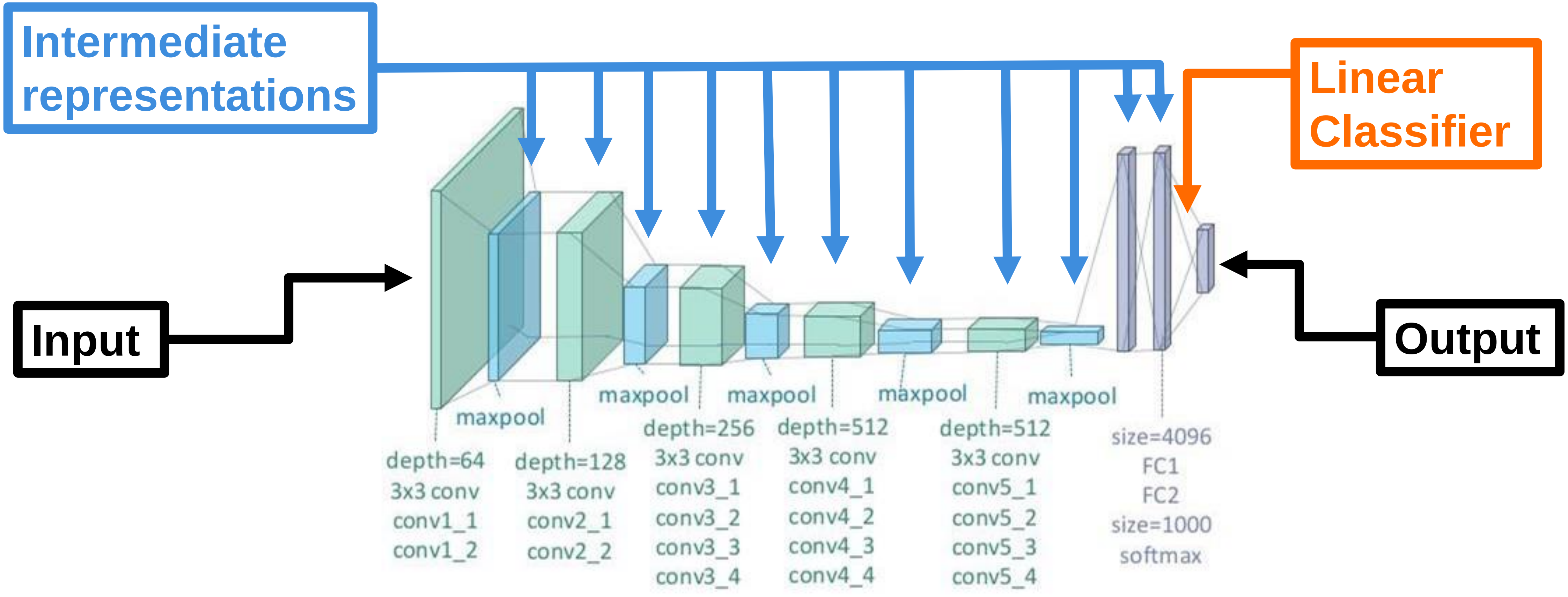
    Fine-tuning pre-trained layers

    We need: **Enough data**



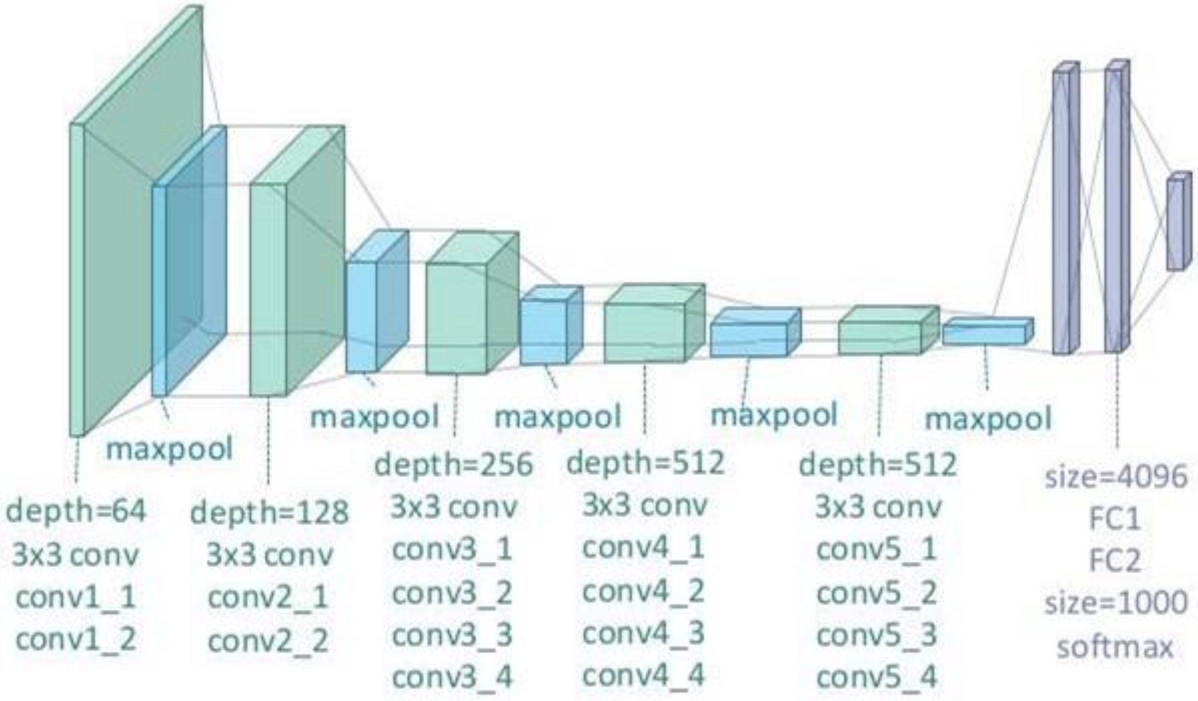
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Beyond the last layer

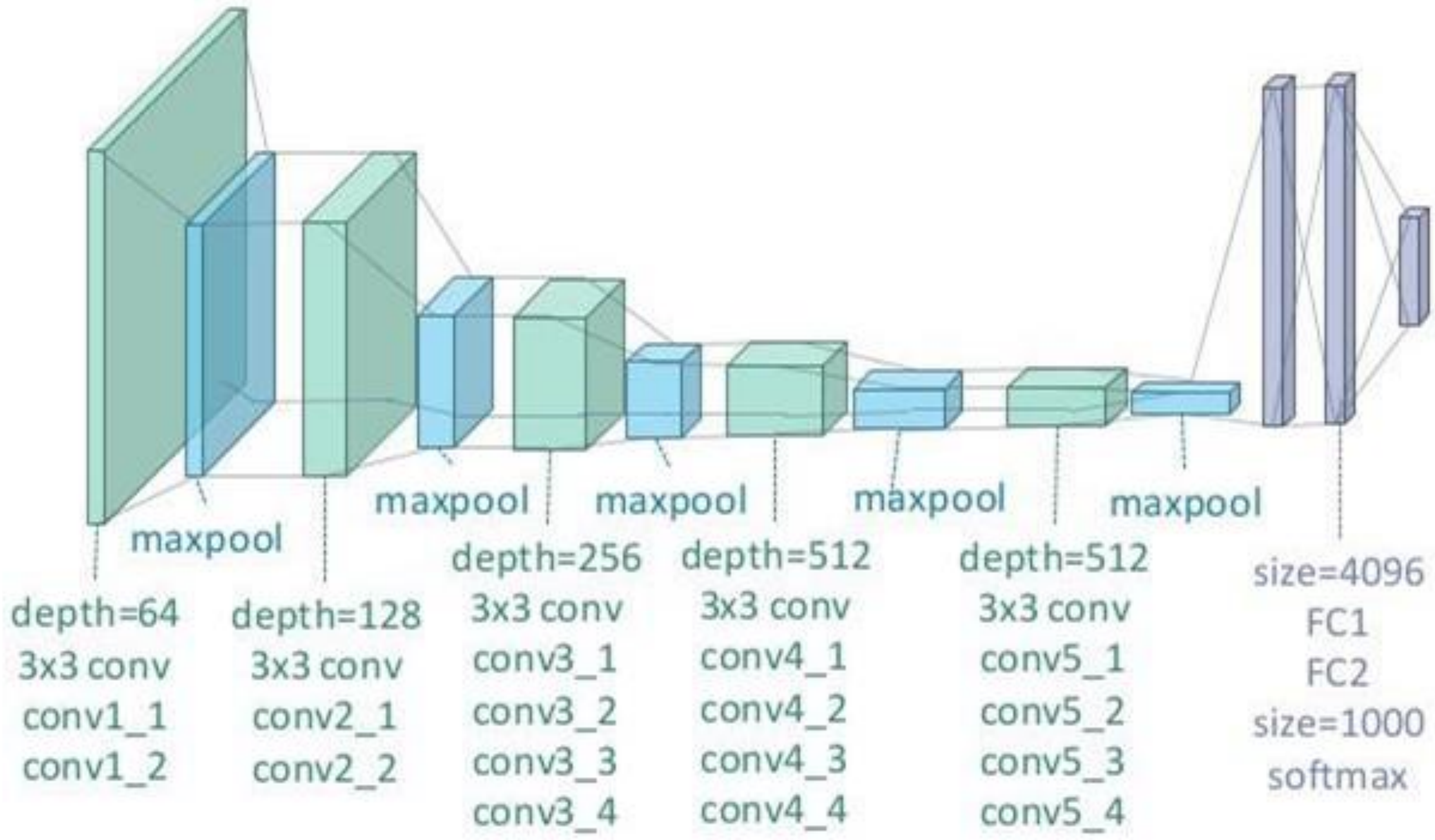


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

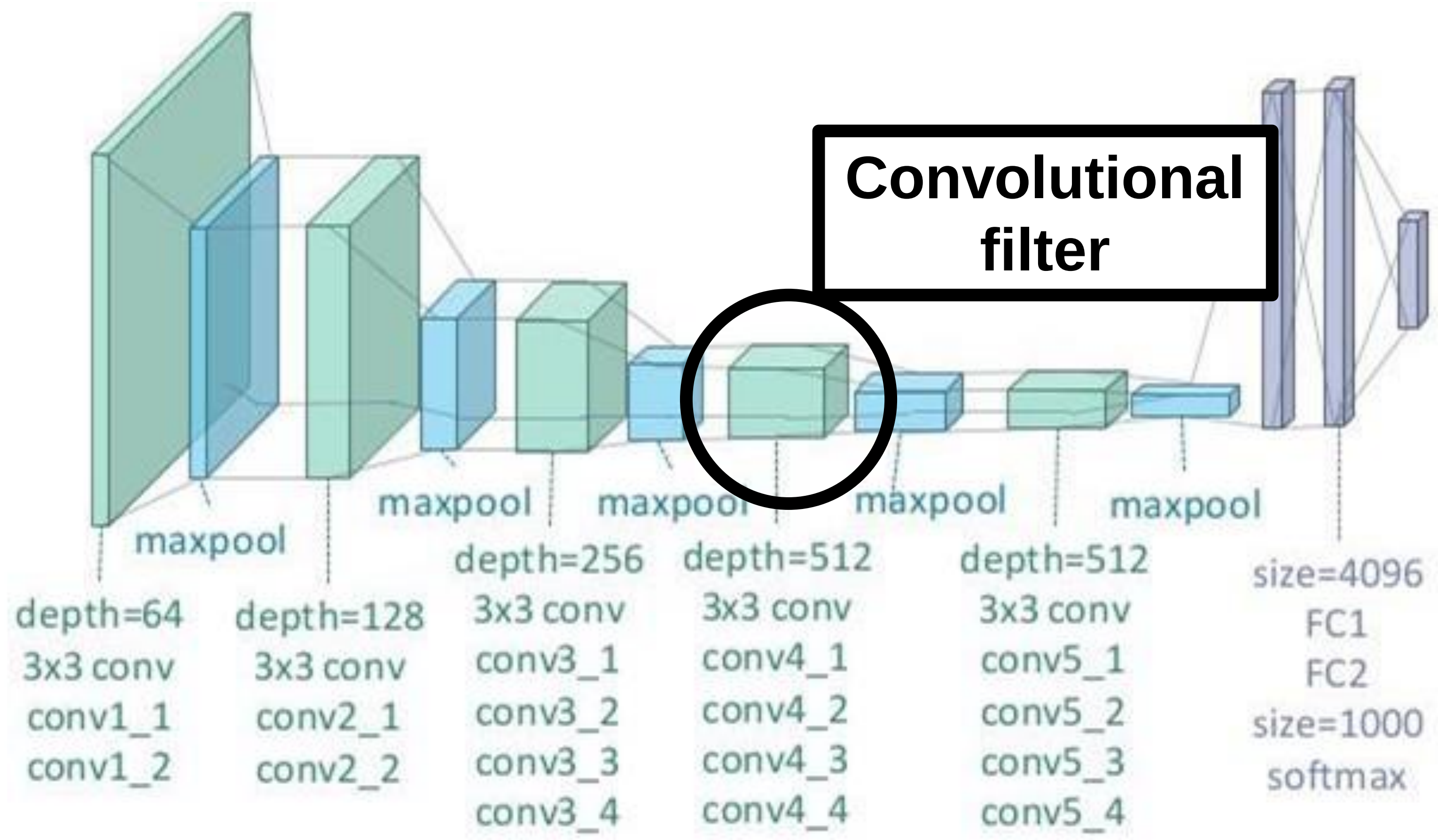
Knowledge inside DNN



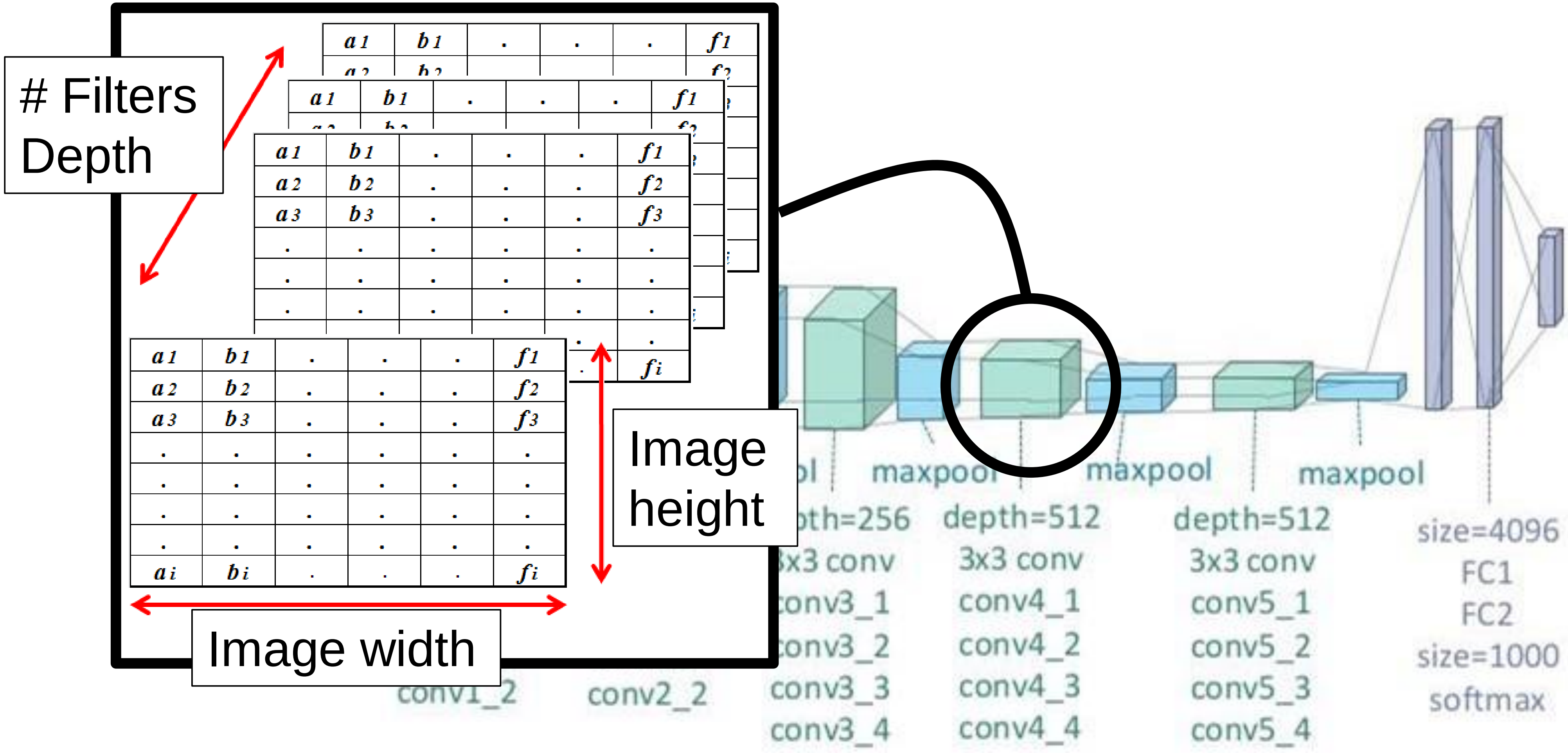
Knowledge inside DNN



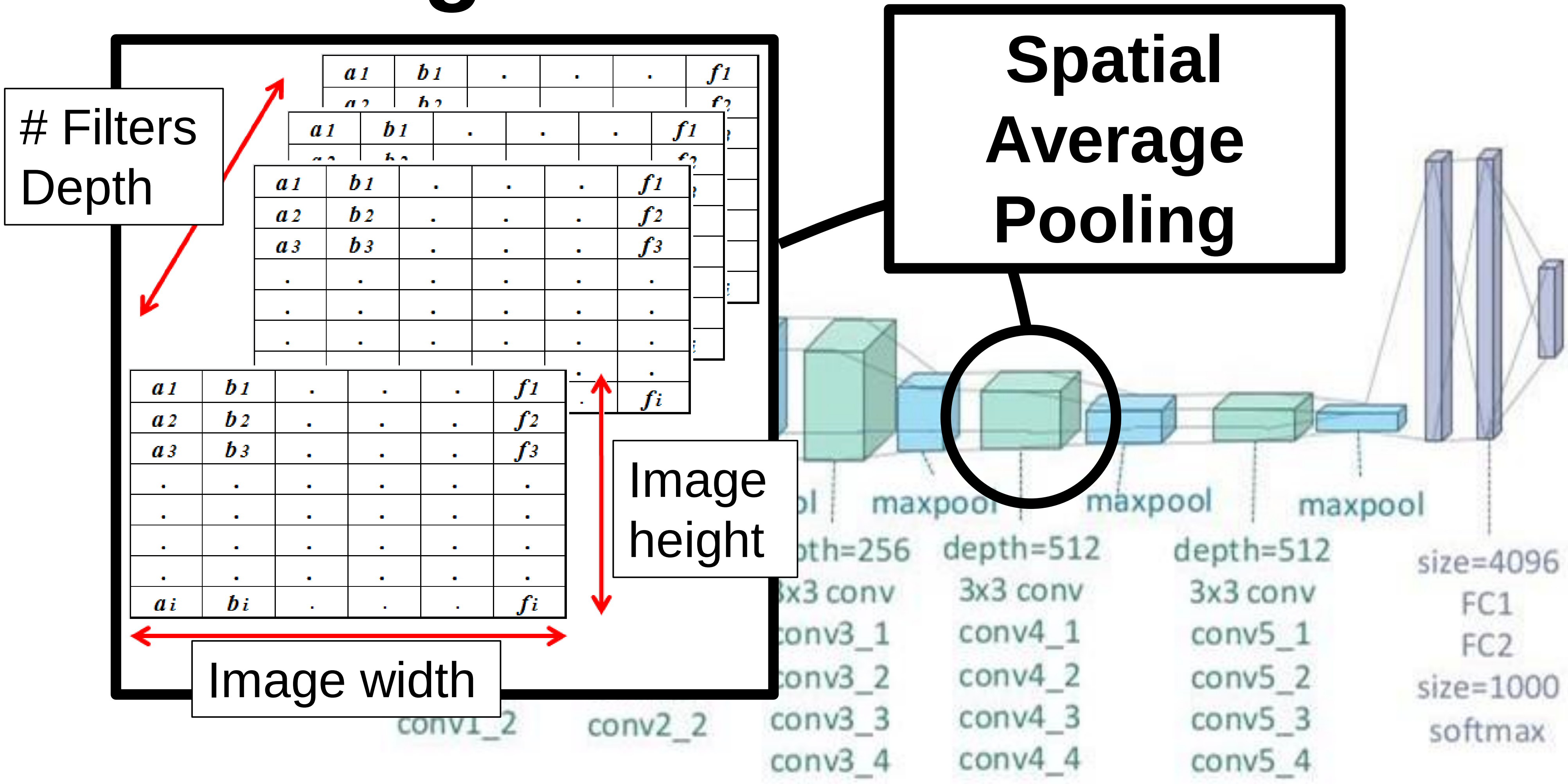
Knowledge inside DNN



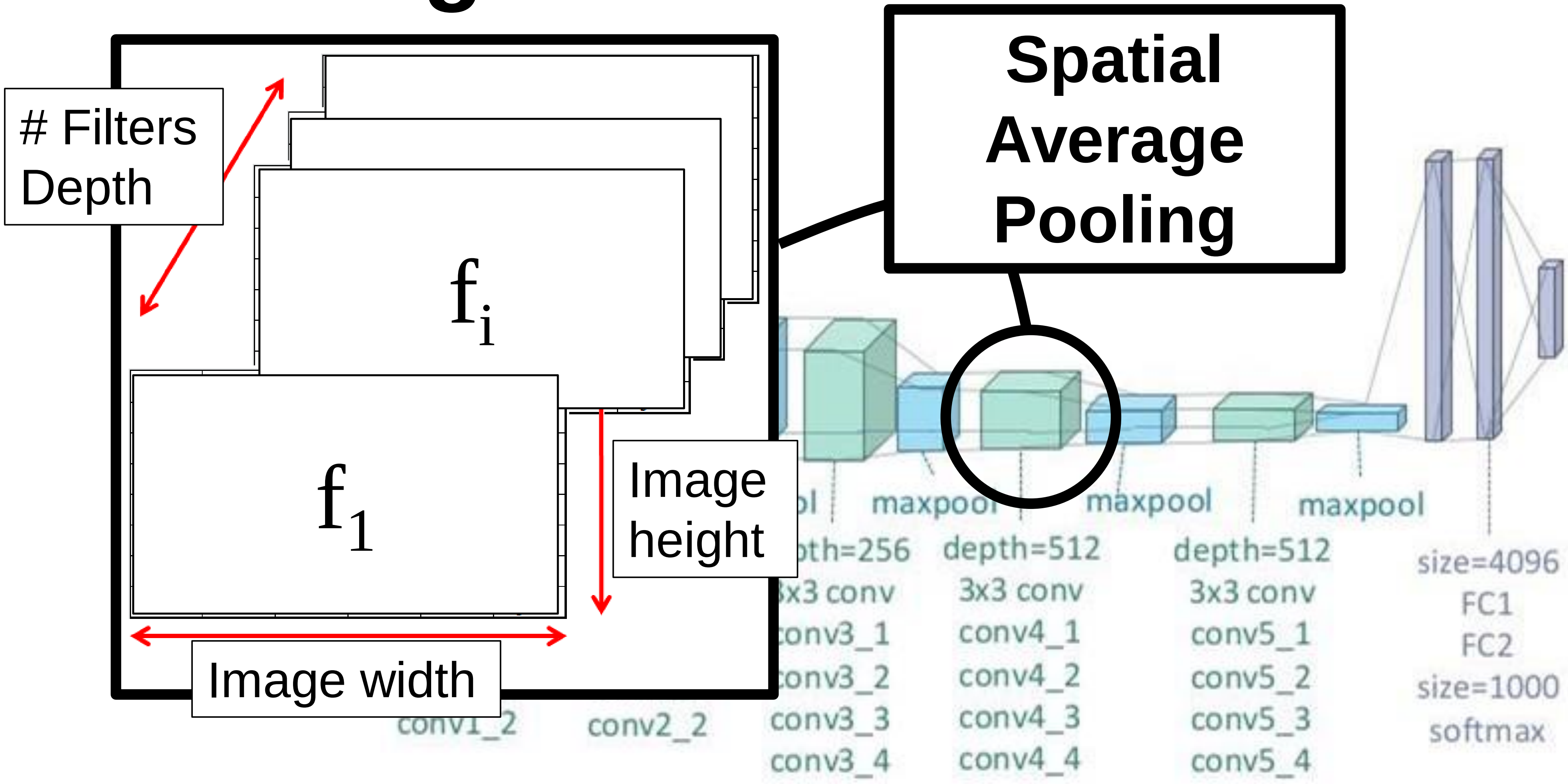
Knowledge inside DNN



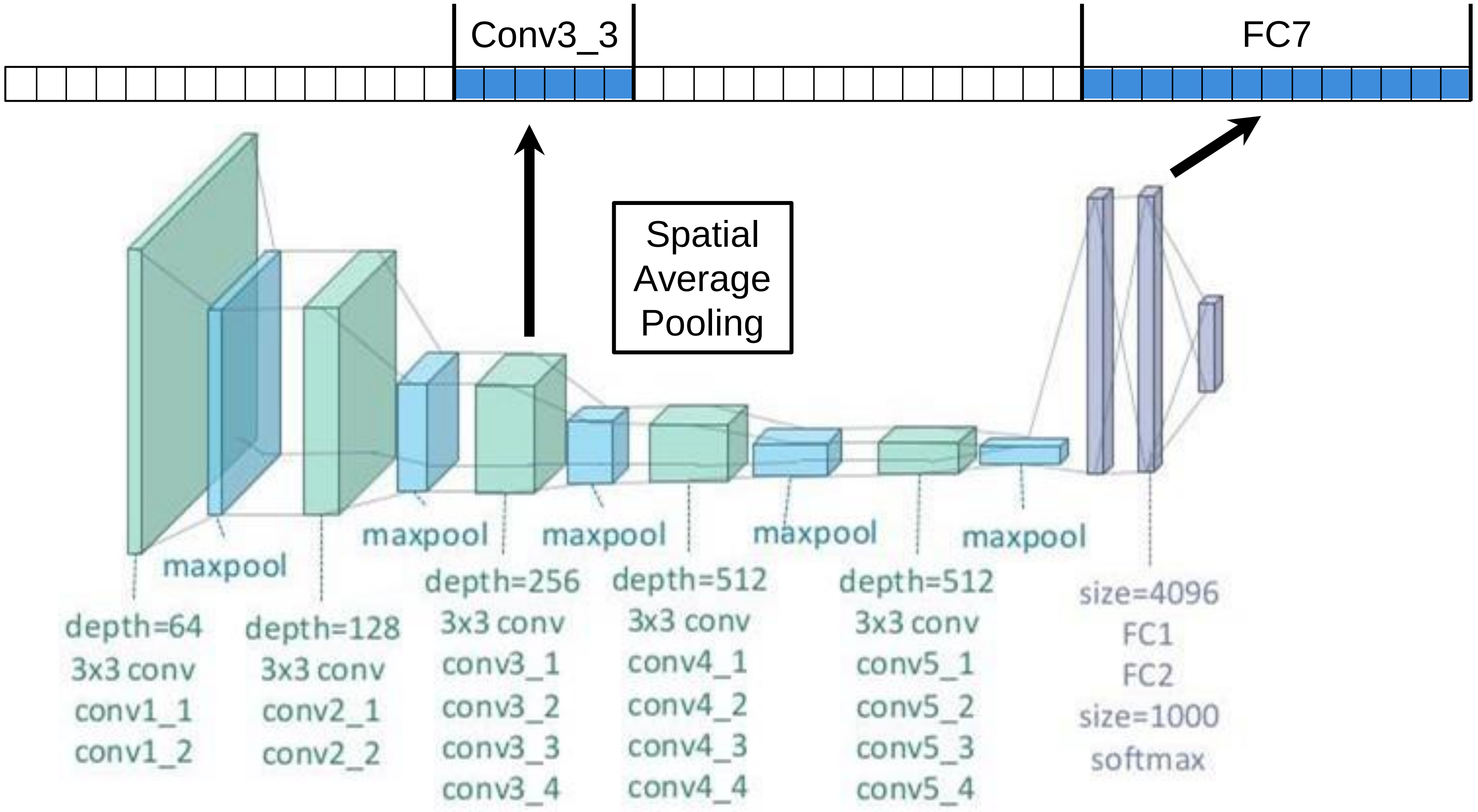
Knowledge inside DNN



Knowledge inside DNN

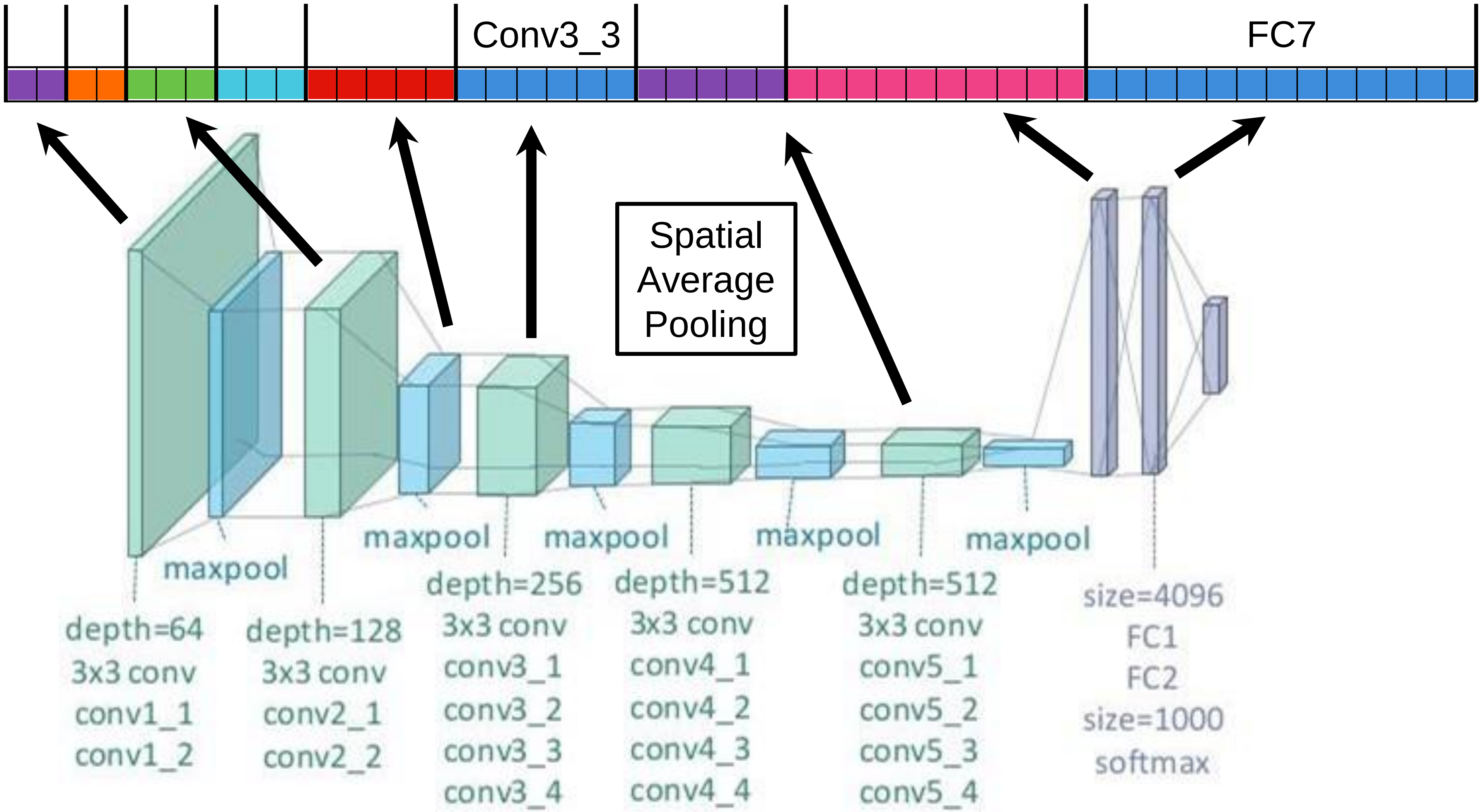


Knowledge inside DNN

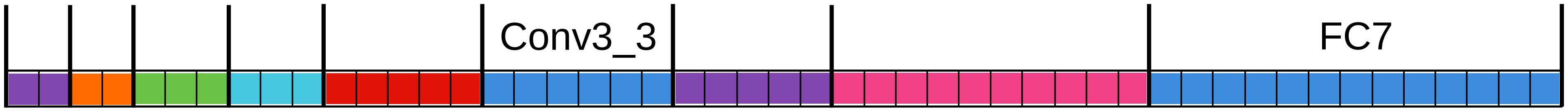


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

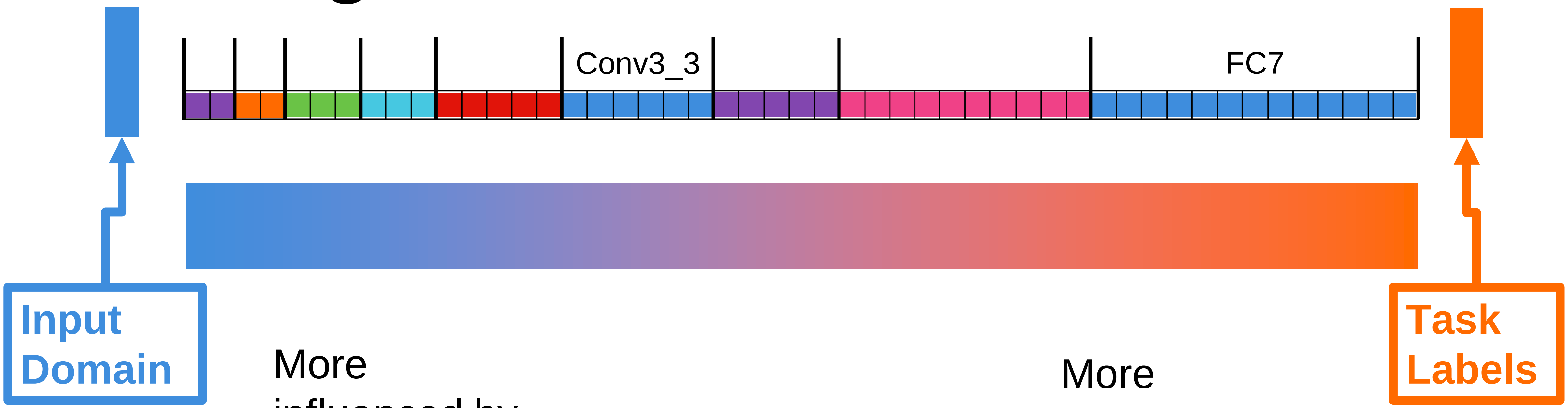
Knowledge inside DNN



Knowledge inside DNN



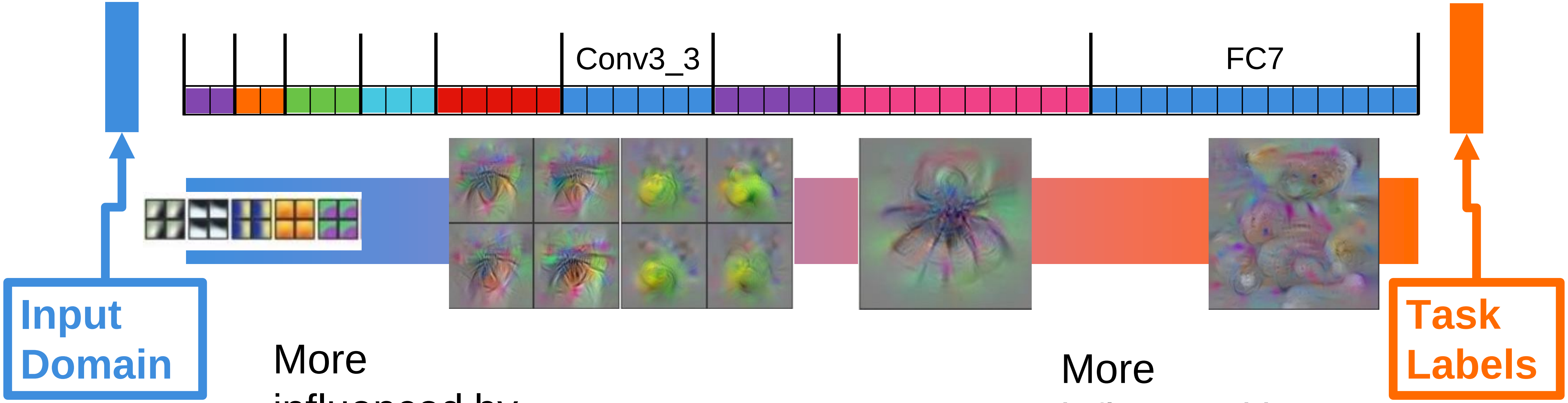
Knowledge **inside** DNN



More influenced by **domain**

More influenced by **task**

Knowledge inside DNN



More influenced by **domain**

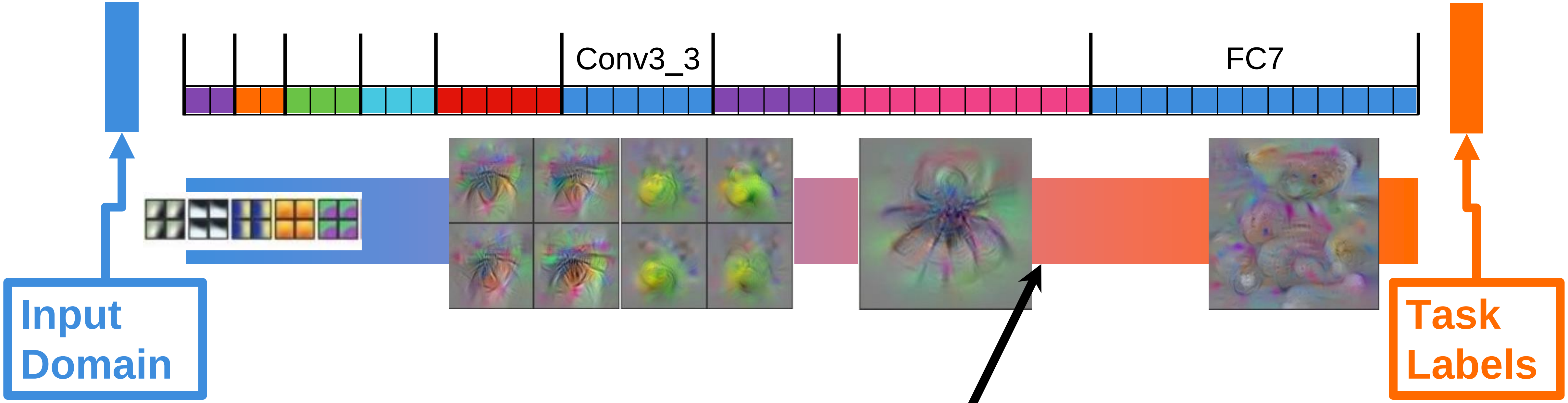
Simple features

More influenced by **task**

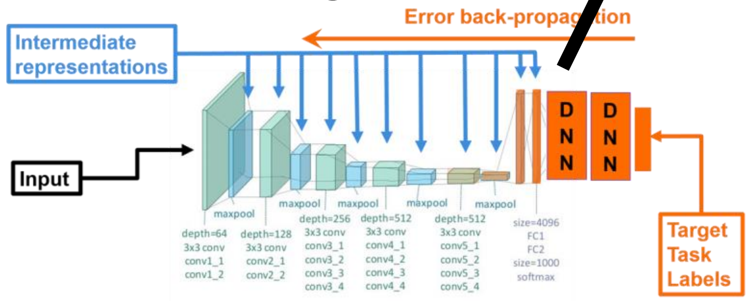
Complex features

Visualizations from: Yosinski, Jason, et al. "Understanding neural networks through deep visualization." *arXiv preprint arXiv:1506.06579* (2015).

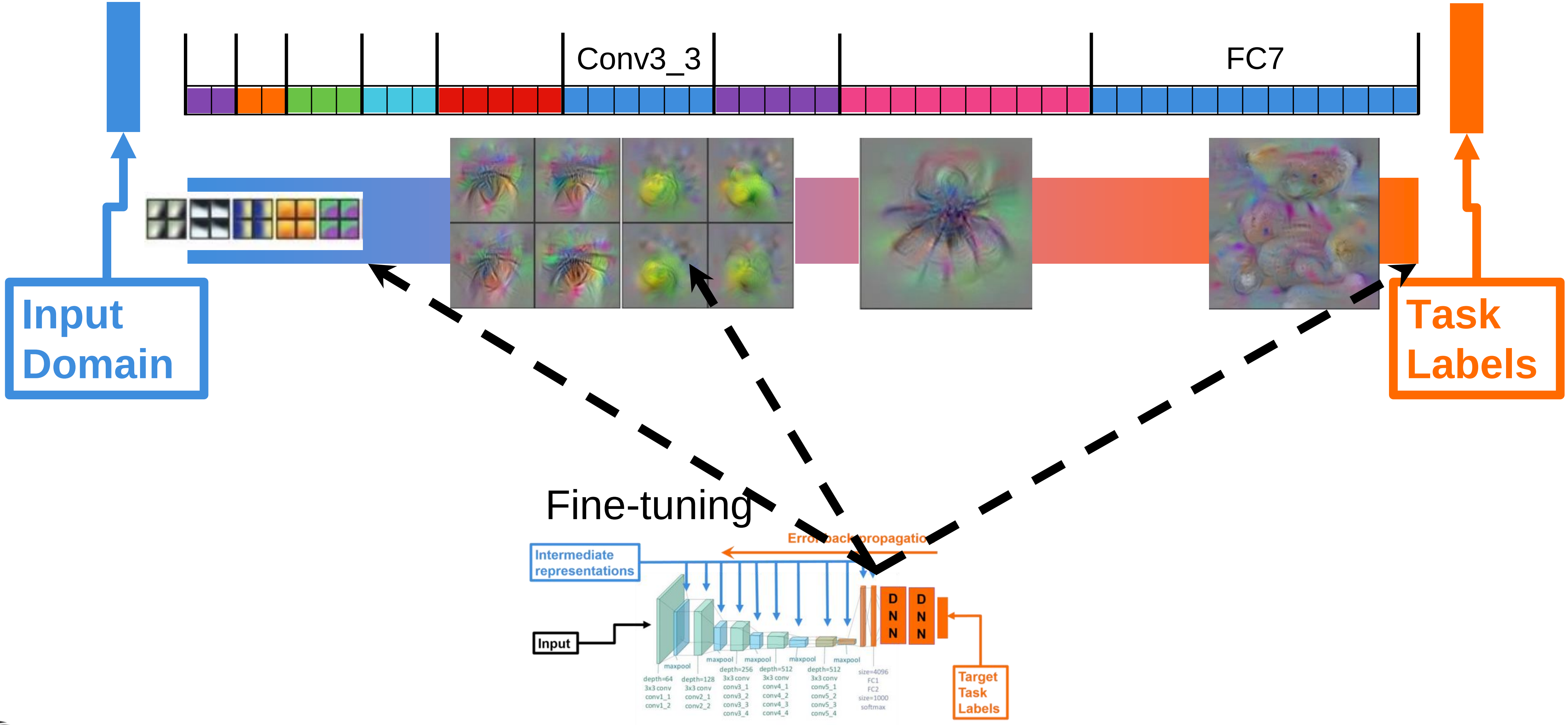
Fine tuning inside DNN?



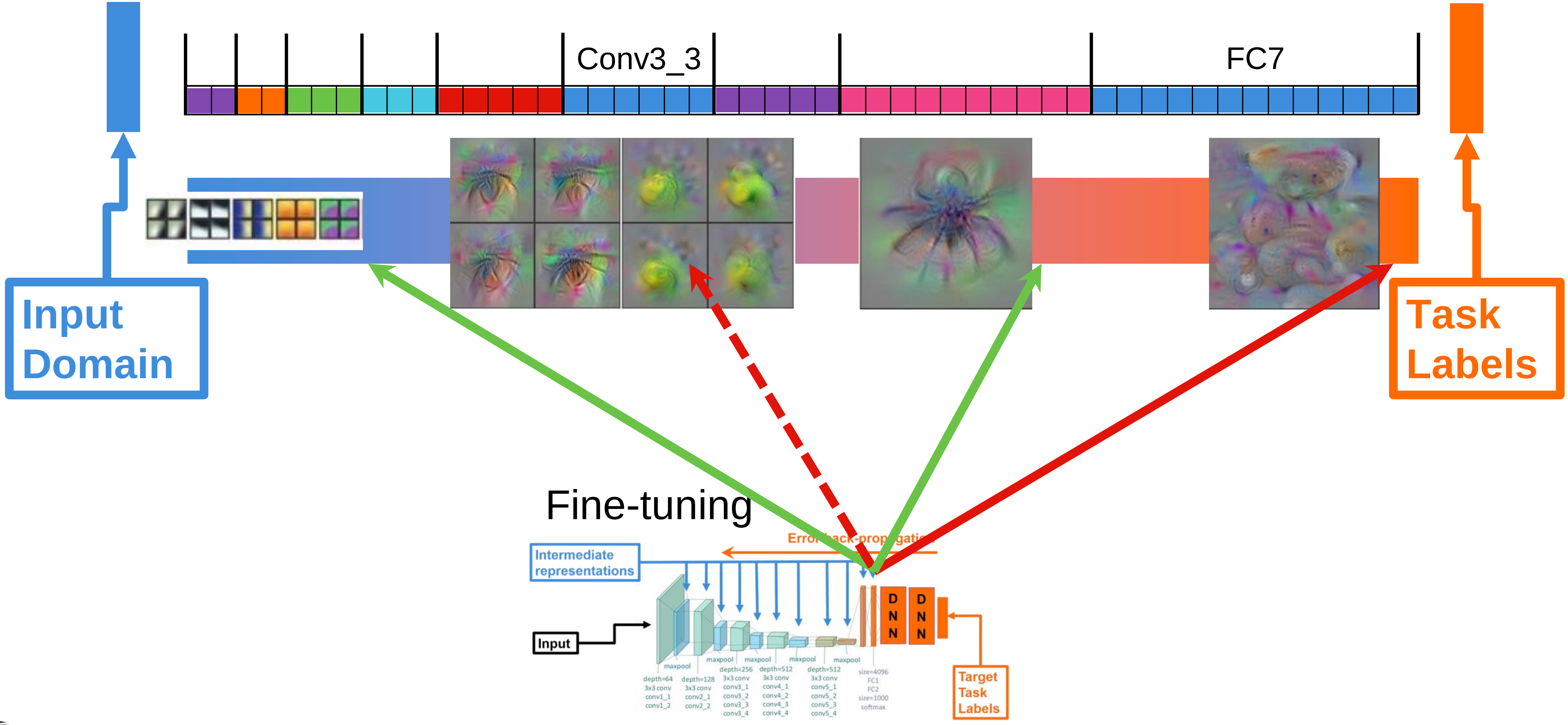
Fine-tuning



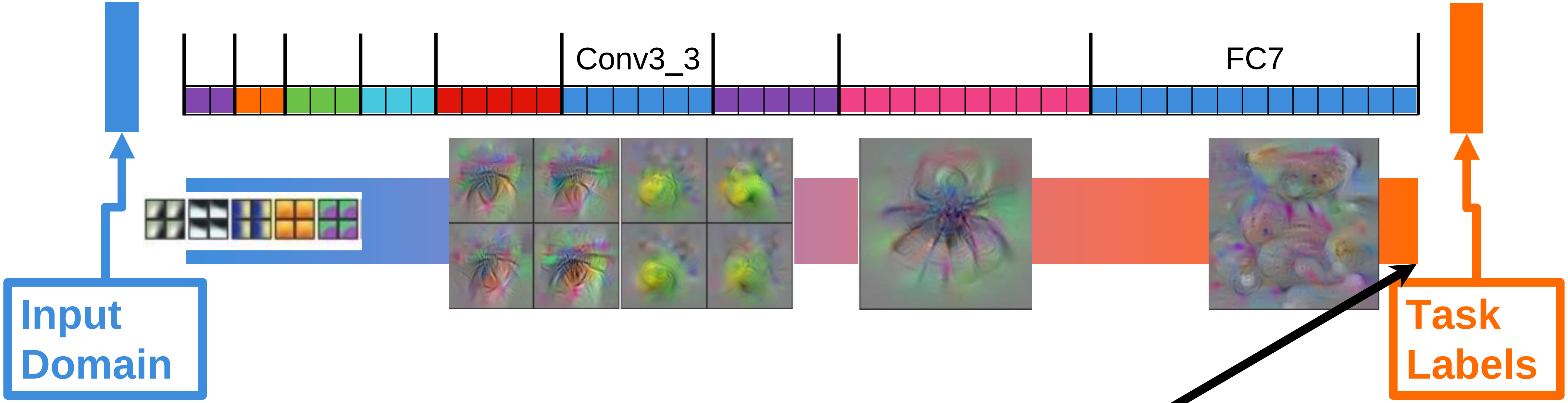
Fine tuning inside DNN?



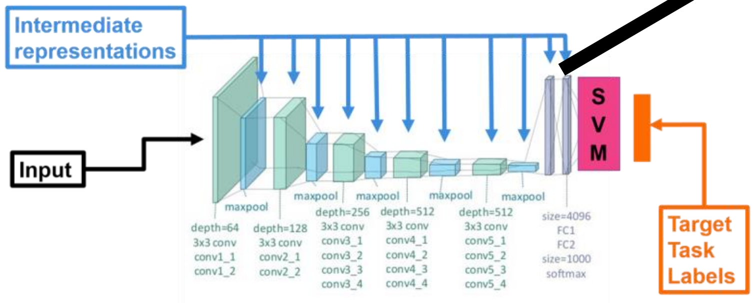
Fine tuning inside DNN?



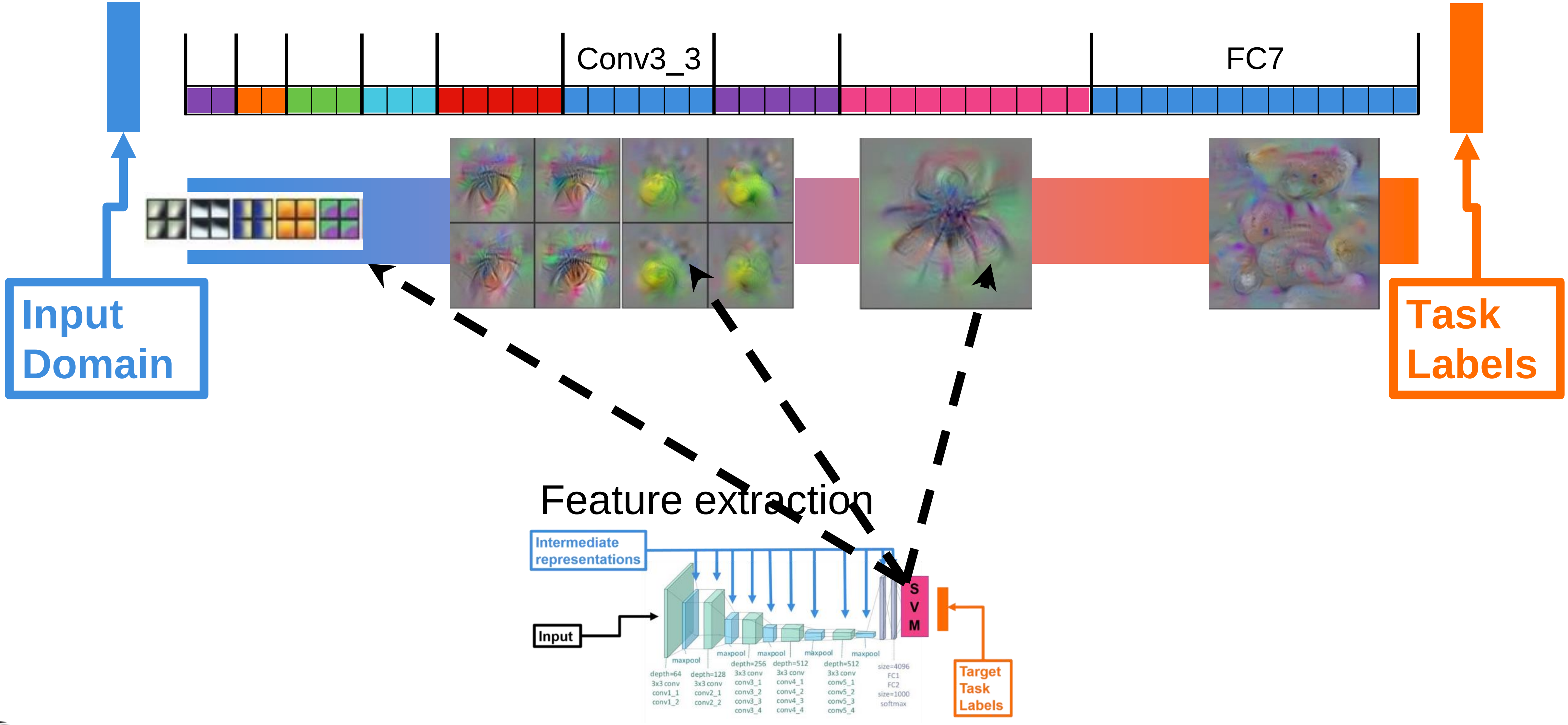
Feature extraction inside DNN?



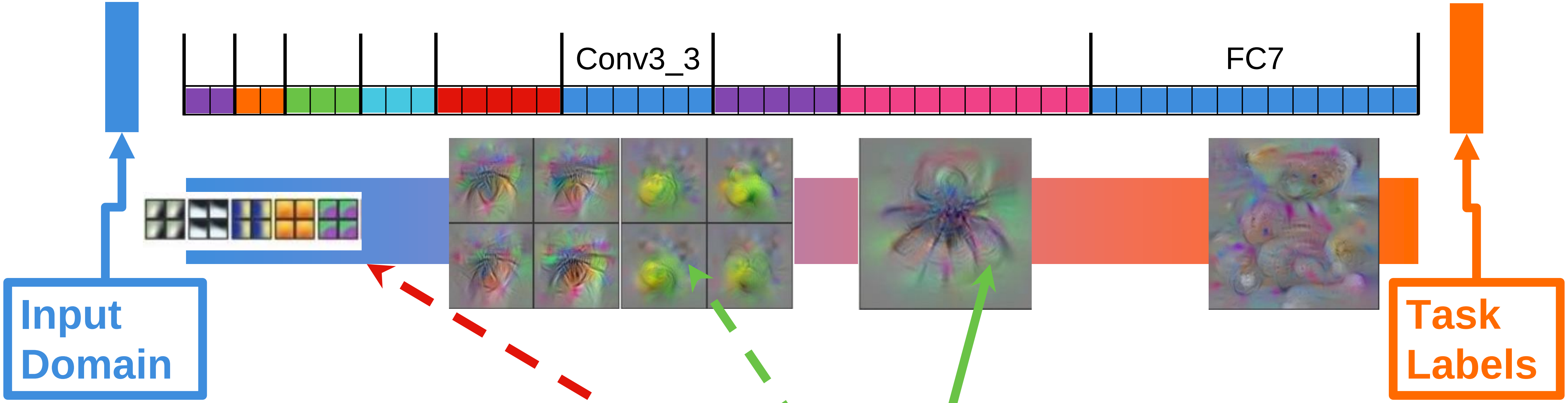
Feature extraction



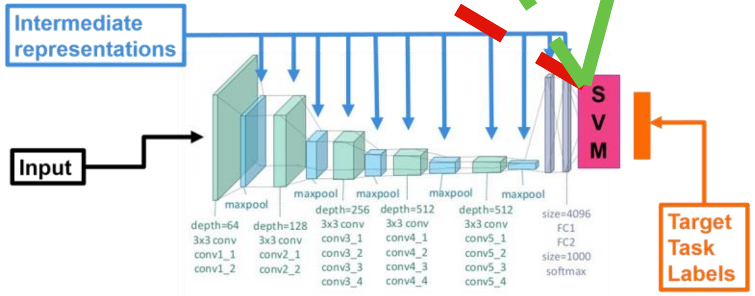
Feature extraction inside DNN?



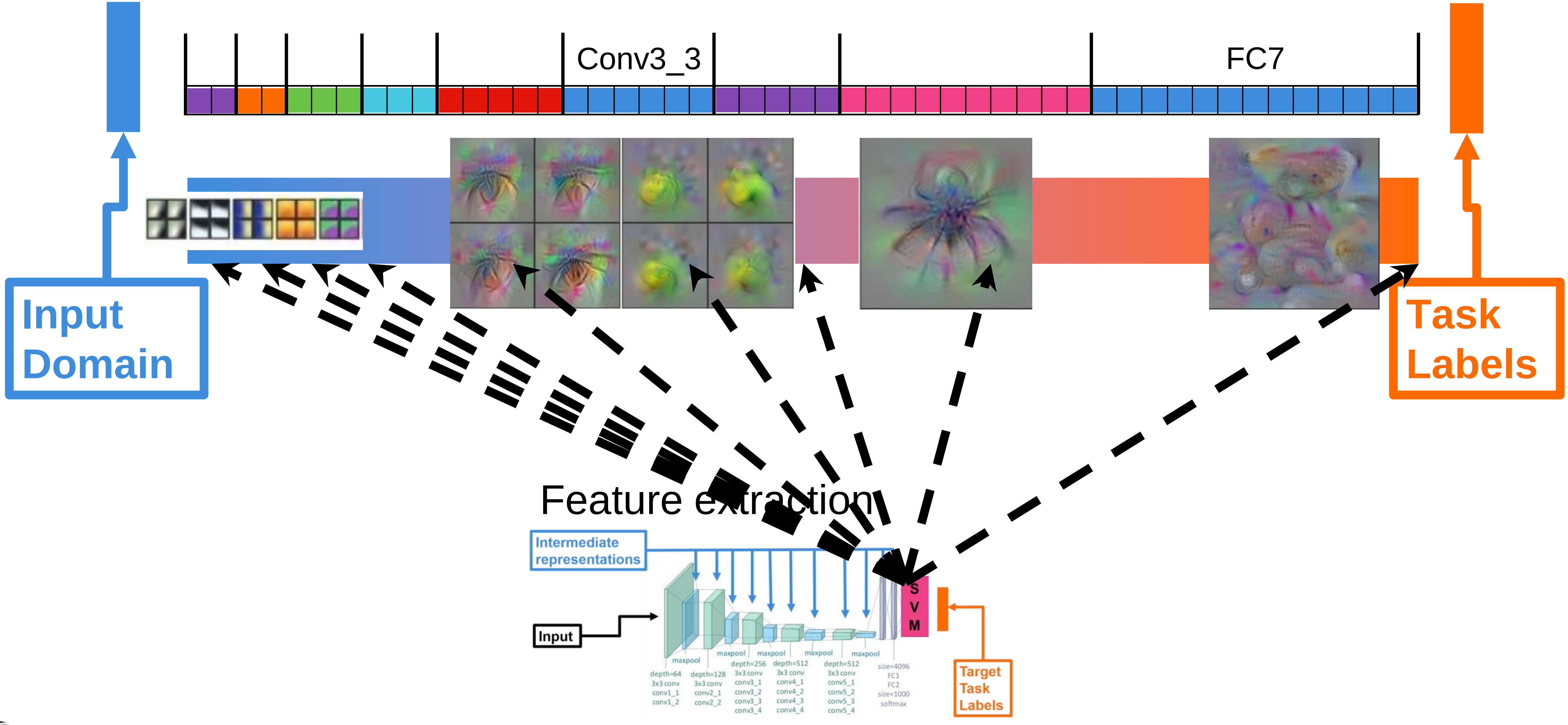
Feature extraction inside DNN?



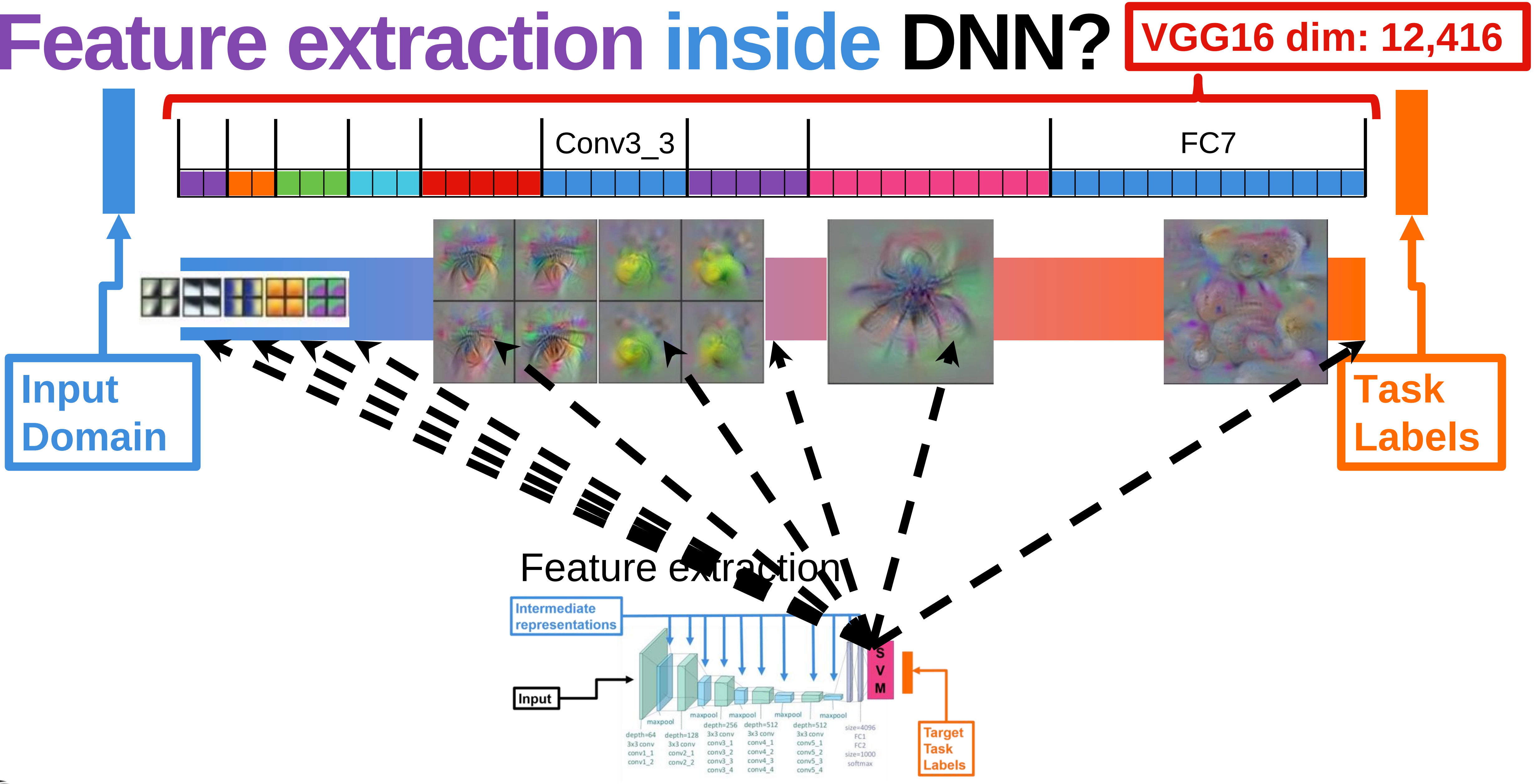
Feature extraction



Feature extraction inside DNN?

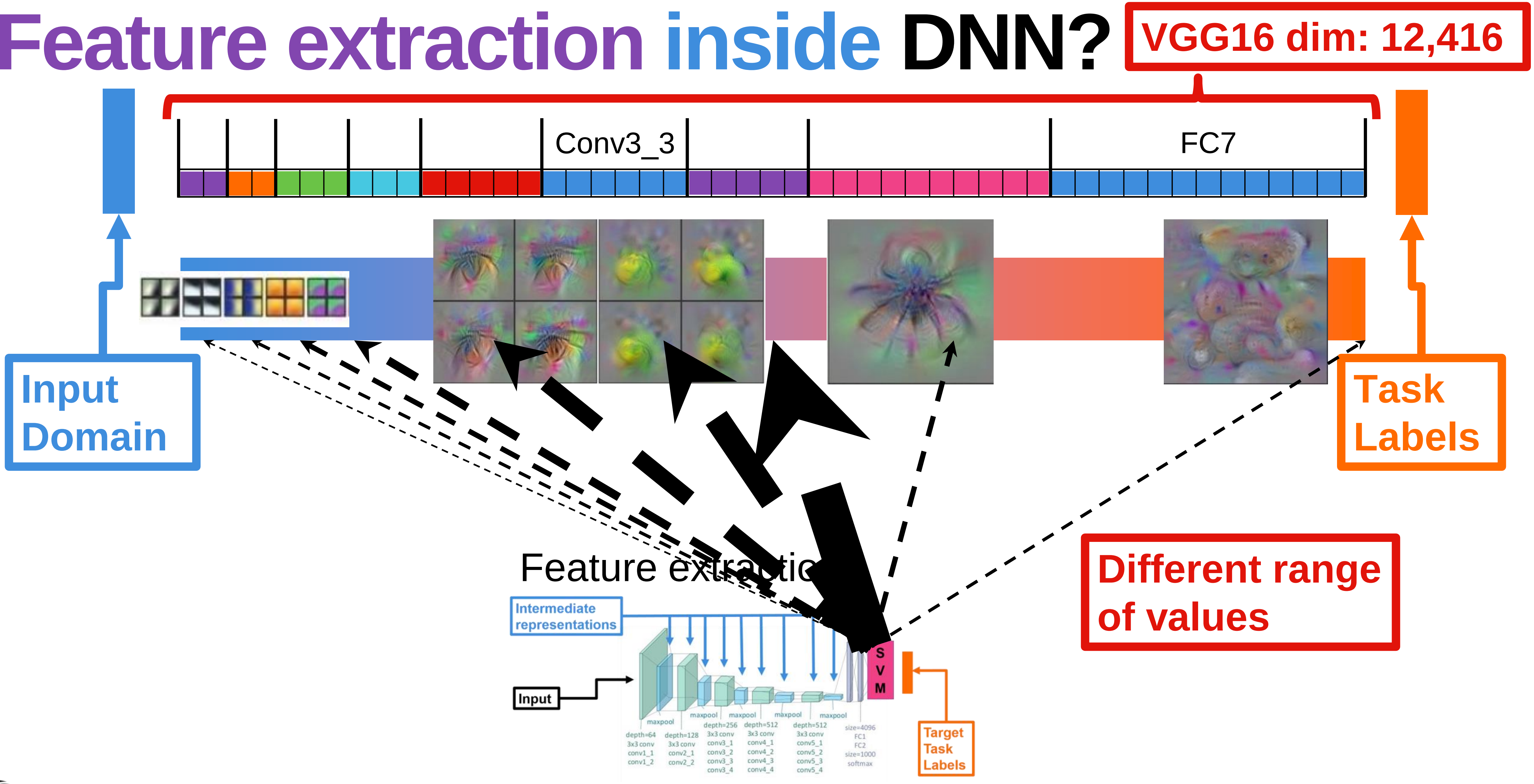


Feature extraction inside DNN?

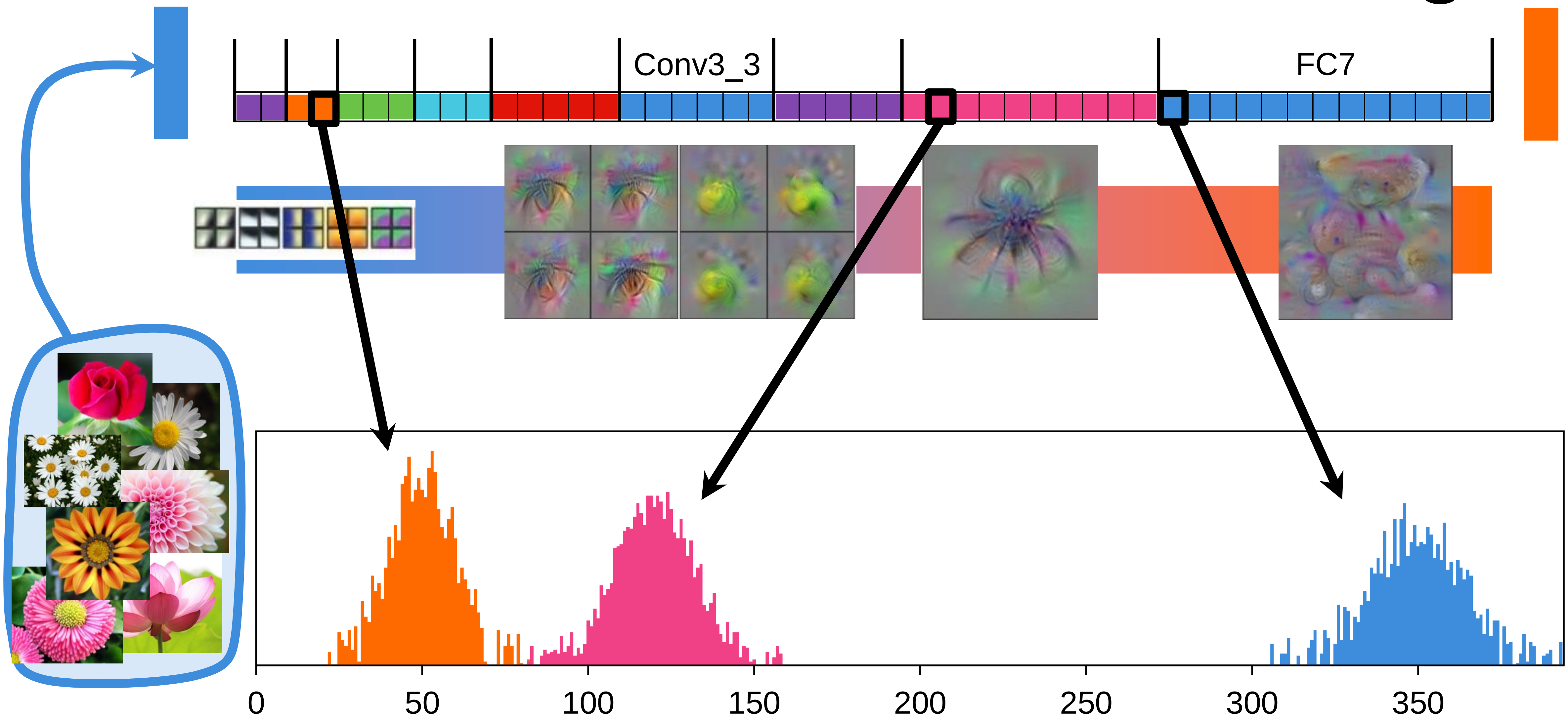


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Feature extraction inside DNN?

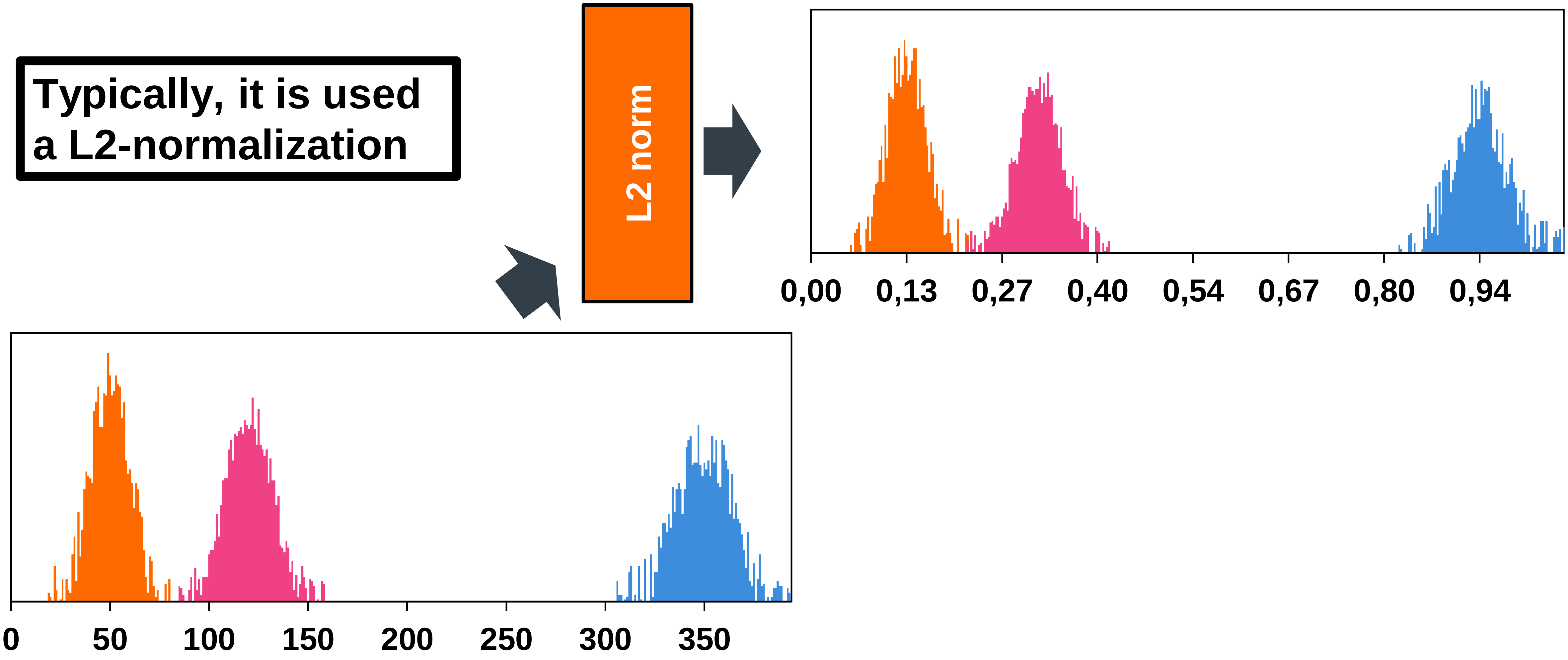


Feature behavior in Transfer Learning



Standardization: features in same range

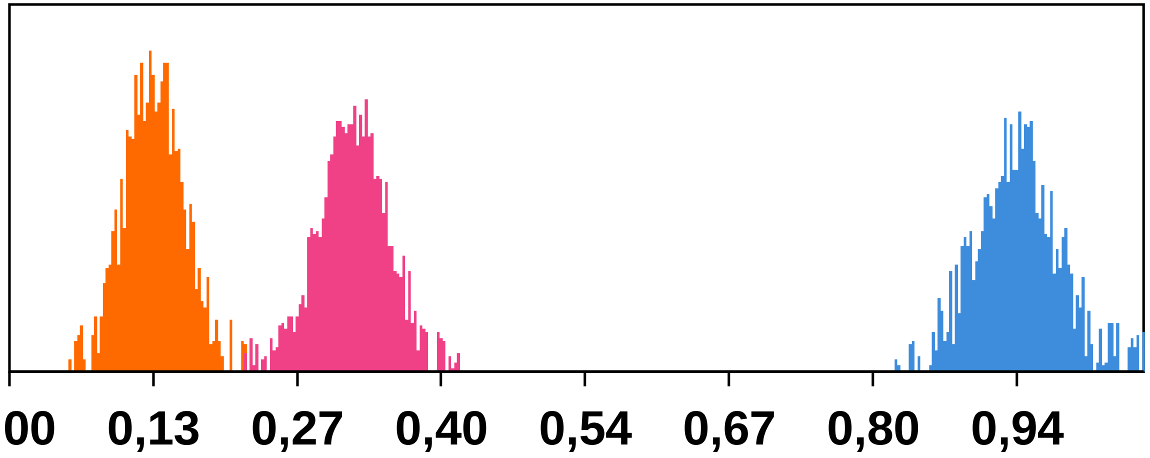
Typically, it is used a L2-normalization

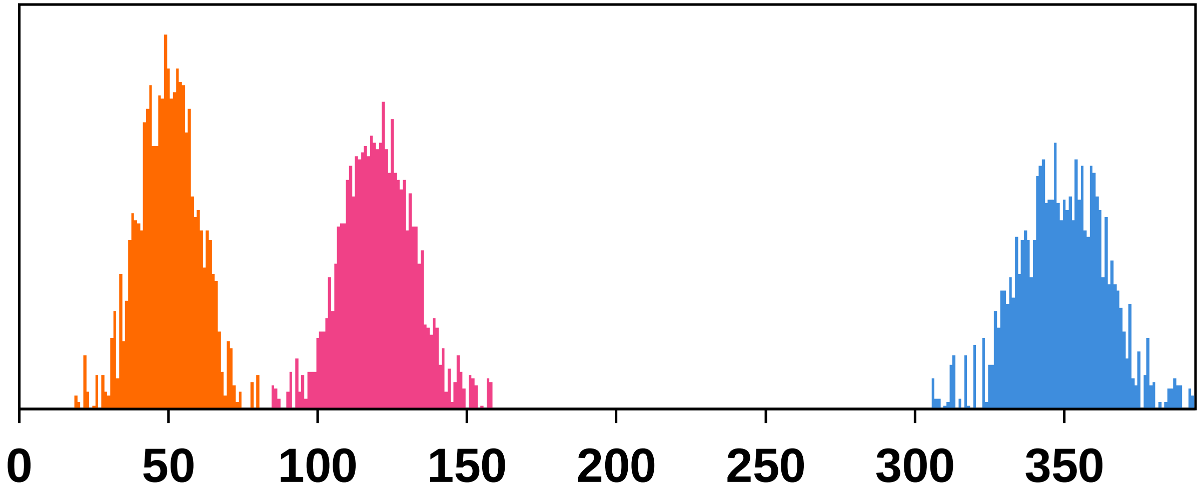


Standardization: features in same range

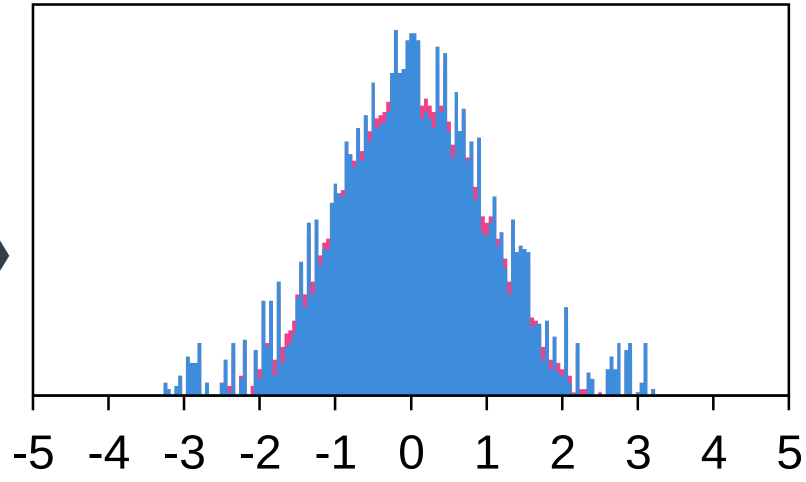
Typically, it is used a L2-normalization

L2 norm





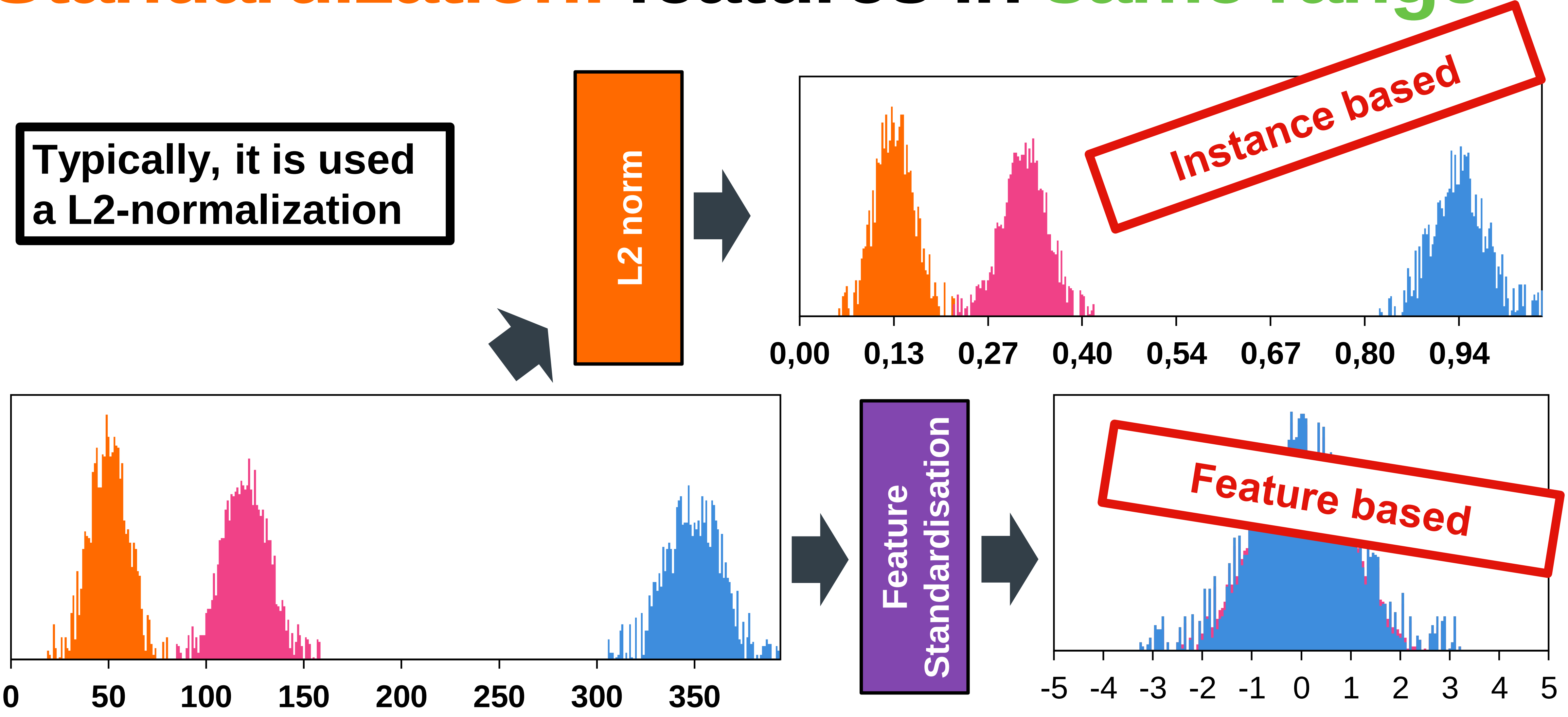
Feature Standardisation



Each feature independently

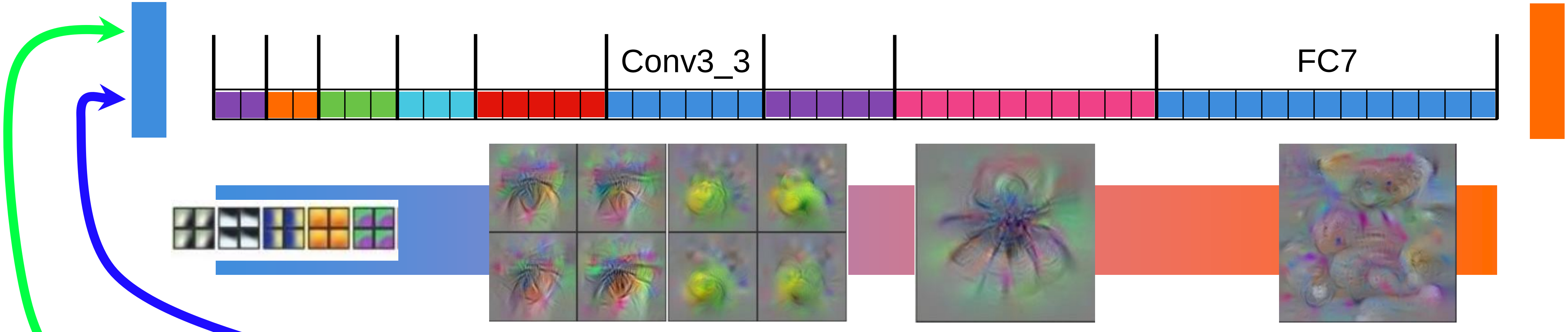
Standardization: features in same range

Typically, it is used a L2-normalization

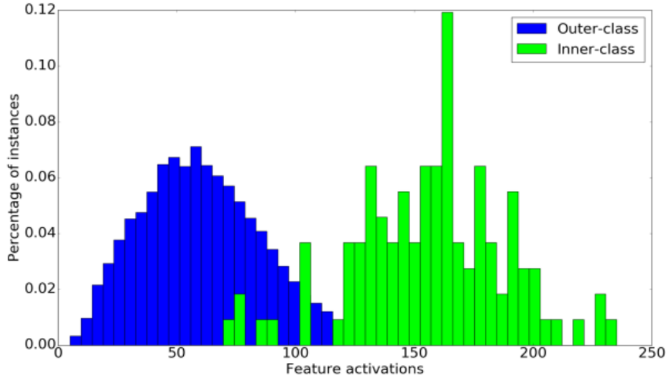


Each feature independently

Standardization: features in context



- Distinct feature behavior for a subset of data wrt rest



Stardardization: features in context

fc7 n1946

Gadwall

Brown Pelican

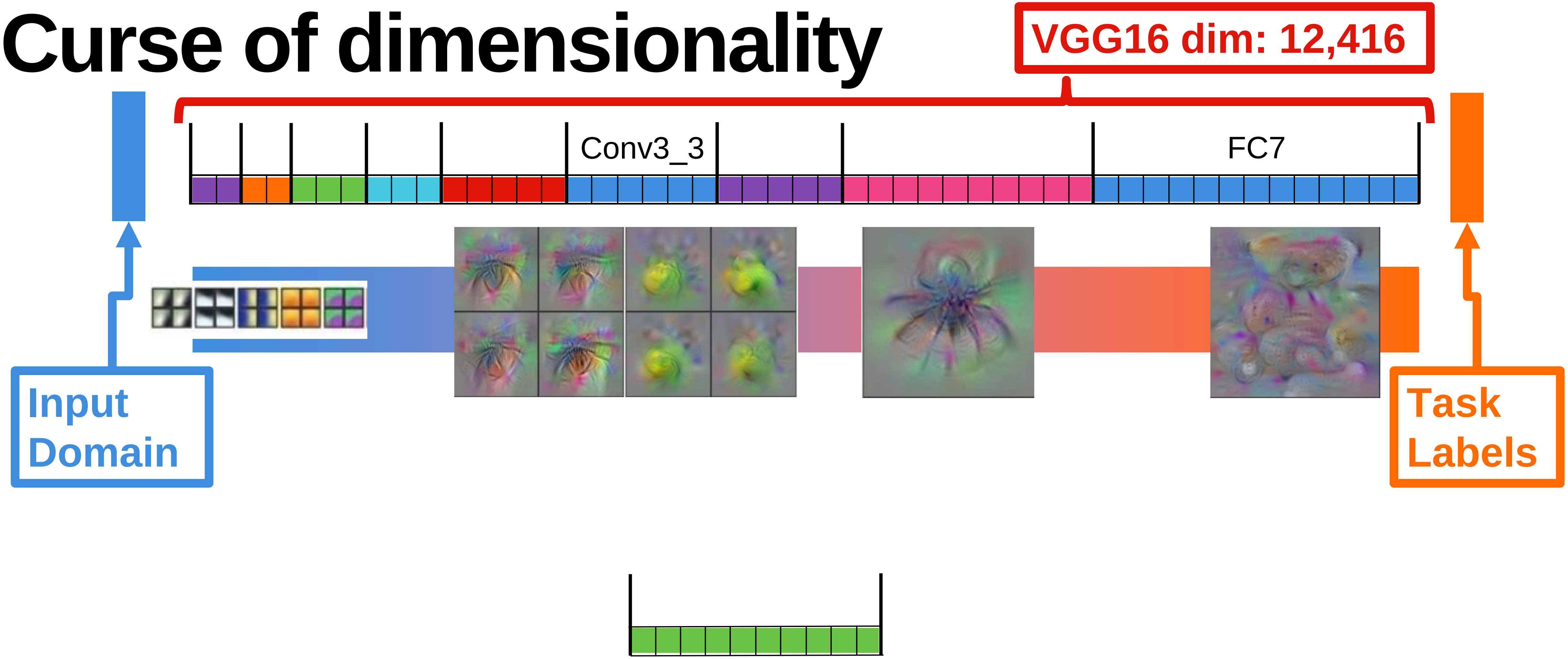
White Pelican

Heermann Gull

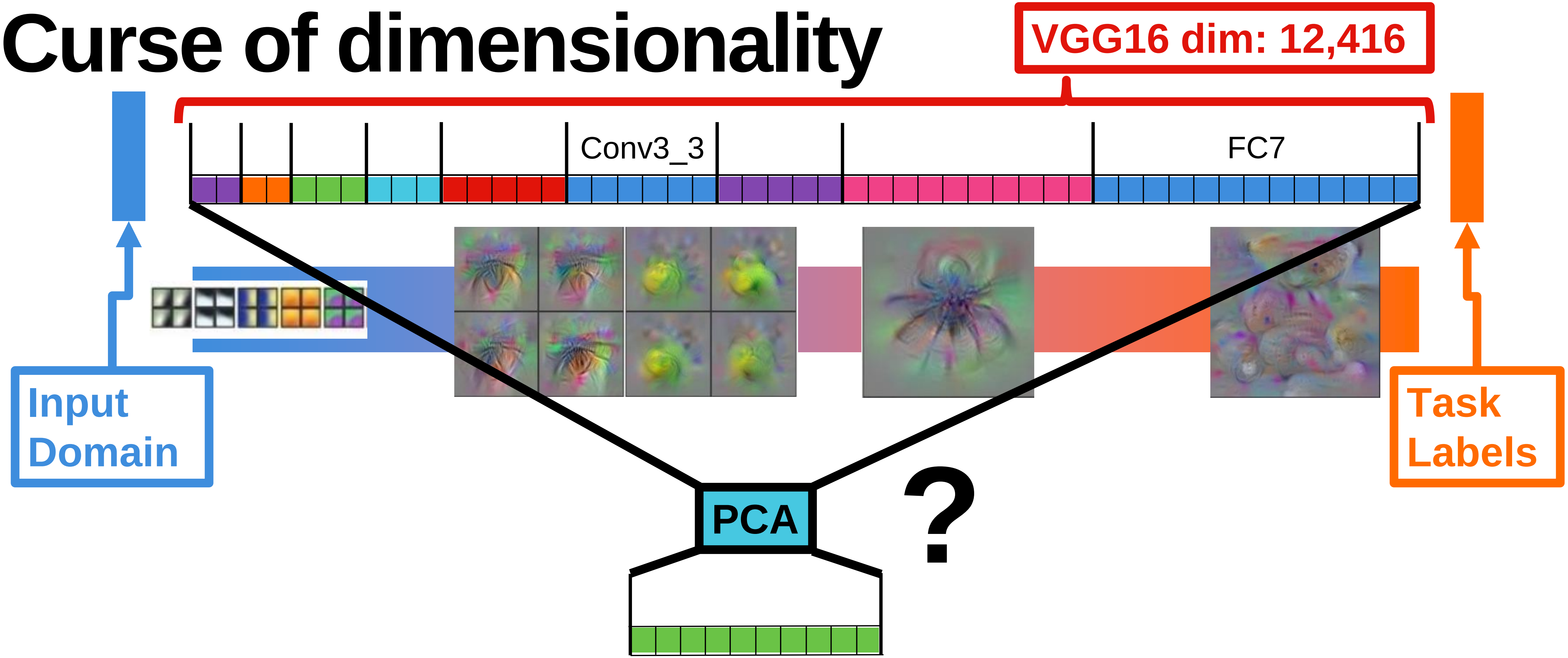
Garcia-Gasulla et al.
On the behavior of convolutional nets for feature extraction. 2017

CUB-200 - **birds**

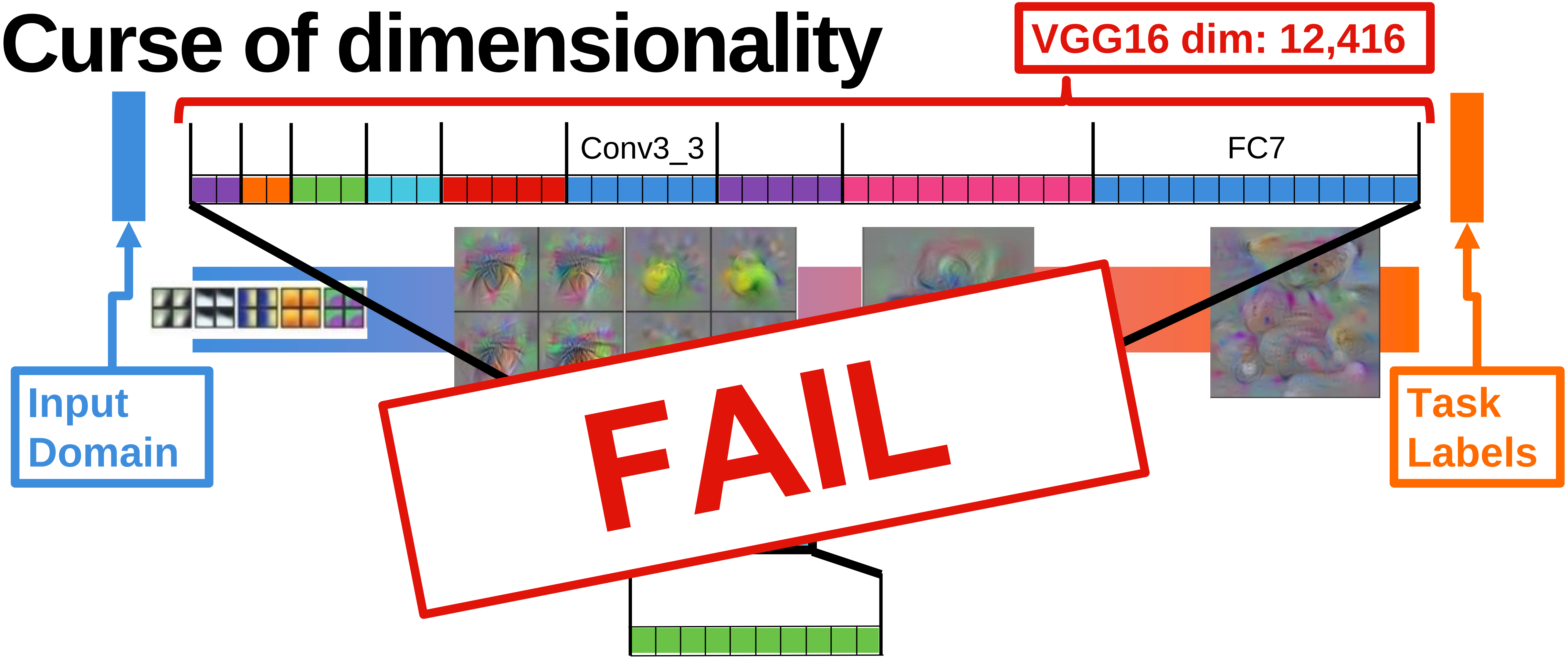
Curse of dimensionality



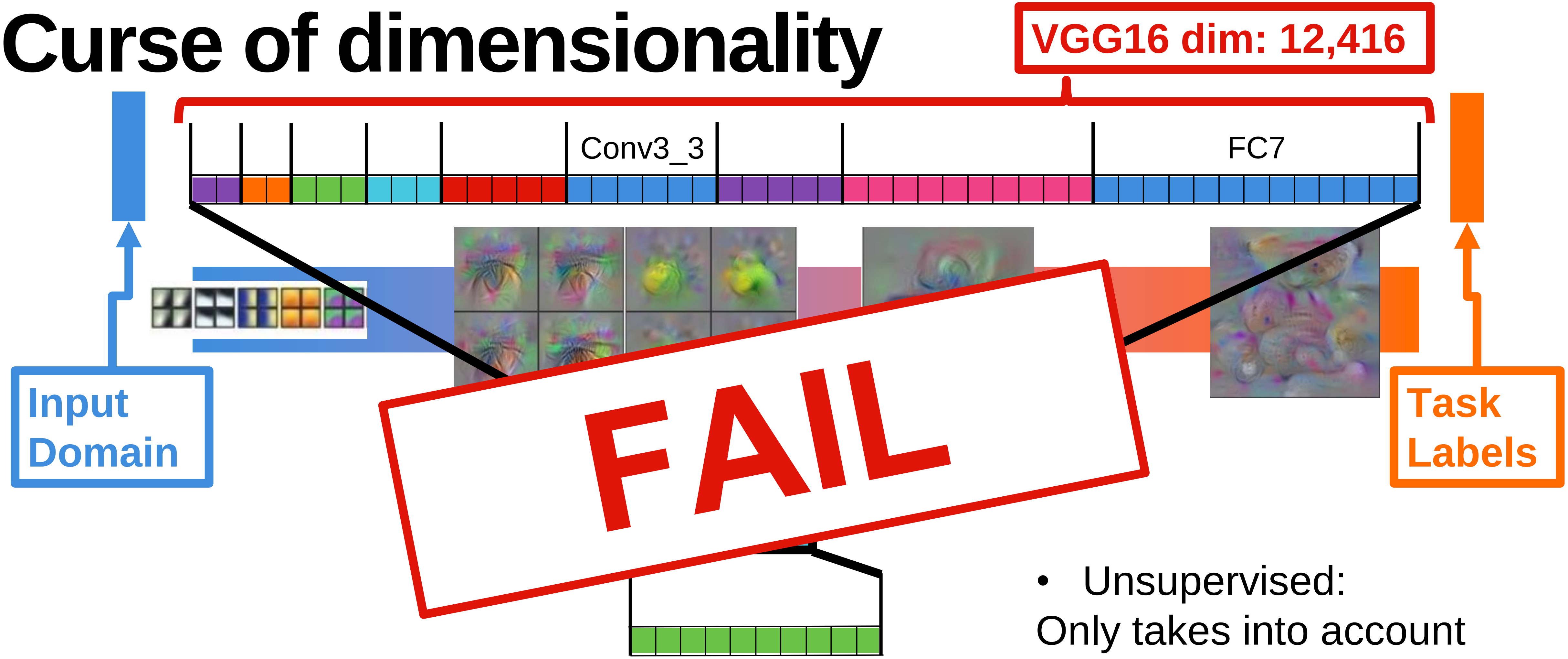
Curse of dimensionality



Curse of dimensionality



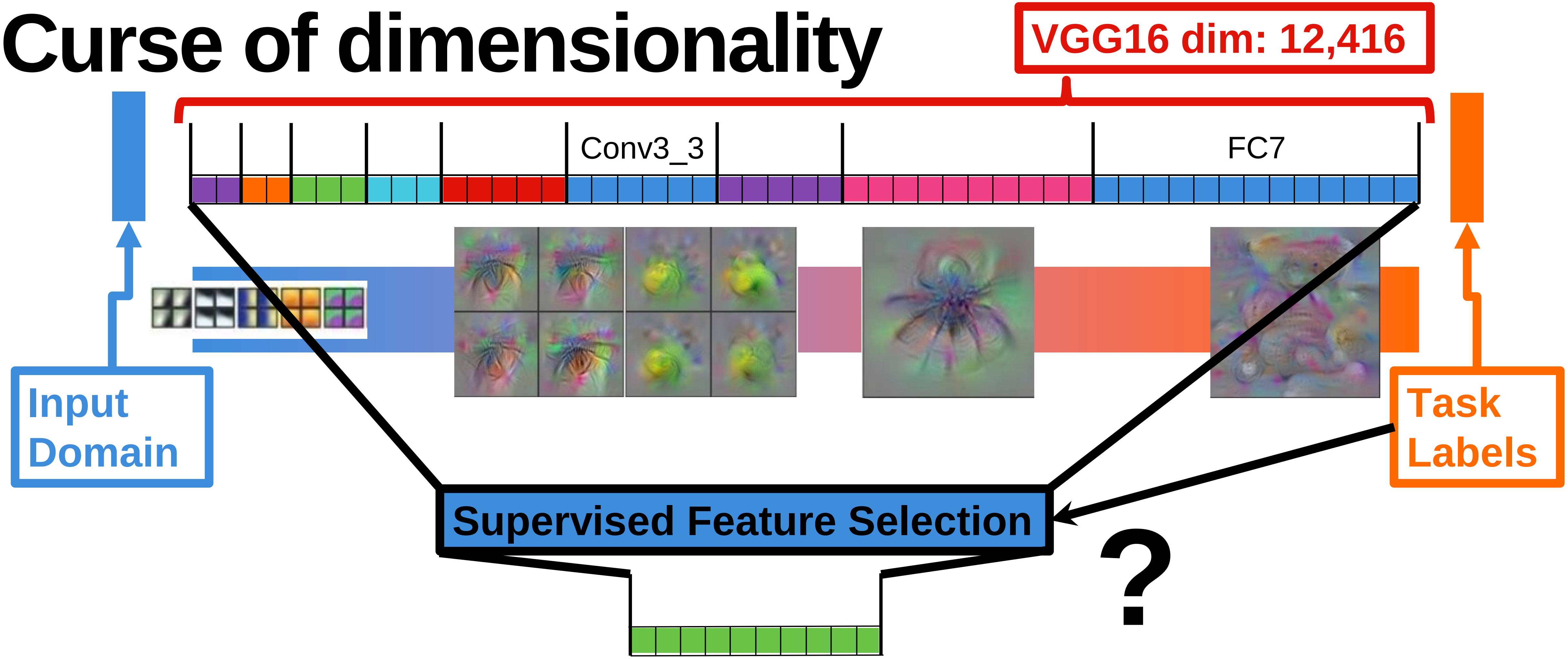
Curse of dimensionality



- Unsupervised:
Only takes into account the **domain**

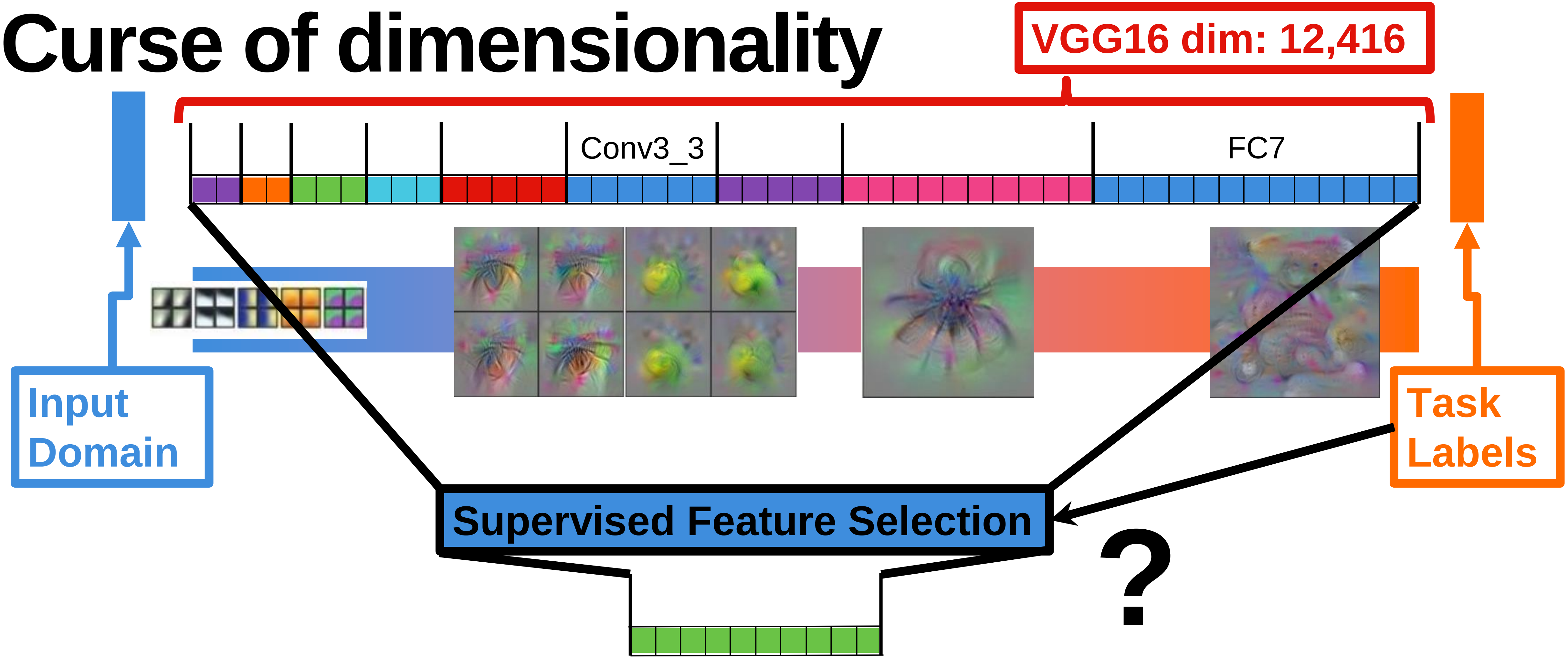
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Curse of dimensionality



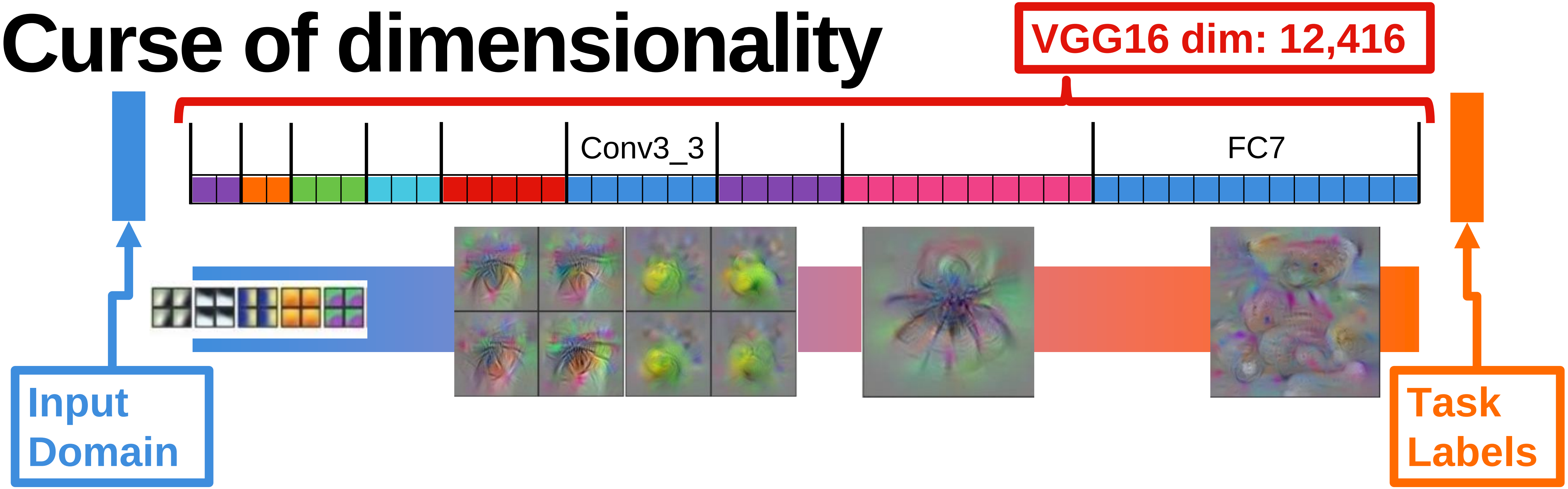
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Curse of dimensionality

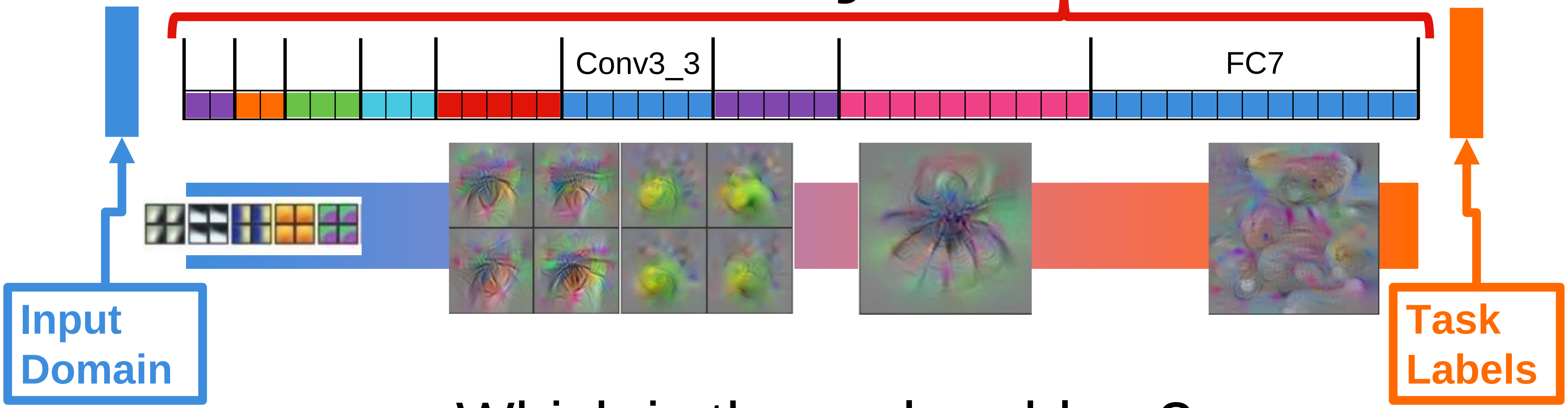


- High computational **cost!**

Curse of dimensionality

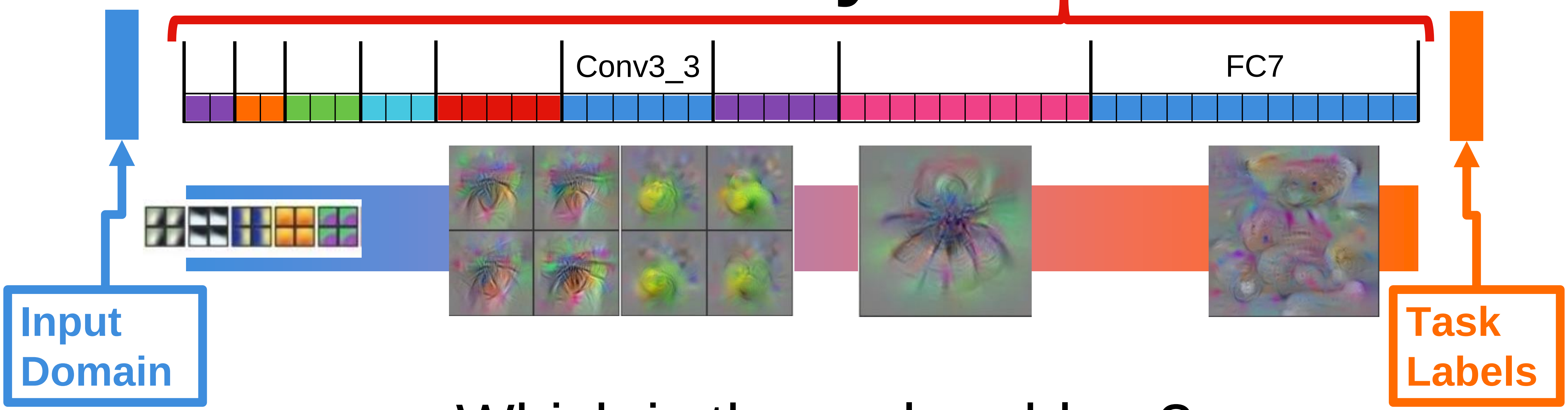


Curse of dimensionality



- Which is the real problem?

Curse of dimensionality

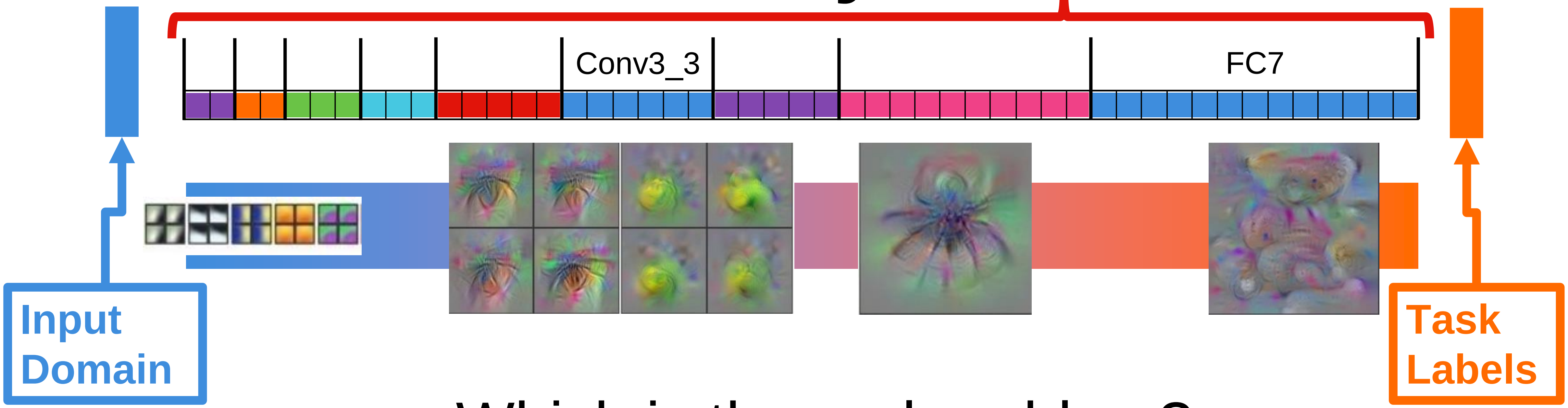


- Which is the real problem?
  - Too many features?
  - Too few images?

Curse of dimensionality

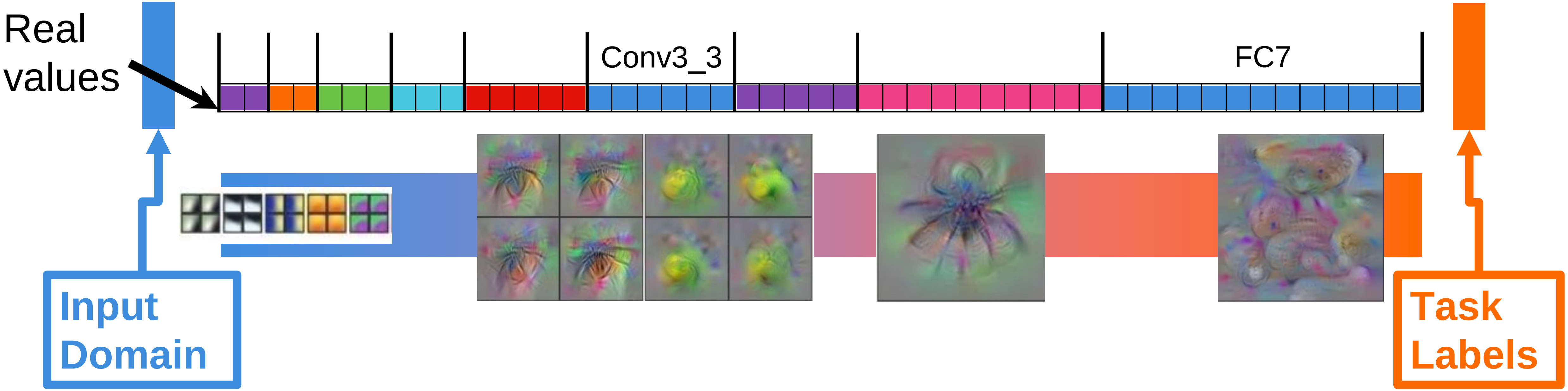
- Which is the real problem?
  - Too many features?
  - ~~Too few images?~~ **A requirement**

Curse of dimensionality

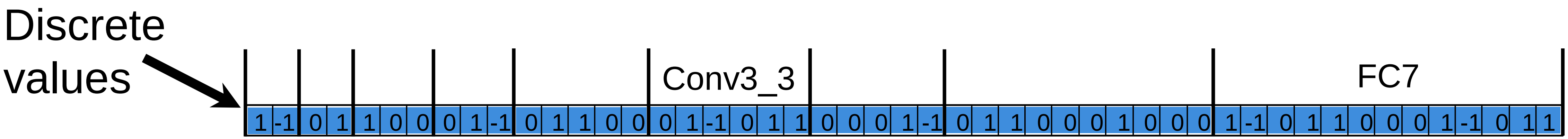


- Which is the real problem?
  - ~~Too many features?~~
  - **Too much information!**
  - ~~Too few images?~~

Meaningful discretization



Quantitization to {-1,0,1} based on feature values

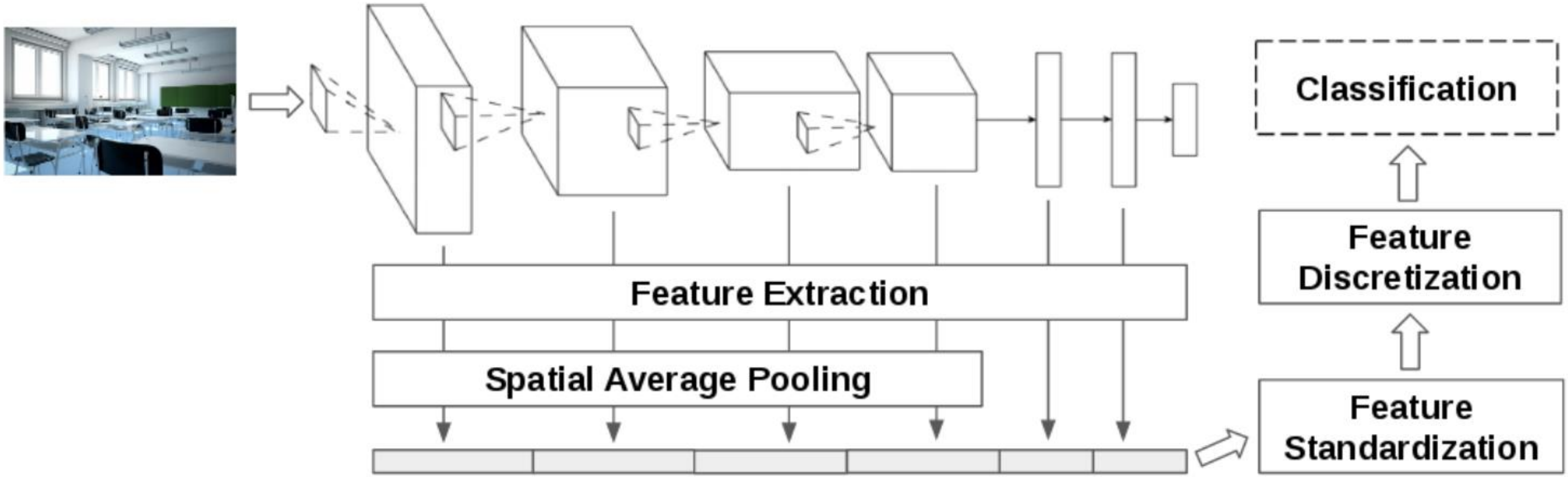


Full-Network Embedding Recipe

Garcia-Gasulla, et al. *An out-of-the-box full-network embedding for convolutional neural networks.* 2018.

1. Spatial Average Pooling
2. Standardisation
3. Discretization

Full Network embedding



FNE - Results: Similar source task

Network pre-trained on **Places2** for mit67 and on **ImageNet** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	sdogs	caltech101	food101	textures	wood
Baseline `fc6`	80.0	65.8	89.5	89.3	78.0	91.4±0.6	61.4±0.2	69.6	70.8±6.6
Baseline `fc7`	81.7	63.2	87.0	89.6	79.3	89.7±0.3	59.1±0.6	69.0	68.9 ±6.8
Full-network	83.6	65.5	93.3	89.2	78.8	91.4±0.6	67.0±0.7	73.0	74.1±6.9
SotA	86.9 [5]	92.3 [10]	97.0 [5]	91.6 [6]	90.3 [5]	93.4 [31]	77.4 [4]	75.5 [17]	- -
ED	✓	✓	✓	✗	✓	✗	✗	✗	-
FT	✓	✓	✓	✓	✓	✓	✓	✗	-

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FNE - Results: Similar source task

Best case scenario!

Network pre-trained on **Places2** for mit67 and on **Image**

Dataset	mit67	cub200	flowers102	cats-dogs	sdogs	caltech101	food101	textures	wood
Baseline `fc6`	80.0	65.8	89.5	89.3	78.0	91.4±0.6	61.4±0.2	69.6	70.8±6.6
Baseline `fc7`	81.7	63.2	87.0	89.6	79.3	89.7±0.3	59.1±0.6	69.0	68.9 ±6.8
Full-network	83.6	65.5	93.3	89.2	78.8	91.4±0.6	67.0±0.7	73.0	74.1±6.9
SotA	86.9 [5]	92.3 [10]	97.0 [5]	91.6 [6]	90.3 [5]	93.4 [31]	77.4 [4]	75.5 [17]	- -
ED	✓	✓	✓	✗	✓	✗	✗	✗	-
FT	✓	✓	✓	✓	✓	✓	✓	✗	-

FNE - Results: Similar source task

Network pre-trained on **Places2** for mit67 and on **ImageNet** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	sdogs	caltech101	food101	textures	wood	
Baseline fc6	80.0	65.8	89.5	89.3	78.0	91.4±0.6	61.4±0.2	69.6	70.8±6.6	+2.9
Baseline fc7	81.7	63.2	87.0	89.6	79.3	89.7±0.3	59.1±0.6	69.0	68.9 ±6.8	+4.2
Full-network	$\boxed{83.6}$	65.5	$\boxed{93.3}$	89.2	78.8	$\boxed{91.4\pm0.6}$	$\boxed{67.0\pm0.7}$	$\boxed{73.0}$	$\boxed{74.1\pm6.9}$	
SotA	86.9 [5]	92.3 [10]	97.0 [5]	91.6 [6]	90.3 [5]	93.4 [31]	77.4 [4]	75.5 [17]	- -	
ED	✓	✓	✓	✗	✓	✗	✗	✗	-	
FT	✓	✓	✓	✓	✓	✓	✓	✗	-	

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

FNE - Results: Similar source task

Network pre-trained on **Places2** for mit67 and on **ImageNet** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	sdogs	caltech101	food101	textures	wood	
Baseline `fc6`	80.0	65.8	89.5	89.3	78.0	91.4±0.6	61.4±0.2	69.6	70.8±6.6	**+2.9**
Baseline `fc7`	81.7	63.2	87.0	89.6	79.3	89.7±0.3	59.1±0.6	69.0	68.9 ±6.8	**+4.2**
Full-network	83.6	**-0.3**	93.3	**-0.4**	**-0.5**	91.4±0.6	67.0±0.7	73.0	74.1±6.9	
SotA	86.9 [5]	92.3 [10]	97.0 [5]	91.6 [6]	90.3 [5]	93.4 [31]	77.4 [4]	75.5 [17]	- -	
ED	✓	✓	✓	✗	✓	✗	✗	✗	-	
FT	✓	✓	✓	✓	✓	✓	✓	✗	-	

FNE - Results: Similar source task

Network pre-trained on **Places2** for mit67 and on **ImageNet** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	sdogs	caltech101	food101	textures	wood	
Baseline `fc6`	80.0	65.8	89.5	89.3	78.0	91.4±0.6	61.4±0.2	69.6	70.8±6.6	+2.9
Baseline `fc7`	81.7	63.2	87.0	89.6	79.3	89.7±0.3	59.1±0.6	69.0	68.9 ±6.8	+4.2
Full-network	83.6	-0.3	93.3	-0.4	-0.5	91.4±0.6	67.0±0.7	73.0	74.1±6.9	
SotA	86.9 [5]	92.3 [10]	97.0 [5]	91.6 [6]	90.3 [5]	93.4 [31]	77.4 [4]	75.5 [17]	- -	
ED	✓	✓	✓	✗	✓	✗	✗	✗	-	
FT	✓	✓	✓	✓	✓	✓	✓	✗	-	

FNE - Results: Dissimilar source task

Network pre-trained on **ImageNet** for mit67 and on **Places2** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	caltech101	textures	wood
Baseline `fc7`	72.2	23.6	73.3	38.7	72.0	55.8	65.3
Full-network	75.5	35.5	88.7	56.2	80.0	65.1	74.0

FNE - Results: Dissimilar source task

Most frequent real-world scenario!

Network pre-trained on **ImageNet** for mit67 and on **Places2** for the rest.

Dataset	mit67	cub200	flowers102	cats-dogs	caltech101	textures	wood
Baseline `fc7`	72.2	23.6	73.3	38.7	72.0	55.8	65.3
Full-network	75.5	35.5	88.7	56.2	80.0	65.1	74.0

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

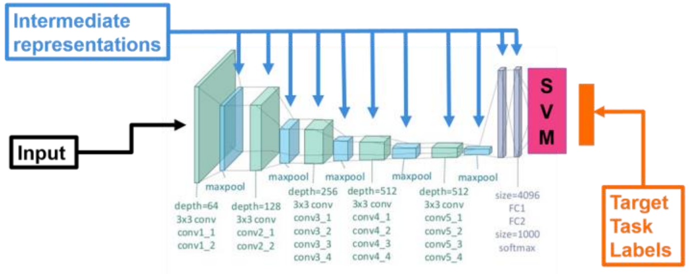
FNE - Results: Dissimilar source task

Network pre-trained on **ImageNet** for mit67 and on **Places2** for the rest.

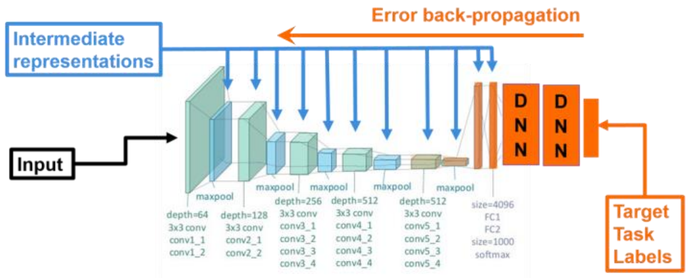
Dataset	mit67	cub200	flowers102	cats-dogs	caltech101	textures	wood
Baseline `fc7`	72.2	23.6	73.3	38.7	72.0	55.8	65.3
Full-network	75.5	35.5	88.7	56.2	80.0	65.1	74.0
	$\boxed{+3.3}$	$\boxed{+11.9}$	$\boxed{+15.4}$	$\boxed{+17.5}$	$\boxed{+8.0}$	$\boxed{+9.3}$	$\boxed{+10.6}$

Simple solutions

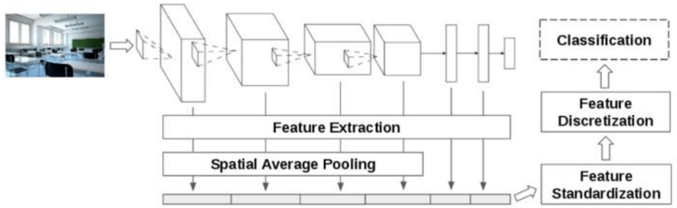
- DNN last layer features + SVM
  (Feature extraction)
  We need: **Similar task and domain**



- Add one or several NN layers +
  Fine-tuning pre-trained layers
  We need: **Enough data**



- Full Network Embedding
  - **Robust to different task and domain**
  - **Works with little data**



Practical tips

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Practical tips

Fine-tuning

- Whenever possible **don't start from scratch.**

Practical tips

Fine-tuning

- Whenever possible **don't start from scratch.**
- **External data** can help prevent overfitting (even from a different problem).

Practical tips

Fine-tuning

- Whenever possible **don’t start from scratch.**
- **External data** can help prevent overfitting (even from a different problem).
- Begin **freezing** as much as possible and proceed with caution (particularly for large models)

Practical tips

Feature extraction

- Easy **baseline** for every problem.

Practical tips

Feature extraction

- Easy **baseline** for every problem.
- **ImageNet,** a model to pre-train them all. (but not always)

Practical tips

Feature extraction

- Easy **baseline** for every problem.
- **ImageNet,** a model to pre-train them all. (but not always)
- Always **normalize** features.

Practical tips

Feature extraction

- Easy **baseline** for every problem.
- **ImageNet,** a model to pre-train them all. (but not always)
- Always **normalize** features.
- If source and target task are closely related:

  → **Last two layers** are your best chance.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Practical tips

Feature extraction

- Easy **baseline** for every problem.
- **ImageNet,** a model to pre-train them all. (but not always)
- Always **normalize** features.
- If source and target task are closely related:

  → **Last two layers** are your best chance.
- If source and target task are quite different:

  → **Try everything**

  → **Use FNE**

Useful References

Don’t start training from scratch

- Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. *Augmenting strong supervision using webdata for fine-grained categorization.* 2015.
- Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. *Bird species categorizationusing pose normalized deep convolutional nets*. 2014.
- Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, and Yunsheng Ma. *Deep-food: Deep learning-based food image recognition for computer-aided dietary assessment.* 2016.

Useful References

Transfer learning 101

- Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. *How transferable are features in deep neural networks?* 2014.
- Dario Garcia-Gasulla, Ferran Parés, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. *On the behavior of convolutional nets for feature extraction.* 2017

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Useful References

Basic feature extraction

- Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. *Factors of transferability for a generic convnet.* 2016.
- Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. *CNN features off-the-shelf: an astounding baseline for recognition.* 2014.
- Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. *Multi-scale orderless pooling of deep convolutional activation features*. 2014.
- Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. *Decaf: A deep convolutional activation feature for generic visual recognition*. 2014.
- Arsalan Mousavian and Jana Kosecka. *Deep convolutional features for image based retrieval and scene categorization*. 2015.

Useful References

Multi-layer feature extraction

- Garcia-Gasulla, Dario, Armand Vilalta, Ferran Parés, Eduard Ayguadé, Jesus Labarta, Ulises Cortés, and Toyotaro Suzumura. *An out-of-the-box full-network embedding for convolutional neural networks.* 2018.
- Vilalta, Armand, Dario Garcia-Gasulla, Ferran Parés, Jonathan Moreno, Eduard Ayguadé Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. *Full-Network Embedding in a Multimodal Embedding Pipeline.* 2017.

Useful References

Advanced fine tuning to solve small tasks

- Weifeng Ge and Yizhou Yu. *Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning.* 2017.
- Marcel Simon and Erik Rodner. *Neural activation constellations: Unsupervised part model discovery with convolutional networks*. 2015.

CTE-Power9

52 computing nodes. Each one:

- 2 x IBM Power9 8335-GTH @ 2.4GHz(total 160 threads)
- 512GB of main memory
- 4 x GPU NVIDIA V100 (Volta) with 16GB each

Available through PRACE and RES

Hands on session 16:00

2 GPUs per account

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

thanks.

dario.garcia@bsc.es
armand.vilalta@bsc.es