## Предобученные нейронные сети для задач CV

#### Цель темы

Узнать, какие базовые модели решают основные задачи компьютерного зрения, и научиться использовать их в построении решений для своих задач.

#### LeNet5

```
LeNet
 Image: 28 (height) × 28 (width) × 1 (channel)
Convolution with 5×5 kernel+2padding:28×28×6
                      √ sigmoid
Pool with 2×2 average kernel+2 stride:14×14×6
Convolution with 5×5 kernel (no pad):10×10×16
                      √ sigmoid
Pool with 2×2 average kernel+2 stride: 5×5×16
                      √ flatten
     Dense: 120 fully connected neurons
                      √ sigmoid
     Dense: 84 fully connected neurons
                     √ sigmoid
     Dense: 10 fully connected neurons
           Output: 1 of 10 classes
```

Схема: Cmgle / en.wikipedia.org

#### **AlexNet**

LeNet	AlexNet
Image: 28 (height) × 28 (width) × 1 (channel)	Image: 224 (height) × 224 (width) × 3 (channels)
Convolution with 5×5 kernel+2padding:28×28×6	Convolution with 11×11 kernel+4 stride: 54×54×96
√ sigmoid	√ ReLu
Pool with 2×2 average kernel+2 stride:14×14×6	Pool with 3×3 max. kernel+2 stride: 26×26×96
Convolution with $5\times5$ kernel (no pad): $10\times10\times16$	Convolution with 5×5 kernel+2 pad:26×26×256
√ sigmoid	√ ReLu
Pool with 2×2 average kernel+2 stride: 5×5×16	Pool with 3×3 max.kernel+2stride:12×12×256
√ flatten	$\downarrow$
Dense: 120 fully connected neurons	Convolution with 3×3 kernel+1 pad:12×12×384
√sigmoid	√ ReLu
Dense: 84 fully connected neurons	Convolution with 3×3 kernel+1 pad:12×12×384
√ sigmoid	√ ReLu
Dense: 10 fully connected neurons	Convolution with 3×3 kernel+1 pad:12×12×256
$\downarrow$	√ ReLu
Output: 1 of 10 classes	Pool with 3×3 max.kernel+2stride:5×5×256
	$\sqrt{flatten}$
	Dense: 4096 fully connected neurons
	√ ReLu, dropout p=0.5
	Dense: 4096 fully connected neurons
	√ ReLu, dropout p=0.5
	Dense: 1000 fully connected neurons
	<u> </u>
	Output: 1 of 1000 classes

Cxeмы: Cmgle / <u>en.wikipedia.org</u>

Полезная литература: <u>ImageNet classification with deep convolutional neural networks</u>

## ImageNet

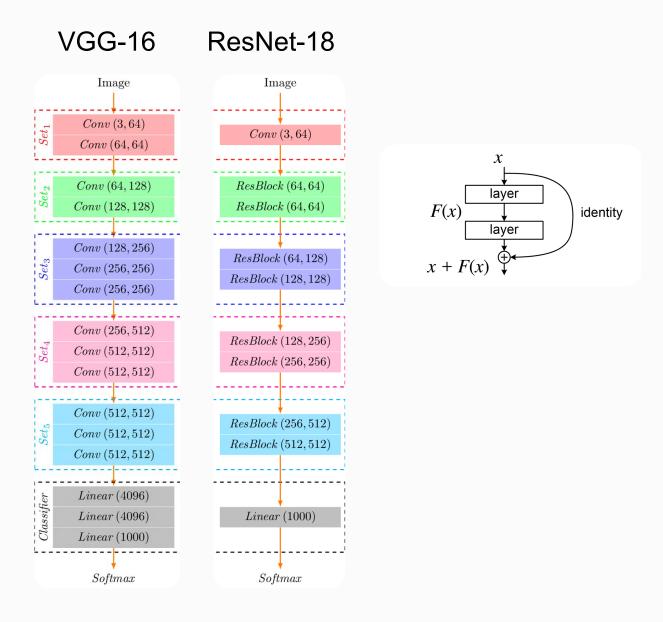


Схема: <u>researchgate.net</u>

Полезная литература: <u>ImageNet Large Scale Visual Recognition Challenge</u>; <u>Very deep convolutional networks for large-scale image recognition</u>

## ImageNet

#### **MobileNet**

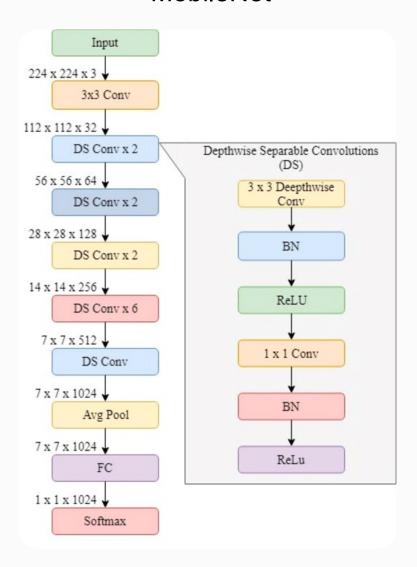


Схема: <u>researchgate.net</u>

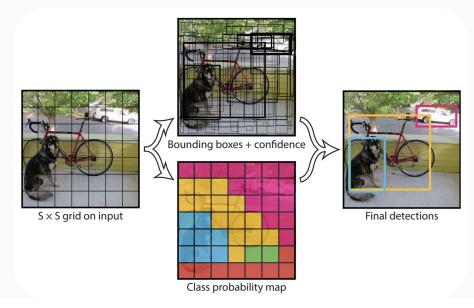
Полезная литература: Mobilenets: Efficient convolutional neural networks for mobile vision applications

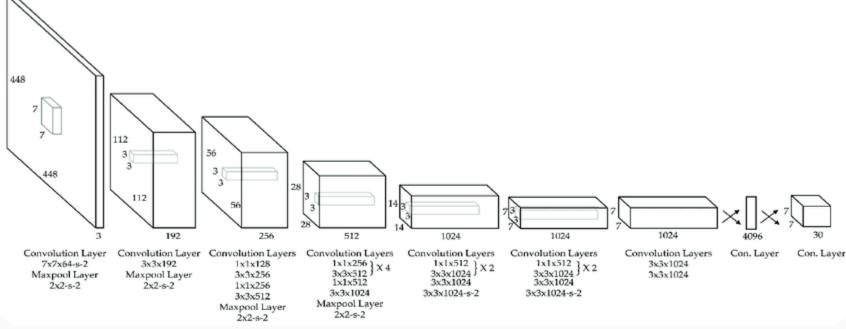
# Использование предобученных моделей

Transfer learning — замена и обучение выходного слоя модели для решения другой задачи.

**Finetuning** — дообучение модели или её части на новых данных для решения такой же задачи в другом домене.

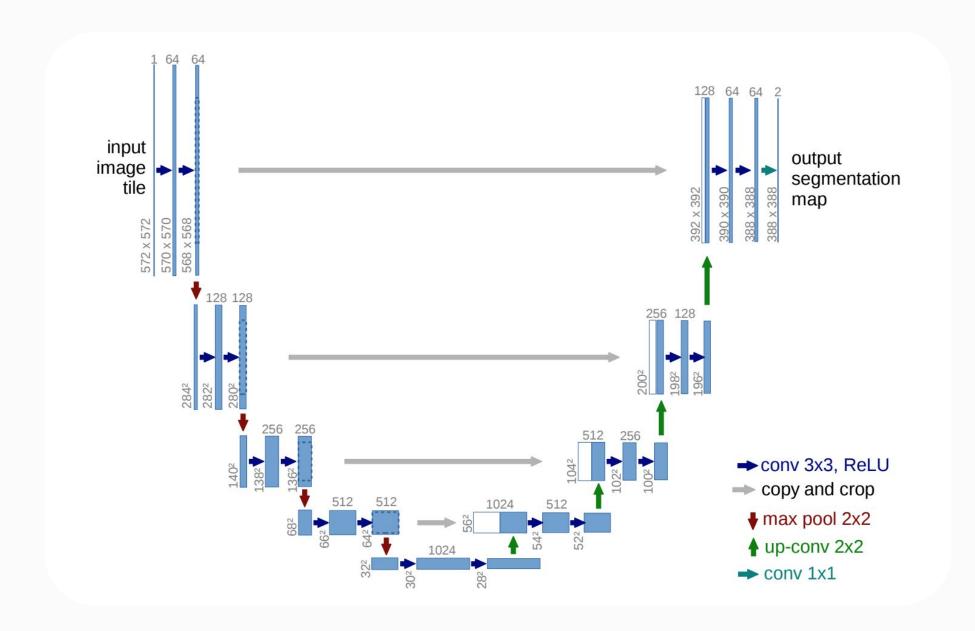
#### Детекция — YOLO





Схемы: Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi / <u>arxiv.org</u>, <u>researchgate.net</u> Полезная литература: <u>You only look once: Unified, real-time object detection</u>

#### Сегментация – UNet



### Генерация

#### **StyleGAN**

#### Latent $\mathbf{z} \in \mathcal{Z}$ Noise Synthesis network gNormalize Const 4×4×512 Mapping network fAdaIN FC Conv 3×3 В ← FC AdaIN FC FC FC Upsample FC Conv 3×3 FC В ← AdaIN Conv 3×3 ⊕← В ← $\mathbf{w} \in \mathcal{W}$ AdaIN

#### Diffusion

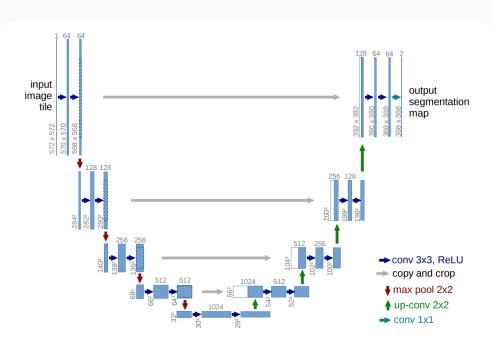


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Схемы: Tero Karras, Samuli Laine, Timo Aila / <u>arxiv.org</u>, Olaf Ronneberger, Philipp Fischer, and Thomas Brox / <u>arxiv.org</u>

Полезная литература: <u>High-Resolution Image Synthesis with Latent Diffusion Models</u>

### Ресурсы

- Papers With Code: The latest in Machine Learning
- <u>arXiv.org e-Print archive</u>
- Google Scholar

## Выводы модуля

- Вы научились :
  - готовить пайплайны обучения нейросетей
  - строить и обучать свои собственные свёрточные сети
  - использовать готовые решения