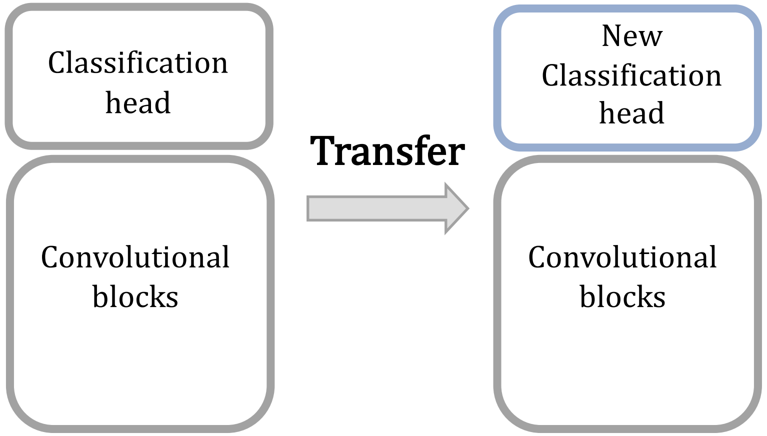
Transfer Learning



CS-EJ3311 - Deep Learning with Python
24.10.-11.12.2022
Aalto University & FiTech.io

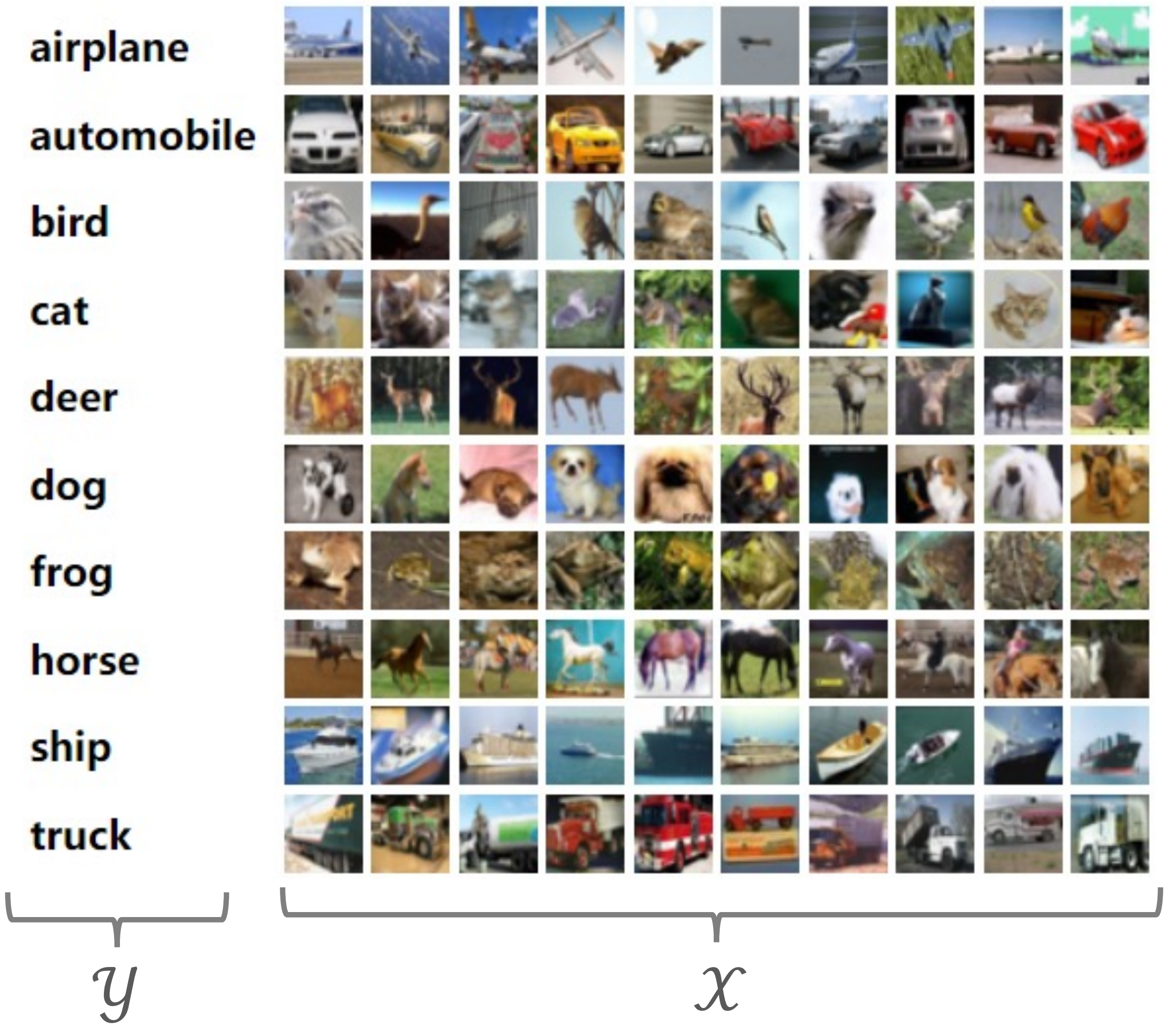
28.11.2022 Shamsi Abdurakhmanova

Deep learning methods requires a lot of:

- ✓ (Labeled) data
- ✓ Computational resources (GPUs)
- ✓ Time
- ✓ DL expertise

Machine learning

- ✓ Feature space $\mathcal{X}$
- ✓ Label space $\mathcal{Y}$



Machine learning

- ✓ Domain $\mathcal{D} = \{\mathcal{X}, \mathrm{P}(\mathcal{X})\}$
- ✓ Task $\mathcal{T} = \{\mathcal{Y}, P(\mathcal{Y}|\mathcal{X})\}$, where $P(\mathcal{Y}|\mathcal{X})$ is a function

ML model trying to learn

Task A (source)

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

Task $\mathcal{T}_s = \{\mathcal{Y}_s, P(\mathcal{Y}_s|\mathcal{X}_s)\}$

Task B (target)

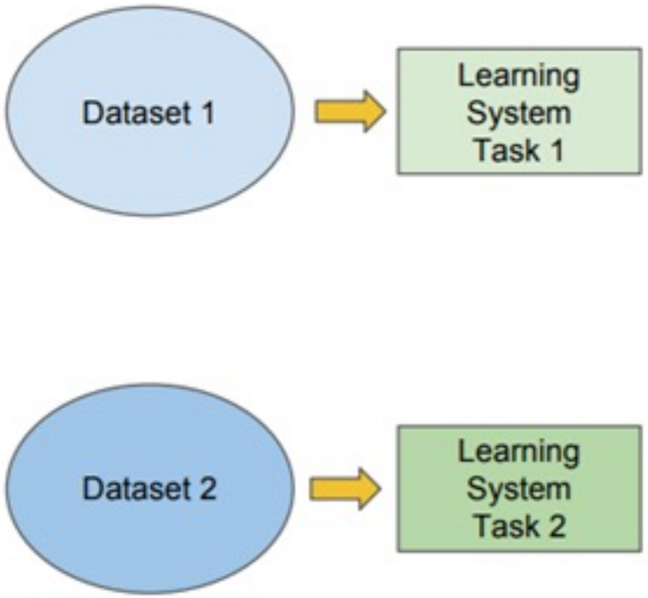
Domain $\mathcal{D}_T = \{\mathcal{X}_\mathrm{T}, \mathrm{P}(\mathcal{X}_T)\}$

Task $\mathcal{T}_T = \{\mathcal{Y}_T, P(\mathcal{Y}_T|\mathcal{X}_T)\}$

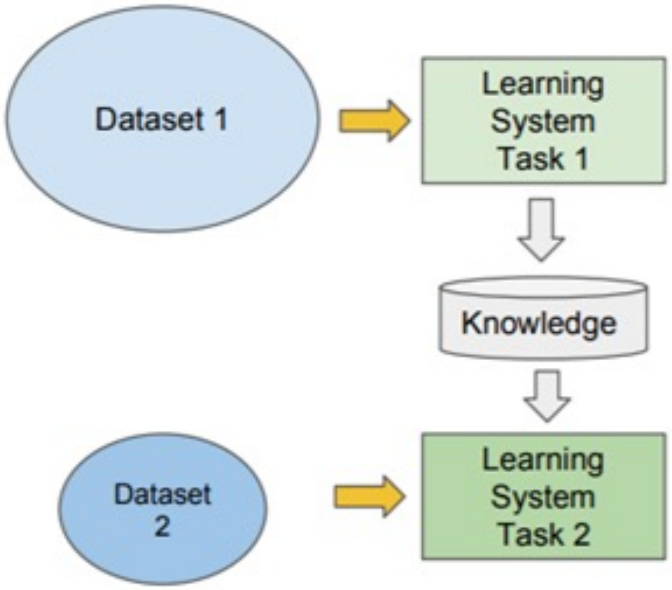
Transfer learning – learn $P(\mathcal{Y}_T|\mathcal{X}_T)$ using knowledge $\mathcal{D}_s, \mathcal{T}_s$

Traditional ML vs Transfer Learning

- Isolated, single task learning:
  - Knowledge is not retained or accumulated. Learning is performed w.o. considering past learned knowledge in other tasks

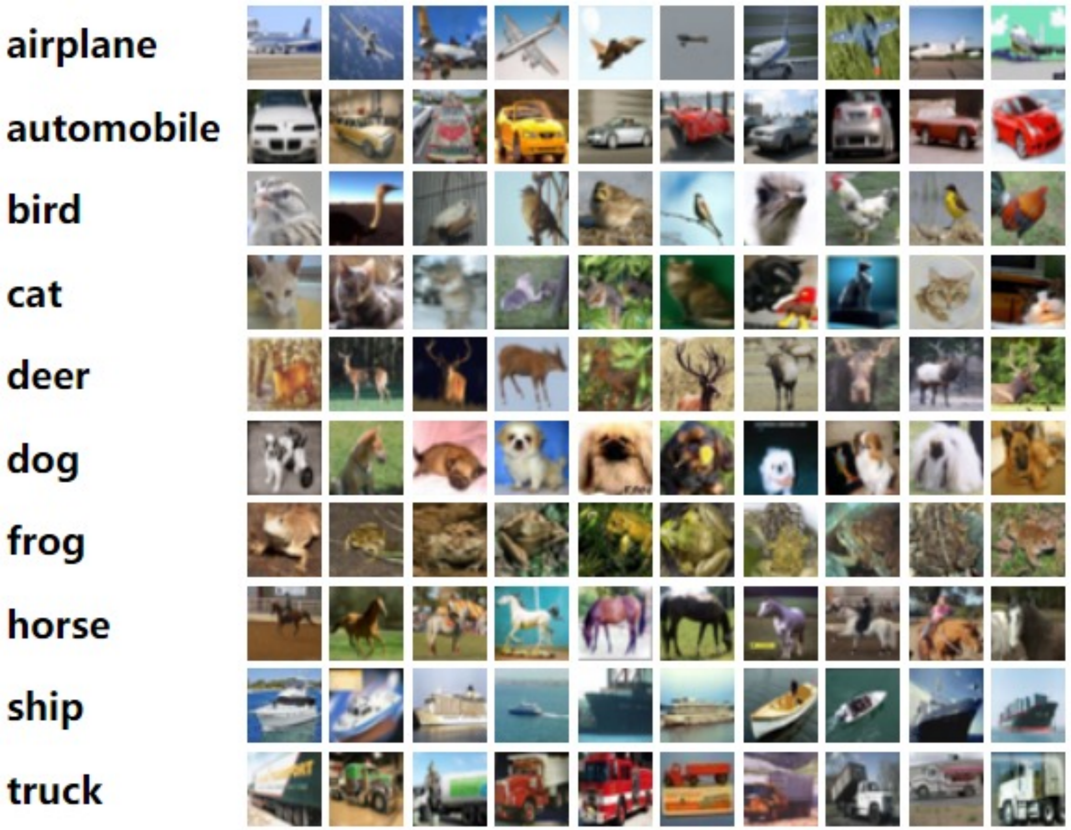


- Learning of a new tasks relies on the previous learned tasks:
  - Learning process can be faster, more accurate and/or need less training data

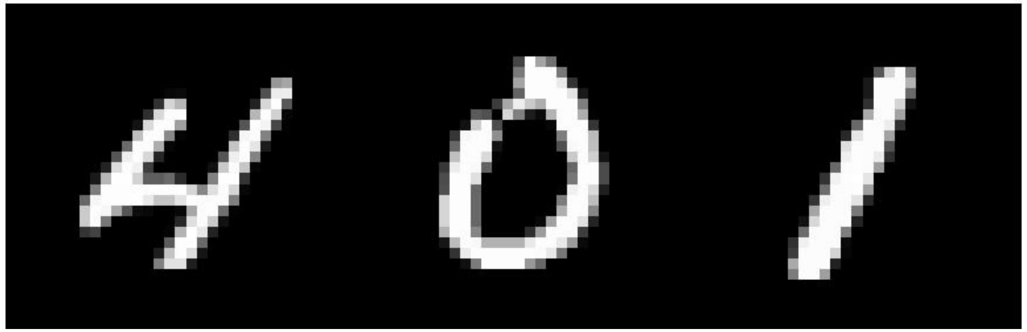


When to use Transfer Learning?

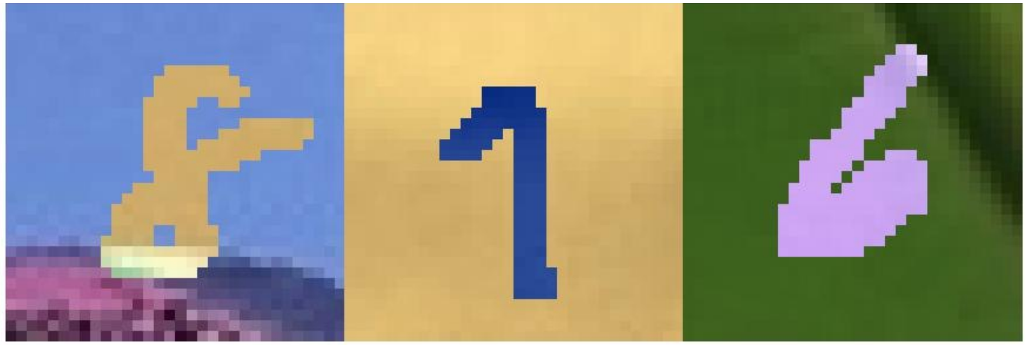
- Task A and B have the similar input, e.g both have images as input (features from A could be helpful in learning task B).
- There is a lot more data for Task A than Task B.



Use transfer learning? Yes

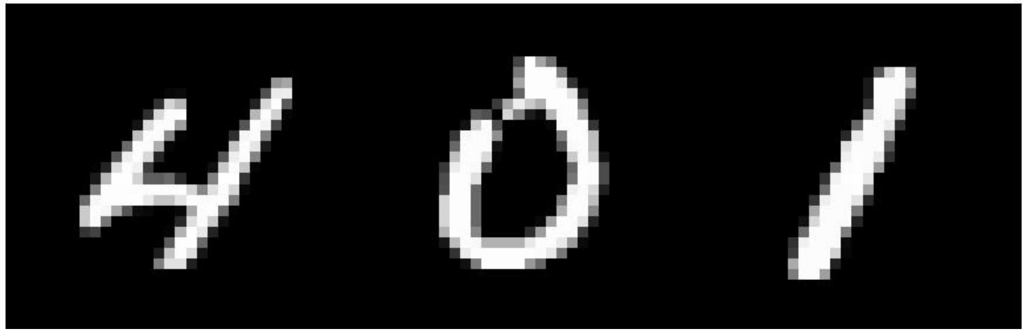


MNIST

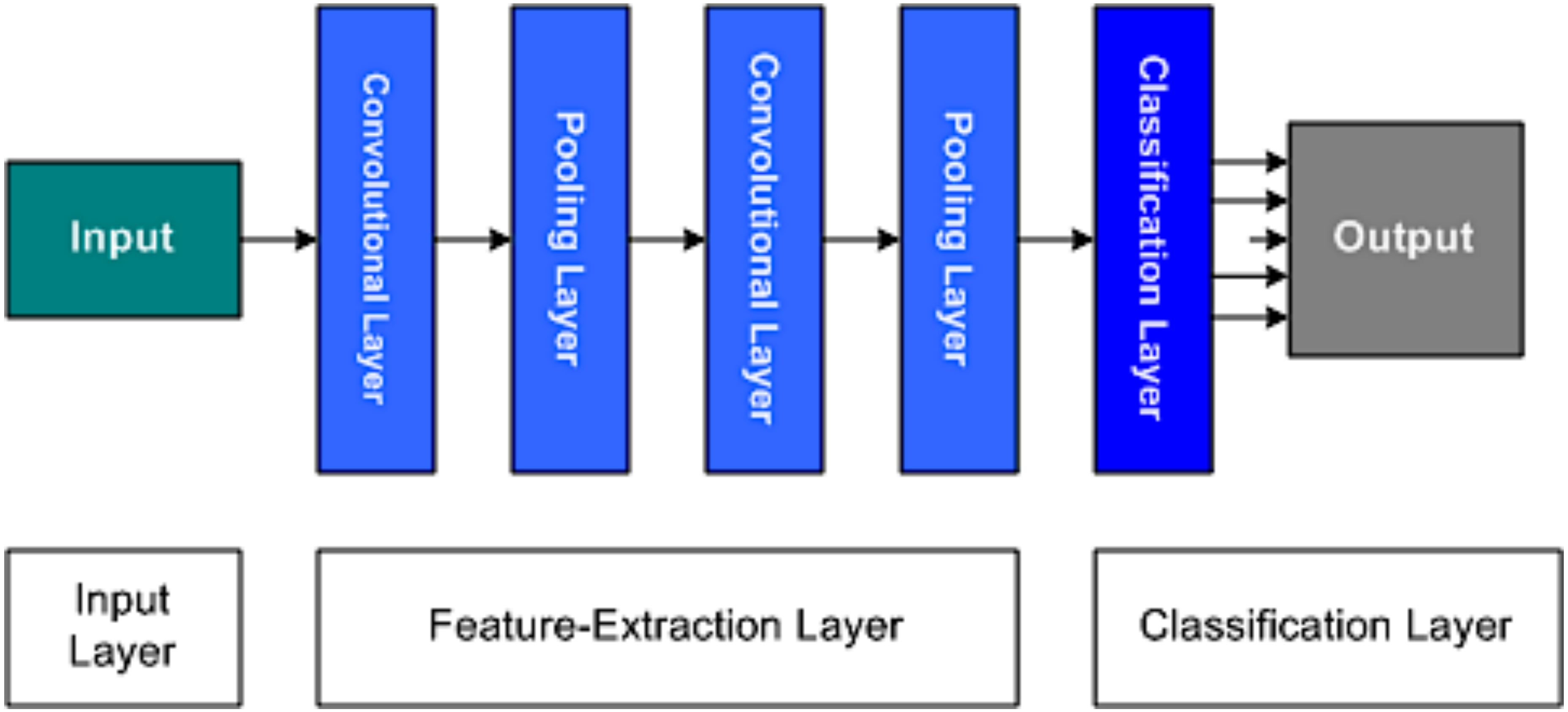


MNIST-M

Use transfer learning? Yes

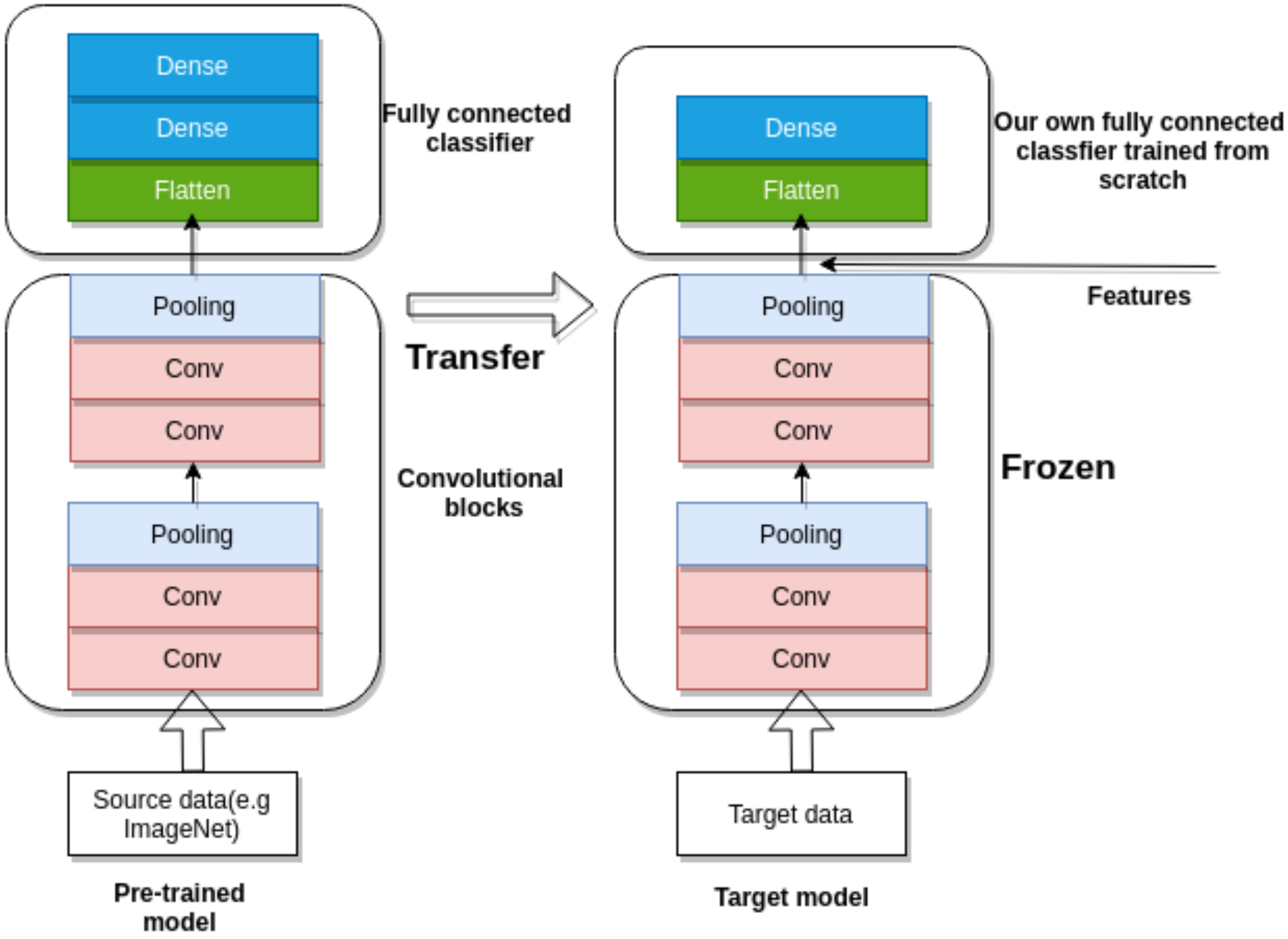


Use transfer learning? No

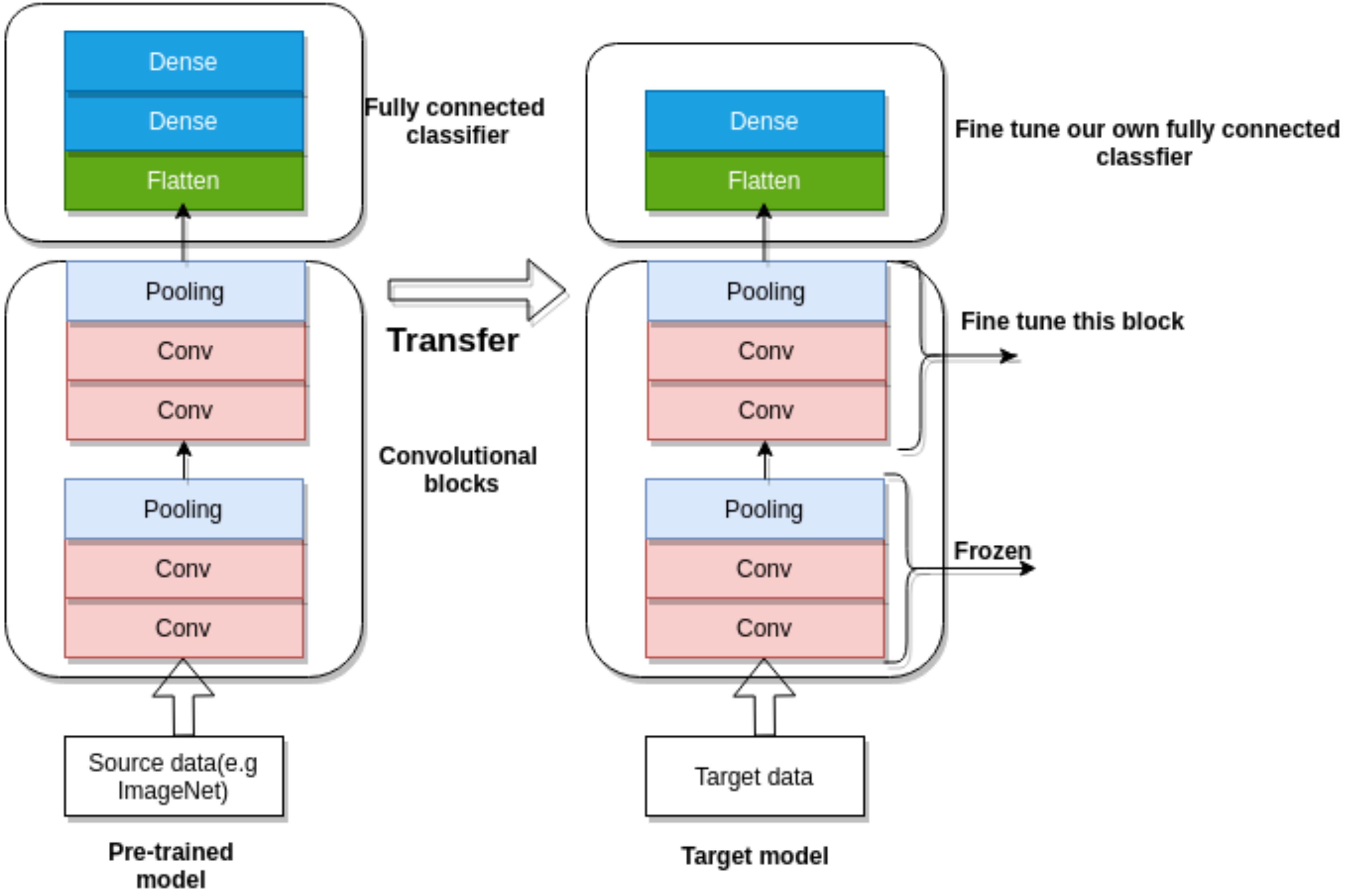


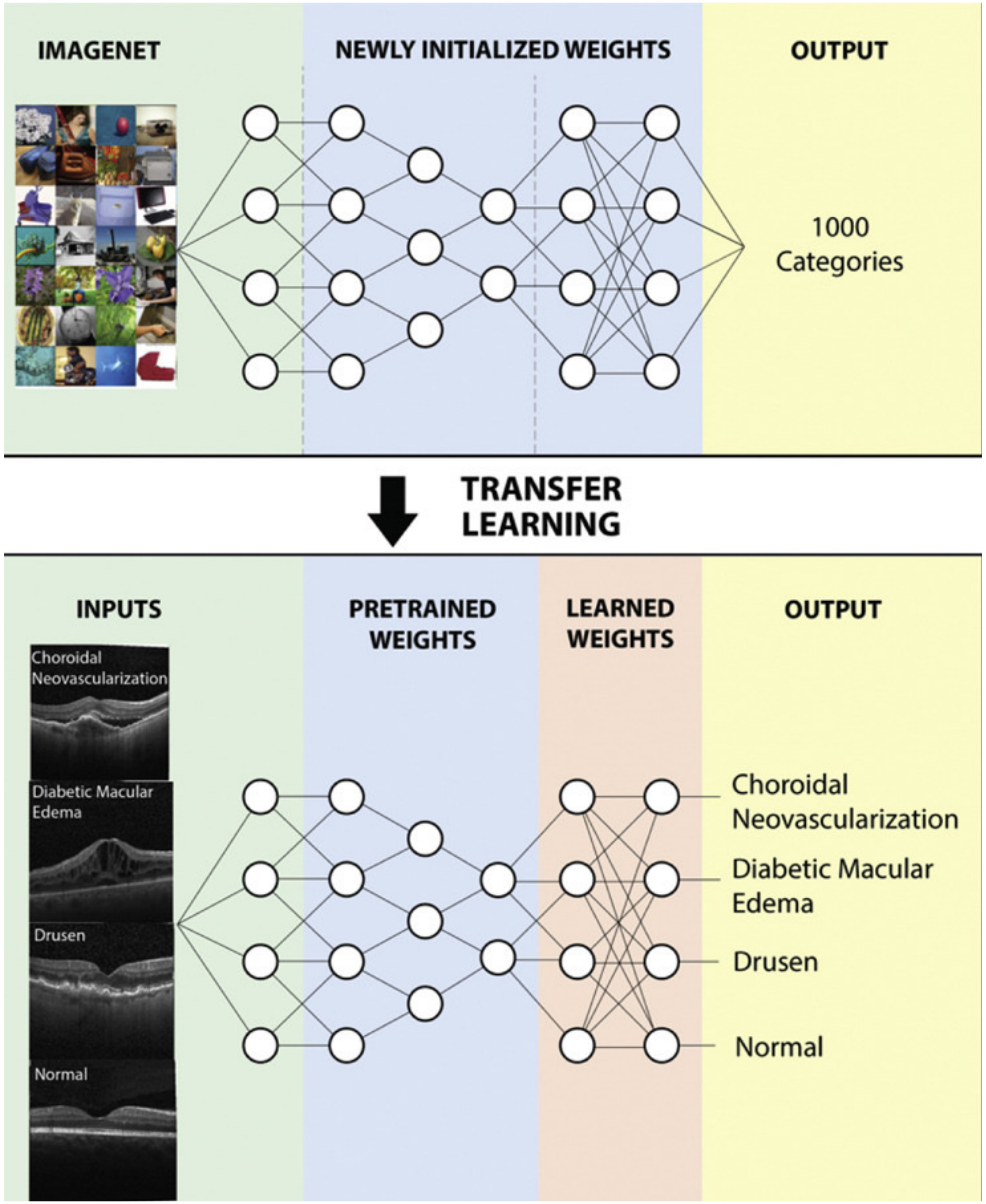
Img source: https://developer.ibm.com/articles/transfer-learning-for-deep-learning/

Pre-trained model for feature extraction



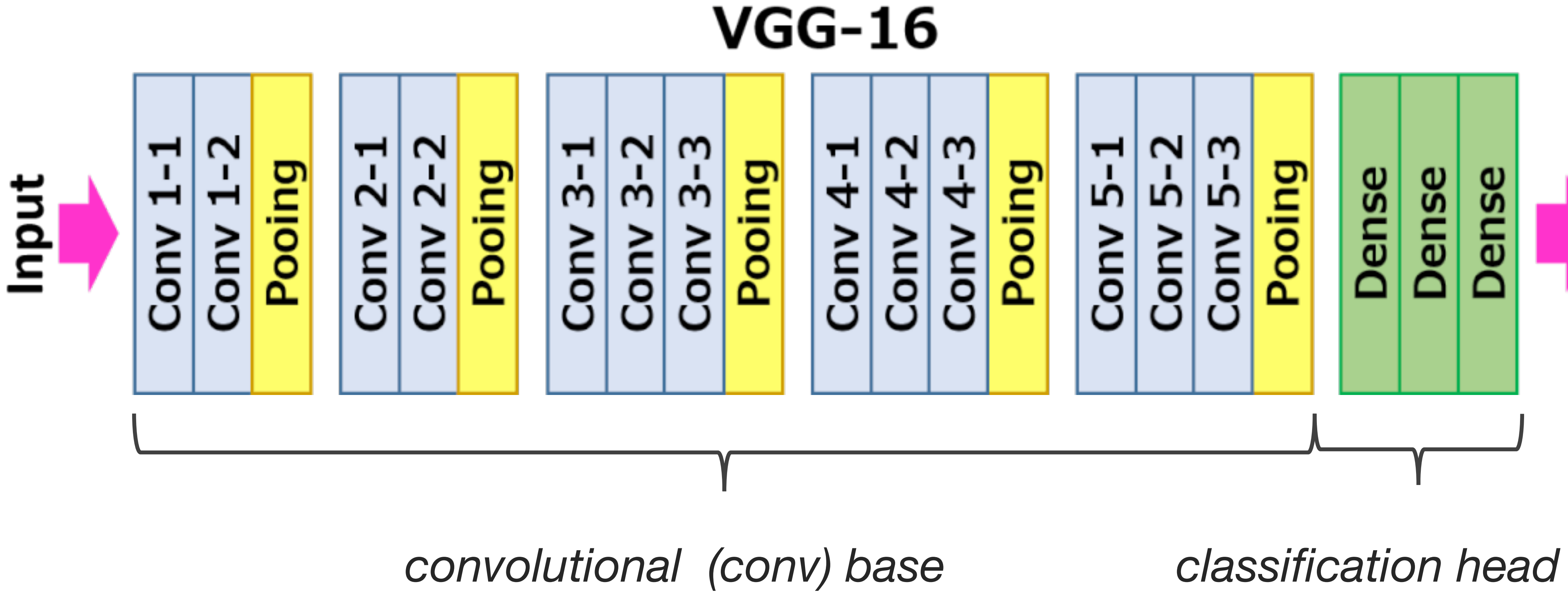
Fine-tuning pre-trained model





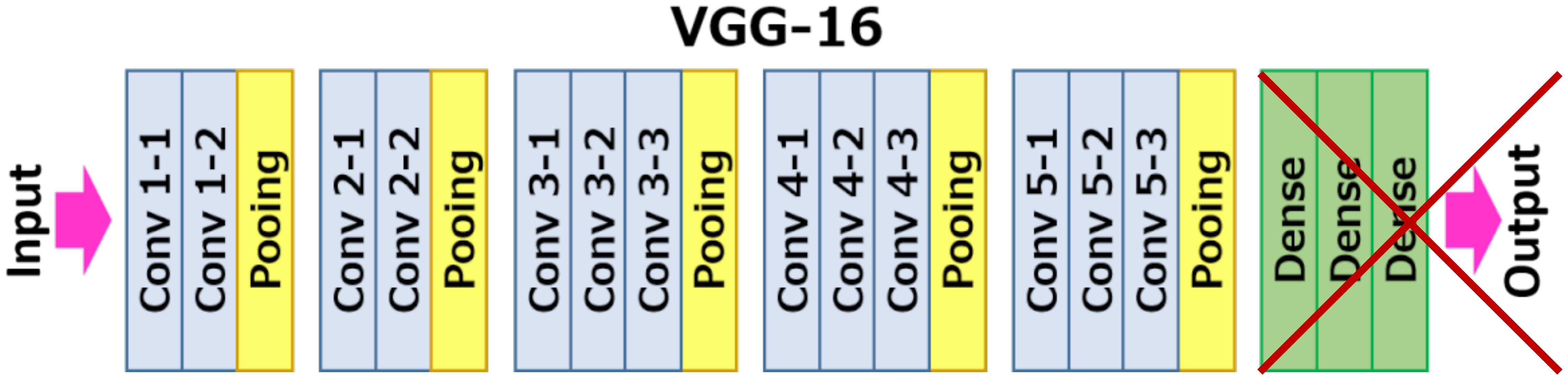
Kermany, D et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*

Pre-trained VGG16 model

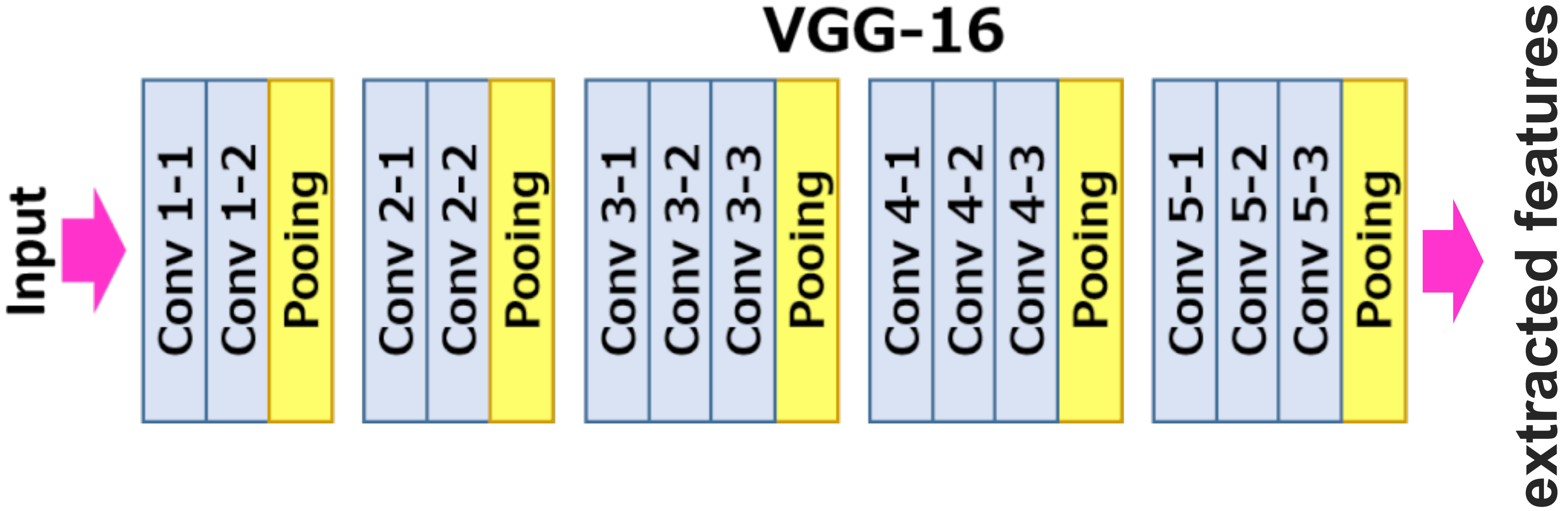


Pre-trained VGG16 as a feature extractor

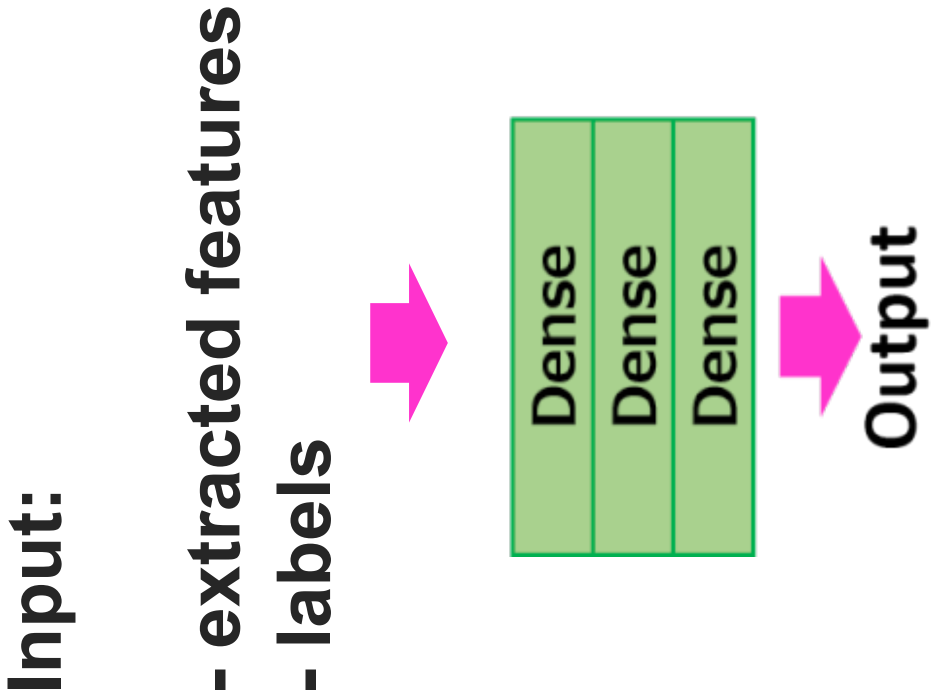
1. Load pre-trained convolutional base



2. Perform feature extraction (params of conv base are frozen)

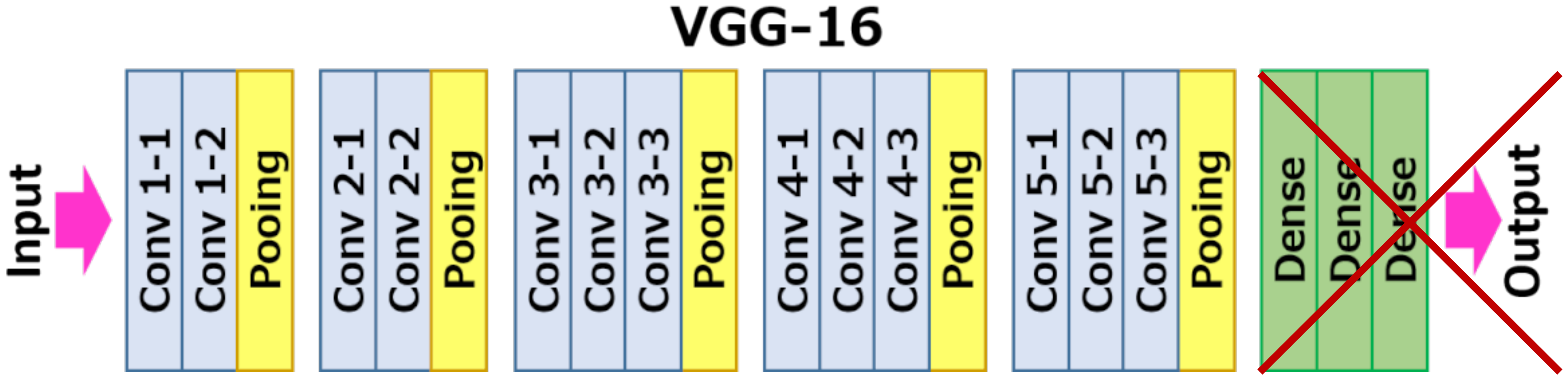


3. Train (new) classification head on extracted features

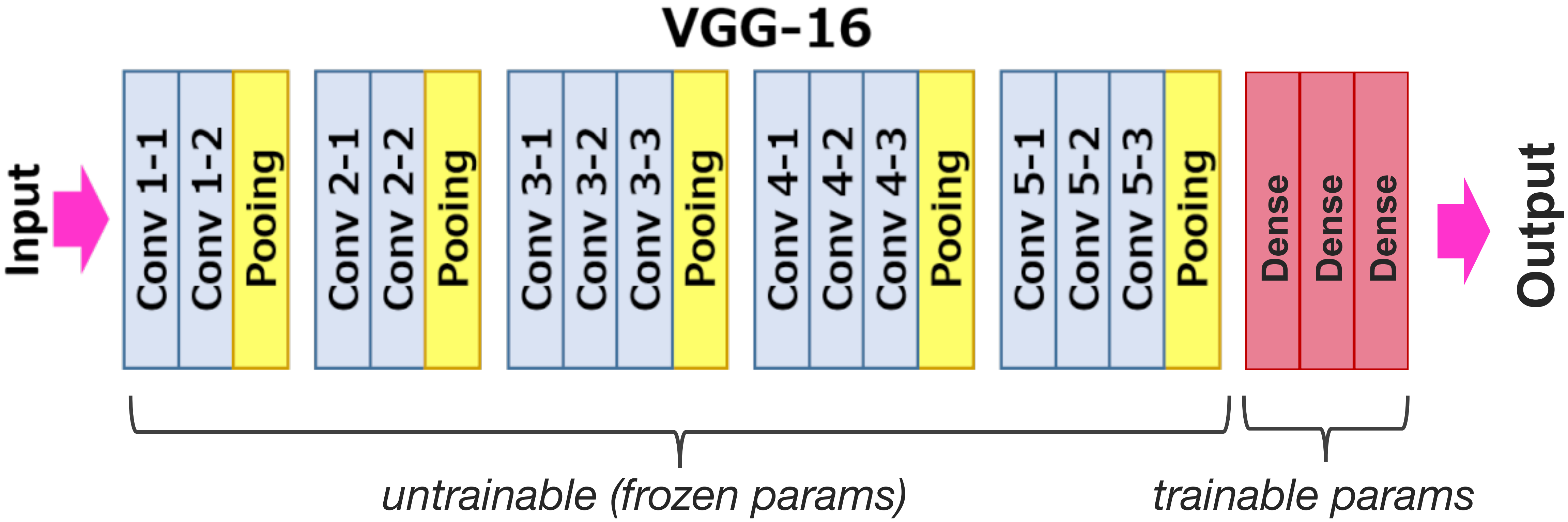


Fine-tuning pre-trained VGG16

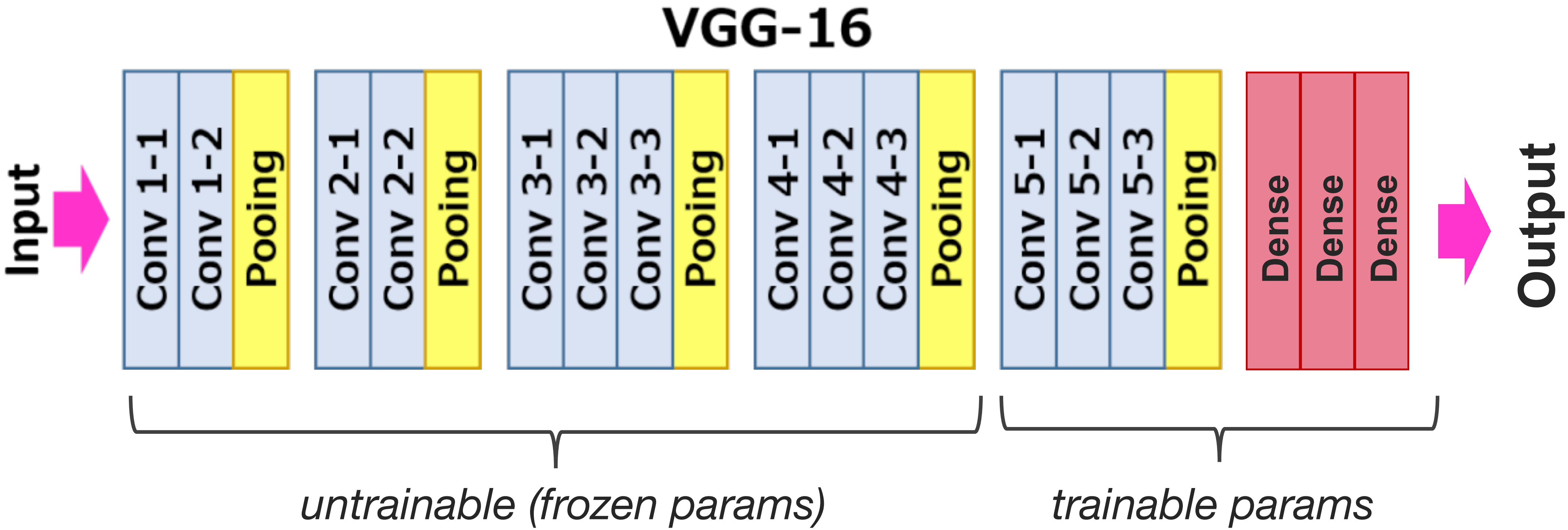
1. Load pre-trained convolutional base



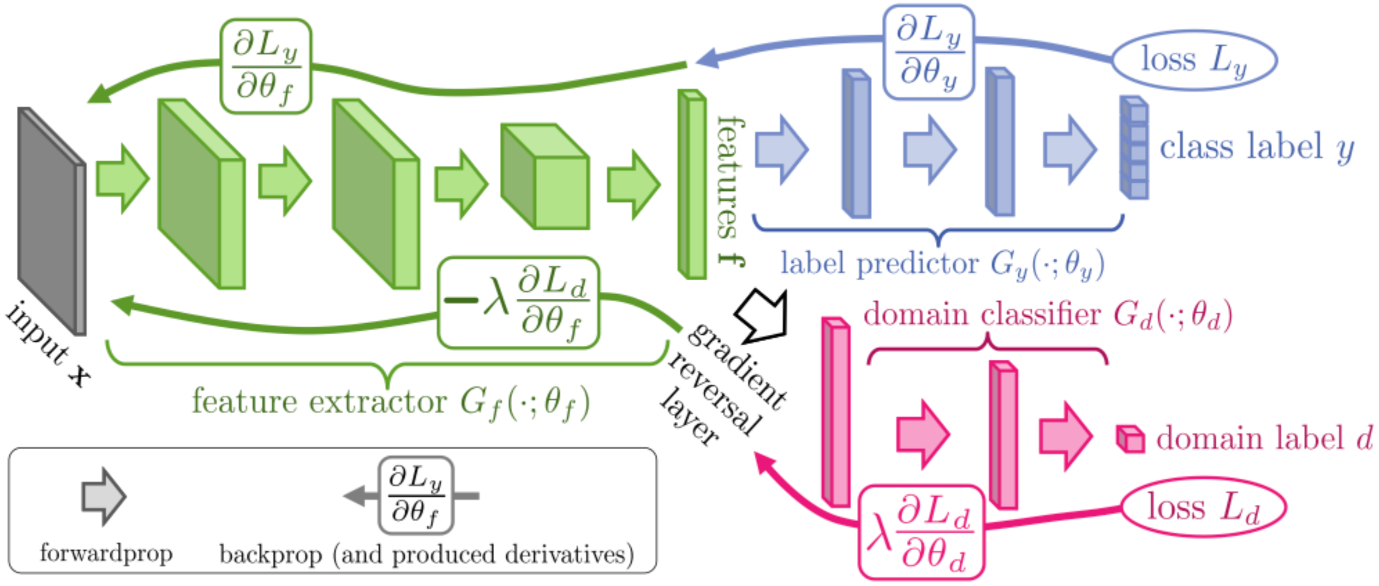
2. *Add and train new clf head (params of conv base are frozen)*



3. Unfreeze last few layers and train together with a clf head



Transfer learning when no labels are available



0
1
2
...
9

MNIST

MNIST-M

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

Transfer learning when no labels are available

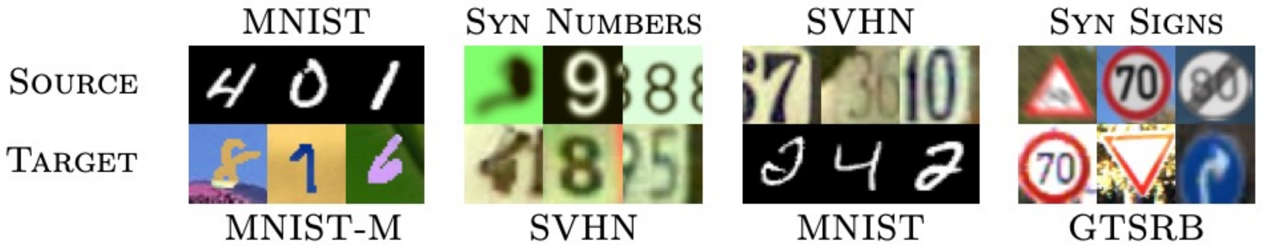


Figure 6: Examples of domain pairs used in the experiments. See Section 5.2.4 for details.

METHOD — SOURCE / TARGET	MNIST / MNIST-M	SYN NUMBERS / SVHN	SVHN / MNIST	SYN SIGNS / GTSRB
SOURCE ONLY	.5225	.8674	.5490	.7900
SA (Fernando et al., 2013)	.5690 (4.1%)	.8644 (−5.5%)	.5932 (9.9%)	.8165 (12.7%)
DANN	**.7666** (52.9%)	**.9109** (79.7%)	**.7385** (42.6%)	**.8865** (46.4%)
TRAIN ON TARGET	.9596	.9220	.9942	.9980

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.