Advanced CNN

Grigory Sapunov  CTO / Intento

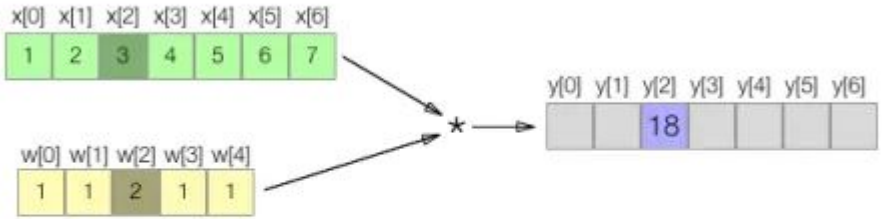
HSE / #8 / 30.10.2018

Использование CNN для других задач

- Трансформация изображений (перенос стиля, синтез изображений, …)
- Обработка видео (spatio-temporal …)
  https://arxiv.org/abs/1606.04698
- Обработка пространственных данных
  http://www.microsoft.com/en-us/research/wp-content/uploads/2016/09/DeepST-SIGSPATIAL2016.pdf
- Обработка звука (ASR, WaveNet, …)
  http://benanne.github.io/2014/08/05/spotify-cnns.html
  https://deepmind.com/blog/wavenet-generative-model-raw-audio/
  http://ronan.collobert.com/pub/matos/2015_cnnspeech_interspeech.pdf
  http://www.microsoft.com/en-us/research/publication/convolutional-neural-networks-for-speech-recognition-2/
- Обработка текстов (классификация, перевод(!), …)
  http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/
  https://arxiv.org/abs/1611.02344

1D Convolution

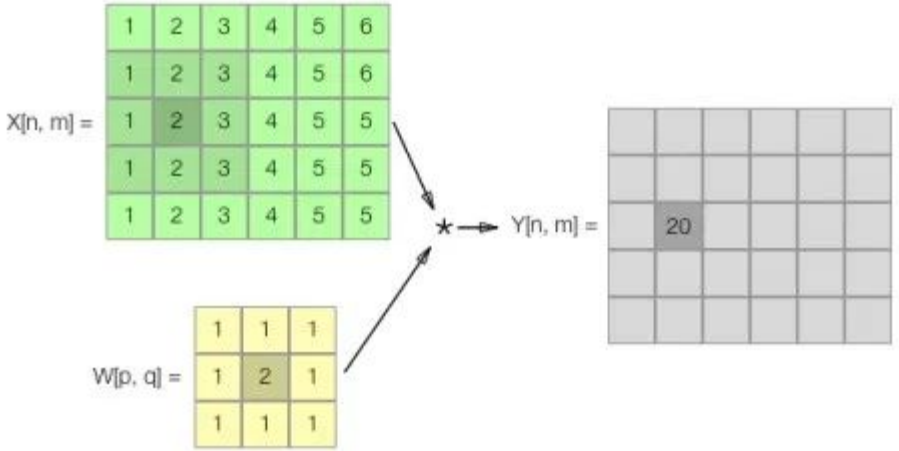
1D



$$y[2] = x[0]w[4] + x[1]w[3] + x[2]w[2] + x[3]w[1] + x[4]w[0] =$$

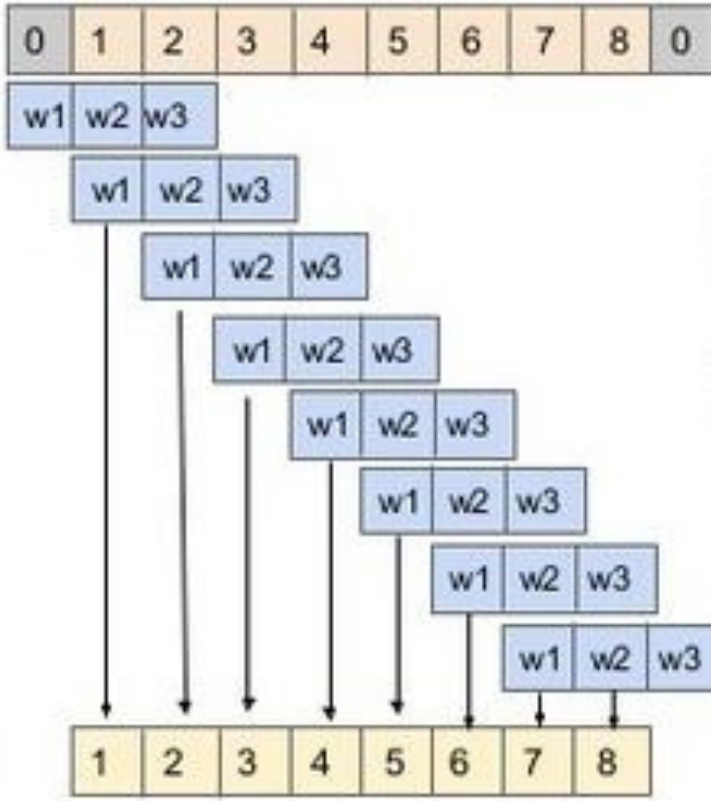
$$1*1 + 2*1 + 3*2 + 4*1 + 5*1 = 18$$

2D



https://www.invasivecode.com/weblog/convolutional-neural-networks-ios-10-macos-sierra/

1D Convolution



https://www.kaggle.com/shivamb/3d-convolutions-understanding-and-implementation

1D Convolution Example: Classification

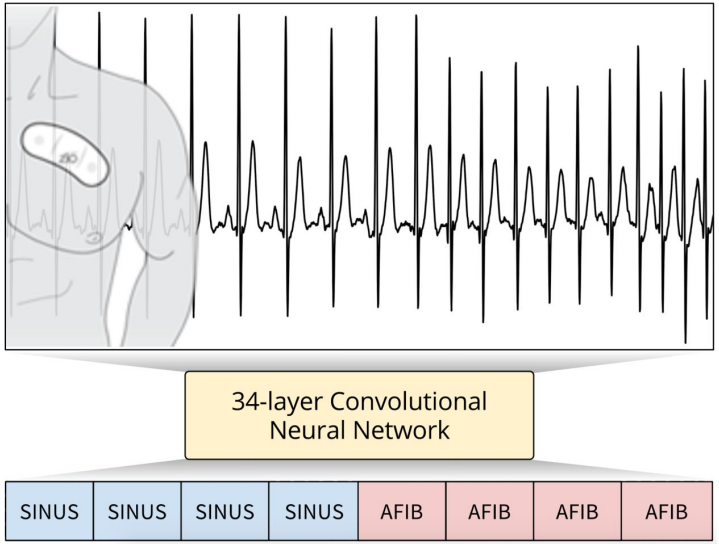
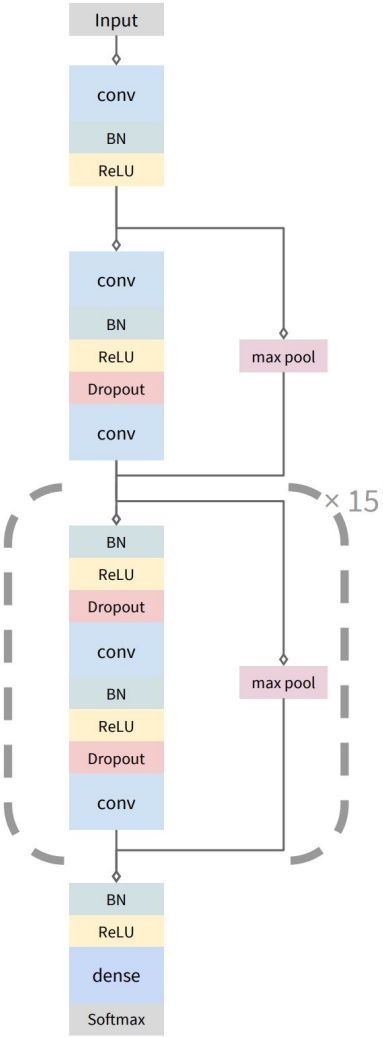


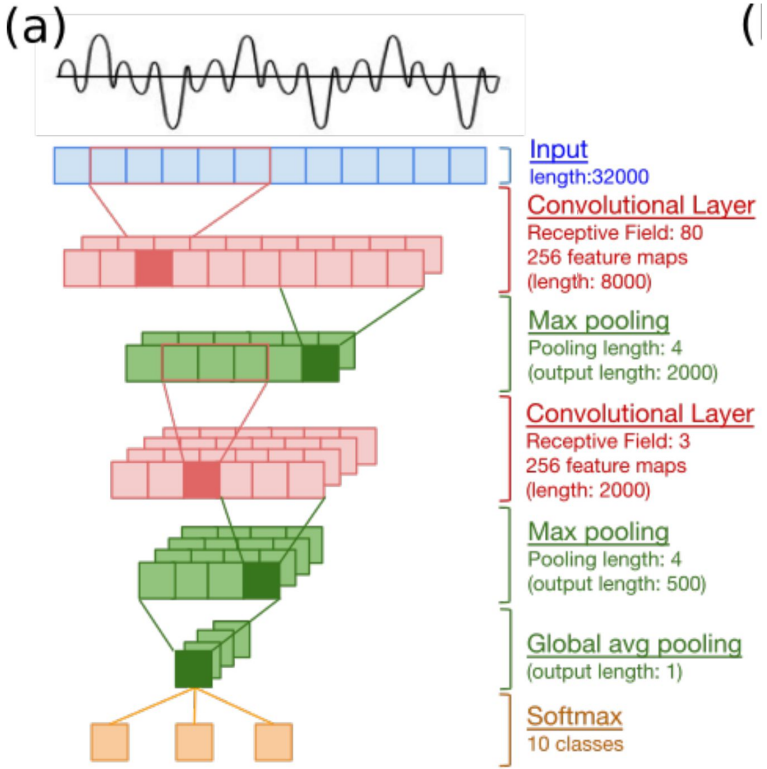
Figure 1. Our trained convolutional neural network correctly detecting the sinus rhythm (SINUS) and Atrial Fibrillation (AFIB) from this ECG recorded with a single-lead wearable heart monitor.



Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks, https://arxiv.org/abs/1707.01836

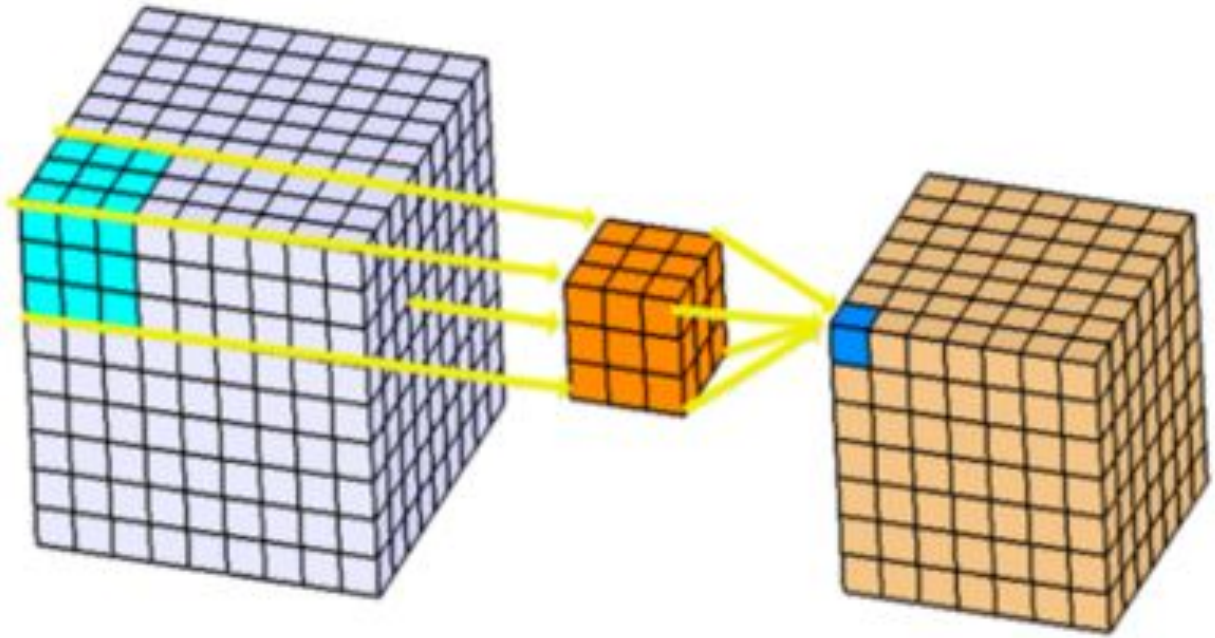
1D Convolution Example: Classification

M3 (0.2M)	M5 (0.5M)	M11 (1.8M)	M18 (3.7M)	M34-res (4M)
Input: 32000x1 time-domain waveform				
[80/4, 256]	[80/4, 128]	[80/4, 64]	[80/4, 64]	[80/4, 48]
Maxpool: 4x1 (output: 2000 × n)				
[3, 256]	[3, 128]	[3, 64] × 2	[3, 64] × 4	$\begin{bmatrix} 3, 48 \\ 3, 48 \end{bmatrix} \times 3$
Maxpool: 4x1 (output: 500×n)				
	[3, 256]	[3, 128] × 2	[3, 128] × 4	$\begin{bmatrix} 3, 96 \\ 3, 96 \end{bmatrix} \times 4$
	Maxpool: 4x1 (output: 125 × n)			
	[3, 512]	[3, 256] × 3	[3, 256] × 4	$\begin{bmatrix} 3, 192 \\ 3, 192 \end{bmatrix} \times 6$
	Maxpool: 4x1 (output: 32 × n)			
		[3, 512] × 2	[3, 512] × 4	$\begin{bmatrix} 3, 384 \\ 3, 384 \end{bmatrix} \times 3$
Global average pooling (output: 1 × n)				
Softmax				



Very Deep Convolutional Neural Networks for Raw Waveforms,
https://arxiv.org/abs/1610.00087

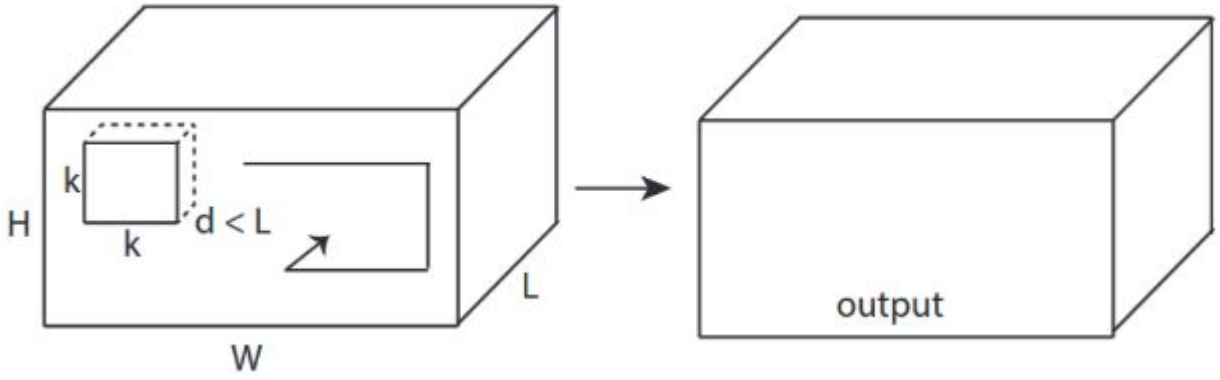
3D Convolution



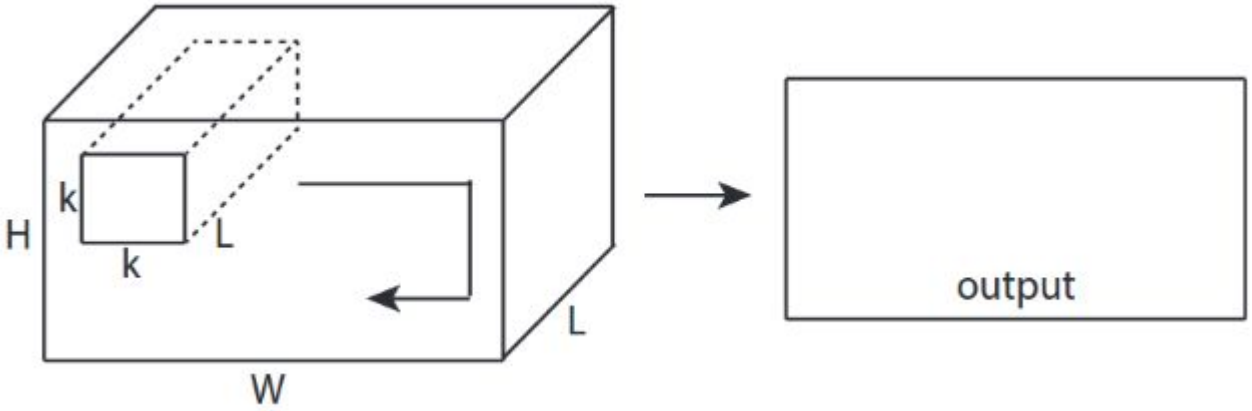
https://www.kaggle.com/shivamb/3d-convolutions-understanding-and-implementation

3D Convolution

3D conv.

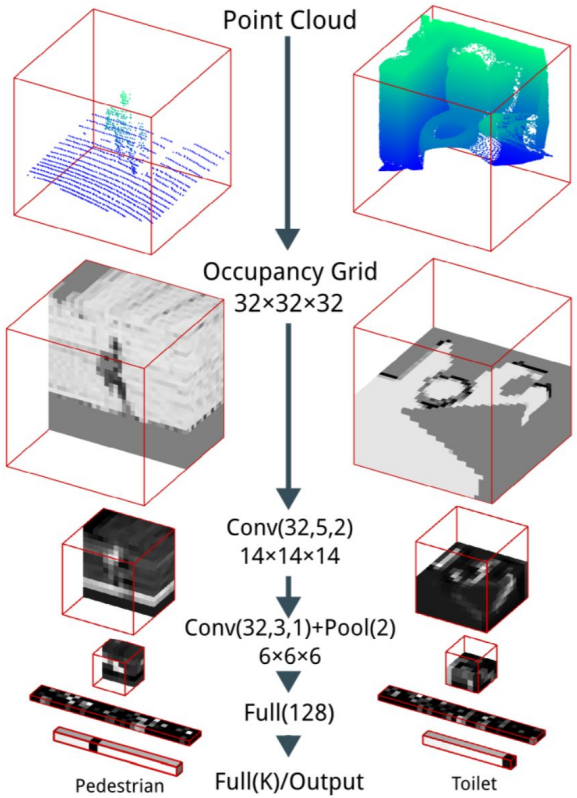


2D conv. with channels



https://stackoverflow.com/questions/42883547/what-do-you-mean-by-1d-2d-and-3d-convolutions-in-cnn

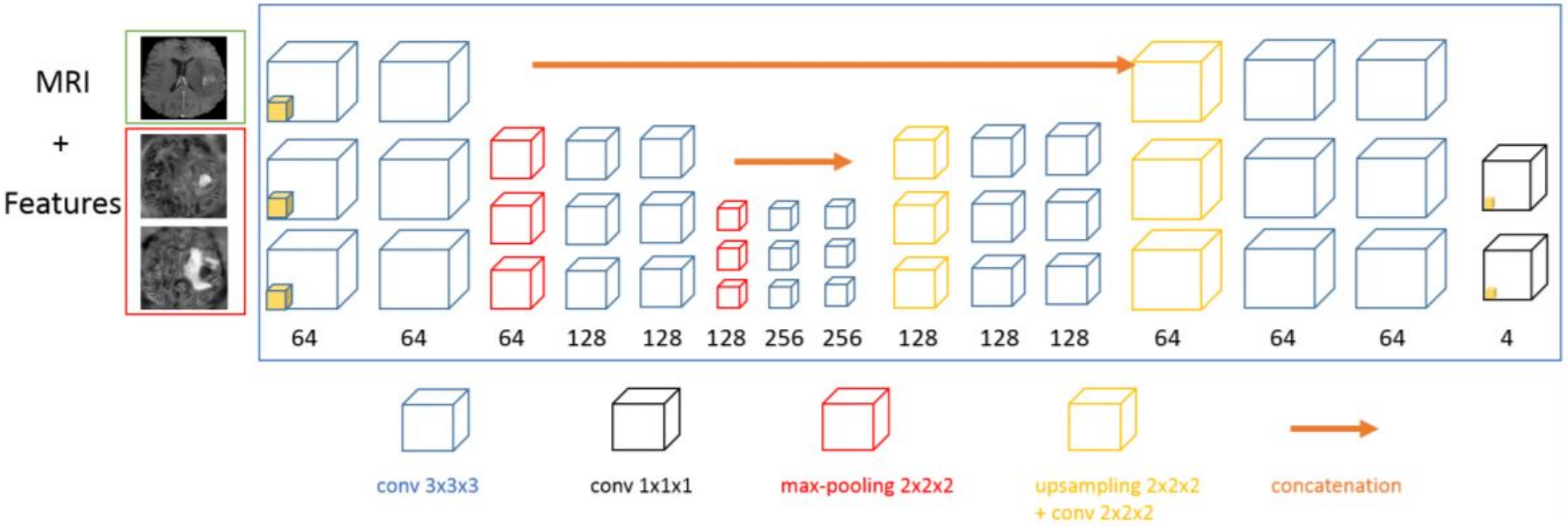
3D Convolution Example: Classification



VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition

https://www.ri.cmu.edu/publications/voxnet-a-3d-convolutional-neural-network-for-real-time-object-recognition/

3D Convolution Example: Segmentation



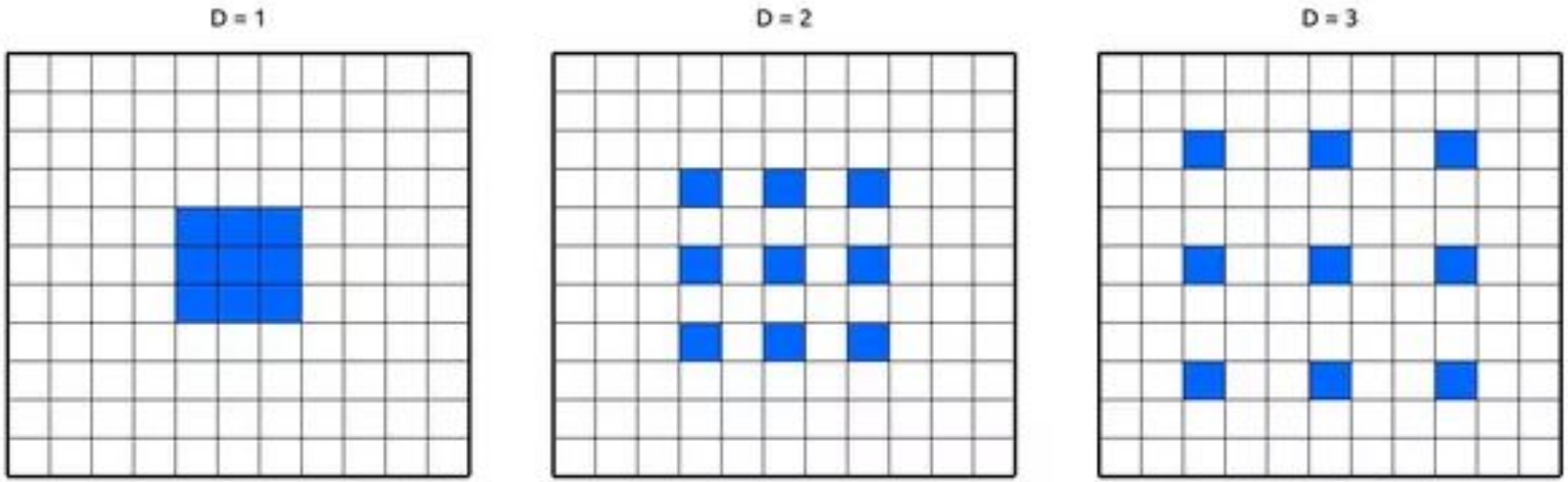
3D Convolutional Neural Networks for Tumor Segmentation using Long-range 2D Context
https://arxiv.org/abs/1807.08599

Convolutions in Keras

- keras.layers.**Conv1D**
  Input: 3D tensor with shape: (batch, steps, channels)
  Output: 3D tensor with shape: (batch, new_steps, filters)
- keras.layers.**Conv2D**
  2D convolution layer (e.g. spatial convolution over images).
  Input: 4D tensor with shape: (batch, channels, rows, cols)
  Output: 4D tensor with shape: (batch, filters, new_rows, new_cols)
- keras.layers.**Conv3D**
  Input: 5D tensor with shape: (batch, channels, conv_dim1, conv_dim2, conv_dim3)
  Output: 5D tensor with shape: (batch, filters, new_conv_dim1, new_conv_dim2, new_conv_dim3)

https://keras.io/layers/convolutional/

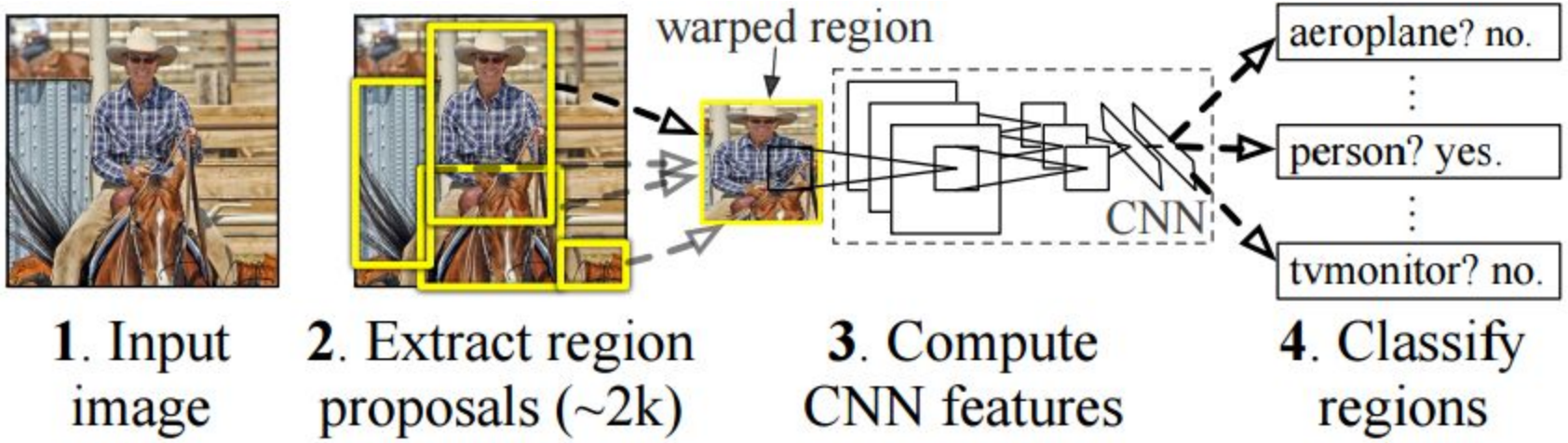
Dilated/Altrous Convolutions



https://www.kaggle.com/shivamb/3d-convolutions-understanding-and-implementation

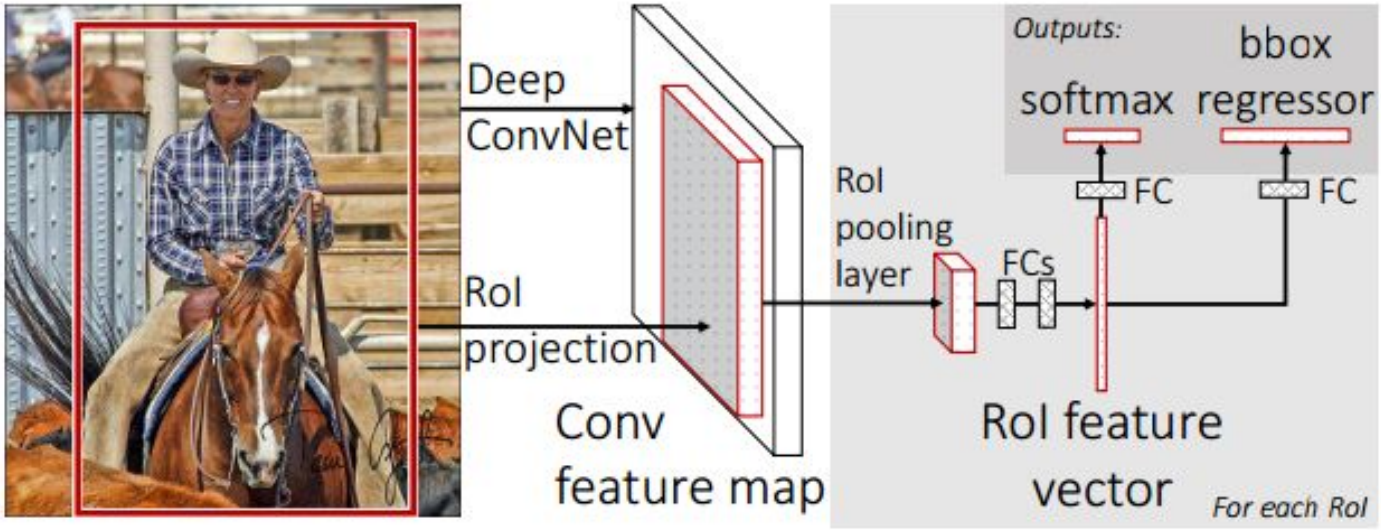
Детекция:
R-CNN, Fast R-CNN, Faster R-CNN

R-CNN: Region-based Convolutional Network



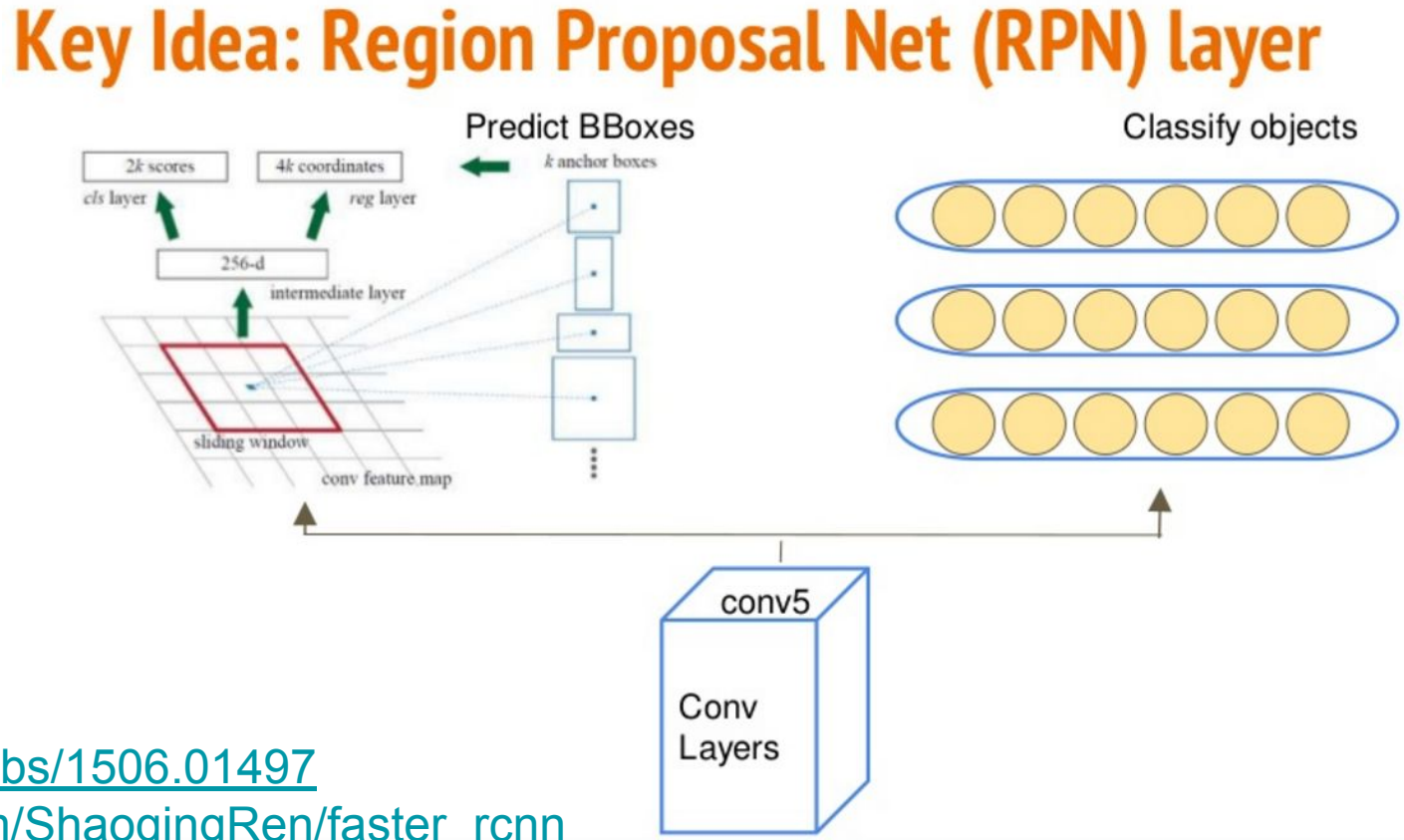
http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/detection.ipynb
https://github.com/rbgirshick/rcnn
https://people.eecs.berkeley.edu/~rbg/papers/pami/rcnn_pami.pdf

Fast R-CNN



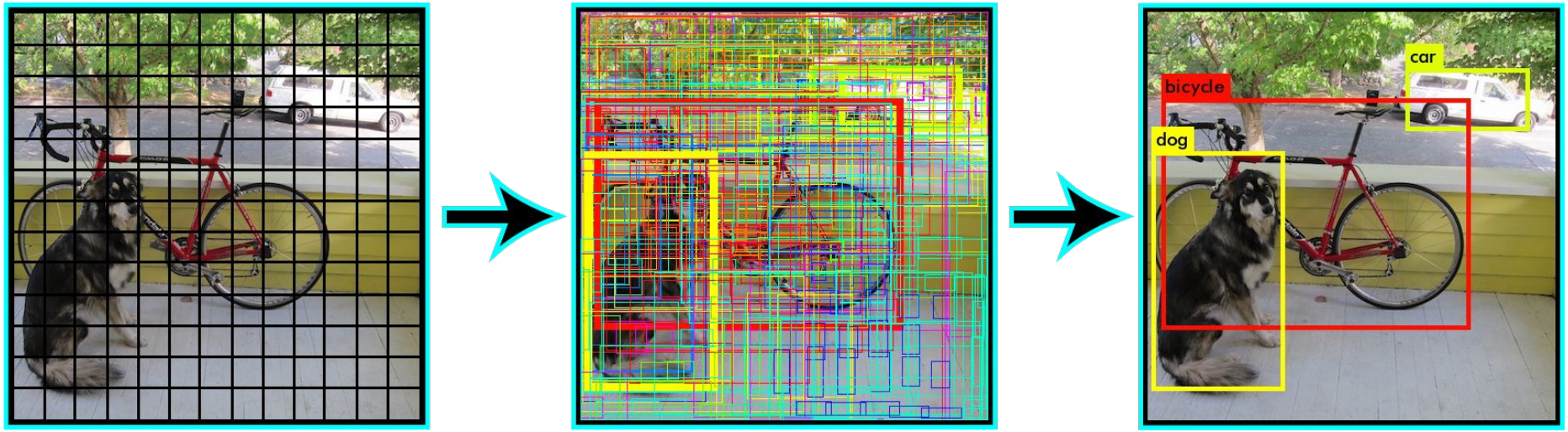
http://tutorial.caffe.berkeleyvision.org/caffe-cvpr15-detection.pdf

Faster R-CNN



https://arxiv.org/abs/1506.01497
https://github.com/ShaoqingRen/faster_rcnn
http://www.slideshare.net/ssuser416c44/faster-rcnn

YOLO: Real-Time Object Detection



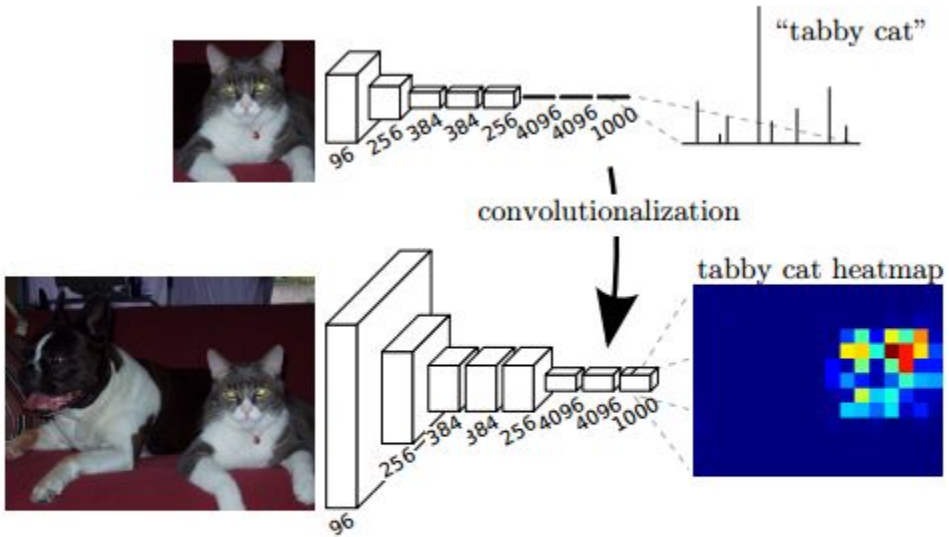
https://pjreddie.com/darknet/yolo/

Fully-convolutional networks (FCN)

Fully-convolutional networks (FCN)

Обычная свёрточная сеть, но без MLP сверху (нет полносвязных слоёв).

Позволяет работать с изображениями произвольного размера и выдавать на выходе тепловую карту классификации.



https://arxiv.org/abs/1411.4038

Fully-convolutional networks (FCN)

Можно преобразовать полносвязный слой на свёрточный

```
layer {
  name: "fc6"
  type: "InnerProduct"
  bottom: "pool5"
  top: "fc6"
  inner_product_param {
    num_output: 4096

  }
}
```

```
layer {
  name: "conv6"
  type: "Convolution"
  bottom: "pool5"
  top: "conv6"
  convolution_param {
    num_output: 4096
    kernel_size: 6
  }
}
```

Table 1: Left: `fc6` definition, Right: equivalent `conv6` definition with a kernel size of 6 because the input to `fc6` is a 6×6 image patch.

https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5

Ресурсы

- https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf
- https://github.com/shelhamer/fcn.berkeleyvision.org
- https://github.com/developmentseed/caffe-fcn

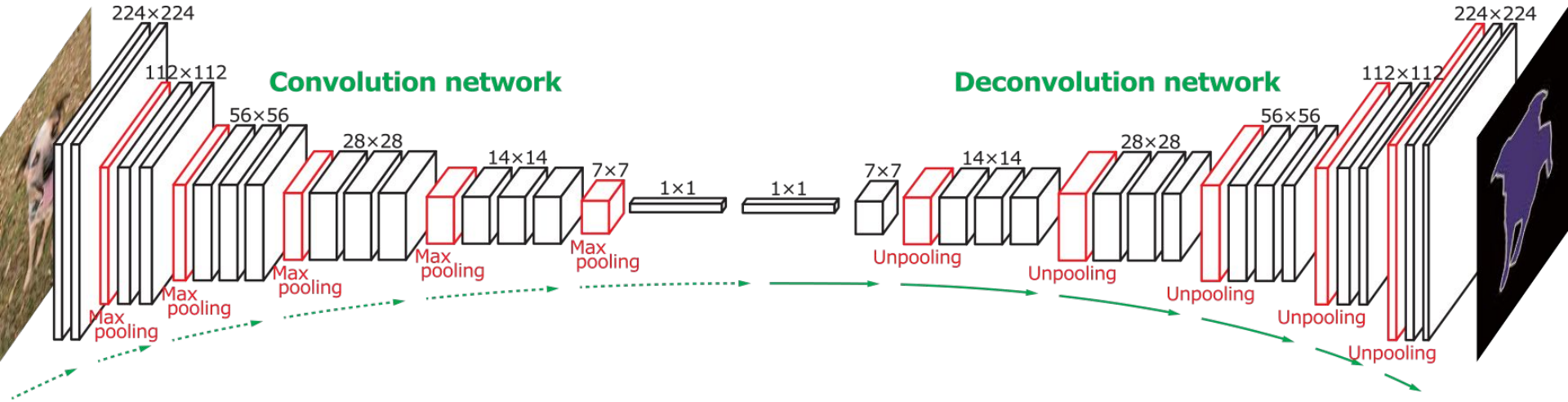
- https://www.quora.com/How-is-Fully-Convolutional-Network-FCN-different-from-the-original-Convolutional-Neural-Network-CNN
- https://www.quora.com/What-are-the-advantages-of-Fully-Convolutional-Networks-over-CNNs
- https://www.quora.com/How-does-the-conversion-of-last-layers-of-CNN-from-fully-connected-to-fully-convolutional-allow-it-to-process-images-of-different-size

Deconvolution networks

Deconvolution networks

Правильнее называть это Transposed convolution, а не Deconvolution (это слово уже занято в цифровой обработке сигналов для обратной операции).

По сути, реализован обучаемый upsampling.



http://cvlab.postech.ac.kr/research/deconvnet/

Deconvolution networks

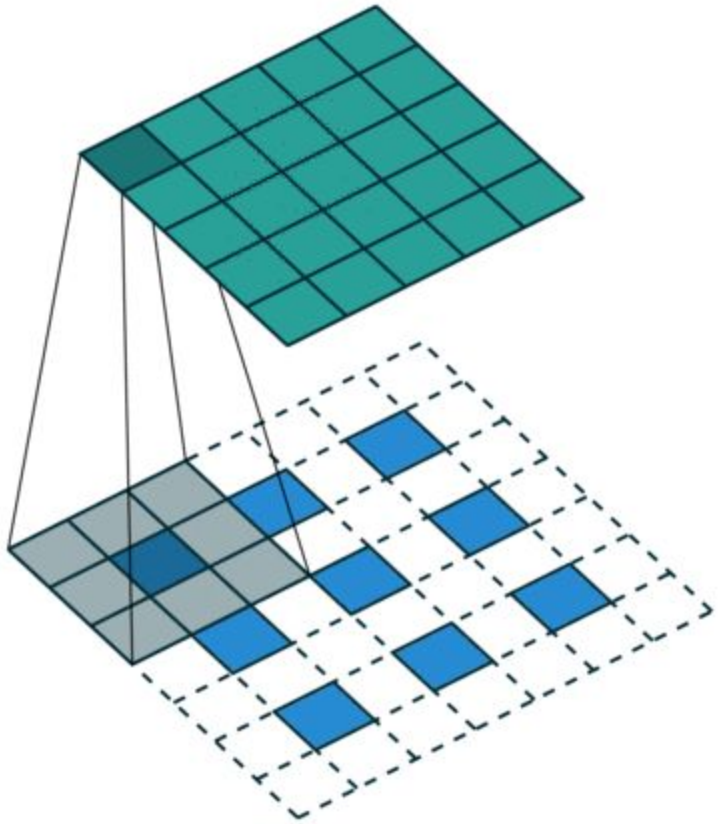
Специальный тип слоя: Deconvolution.
Параметр stride задаёт степень увеличения.

```
layer {
  name: "upscore"
  type: "Deconvolution"
  bottom: "score"
  top: "upscore"
  convolution_param {
    num_output: 12 # set this to number of classes
    kernel_size: 63
    stride: 32
  }
}
```

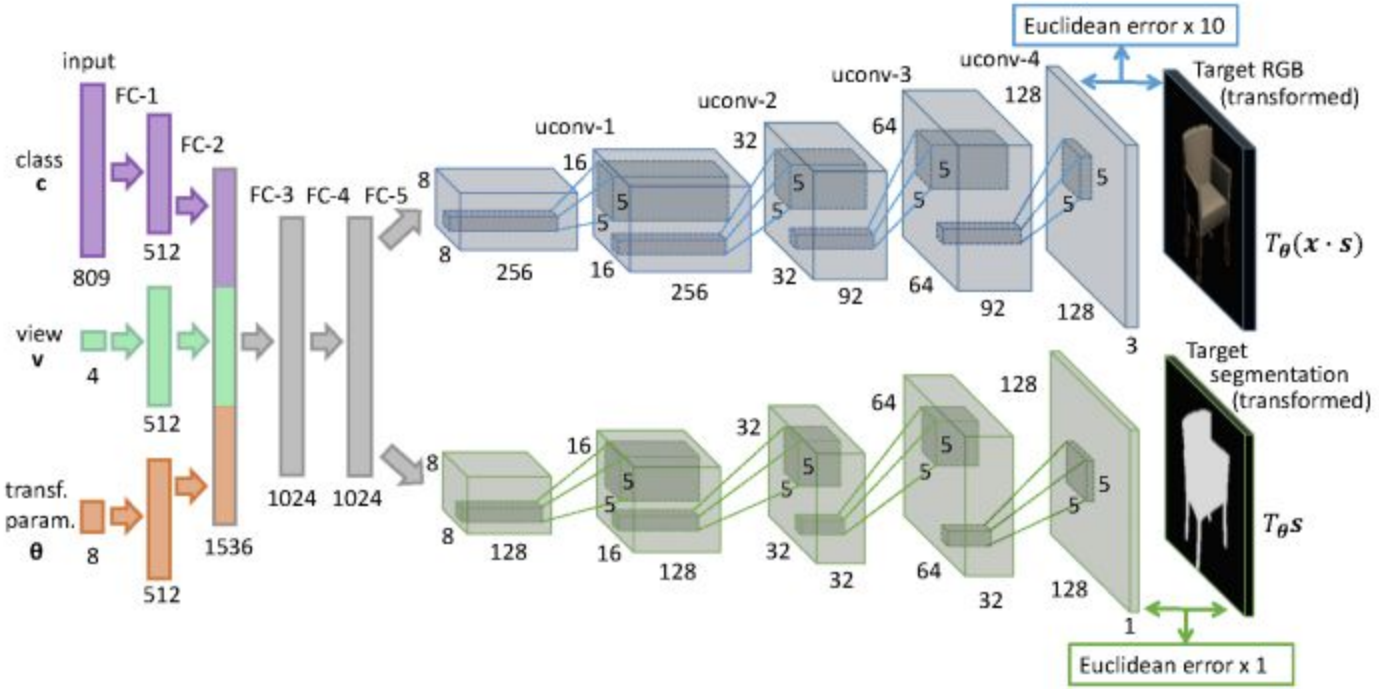
https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5

Как работает Transposed convolution

1) Делаем padding нулями (unpooling)
2) Применяем convolution



Генерация изображений



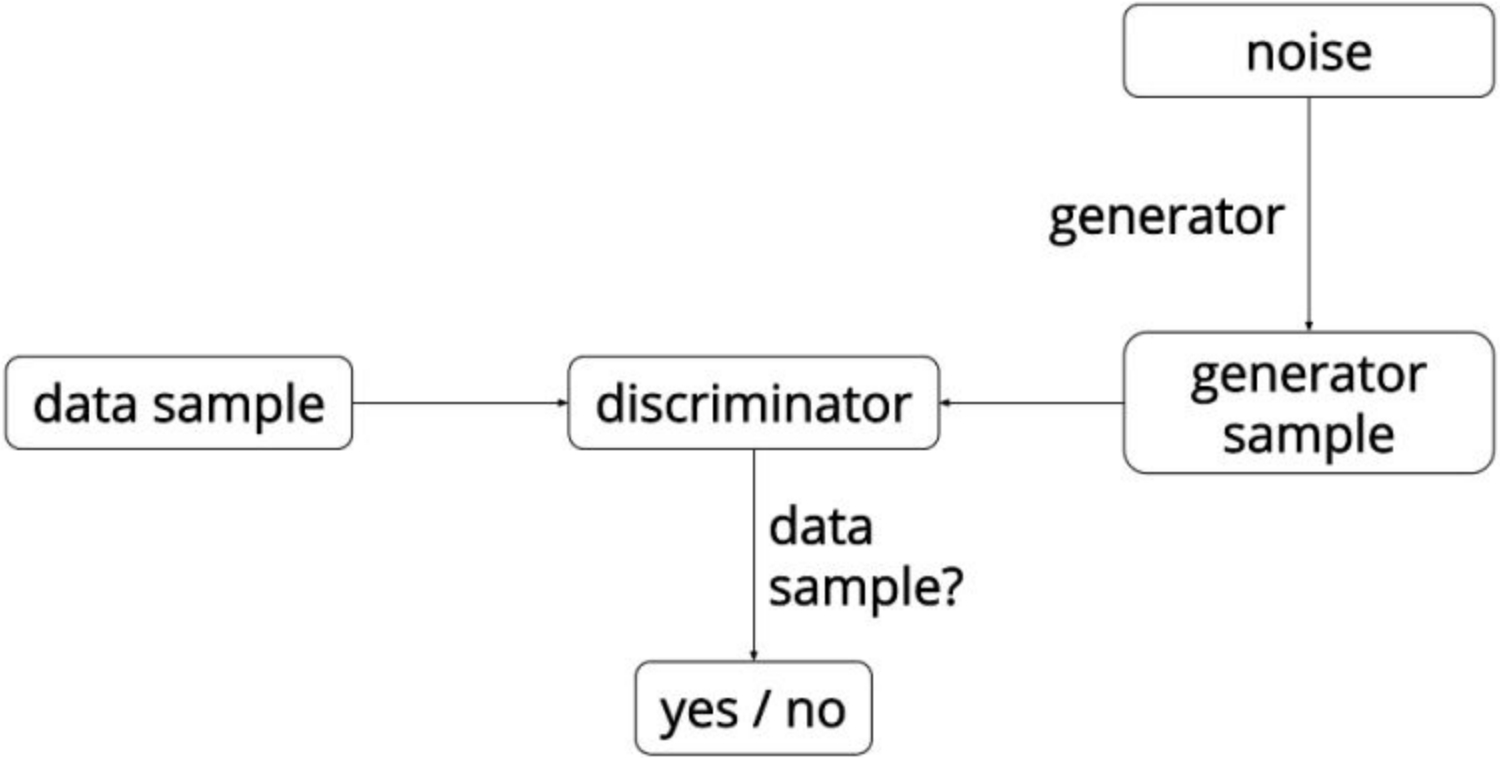
https://arxiv.org/abs/1411.5928

Ресурсы

- https://devblogs.nvidia.com/parallelforall/image-segmentation-using-digits-5
- https://github.com/vdumoulin/conv_arithmetic
- A guide to convolution arithmetic for deep learning https://arxiv.org/abs/1603.07285
- https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d
- https://towardsdatascience.com/transpose-convolution-77818e55a123
- https://www.quora.com/How-does-a-deconvolutional-neural-network-work
- http://stackoverflow.com/questions/35049197/how-does-the-unpooling-and-deconvolution-work-in-deconvnet
- Кейс с убиранием очков https://blog.insightdatascience.com/isee-removing-eyeglasses-from-faces-using-deep-learning-d4e7d935376f

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)



https://arxiv.org/abs/1411.5928

Generative Adversarial Networks (GANs)

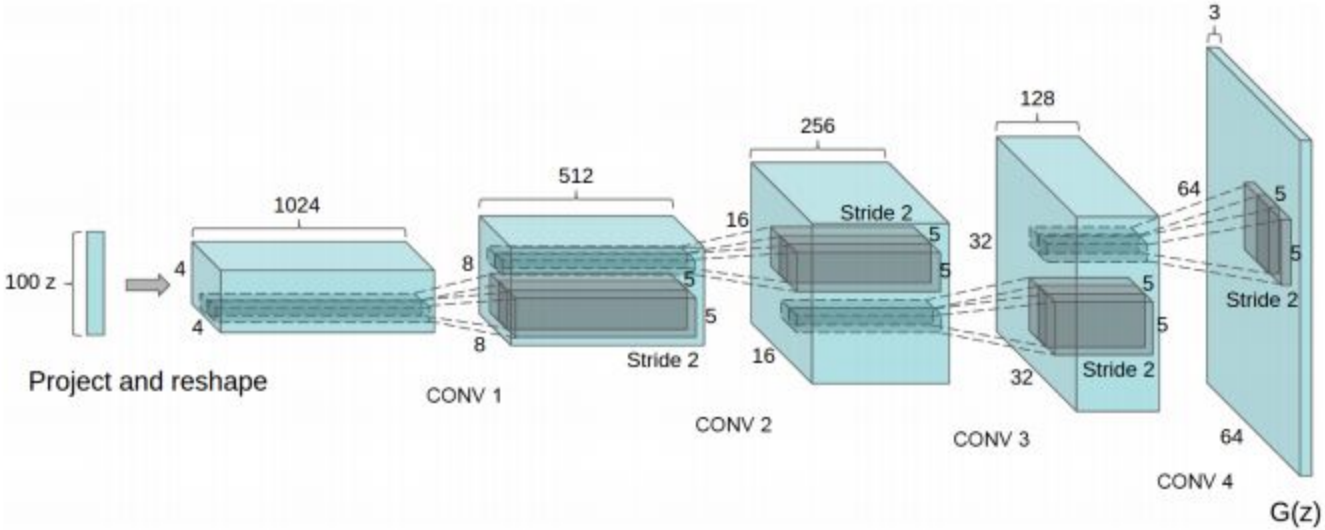


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

https://arxiv.org/pdf/1511.06434v2.pdf

Generative Adversarial Networks (GANs)

Figure 2: Generated bedrooms after one training pass through the dataset. Theoretically, the model could learn to memorize training examples, but this is experimentally unlikely as we train with a small learning rate and minibatch SGD. We are aware of no prior empirical evidence demonstrating memorization with SGD and a small learning rate.

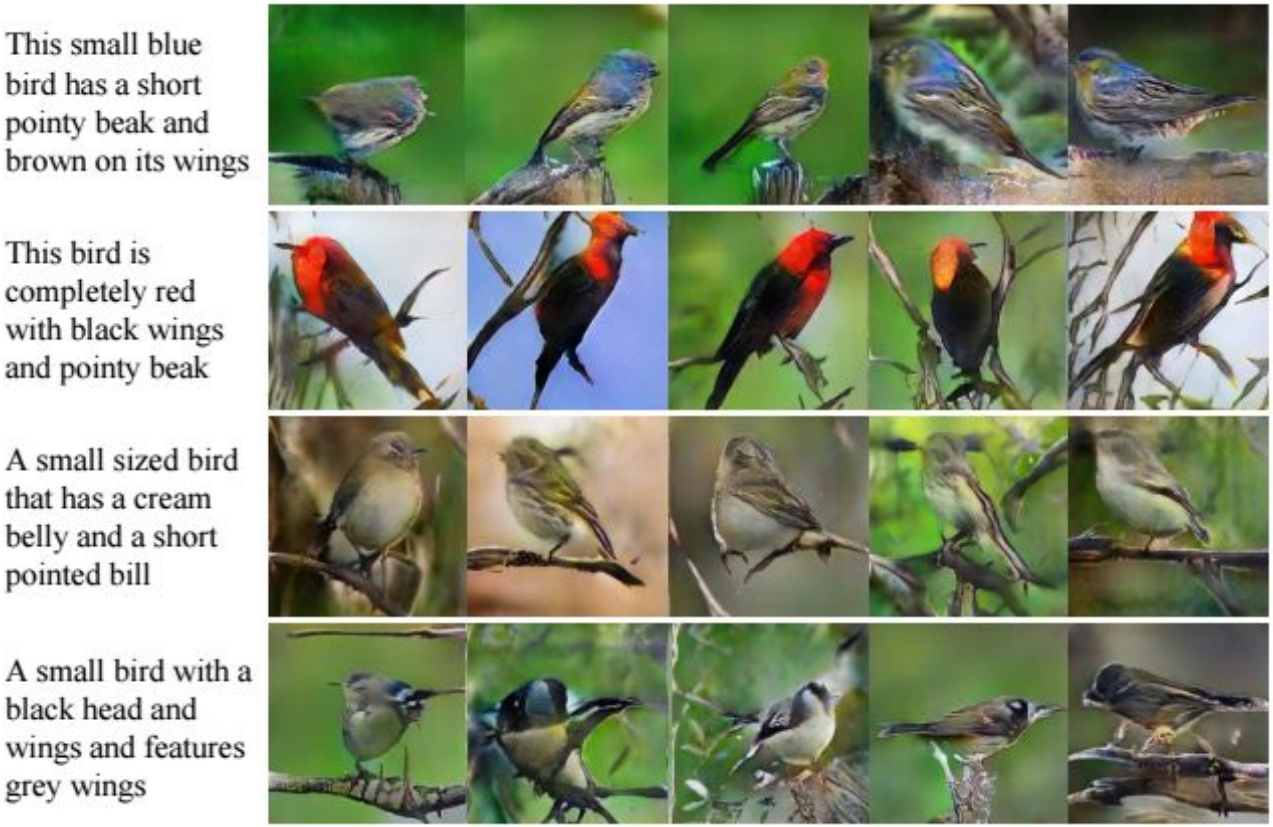
https://arxiv.org/pdf/1511.06434v2.pdf

Generative Adversarial Networks (GANs)

volcano

http://www.evolvingai.org/ppgn

Example: StackGAN



StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, https://arxiv.org/abs/1612.03242

Example: Progressive Growing of GANs

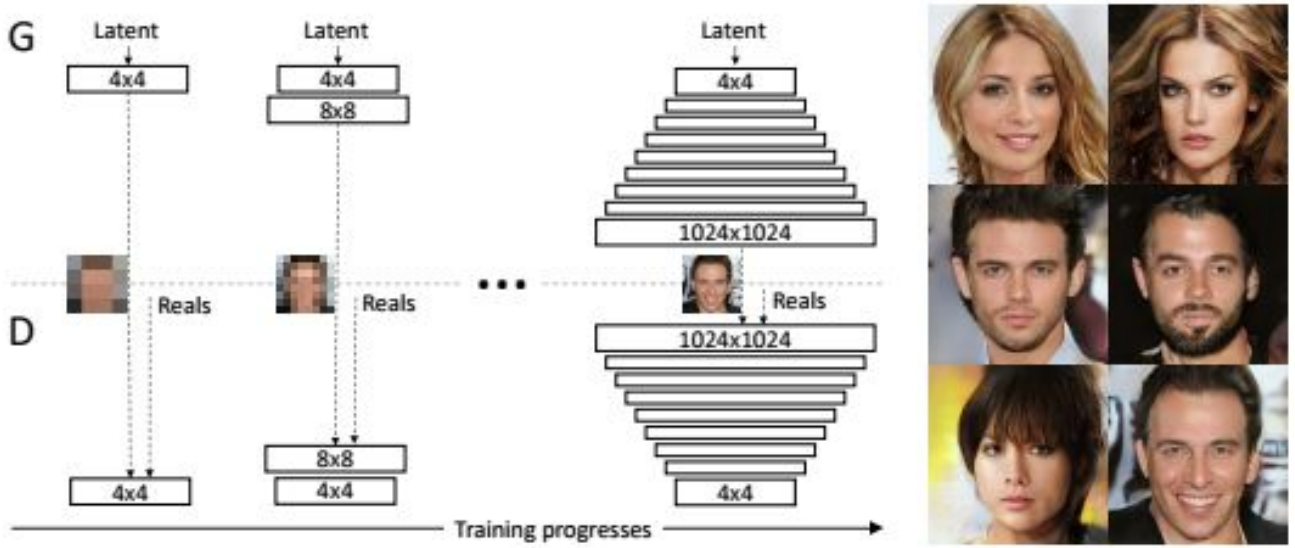


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $N \times N$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

Progressive Growing of GANs for Improved Quality, Stability, and Variation,
https://arxiv.org/abs/1710.10196
https://www.youtube.com/watch?v=XOxxPcy5Gr4

Ресурсы

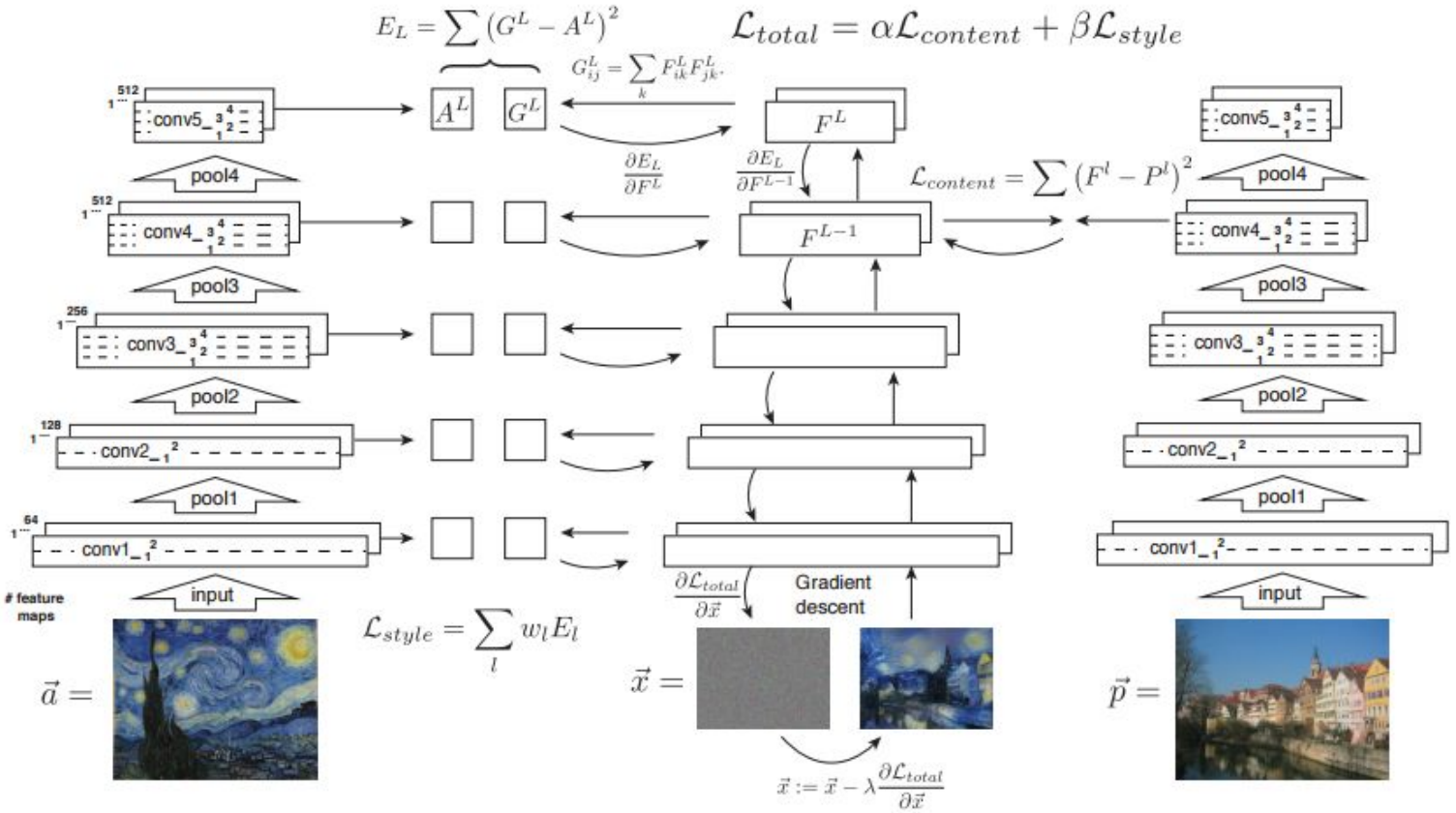
- https://openai.com/blog/generative-models/
- https://www.quora.com/What-are-Generative-Adversarial-Networks
- Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
  https://arxiv.org/abs/1511.06434
- http://blog.aylien.com/introduction-generative-adversarial-networks-code-tensorflow/
- Plug & Play Generative Networks
  http://www.evolvingai.org/ppgn

Кейс: Перенос стиля

Неклассические задачи: перенос стиля

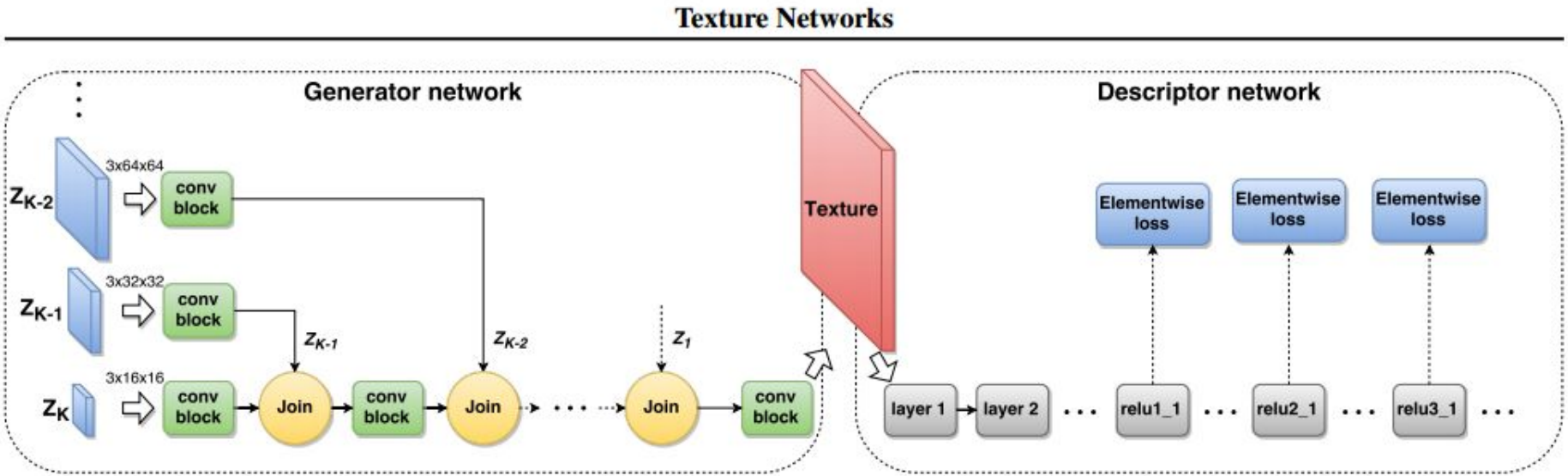
https://arxiv.org/abs/1508.06576

Перенос стиля: оригинальный алгоритм



https://arxiv.org/abs/1508.06576

Перенос стиля: быстрый алгоритм



https://arxiv.org/abs/1603.03417
https://research.googleblog.com/2016/10/supercharging-style-transfer.html

Реализации

Классический алгоритм:

- https://github.com/jcjohnson/neural-style (Torch)
- https://github.com/fzliu/style-transfer (Caffe)
- https://github.com/titu1994/Neural-Style-Transfer (Keras)

Быстрые алгоритмы:

- https://github.com/jcjohnson/fast-neural-style (Torch)
- https://github.com/yusuketomoto/chainer-fast-neuralstyle (Python/Chainer)
- https://github.com/lengstrom/fast-style-transfer (Python/TF)
- https://github.com/DmitryUlyanov/texture_nets (Torch)

Doodle:

- https://github.com/alexjc/neural-doodle