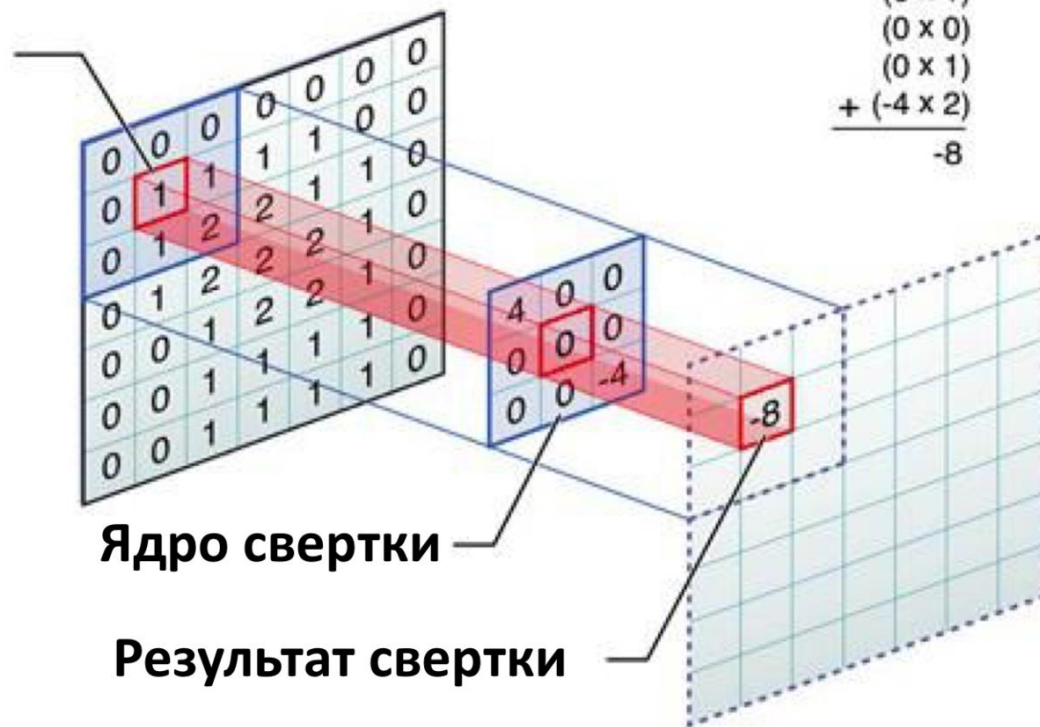
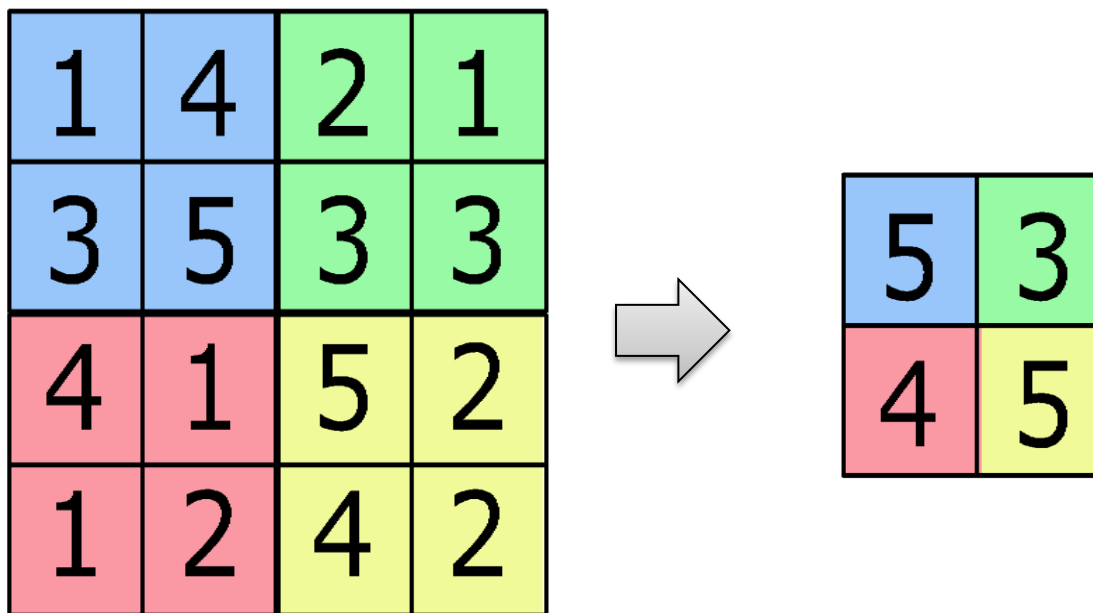


Точка приложения  
свертки



$$\begin{array}{r}
 (4 \times 0) \\
 (0 \times 0) \\
 (0 \times 0) \\
 (0 \times 0) \\
 (0 \times 1) \\
 (0 \times 1) \\
 (0 \times 0) \\
 (0 \times 1) \\
 + (-4 \times 2) \\
 \hline
 -8
 \end{array}$$

# Subsampling (pooling)

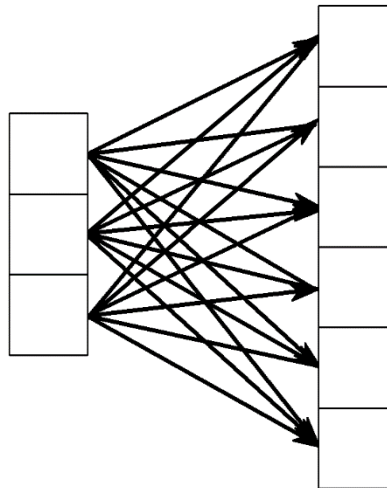


input: 4x4

pool (2,2)

out: 2x2

# Fully Connected (Inner product)

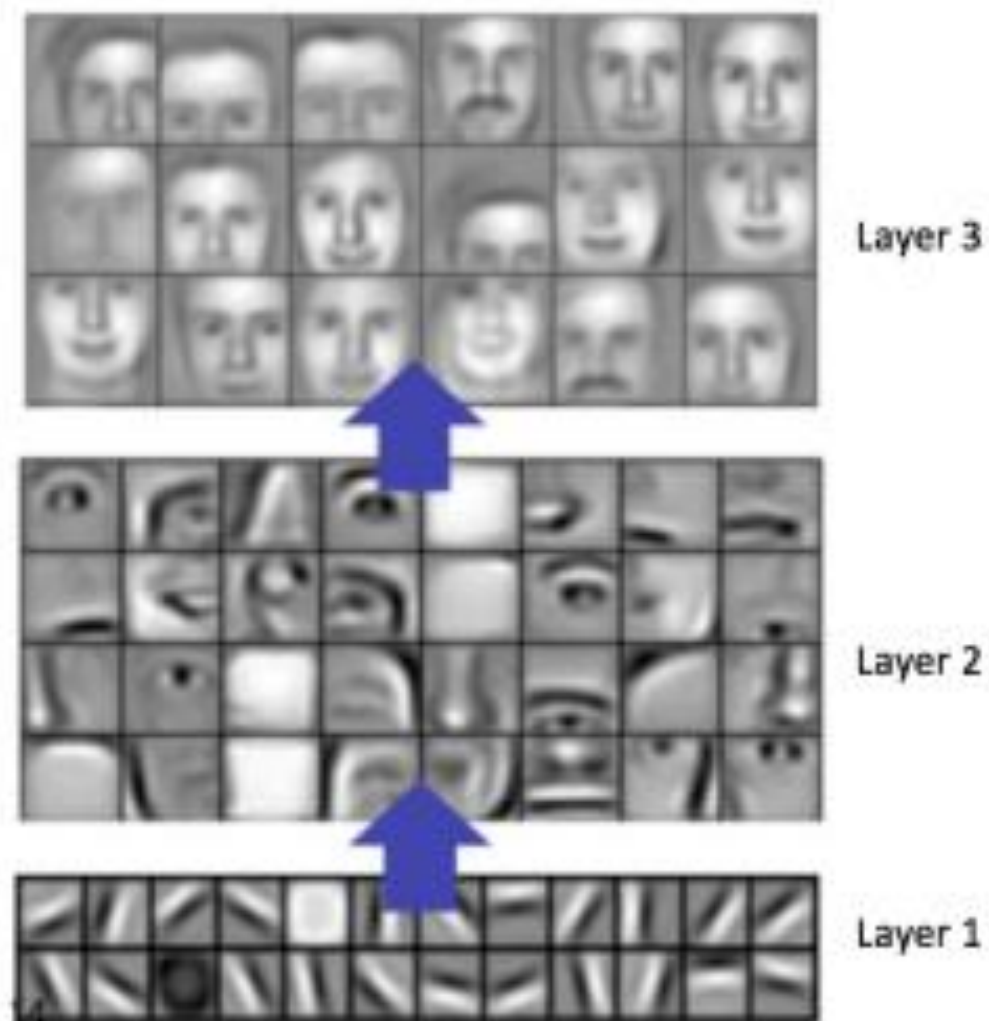


input: 3

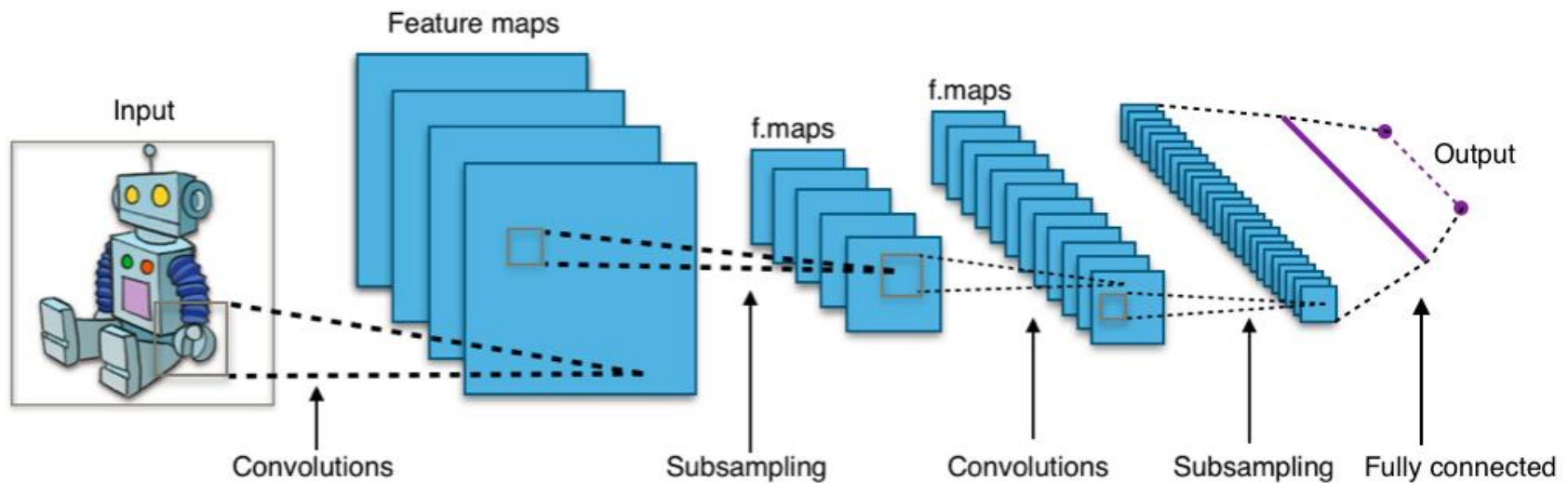
fc (6)

out: 6

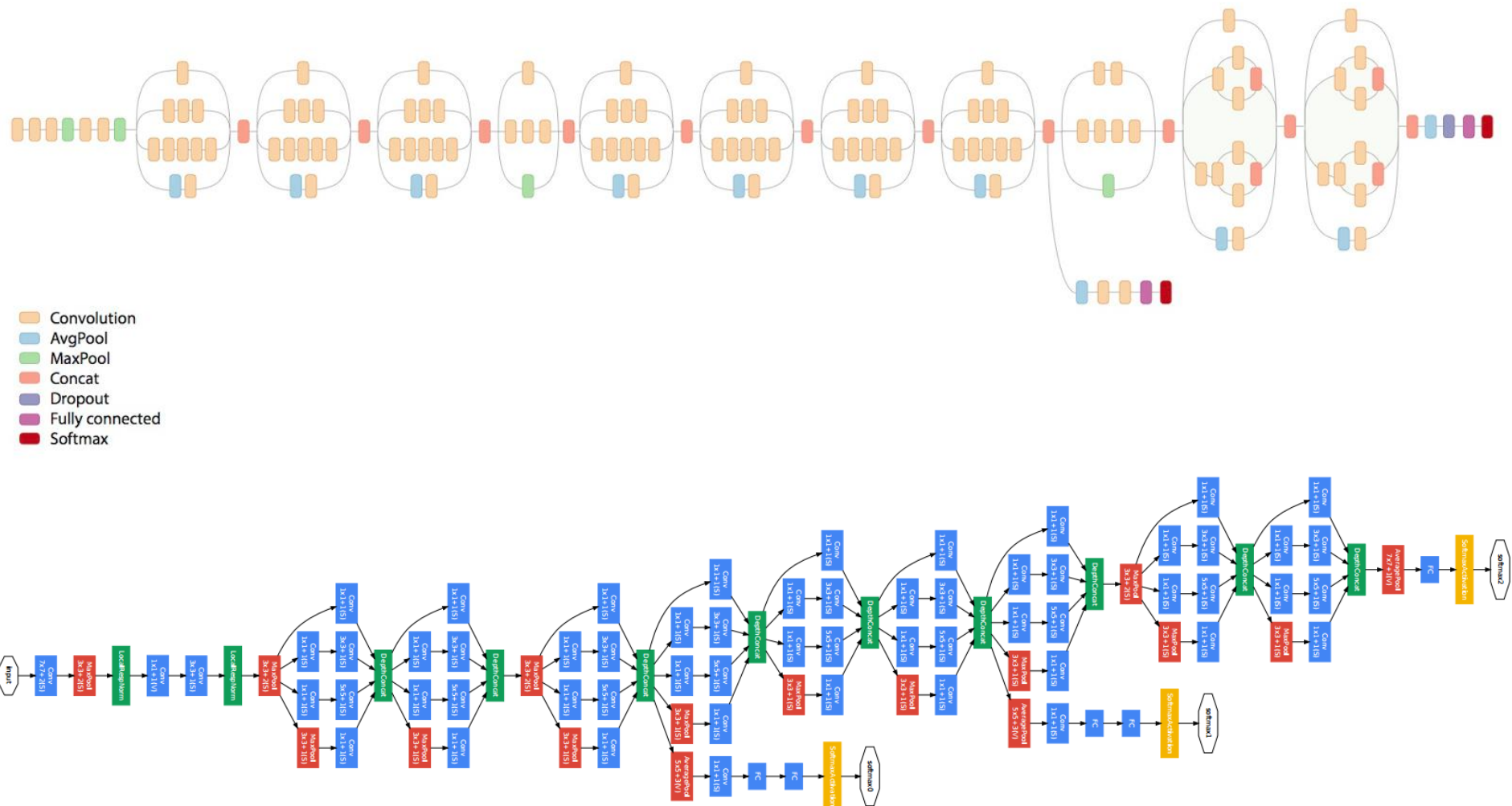
# Визуализация свёрточных слоев



# Типовая архитектура



# Современные сети (Inception architecture)





# Укрупненные строительные блоки

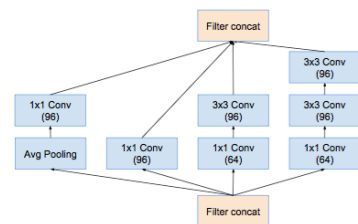


Figure 4. The schema for  $35 \times 35$  grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 9.

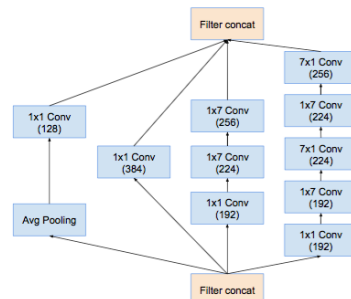


Figure 5. The schema for  $17 \times 17$  grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 9.

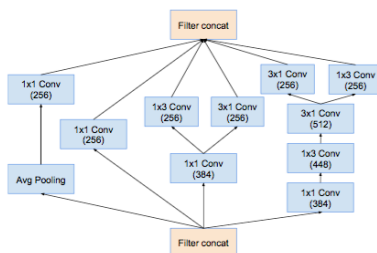


Figure 6. The schema for  $8 \times 8$  grid modules of the pure Inception-v4 network. This is the Inception-C block of Figure 9.

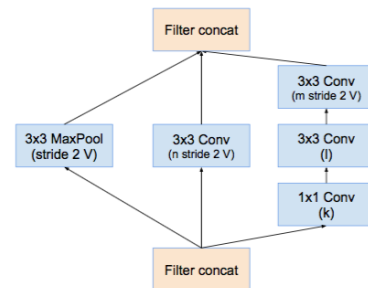


Figure 7. The schema for  $35 \times 35$  to  $17 \times 17$  reduction module. Different variants of this blocks (with various number of filters) are used in Figure 9 and 15 in each of the new Inception-v4, -ResNet-v1, -ResNet-v2) variants presented in this paper. The  $k, l, m, n$  numbers represent filter bank sizes which can be looked up in Table 1.

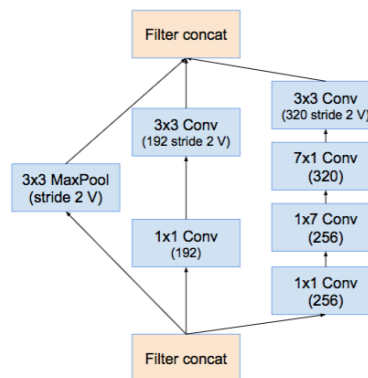
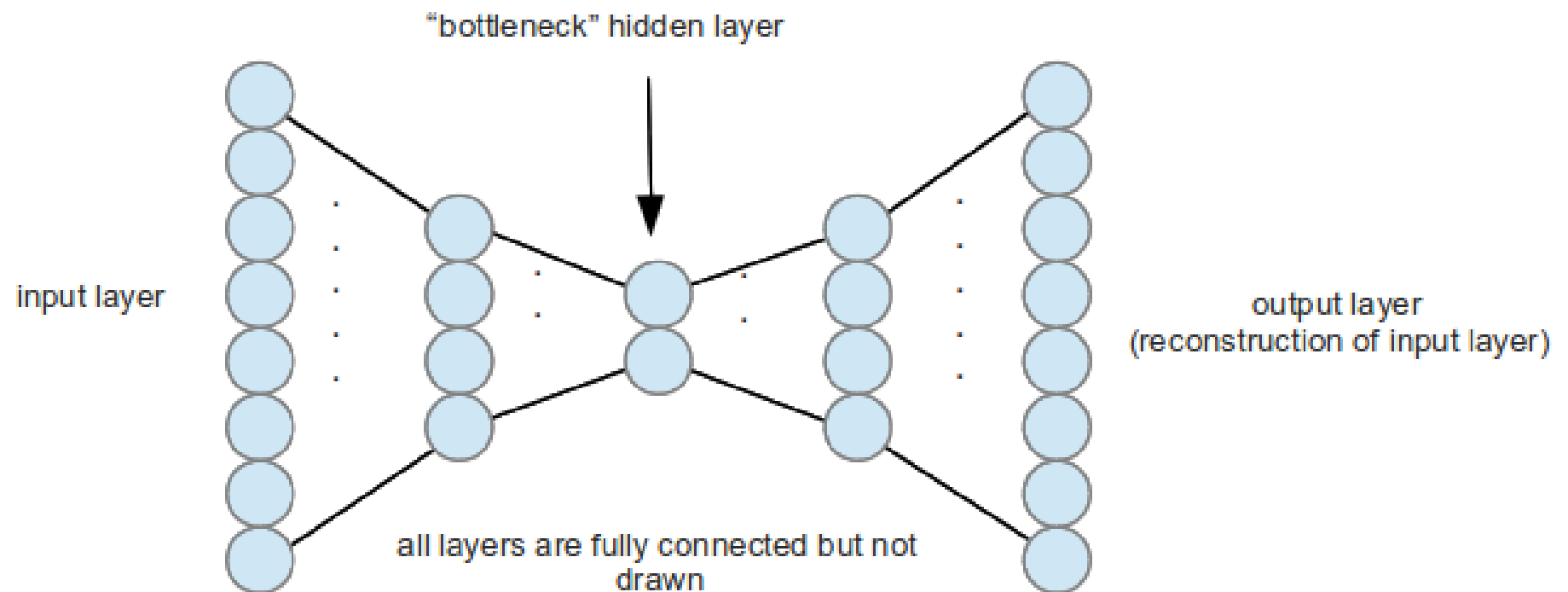


Figure 8. The schema for  $17 \times 17$  to  $8 \times 8$  grid-reduction module. This is the reduction module used by the pure Inception-v4 network in Figure 9.

# Autoencoder

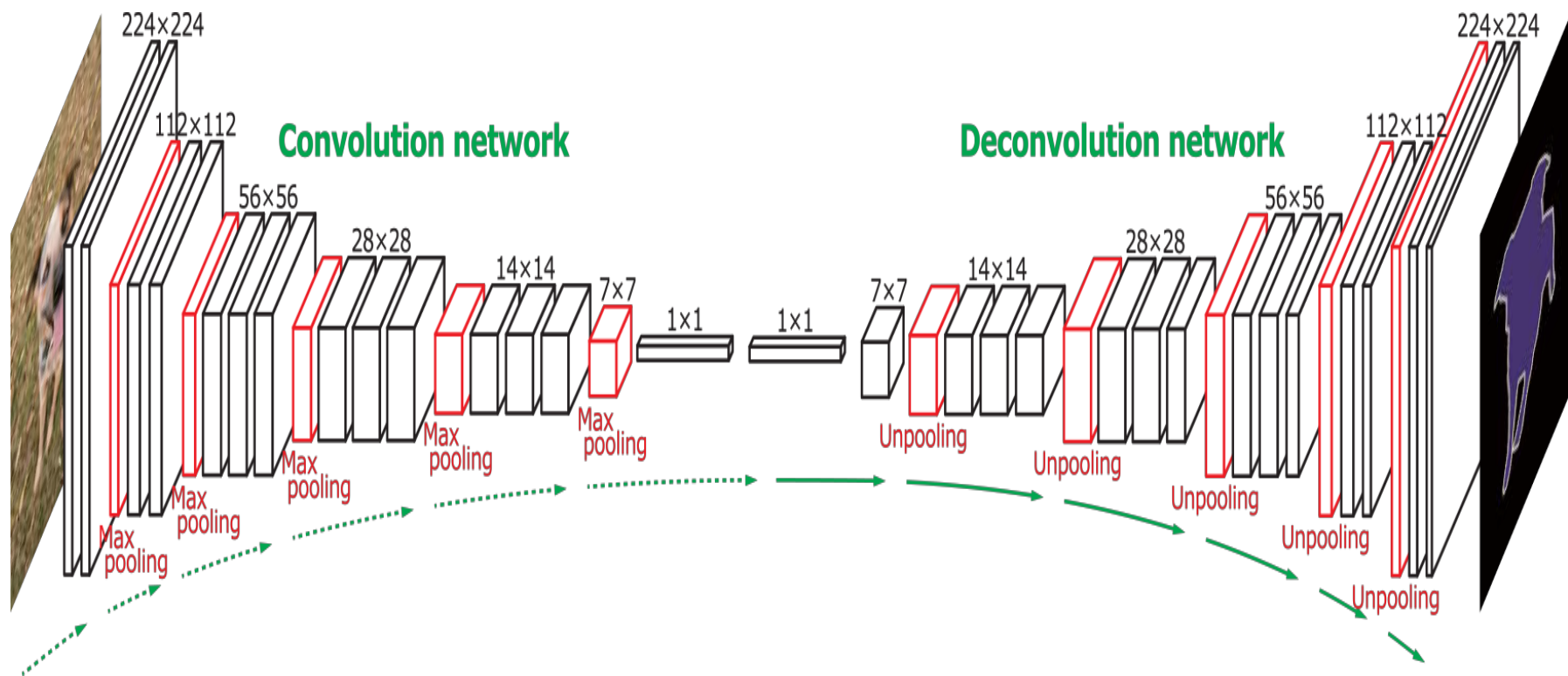




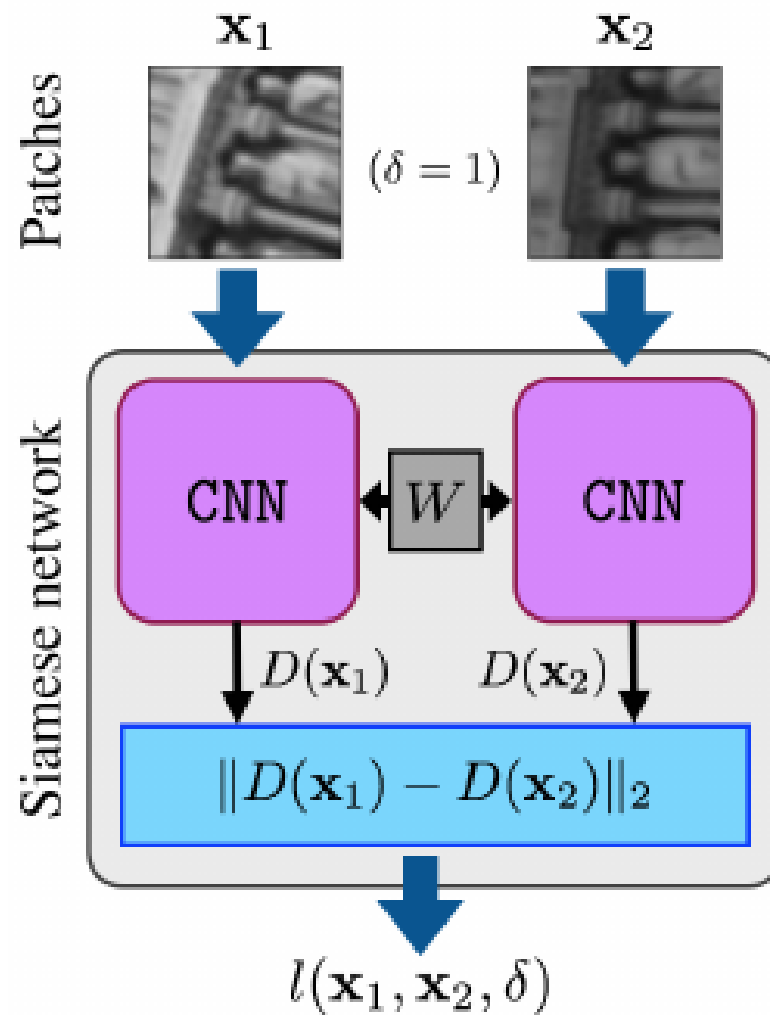
# Clustering with autoencoder

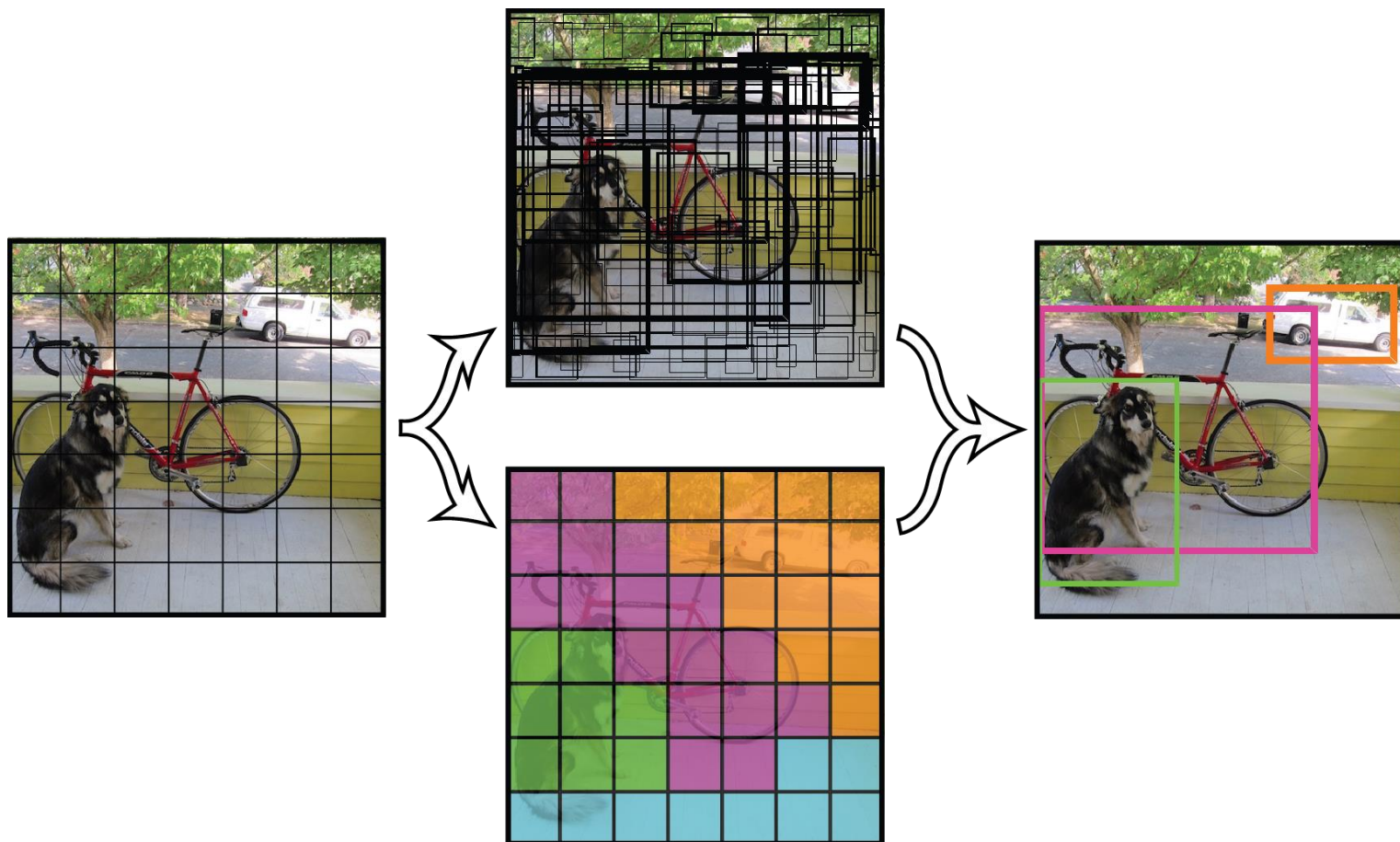


# Fully Convolutional Network

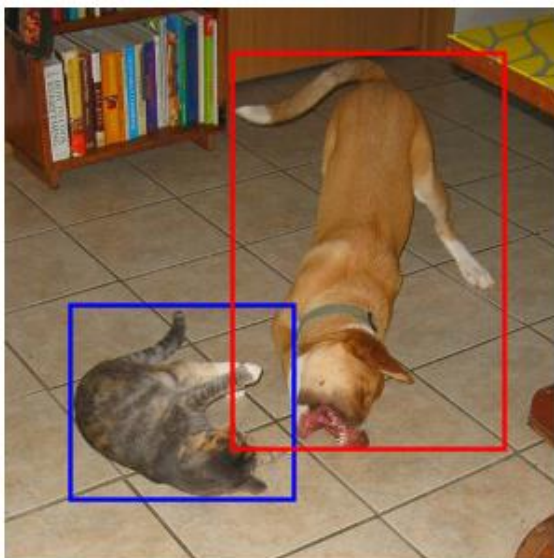


# Siamese Network

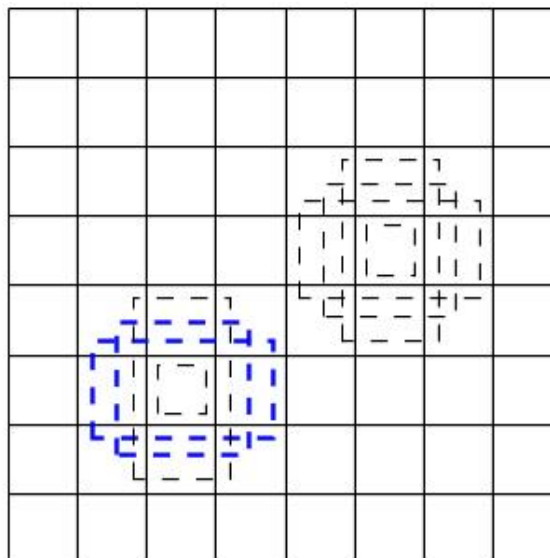




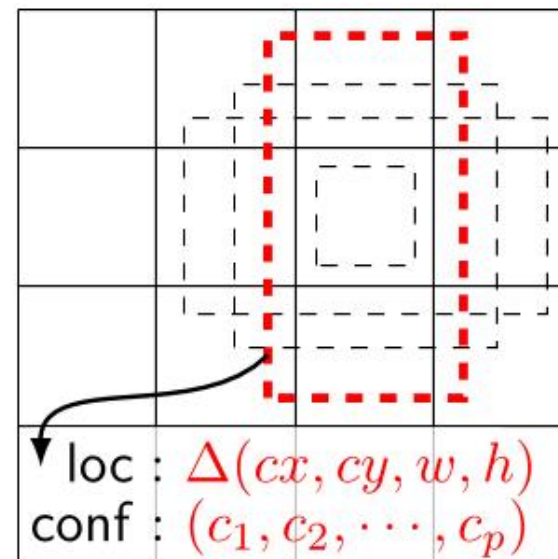
# SSD и YOLO network



(a) Image with GT boxes



(b)  $8 \times 8$  feature map

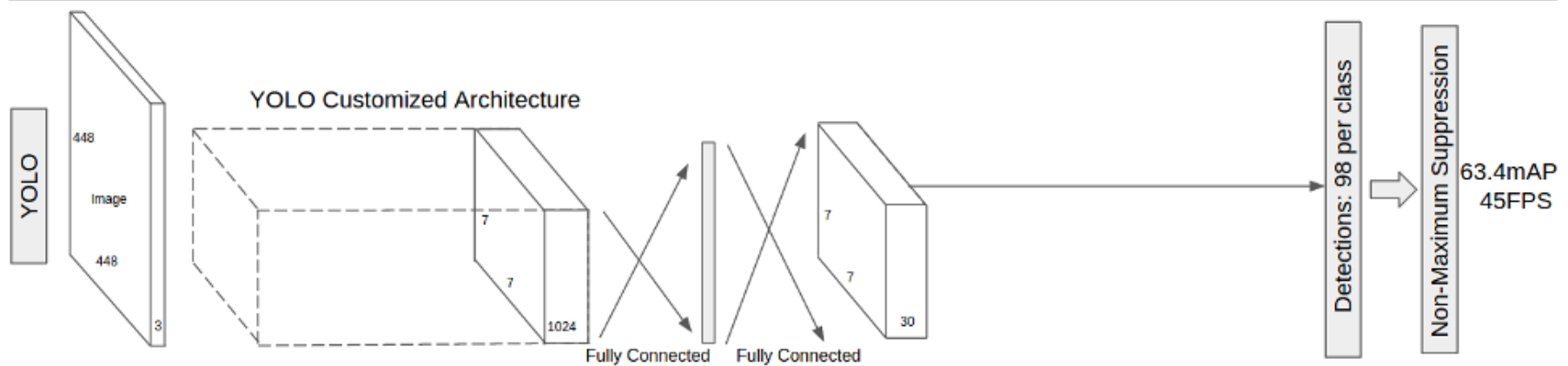
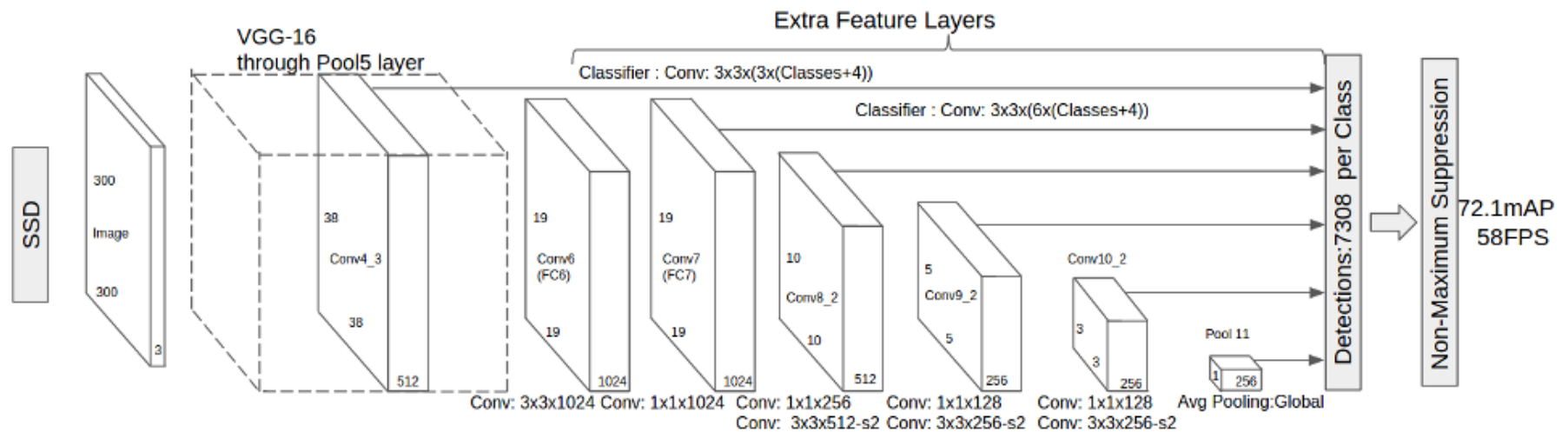


loc :  $\Delta(cx, cy, w, h)$   
conf :  $(c_1, c_2, \dots, c_p)$

(c)  $4 \times 4$  feature map



# SSD и YOLO Network





# Фреймворки для СНС

Theano  
Torch  
Caffe  
Pylearn2  
Tensorflow  
MXNet  
Lasagne  
Keras  
Chainer  
DeepLearnToolbox  
Cuda-Convnet  
RNNLM  
... (over 9000)



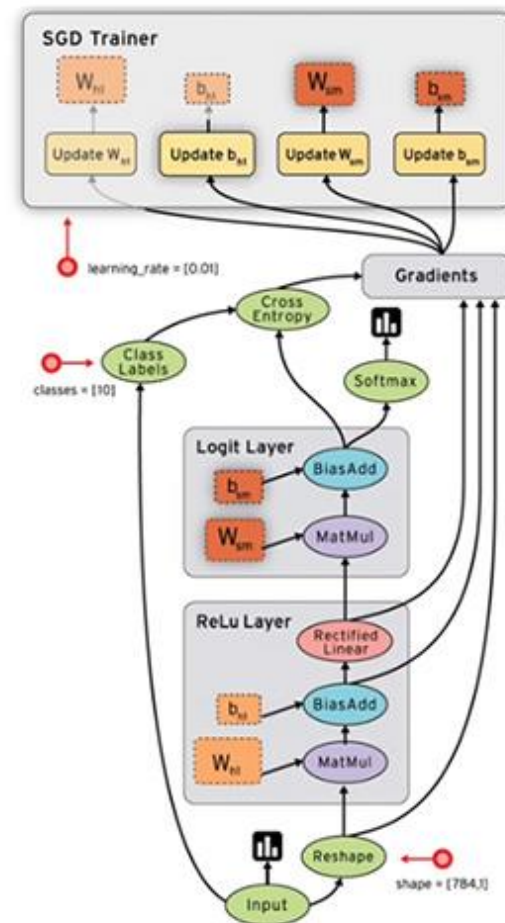
# Caffe

- C++, Python
- Структура сети в текстовом файле
- cuDNN, (а еще CL, MKL)
- nVidia DIGITS

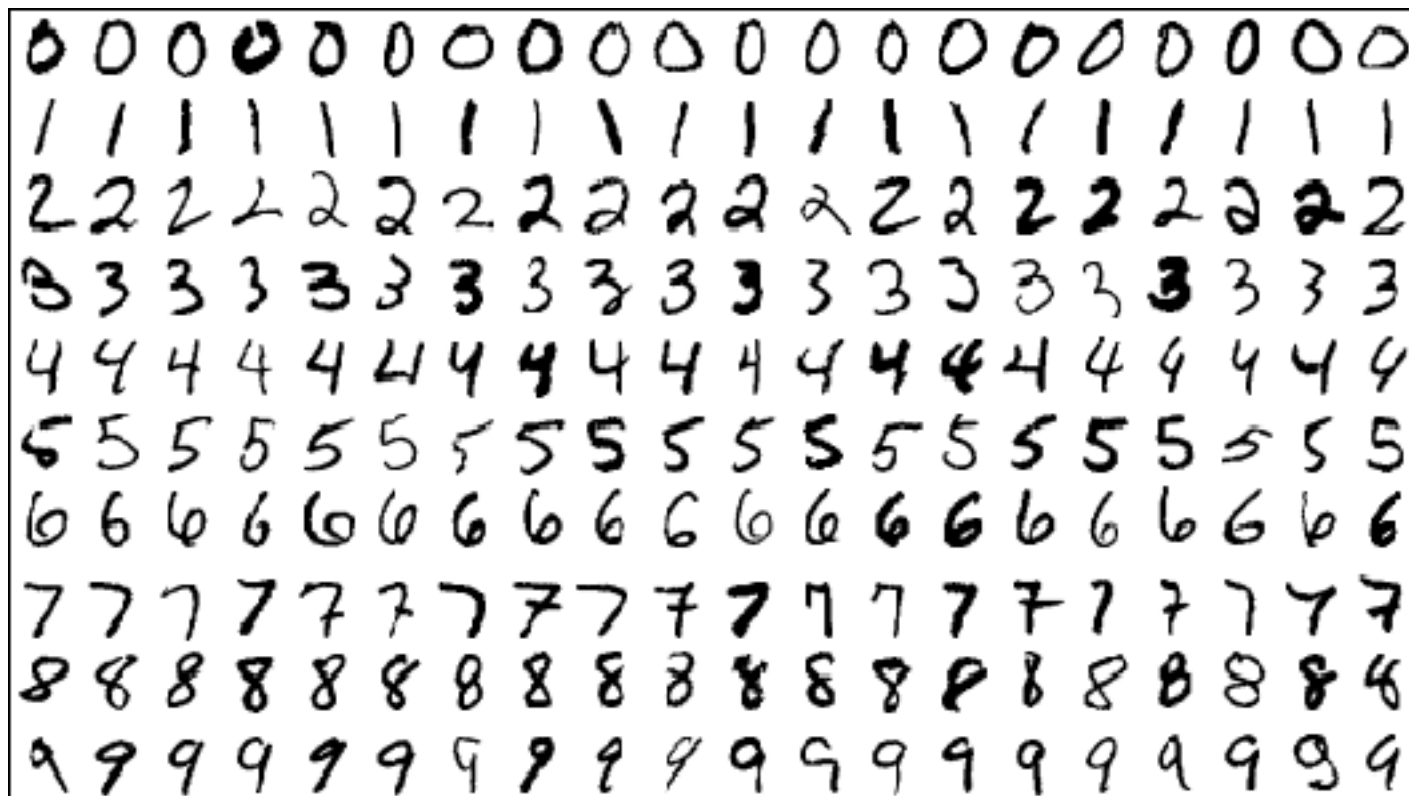


Maximally accurate	Maximally specific
espresso	2.23192
coffee	2.19914
beverage	1.93214
liquid	1.89367
fluid	1.85519

# Tensorflow



# MNIST dataset



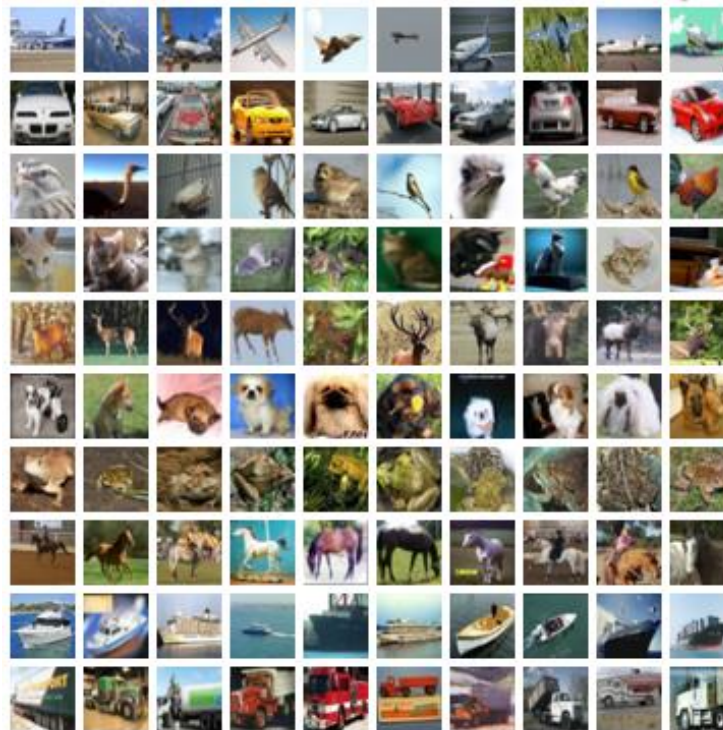


# Сравнение классификаторов на MNIST

Type	Classifier	Distortion	<a href="#">Preprocessing</a>	Error rate (%)
<a href="#">K-Nearest Neighbors</a>	K-NN with non-linear deformation (P2DHMDM)	None	Shiftable edges	0.52 <sup>[14]</sup>
<a href="#">Boosted Stumps</a>	Product of stumps on <a href="#">Haar features</a>	None	Haar features	0.87 <sup>[15]</sup>
Non-Linear Classifier	40 PCA + quadratic classifier	None	None	3.3 <sup>[9]</sup>
<a href="#">Support vector machine</a>	Virtual <a href="#">SVM</a> , deg-9 poly, 2-pixel jittered	None	Deskewing	0.56 <sup>[16]</sup>
<a href="#">Neural network</a>	2-layer 784-800-10	None	None	1.6 <sup>[17]</sup>
<a href="#">Neural network</a>	2-layer 784-800-10	<a href="#">elastic distortions</a>	None	0.7 <sup>[17]</sup>
Deep <a href="#">neural network</a>	6-layer 784-2500-2000-1500-1000-500-10	<a href="#">elastic distortions</a>	None	0.35 <sup>[18]</sup>
<a href="#">Convolutional neural network</a>	Committee of 35 conv. net, 1-20-P-40-P-150-10	<a href="#">elastic distortions</a>	Width normalizations	0.23

# CIFAR 10 Dataset

Airplane  
Automobile  
Bird  
Cat  
Deer  
Dog  
Frog  
Horse  
Ship  
Truck

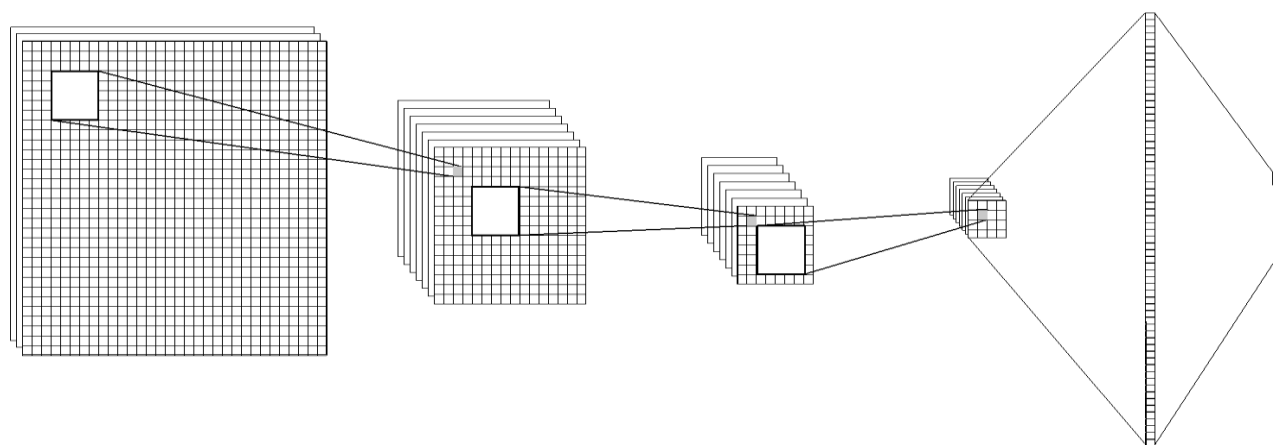


32x32 color images  
10 classes  
60000 images  
50000 training,  
10000 testing

\*Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009



# Архитектура сети



data:	input:	feature 1:	feature 2:	feature 3:	inner:	out:
	32x32x3	16x16x32	8x8x32	4x4x64	64	10
operation:	conv 5x5x32 pool 2x2	conv 5x5x32 pool 2x2	conv 5x5x64 pool 2x2	fc 64	fc 10 softmax	