
بسم الله الرحمن الرحیم

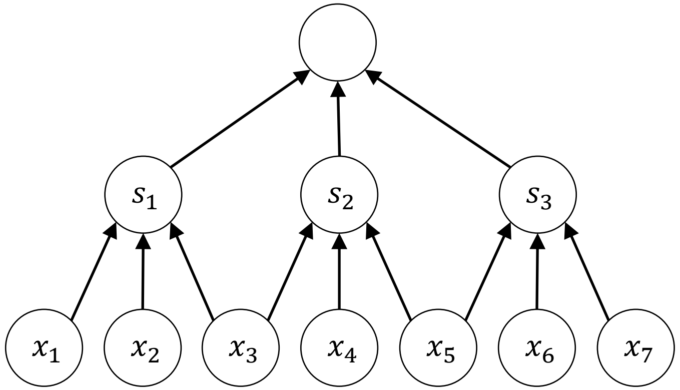
Deep Learning

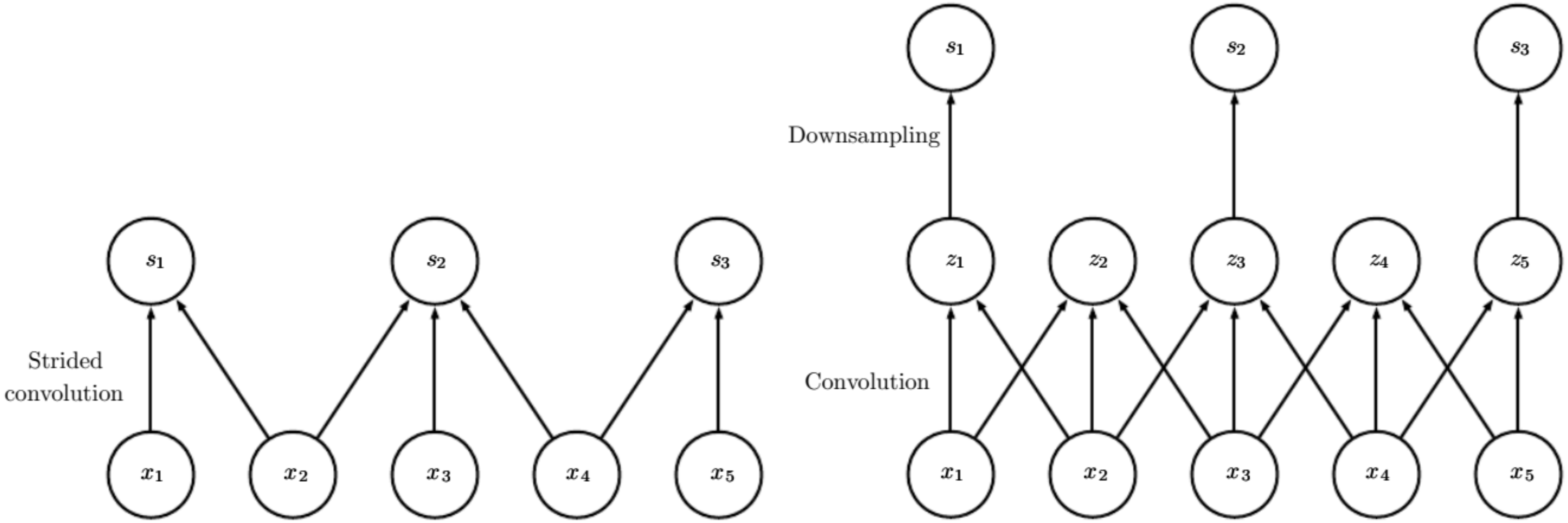
Mohammad Reza Mohammadi

2021

Stride

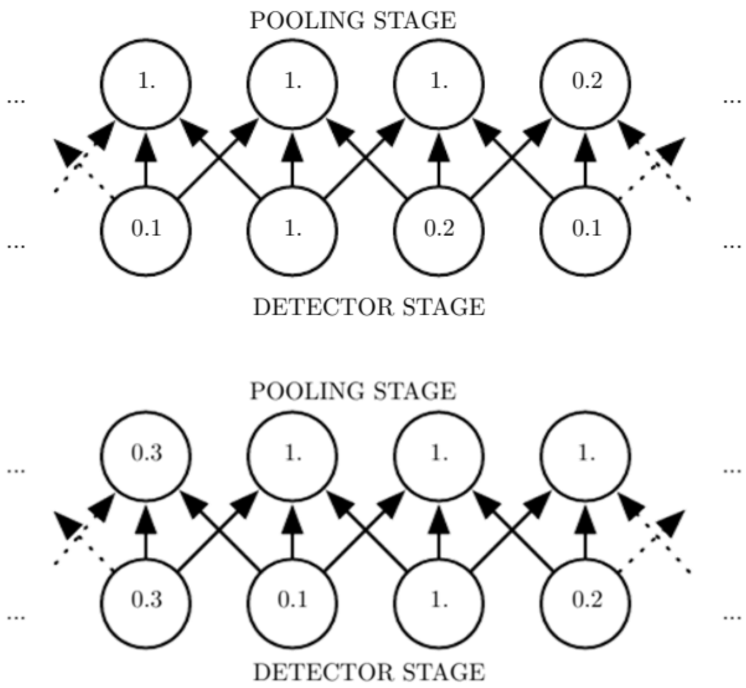
- We may want to skip over some positions of the kernel in order to reduce the computational cost
  - at the expense of not extracting our features as finely
  - we can think of this as downsampling
  - this increases the receptive field





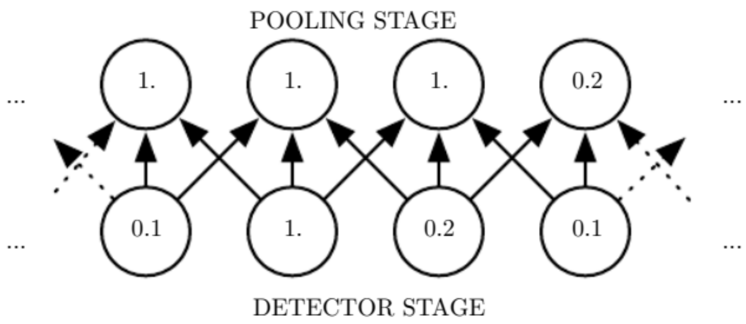
Pooling

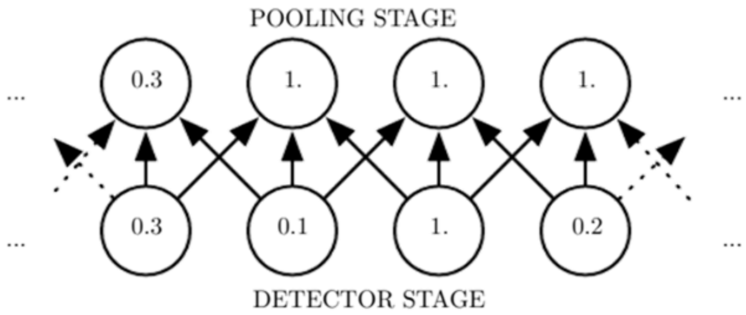
- A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs
  - For example, the max pooling operation reports the maximum output within a rectangular neighborhood
  - Other popular pooling functions include the average, the L2 norm, the standard deviation, or a weighted average
- In all cases, pooling helps to make the representation become approximately invariant to small translations of the input

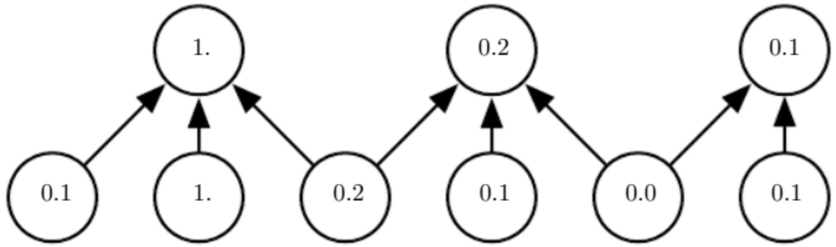


Pooling + Stride

- A pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs
- We can stride after pooling to improve the computational efficiency
  - this reduction in the input size of fully connected layers can also result in improved statistical efficiency and reduced memory requirements for storing the parameters

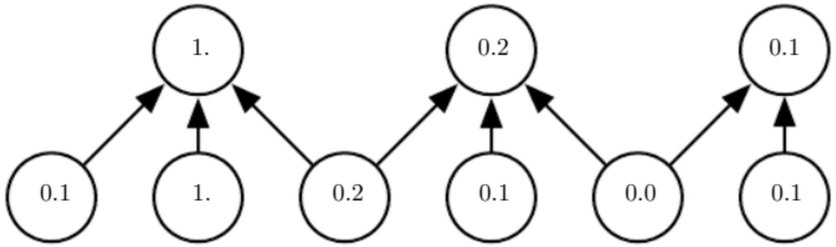


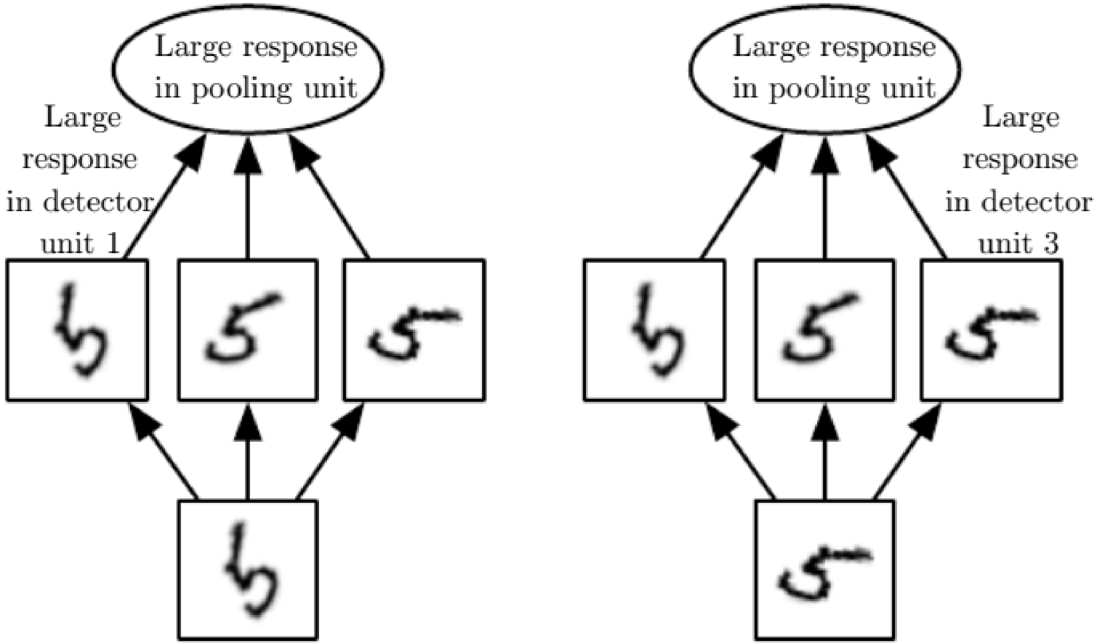




Pooling over outputs

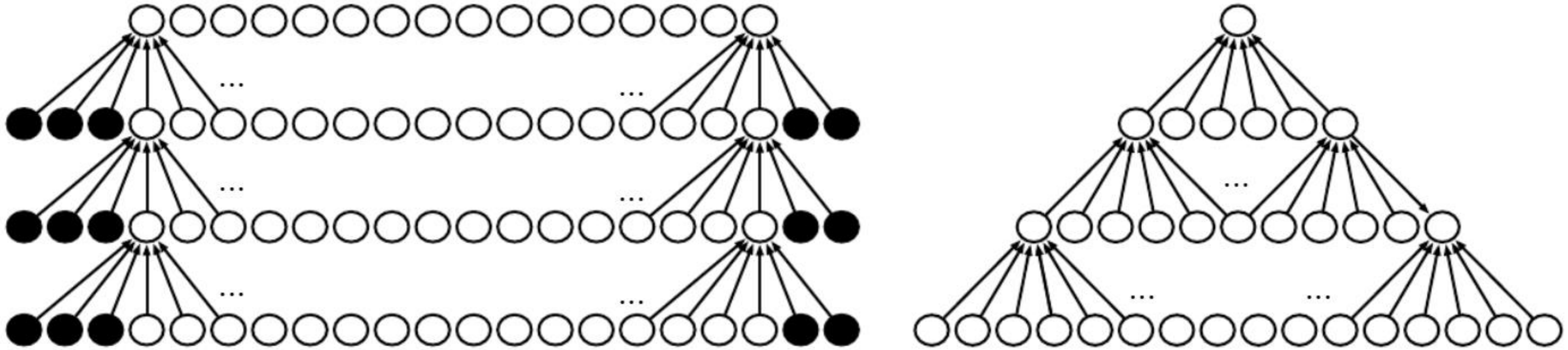
- Pooling over spatial regions produces invariance to translation
- If we pool over the outputs of separately parametrized convolutions, the features can learn which transformations to become invariant to





Padding

- The width of the representation shrinks by one pixel less than the kernel width at each layer
- Zero padding the input allows us to control the kernel width and the size of the output independently



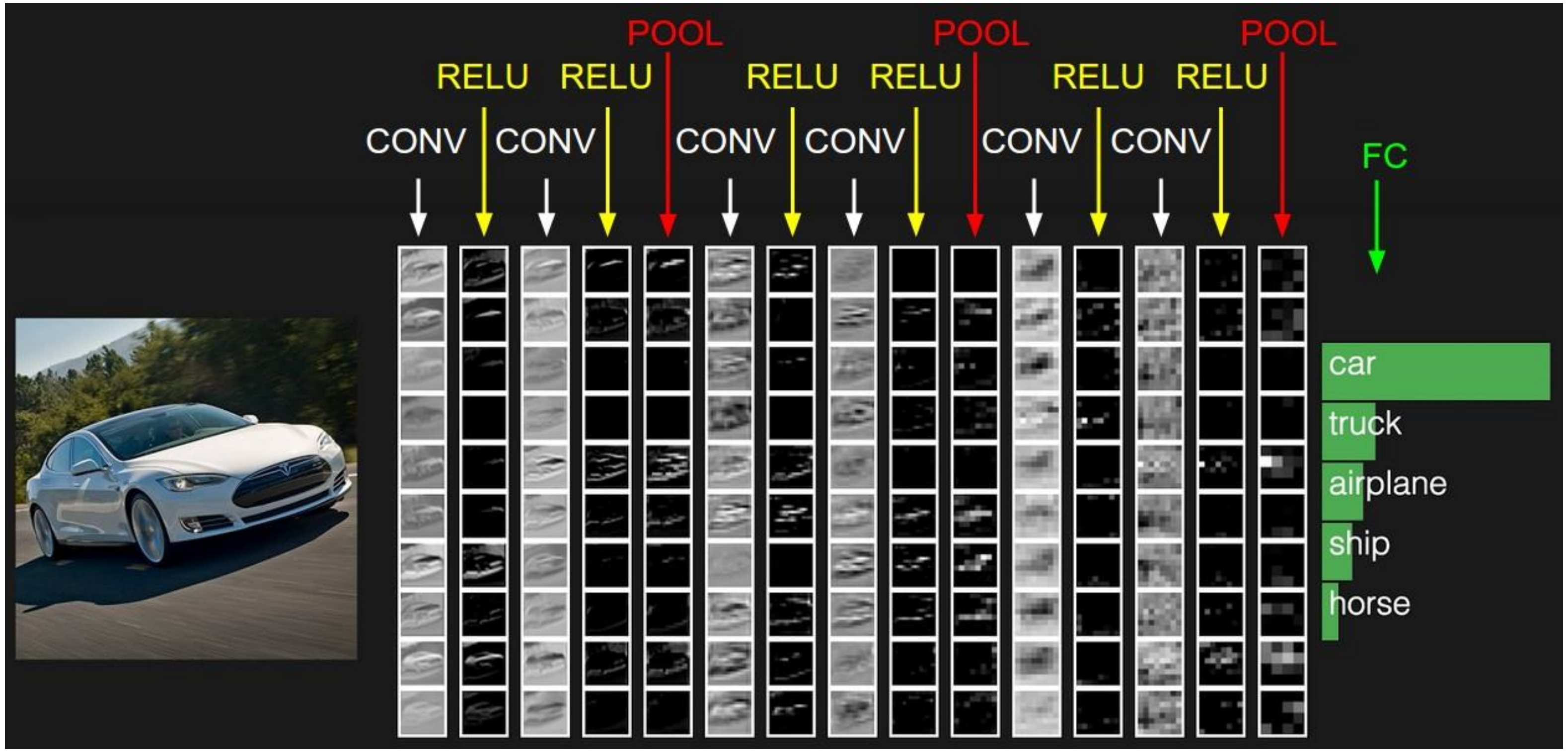
Example: cat or dog?

- A Kaggle competition from 2013
  - https://www.kaggle.com/c/dogs-vs-cats/data
  - https://drive.iust.ac.ir/index.php/s/pN26XPnjaK9DGws
  - It contains 25000 images, 12500 in each class
  - We use 4000 images in total:
    - 2000 for training (50%, 50%)
    - 1000 for validation
    - 1000 for test

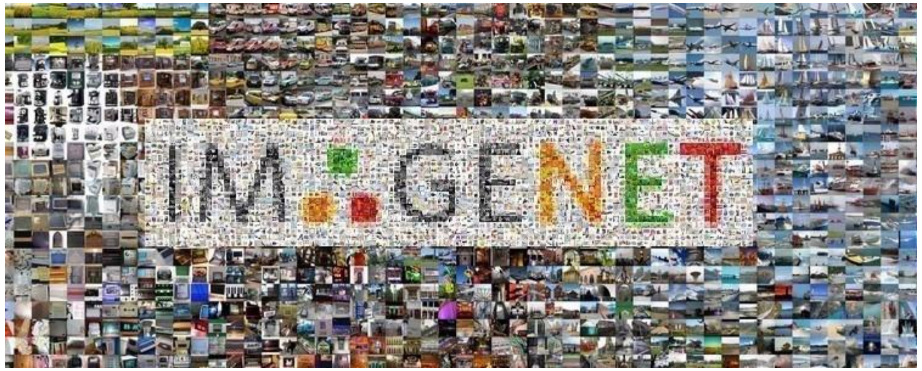
Building the network

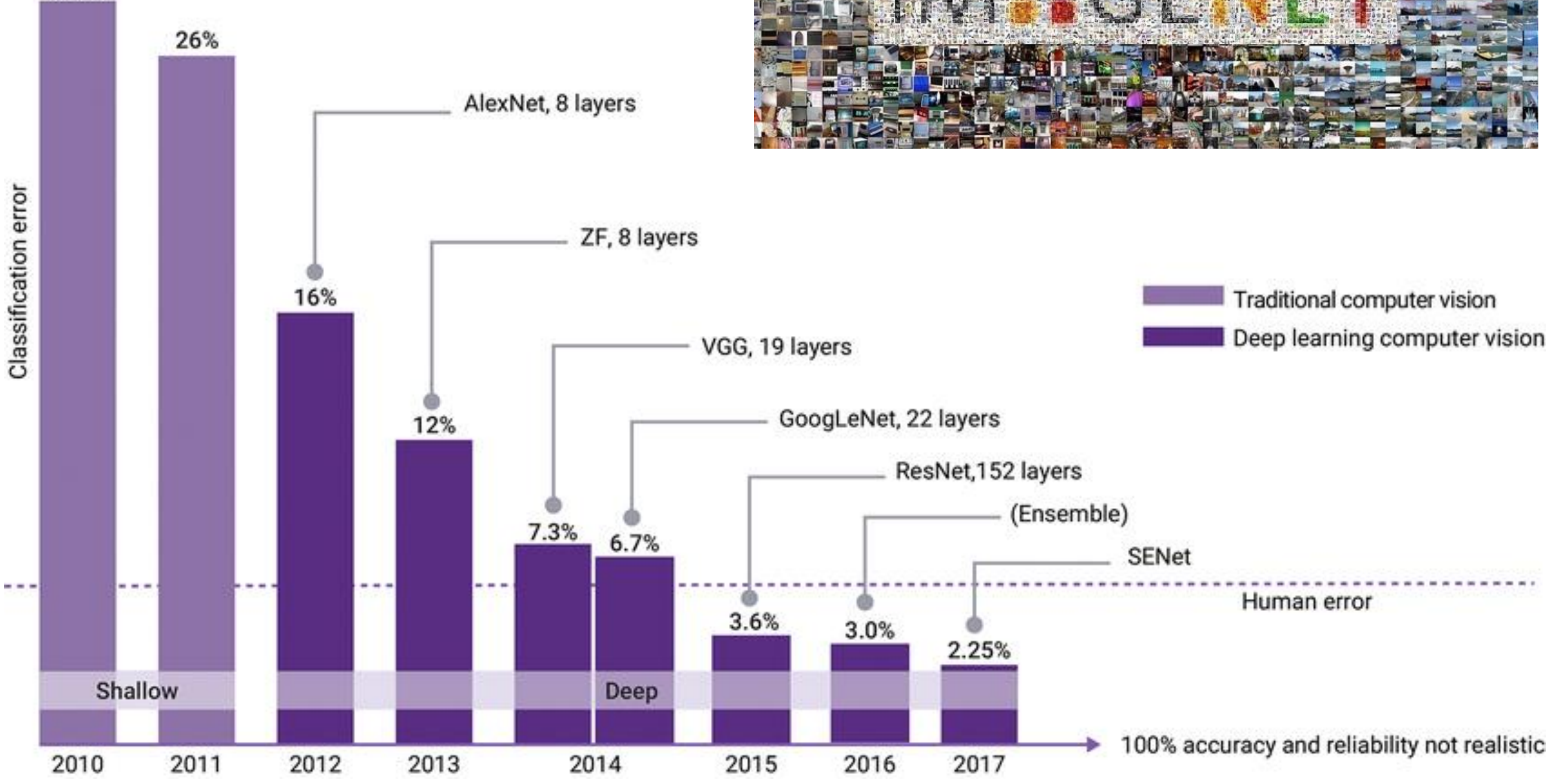
- Bigger images (than CIFAR) and a more complex problem, you'll make your network larger

- Larger input, more strides!
- Typically, the depth of the feature maps progressively increases in the network, whereas the size of the feature maps decreases



ILSVRC results





Keras Applications

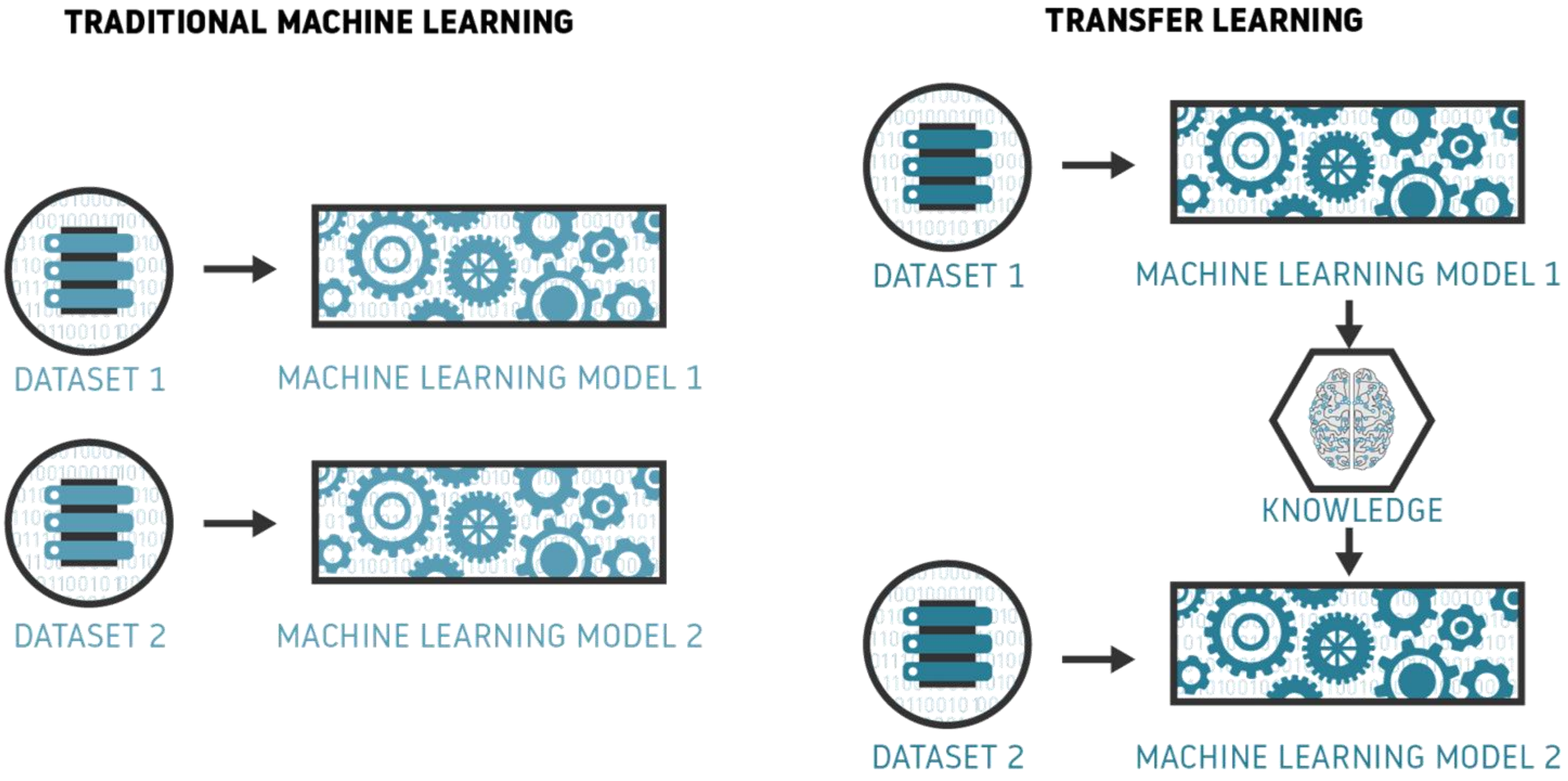
keras.io/api/applications/

- Metrics
- Losses
- Built-in small datasets
- **Keras Applications**
- Utilities
- Code examples
- Why choose Keras?
- Community & governance
- Contributing to Keras

Available models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-

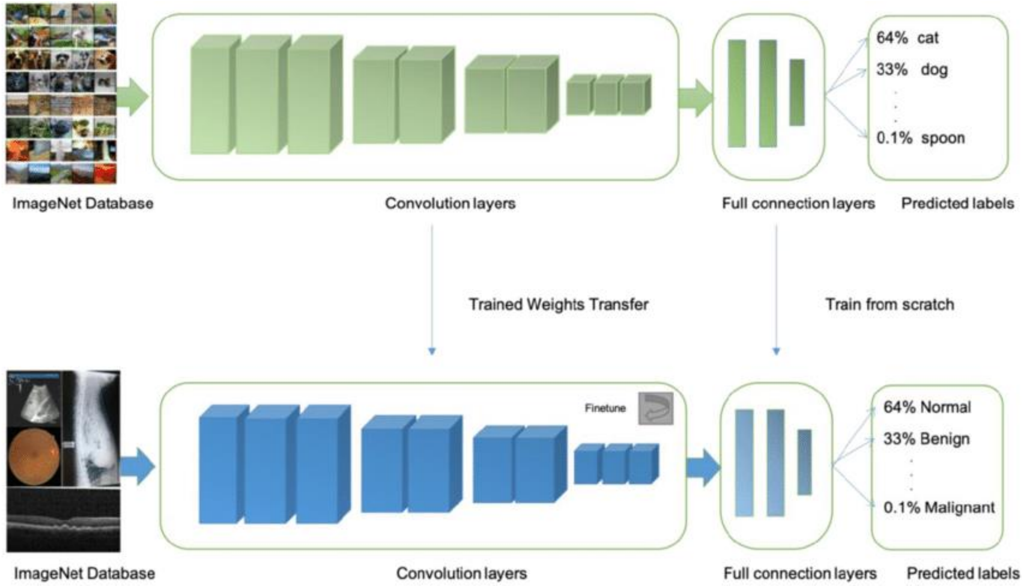
Transfer Learning



https://datascience.aero/transfer-learning-aviation/

Pretrained convnet

- A common and highly effective approach to deep learning on small image datasets is to use a pretrained network
- A pretrained network is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task

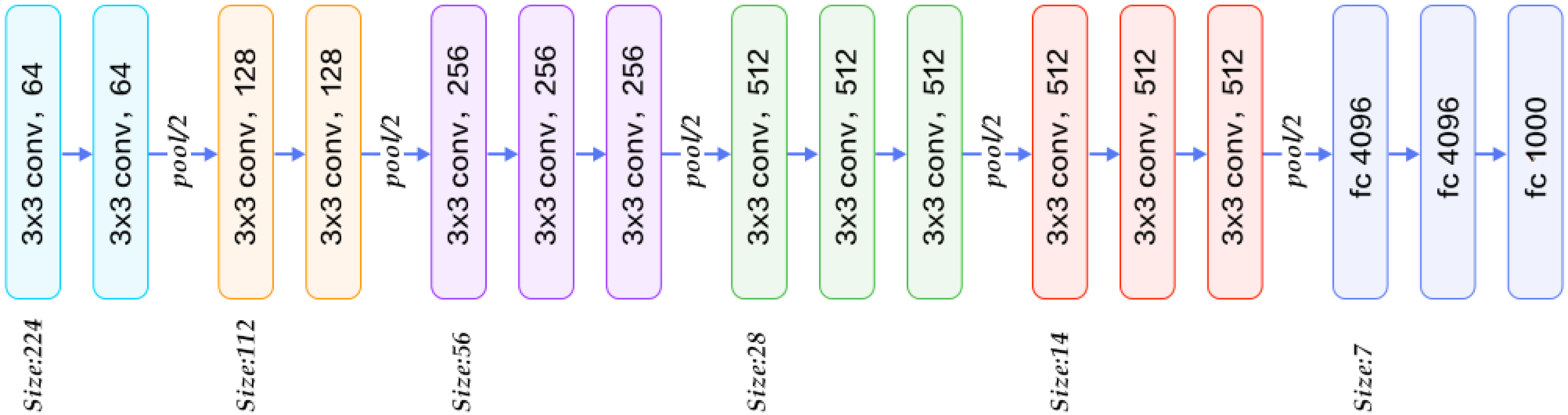


Pretrained convnet

- A common and highly effective approach to deep learning on small image datasets is to use a pretrained network
- A pretrained network is a saved network that was previously trained on a large dataset, typically on a large-scale image-classification task
- If this original dataset is large enough and general enough, then the spatial hierarchy of features learned by the pretrained network can effectively act as a generic model of the visual world
  - its features can prove useful for many different computer vision problems
- Such portability of learned features across different problems is a key advantage of deep learning compared to many older approaches

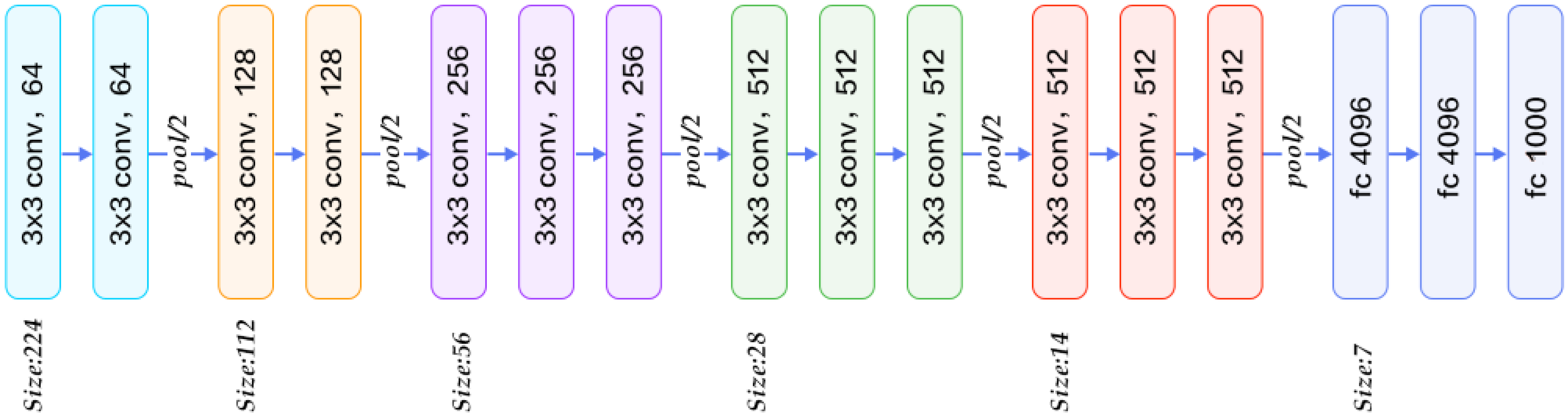
Pretrained convnet

- Let's consider a large convnet trained on the ImageNet dataset
  - 1.4 million labeled images and 1,000 different classes (including many animals)
- We expect to perform well on the dogs-versus-cats classification problem
- We'll use the VGG16 architecture



Pretrained convnet

- There are two typical ways to use a pre-trained network:
  - Feature extraction
  - Fine tuning



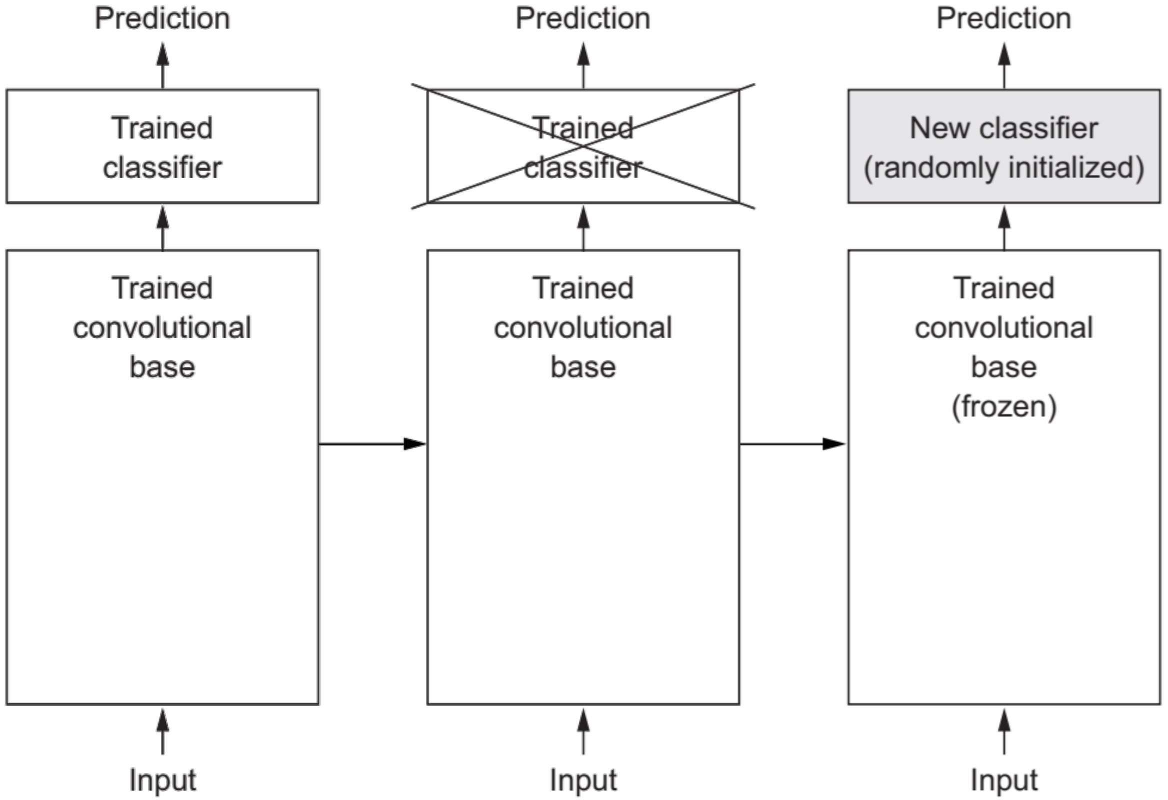
Feature extraction

- Using the representations learned by a previous network to extract interesting features from new samples
- These features are then run through a new classifier, which is trained from scratch
- ConvNets for image classification usually comprise two parts:
  - A series of pooling and convolution layers – convolutional base
  - A densely connected classifier

Feature extraction

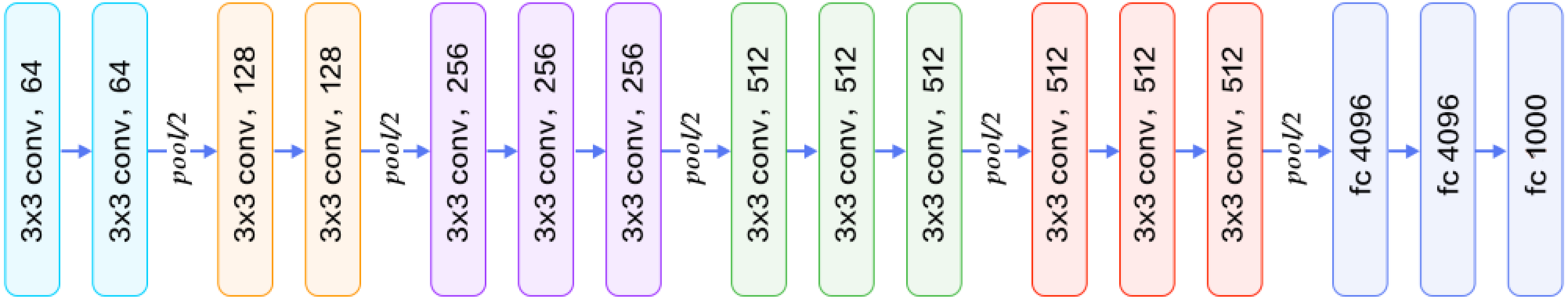
- In the case of ConvNets, feature extraction consists of taking the convolutional base of a previously trained network, running the new data through it, and training a new classifier on top of the output

- Could we reuse the densely connected classifier as well?



Feature Extraction

- The feature maps of a convnet are presence maps of generic concepts over an image
- The representations learned by the dense classifier will be specific to the set of classes on which the model was trained



Feature Extraction

- Level of generality (and therefore reusability) of the representations in ConvNets depends on the depth of the layer in the model
  - Earlier layers extract local, highly generic feature maps (such as visual edges, colors, and textures)
  - Later extract more-abstract concepts (such as "cat ear" or "dog eye")
- If the test dataset is substantially different, it is better to use only the first few layers