Introduction to Machine Learning

Module 2B: Modern Convolutional Neural Networks

Instructor: Tugce Gurbuz

July 15th 2022

Let's work on cooler CNNs!

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

Let's work on cooler CNNs!

Success -> (1) large scale CNNs and (2) transfer learning

Let's work on cooler CNNs!

Success -> **(1) large scale CNNs** and (2) transfer learning

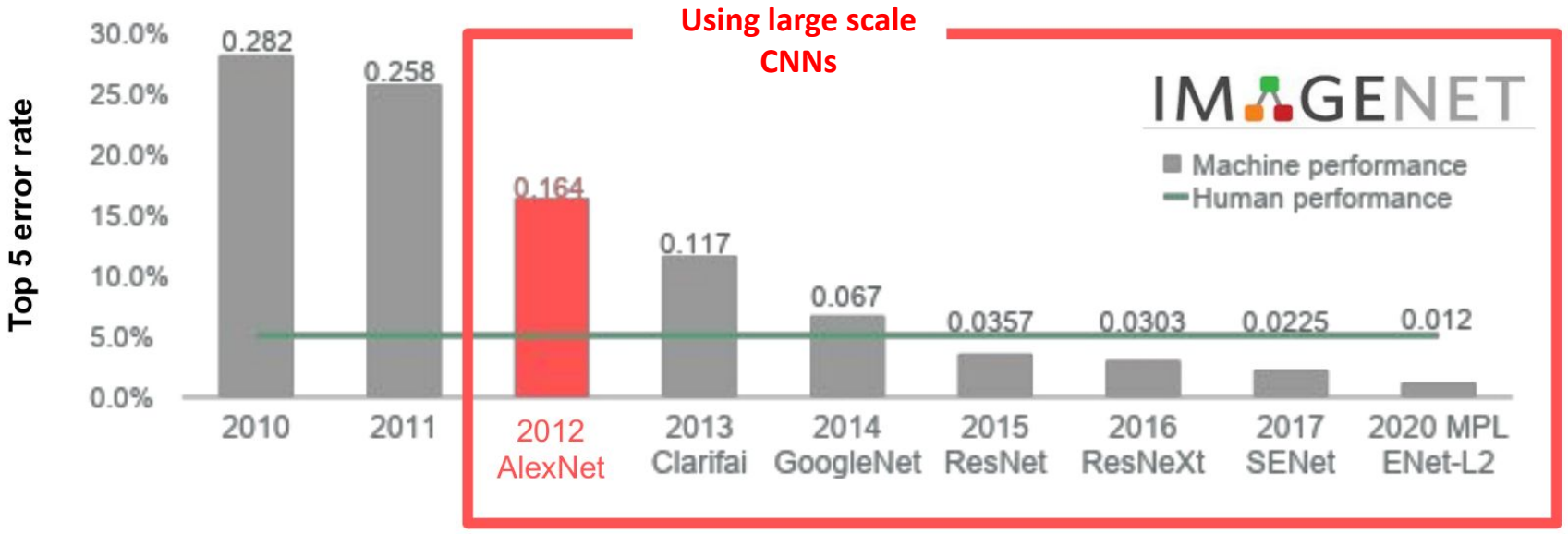
Let's work on cooler CNNs!

Success -> **(1) large scale CNNs** and (2) transfer learning

IMAGENET

Let's work on cooler CNNs!

Success -> **(1) large scale CNNs** and (2) transfer learning

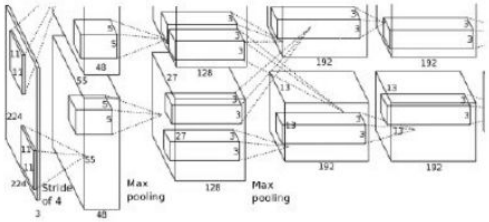


Let's work on cooler CNNs!

Success -> **(1) large scale CNNs** and (2) transfer learning

Big Data: ImageNet

+



Deep Convolutional Neural Network

+

Backprop on GPU

+ **A number of small tweaks**
Sigmoid -> ReLU, batch normalization, dropout

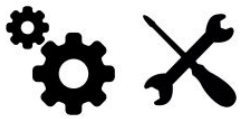
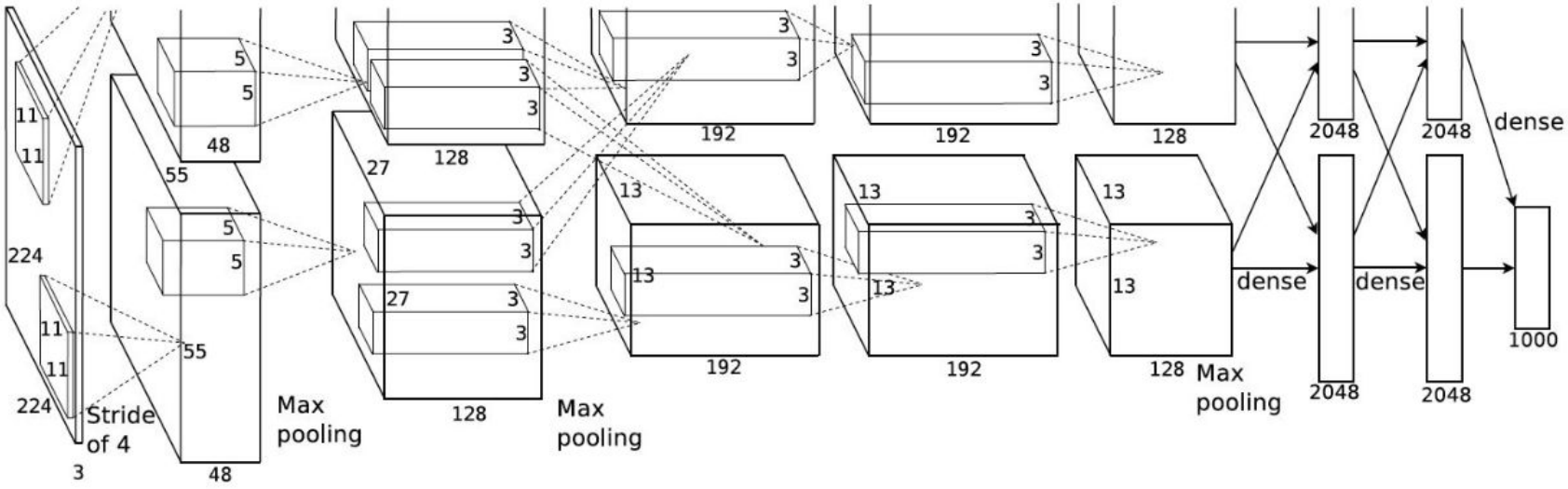


Image credit: http://www.andreykurenkov.com/writing/ai/a-brief-history-of-neural-nets-and-deep-learning-part-4/

AlexNet

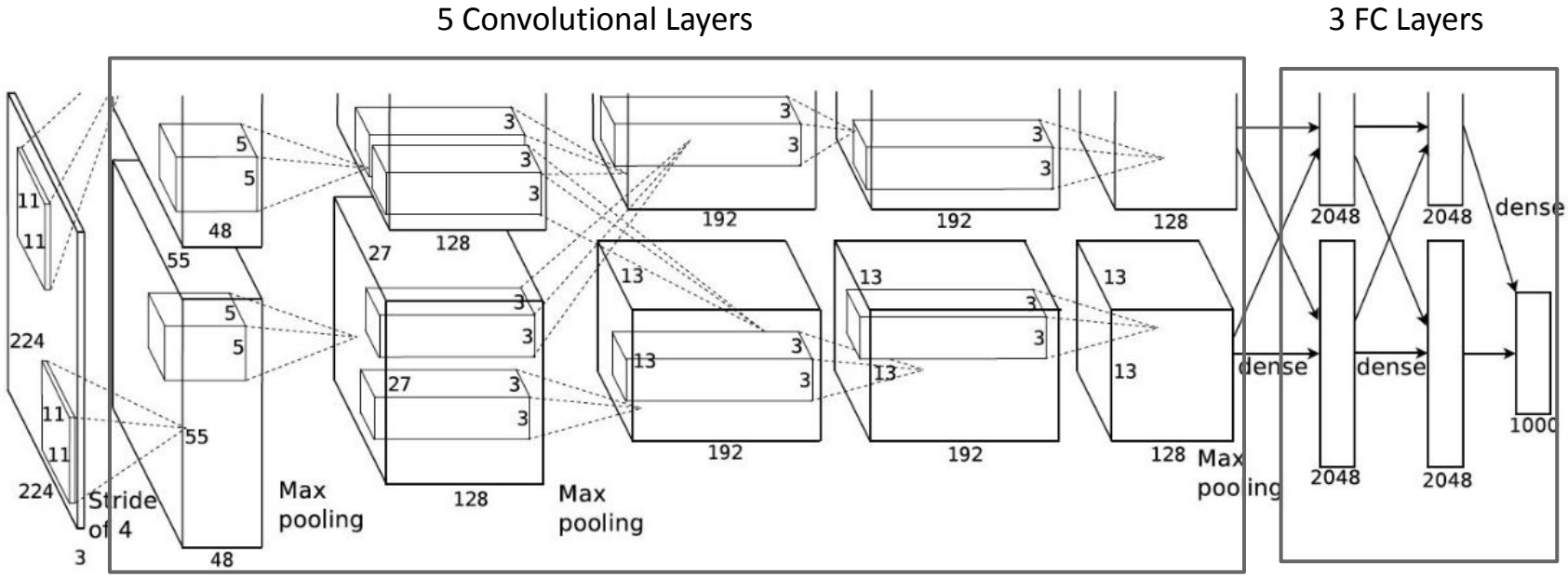


Alex Krizhevsky, Ilya Sutskeever, Geoffrey Hinton

"ImageNet classification with deep convolutional neural networks." NeurIPS 2012

```
torchvision.models.alexnet()
```

AlexNet



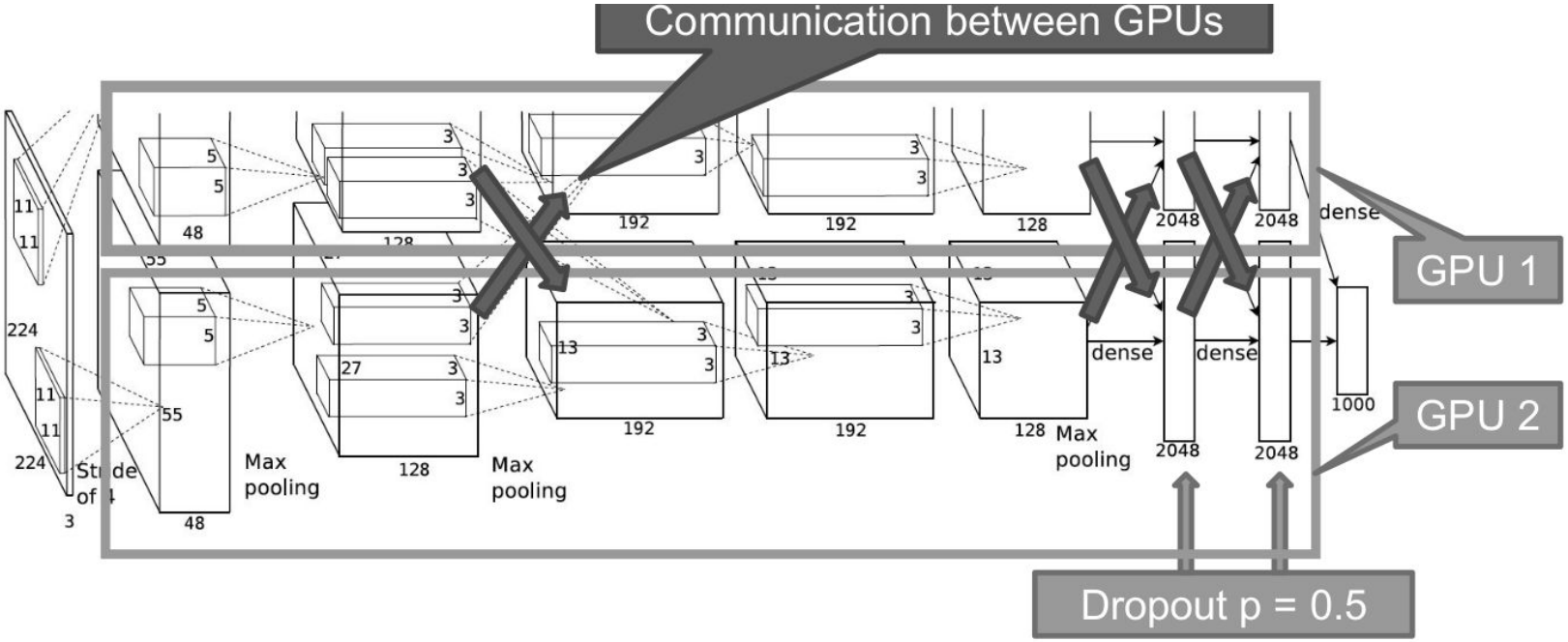
Alex Krizhevsky, Ilya Sutskeever, Geoffrey Hinton

"ImageNet classification with deep convolutional neural networks." NeurIPS 2012

- Total number of parameters: 60M
- Trained on 2 GPUs

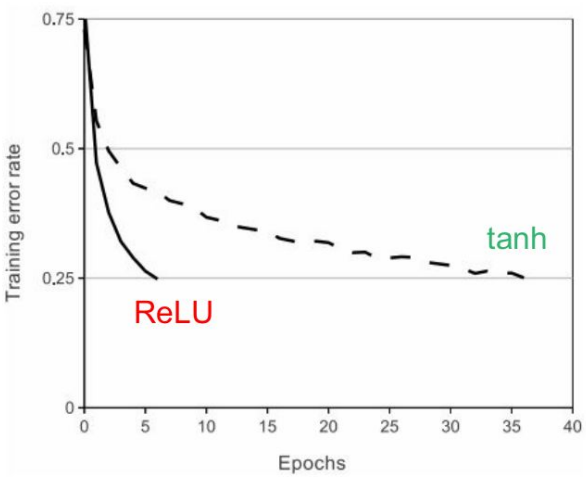
```
torchvision.models.alexnet()
```

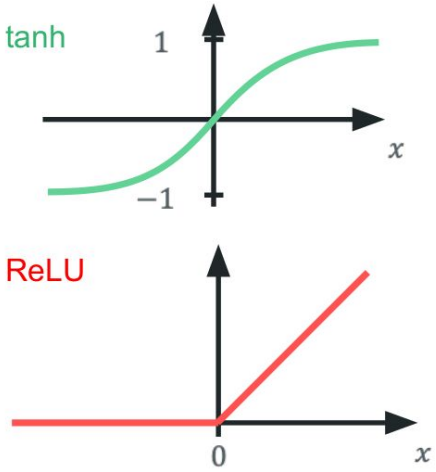
AlexNet



torchvision.models.alexnet()

AlexNet





Krizhevsky, Sutskever, Hinton, NeurIPS 2012

torchvision.models.alexnet()

AlexNet

Batch normalization -> normalizing the input batch

- If the distribution of the inputs to every layer is the same, the network is efficient.

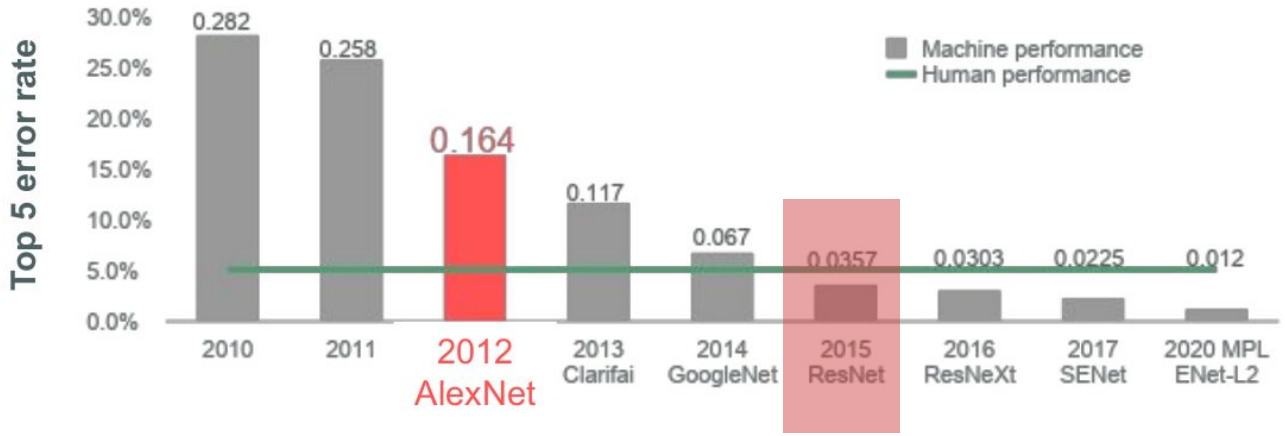
```
torchvision.models.alexnet()
```

AlexNet

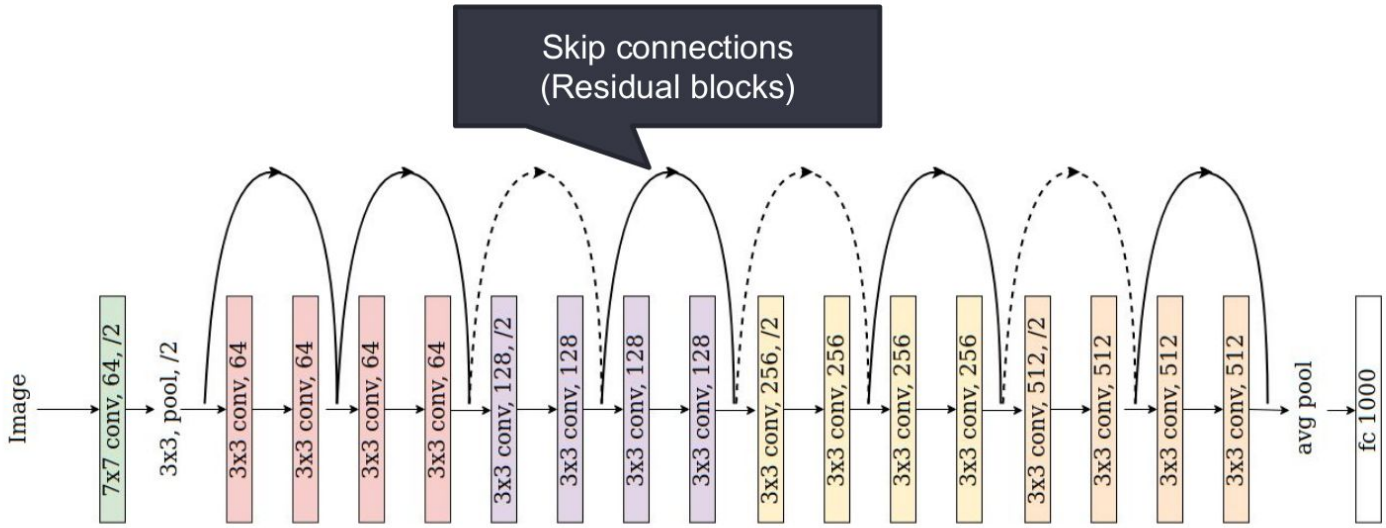
Let’s practice AlexNet in Section-1 of tutorial-1!

```
torchvision.models.alexnet()
```

ConvNets After AlexNet: ResNet



ResNet

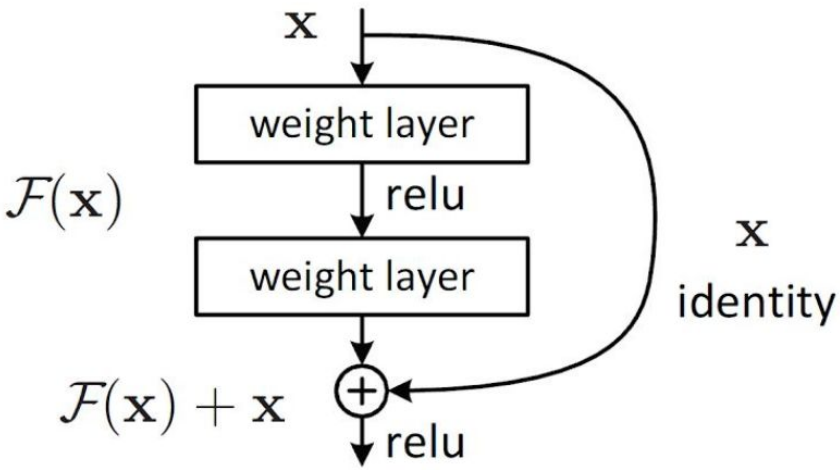


He et al., *CVPR* 2016

```
torchvision.models.resnet18()
…
torchvision.models.resnet152()
```

ResNet

“Skip connections”



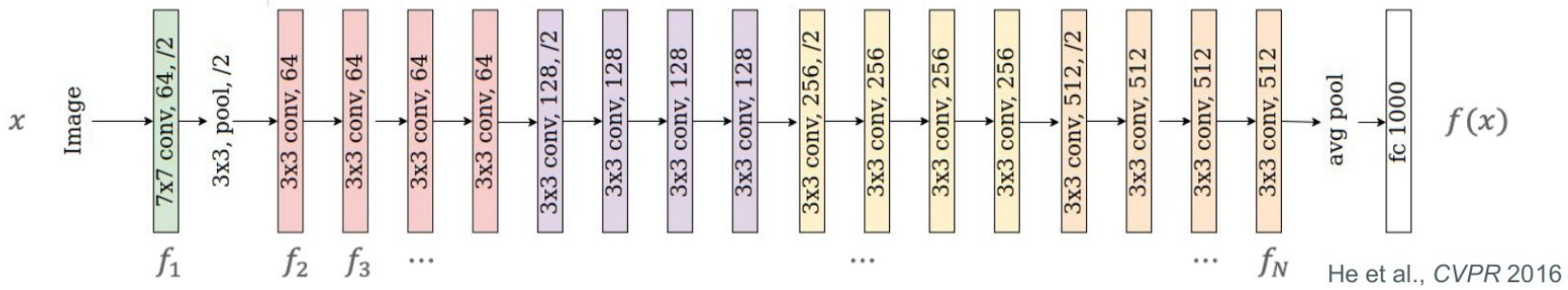
Better gradient flow because of “skip connections”

```
torchvision.models.resnet18()
…
torchvision.models.resnet152()
```

ResNet

“Skip connections” avoid vanishing gradients <3

$$f(x) = f_1\left(f_2(\cdots f_N(x))\right) \quad \Rightarrow \quad f'(x) = f_1'\left(f_2(\cdots)\right) \cdot f_2'(\cdots) \cdot \ldots \cdot f_N'(x) \qquad \textit{(chain rule)}$$

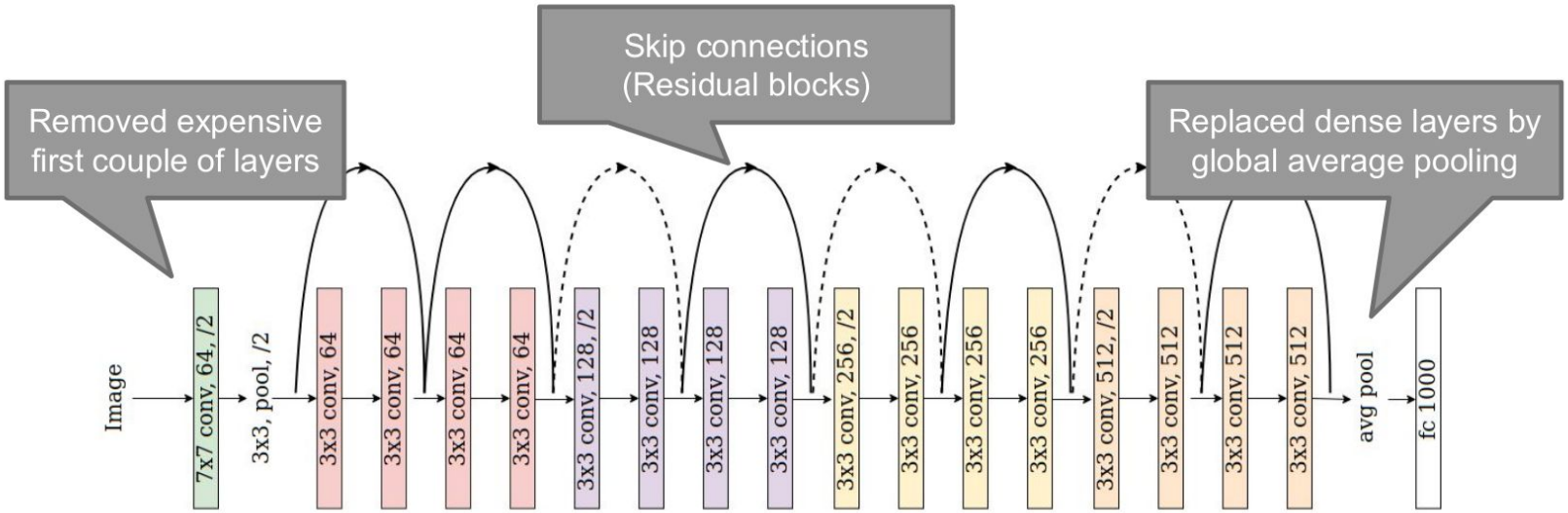


He et al., *CVPR* 2016

```
torchvision.models.resnet18()
…
torchvision.models.resnet152()
```

ResNet

“Skip connections” avoid vanishing gradients <3



He et al., *CVPR* 2016

```
torchvision.models.resnet18()
…
torchvision.models.resnet152()
```

ResNet

Let’s practice ResNet in Section-2 of tutorial-1!

```
torchvision.models.resnet18()
…
torchvision.models.resnet152()
```

Transfer Learning

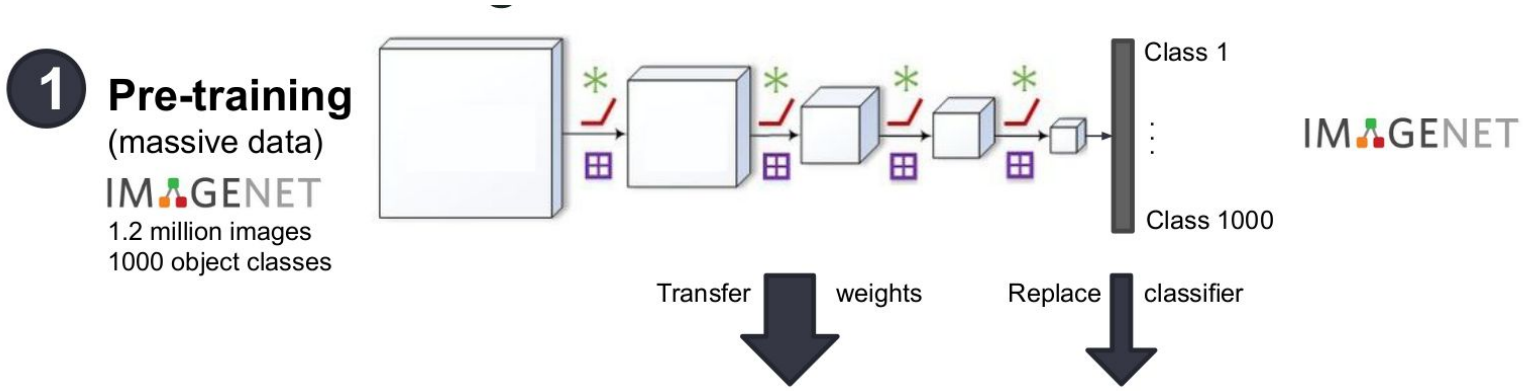
Success -> (1) large scale CNNs and **(2) transfer learning**

Transfer Learning

Success -> (1) large scale CNNs and **(2) transfer learning**

Transfer Learning

Success -> (1) large scale CNNs and **(2) transfer learning**



Transfer Learning

Success -> (1) large scale CNNs and **(2) transfer learning**

Transfer Learning

Success -> (1) large scale CNNs and **(2) transfer learning**

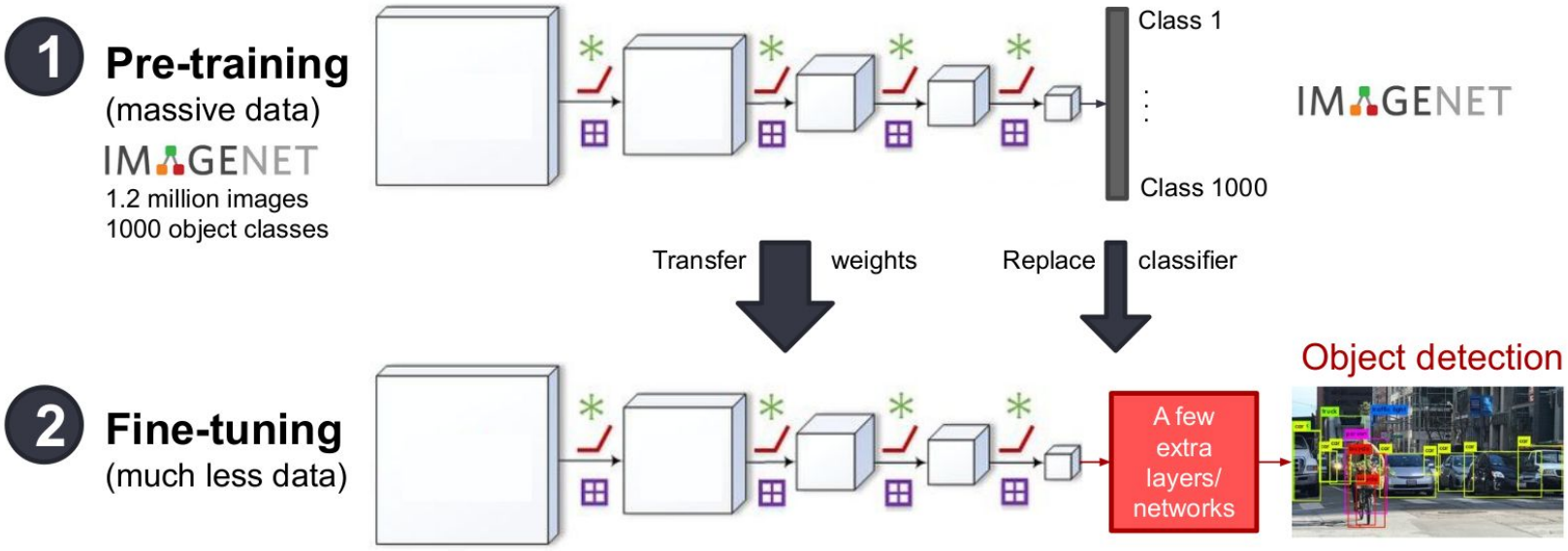
Option 1: Train only classification layer, freeze backbone
(sometimes referred to as the “linear evaluation protocol”)

- Fast & simple

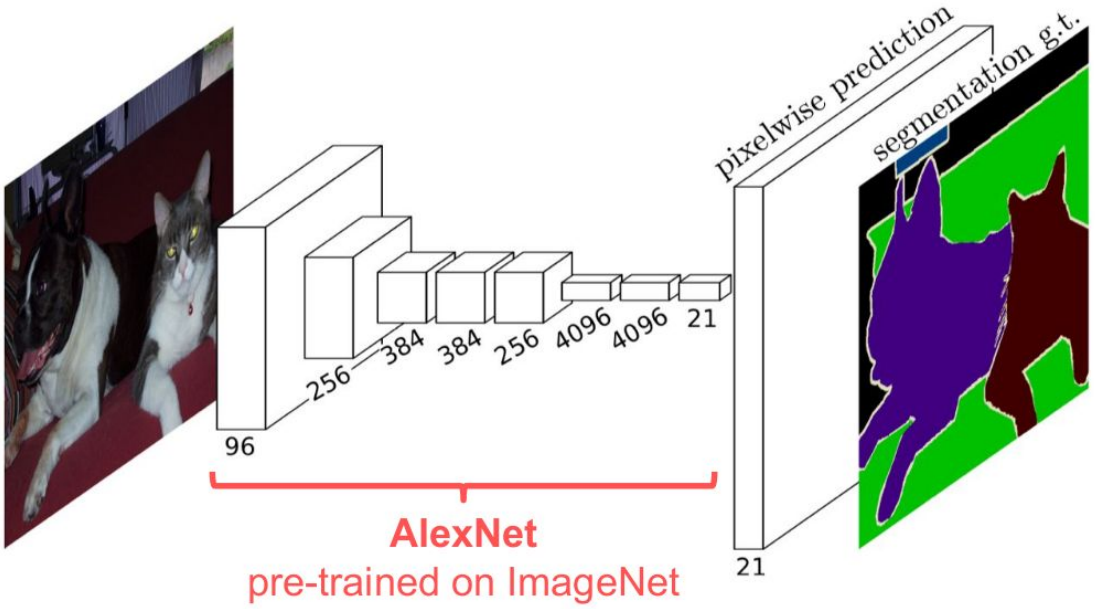
Option 2: Train classification layer, fine-tune backbone at the same time

- Slower, but can adapt feature extraction to dataset statistics

Transfer Learning Is Beyond Classification



Transfer Learning Is Beyond Classification



Long, Shelhamer, Darrell, *CVPR* 2015

Transfer Learning

Let’s practice transfer learning on the last section of the tutorial-1!

Then, we will continue with tutorial-2 to perform face recognition with modern CNNs and also have a discussion about ethical part of AI