

Лекция №11

# Введение в машинное обучение

Спасёнов Алексей

# Лекция 11



---

Не забываем отмечаться и оставлять отзывы



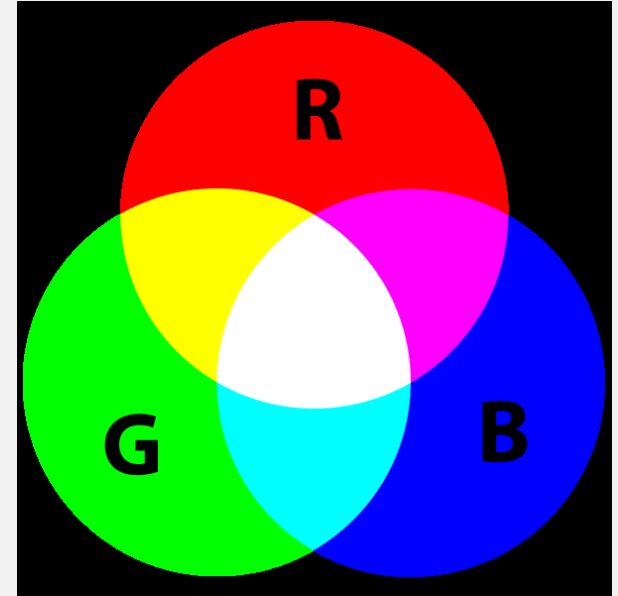
## Содержание лекции

1. Задачи обработки изображений
2. Свёрточные нейронные сети
3. Свёрточные слои, операции пулинга
4. Регуляризации нейронных сетей. Dropout
5. Transfer learning



Параметры изображения:

1. **Размеры:**  $(w, h)$
2. **Каналы:**  $c$
3. **Динамический диапазон:**  $d$



Типичные изображения:

**Цветные**(RGB)  $-(w, h, 3)$ , 8 bit

**Черно-белые**(Grayscale)  $-(w, h)$  или  $(w, h, 1)$ , 8 bit

# Задачи обработки изображений

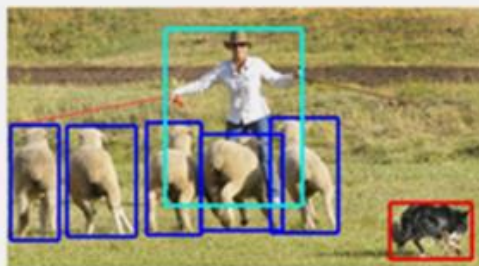


Основные задачи обработки изображений:

1. Классификация объектов
2. Детектирование объектов
3. Сегментация объектов



(a) classification



(b) detection



(c) segmentation

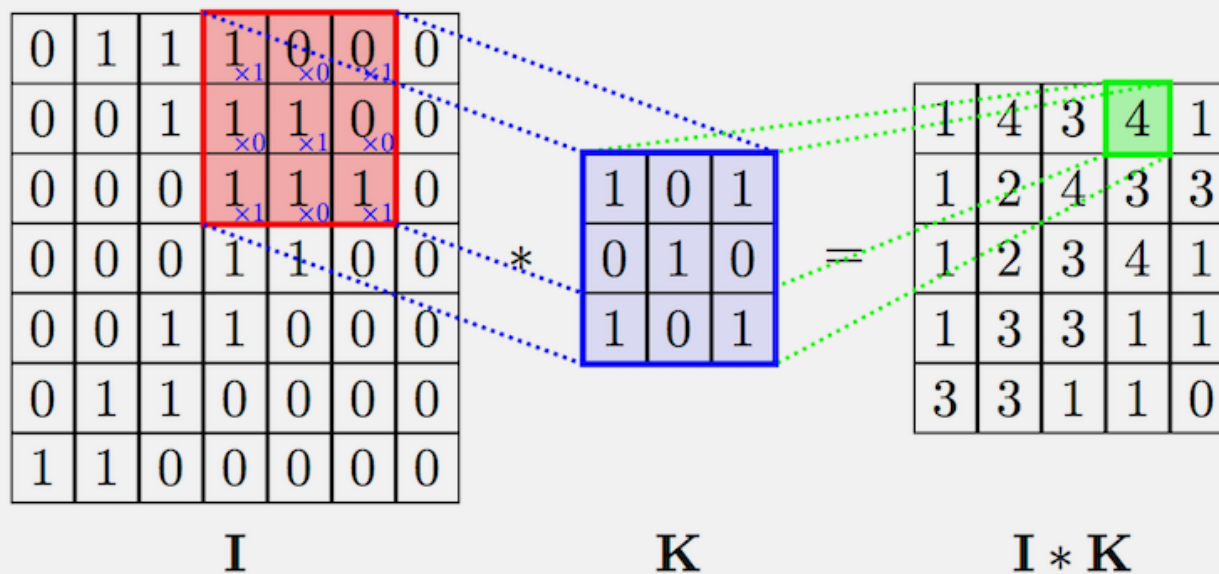
# Фильтрация, операция свертки



Дано:

Исходное изображение **I**, массив (w, h)

Ядро свертки **K**, массив (x, y),  $x < w$ ,  $y < h$



# Поиск границ, оператор Собеля (пример фильтрации)



Идея: посчитаем градиент яркости изображения

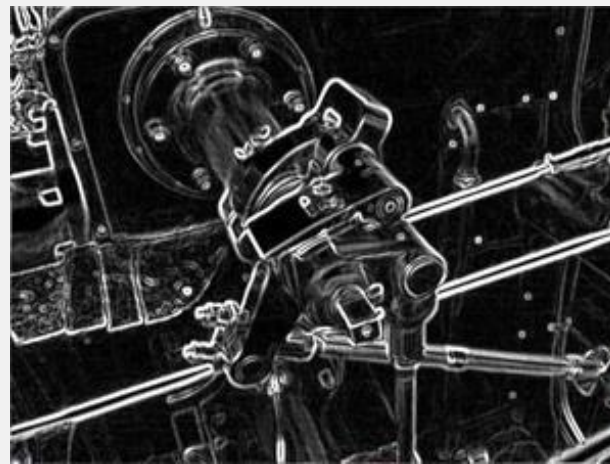
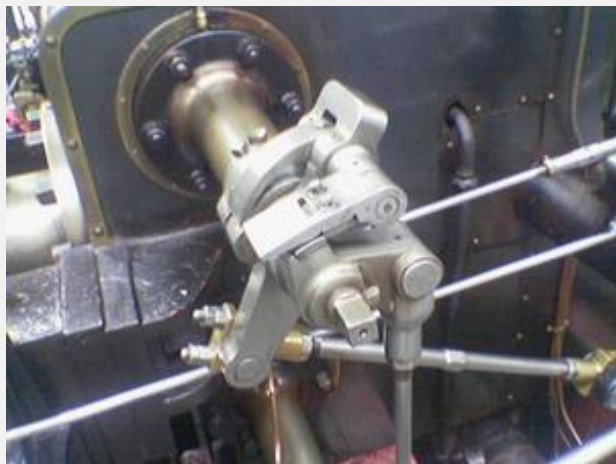
$$\mathbf{G}_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * \mathbf{A}$$

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

# Оператор Собеля , поиск границ



Пример работы оператора Собеля

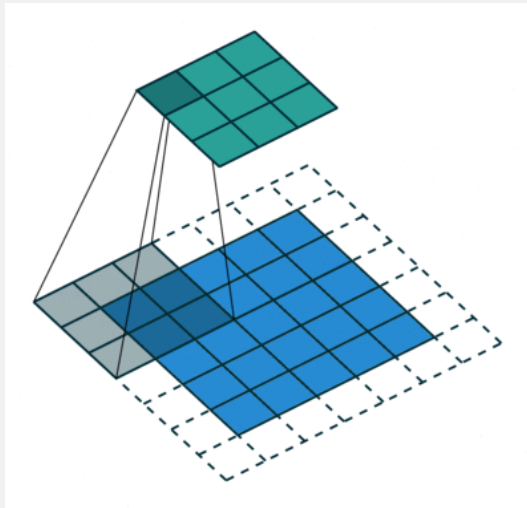




# Padding + операция свертки



## Convolution



Input		Kernel		Output																																													
<table><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>2</td><td>0</td></tr><tr><td>0</td><td>3</td><td>4</td><td>5</td><td>0</td></tr><tr><td>0</td><td>6</td><td>7</td><td>8</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	1	2	0	0	3	4	5	0	0	6	7	8	0	0	0	0	0	0	*	<table><tr><td>0</td><td>1</td></tr><tr><td>2</td><td>3</td></tr></table>	0	1	2	3	=	<table><tr><td>0</td><td>3</td><td>8</td><td>4</td></tr><tr><td>9</td><td>19</td><td>25</td><td>10</td></tr><tr><td>21</td><td>37</td><td>43</td><td>16</td></tr><tr><td>6</td><td>7</td><td>8</td><td>0</td></tr></table>	0	3	8	4	9	19	25	10	21	37	43	16	6	7	8	0
0	0	0	0	0																																													
0	0	1	2	0																																													
0	3	4	5	0																																													
0	6	7	8	0																																													
0	0	0	0	0																																													
0	1																																																
2	3																																																
0	3	8	4																																														
9	19	25	10																																														
21	37	43	16																																														
6	7	8	0																																														

**Padding** добавляет к краям поддельные пиксели (обычно нулевого значения).

Таким образом, ядро при проскальзывании позволяет неподдельным пикселям оказываться в своем центре, а затем распространяется на поддельные пиксели за пределами края, создавая выходную матрицу того же размера, что и входная.

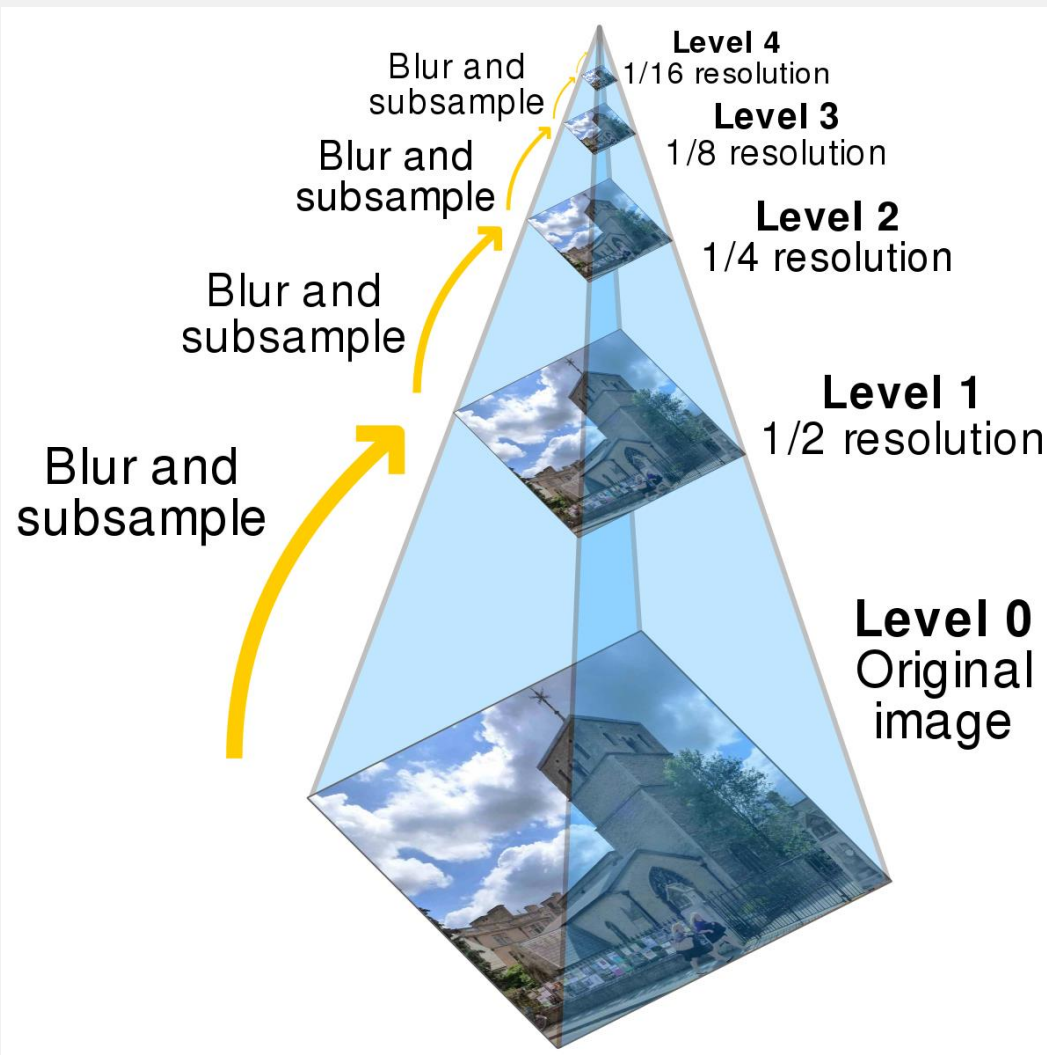
# Template Matching



Проходим по изображению сравнивая эталонное изображение с фрагментами исследуемого каким-либо образом



# Проблема масштаба, пирамиды изображений



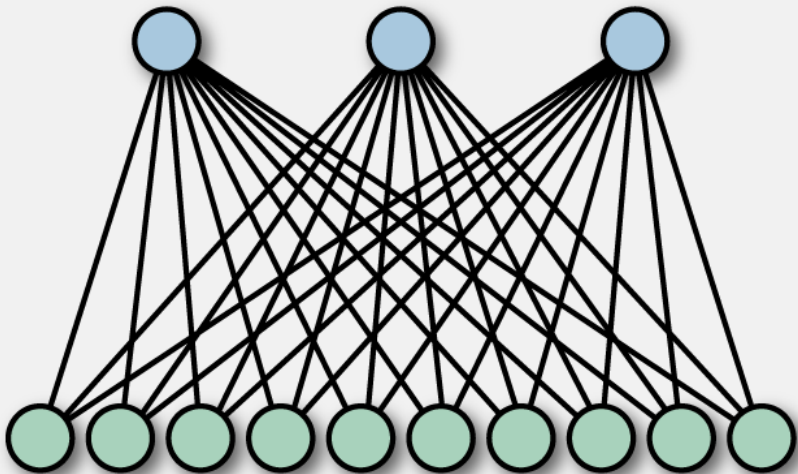
Пирамиды изображений позволяют выполнять Template Matching в различных масштабах

# Свёрточные слои

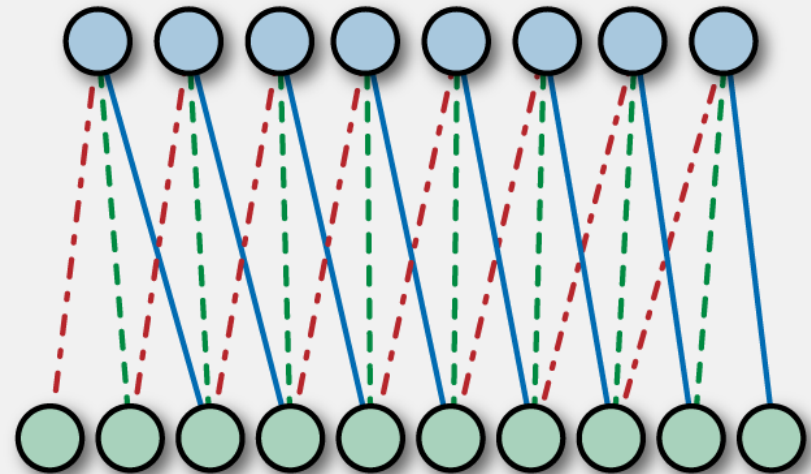


**Свёрточный слой** - это **полносвязный слой** в котором мы игнорируем не локальные связи, а локальные связи считаем общими для всех нейронов

Fully Connected



Convolutional Layer

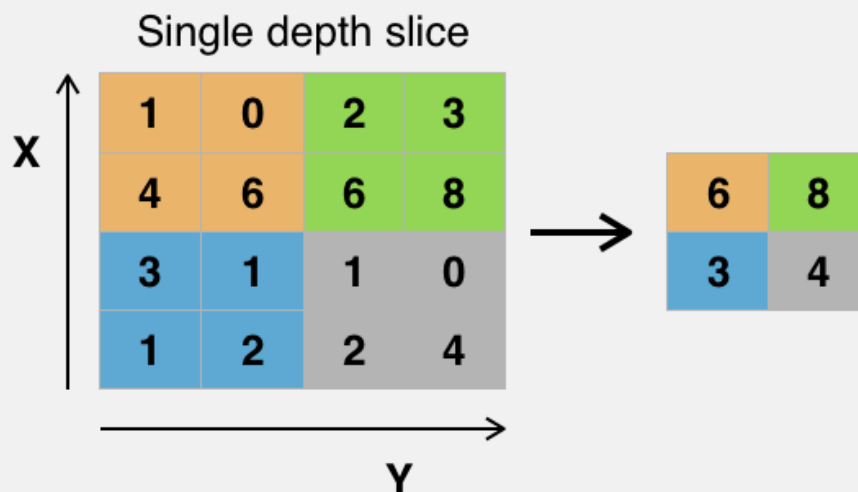


# Слой пулинга



Слой **пулинга** (pooling) -это слой снижения размерности карт признаков

Для снижения размерности как правило используют функцию max



# Нейронные сети. CNN. LeNet-5



Основные параметры сети:

- Перед подачей в сеть изображения стандартизировали
- Свертки 5x5
- Пулинг с функцией sum
- Сигмоидальная функция активации

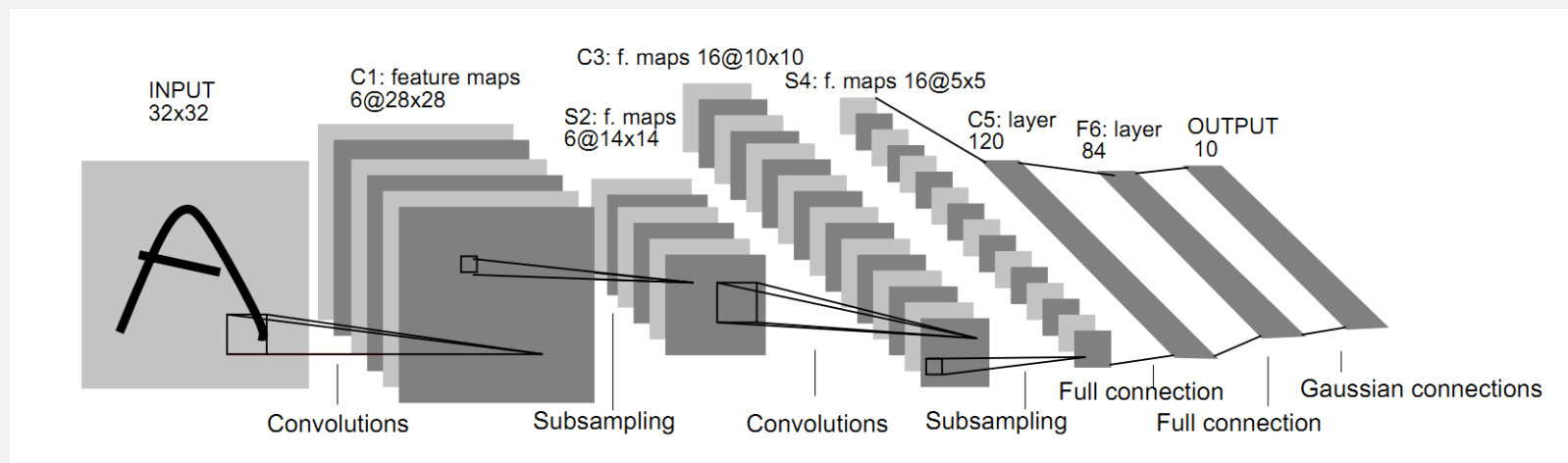
LeCun et al., 1998

<http://yann.lecun.com/exdb/lenet/>

# Нейронные сети. CNN. LeNet-5



## Архитектура сети:



Карты признаков

# Что такое ImageNet?



**ImageNet** - проект, направленный на (ручную) маркировку и категоризацию изображений в примерно 22 000 отдельных категорий объектов для задач исследований в области компьютерного зрения.

В рамках проекта **ImageNet** ежегодно проводится соревнование ImageNet Large Scale Visual Recognition Challenge (**ILSVRC**)













**Задача соревнования** - построить модель, для классификации изображений в 1000 отдельных категорий объектов. Эти 1000 категорий изображений представляют классы предметов, с которыми мы сталкиваемся в повседневной жизни, такие как виды собак, кошек, различные предметы быта, типы транспортных средств и многое другое.

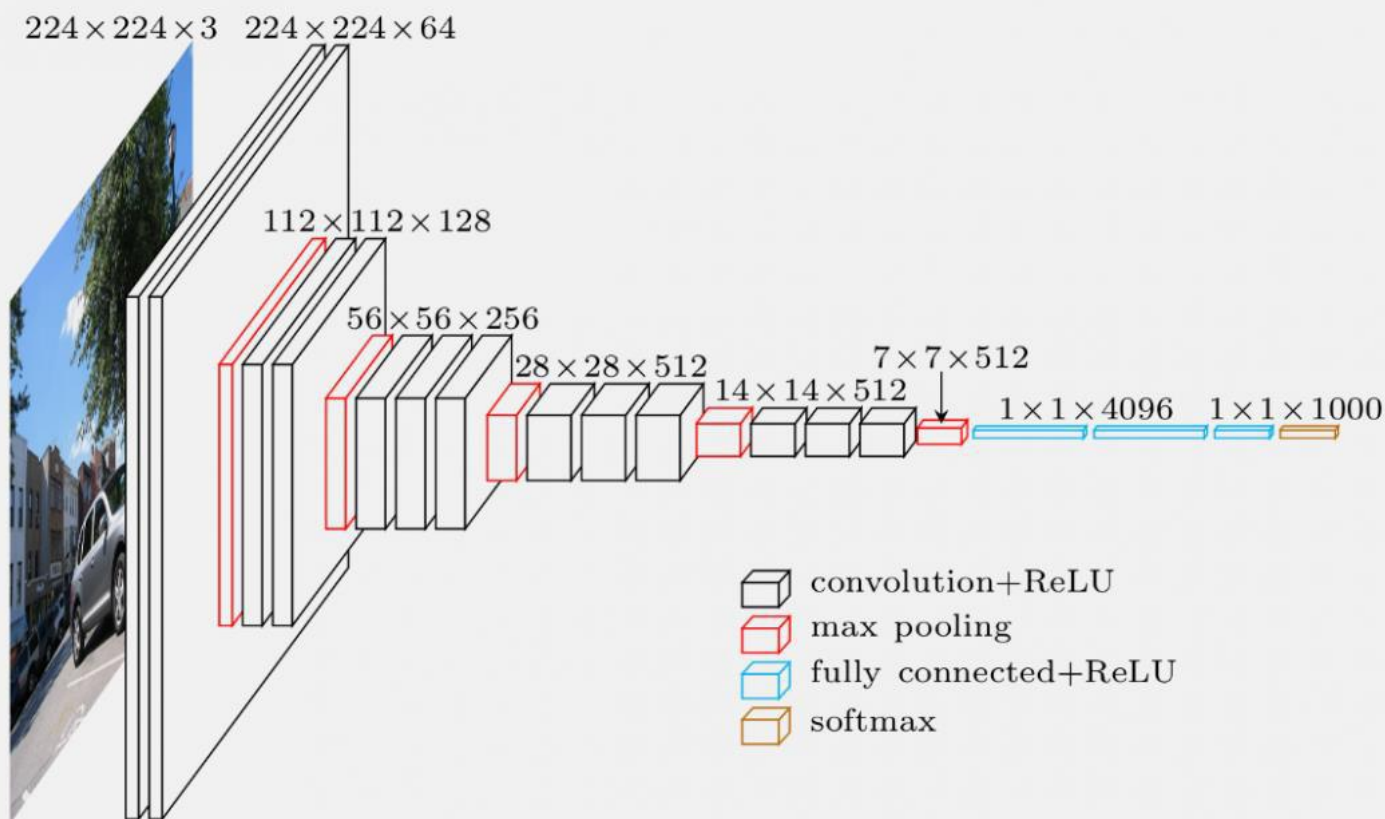
1. **Train:** ~1.2 миллиона изображений
2. **Test:** ~100 тысяч изображений



			
<b>mite</b>	<b>container ship</b>	<b>motor scooter</b>	<b>leopard</b>
<div> <div></div> <div>mite</div> <div>black widow</div> <div>cockroach</div> <div>tick</div> <div>starfish</div> </div>	<div> <div></div> <div>container ship</div> <div>lifeboat</div> <div>amphibian</div> <div>fireboat</div> <div>drilling platform</div> </div>	<div> <div></div> <div>motor scooter</div> <div>go-kart</div> <div>moped</div> <div>bumper car</div> <div>golfcart</div> </div>	<div> <div></div> <div>leopard</div> <div>jaguar</div> <div>cheetah</div> <div>snow leopard</div> <div>Egyptian cat</div> </div>
			
<b>grille</b>	<b>mushroom</b>	<b>cherry</b>	<b>Madagascar cat</b>
<div> <div></div> <div>convertible</div> <div>grille</div> <div>pickup</div> <div>beach wagon</div> <div>fire engine</div> </div>	<div> <div></div> <div>agaric</div> <div>mushroom</div> <div>jelly fungus</div> <div>gill fungus</div> <div>dead-man's-fingers</div> </div>	<div> <div></div> <div>dalmatian</div> <div>grape</div> <div>elderberry</div> <div>ffordshire bullterrier</div> <div>currant</div> </div>	<div> <div></div> <div>squirrel monkey</div> <div>spider monkey</div> <div>titi</div> <div>indri</div> <div>howler monkey</div> </div>



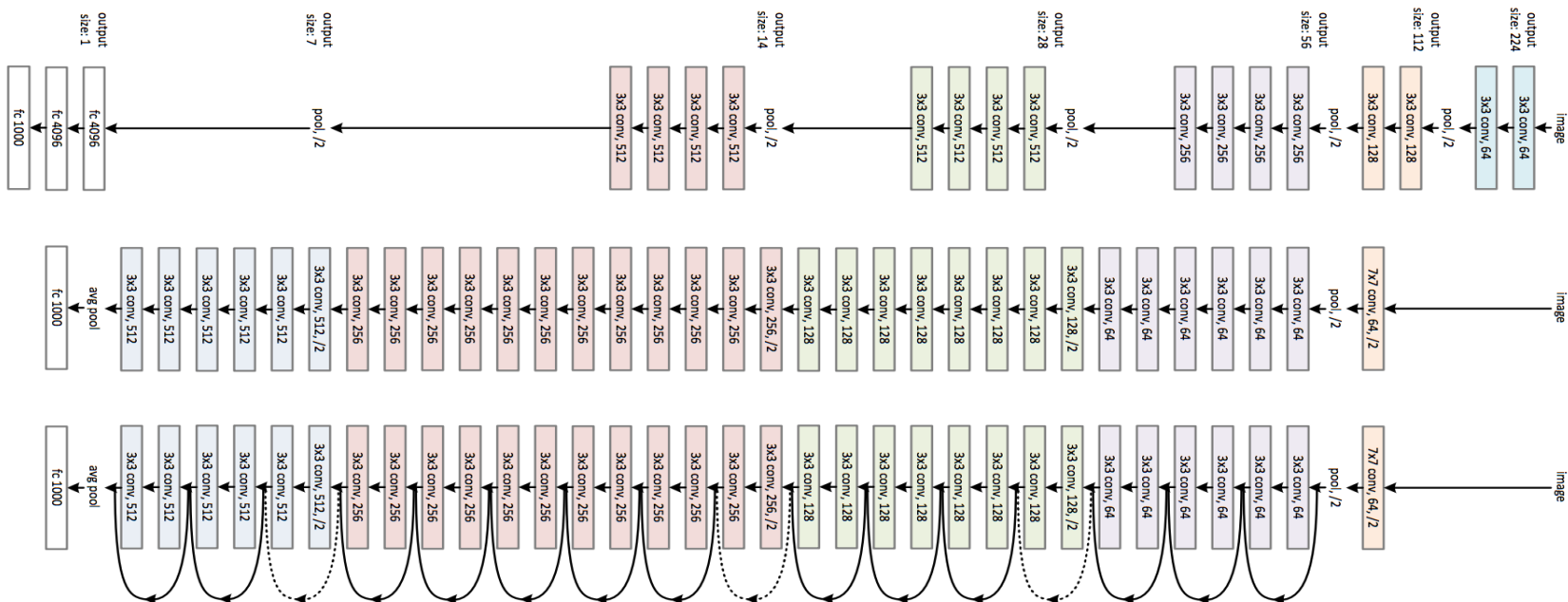
## Архитектура сети VGG-16 (победитель ILSVRC 2014)



# Нейронные сети. CNN.



## VGG 16

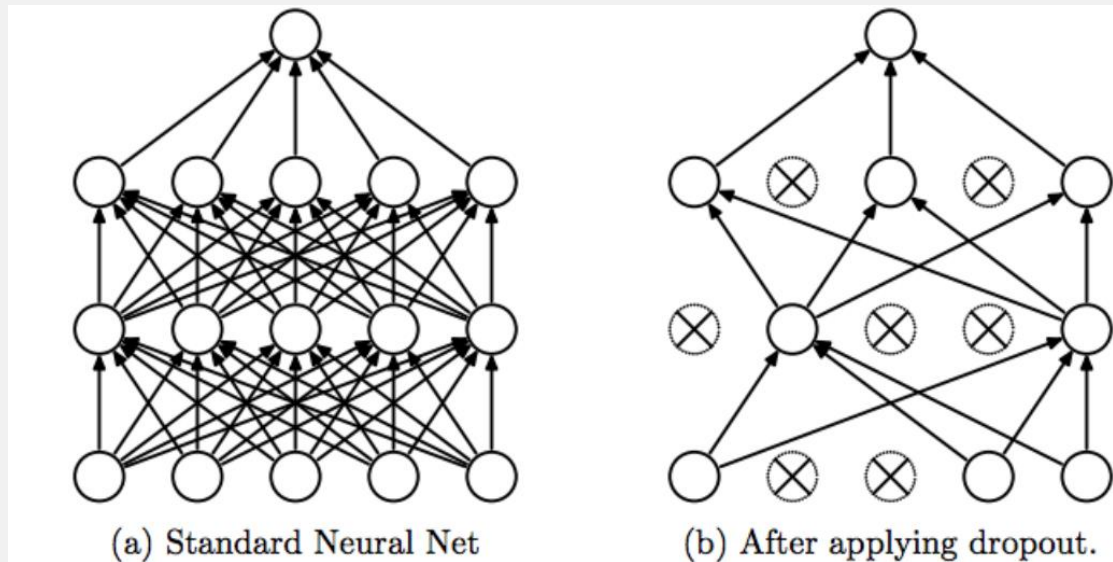


## ResNet

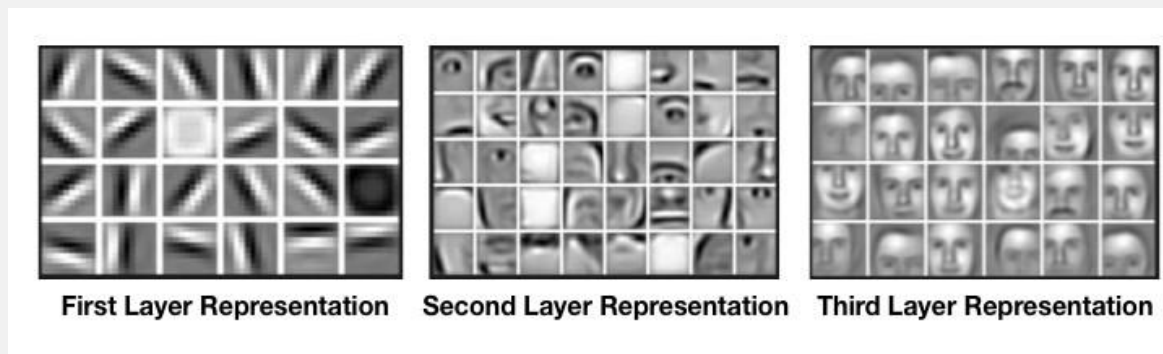
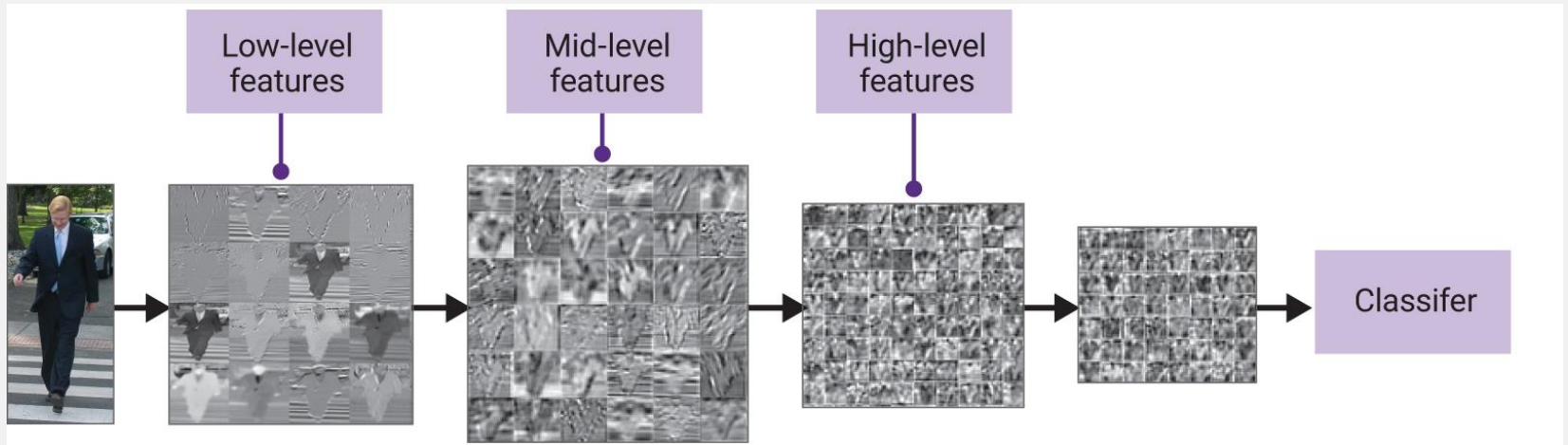
# Регуляризация NN. Dropout



Суть метода заключается в том, что в процессе обучения выбирается слой, из которого случайным образом выбрасывается определённое количество нейронов, которые выключаются из дальнейших вычислений.

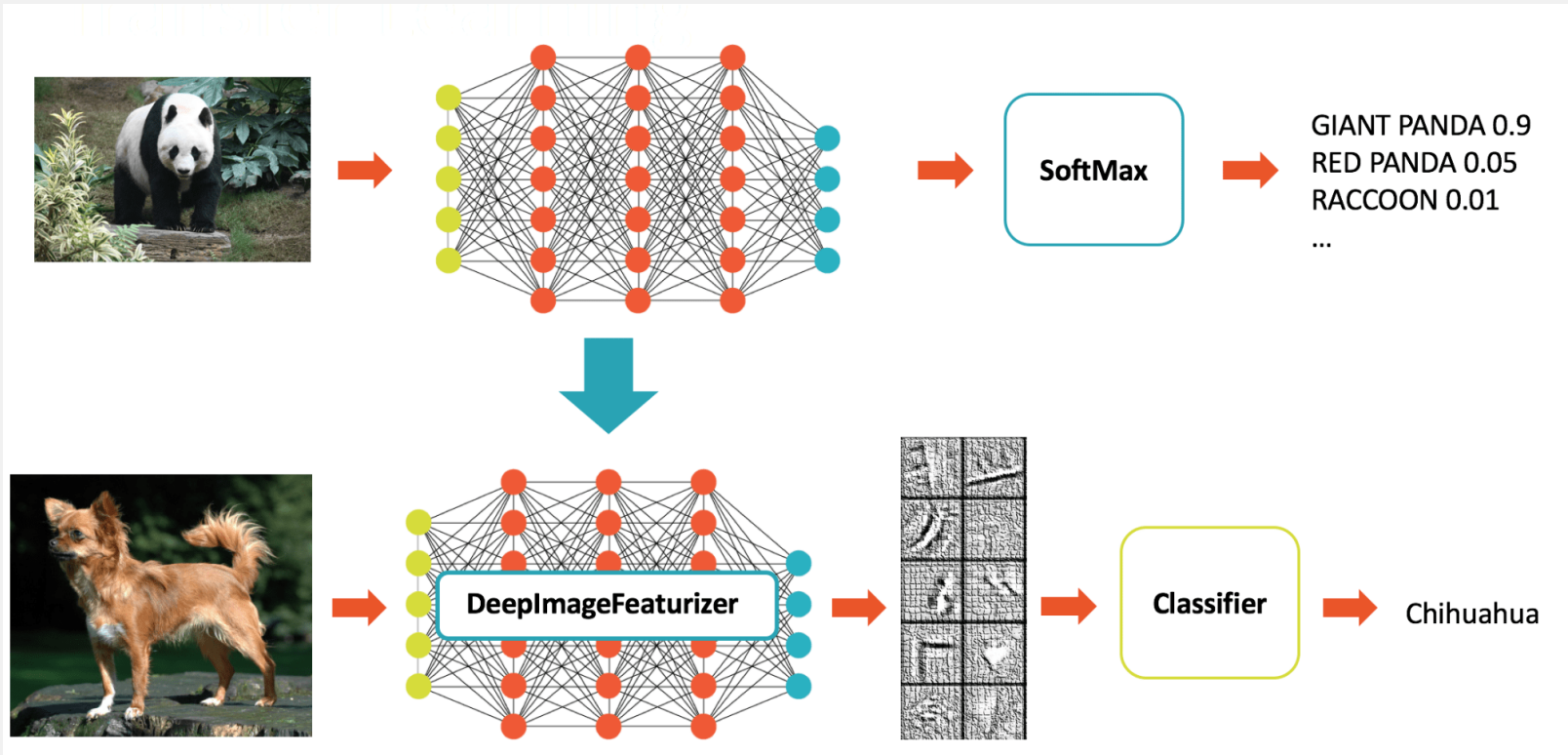


# CNN. Image Embedding





# Transfer learning





**Спасибо за  
внимание!**

**Спасёнов Алексей**

[a.spasenov@corp.mail.ru](mailto:a.spasenov@corp.mail.ru)