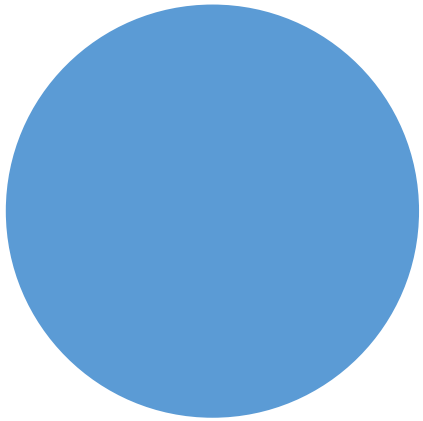
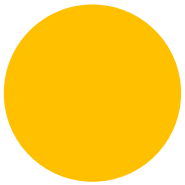
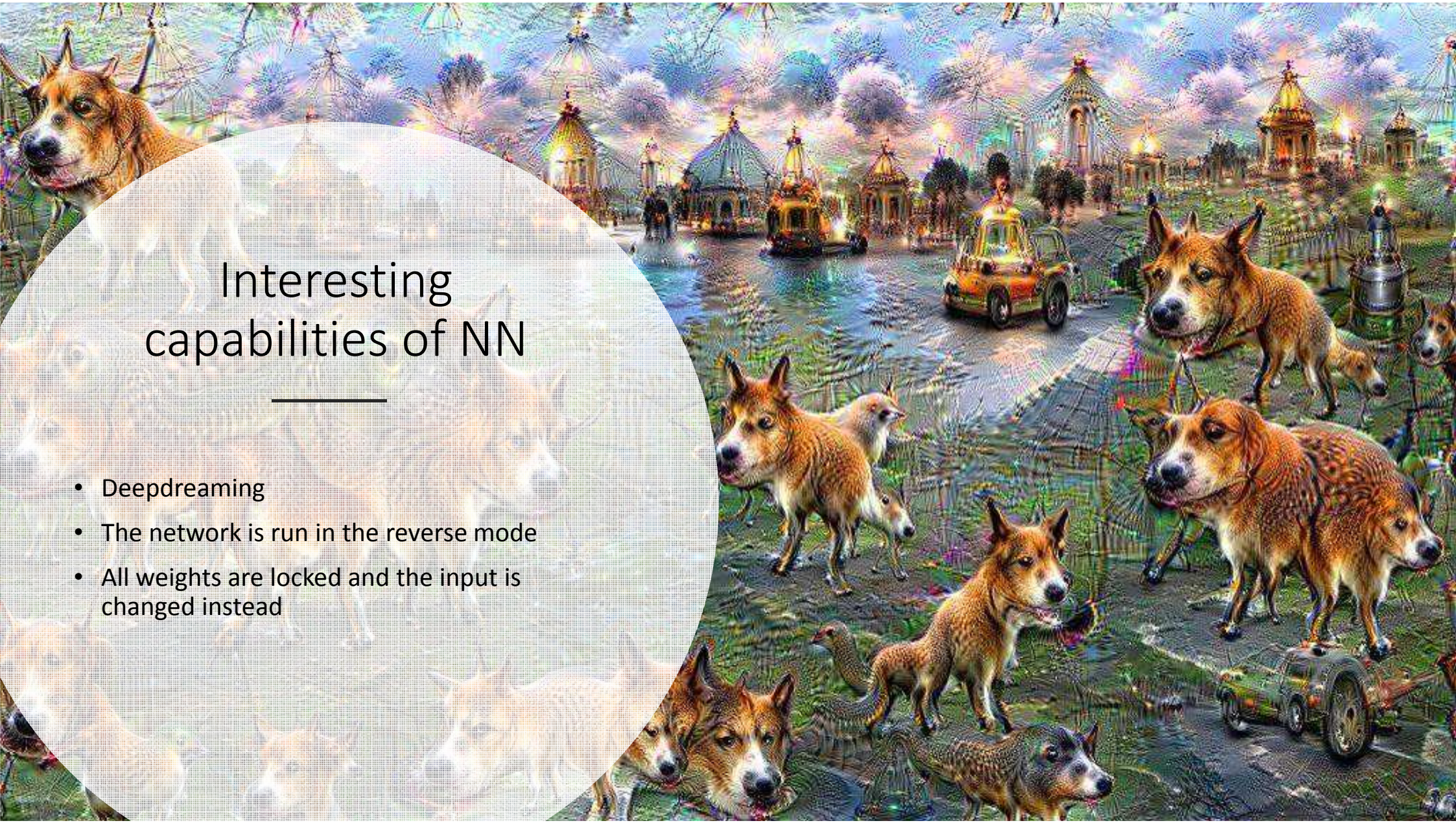



Deep Neural Networks State-of-the-art

NSES-Lecture

Miroslav Hlaváč



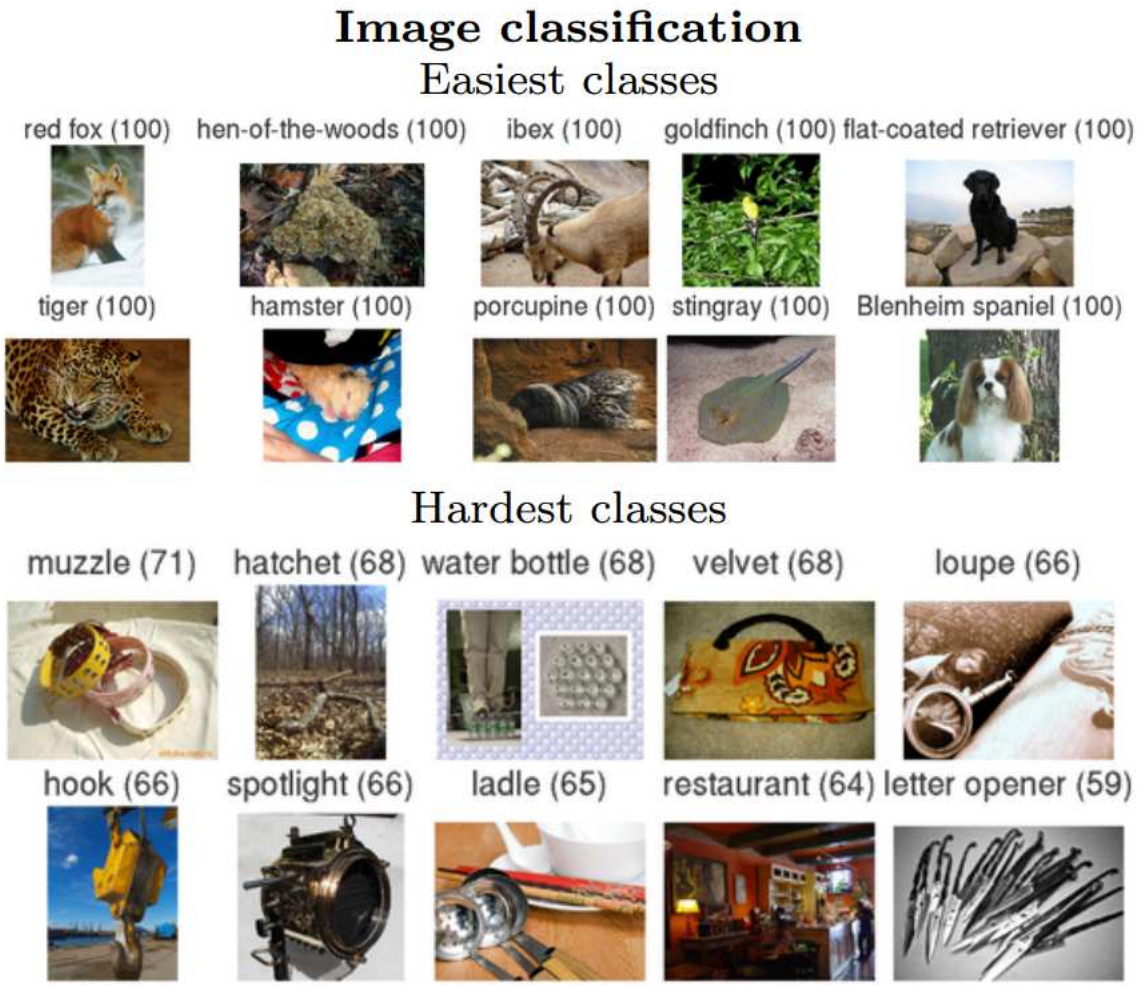
Interesting capabilities of NN

- Deepdreaming
- The network is run in the reverse mode
- All weights are locked and the input is changed instead

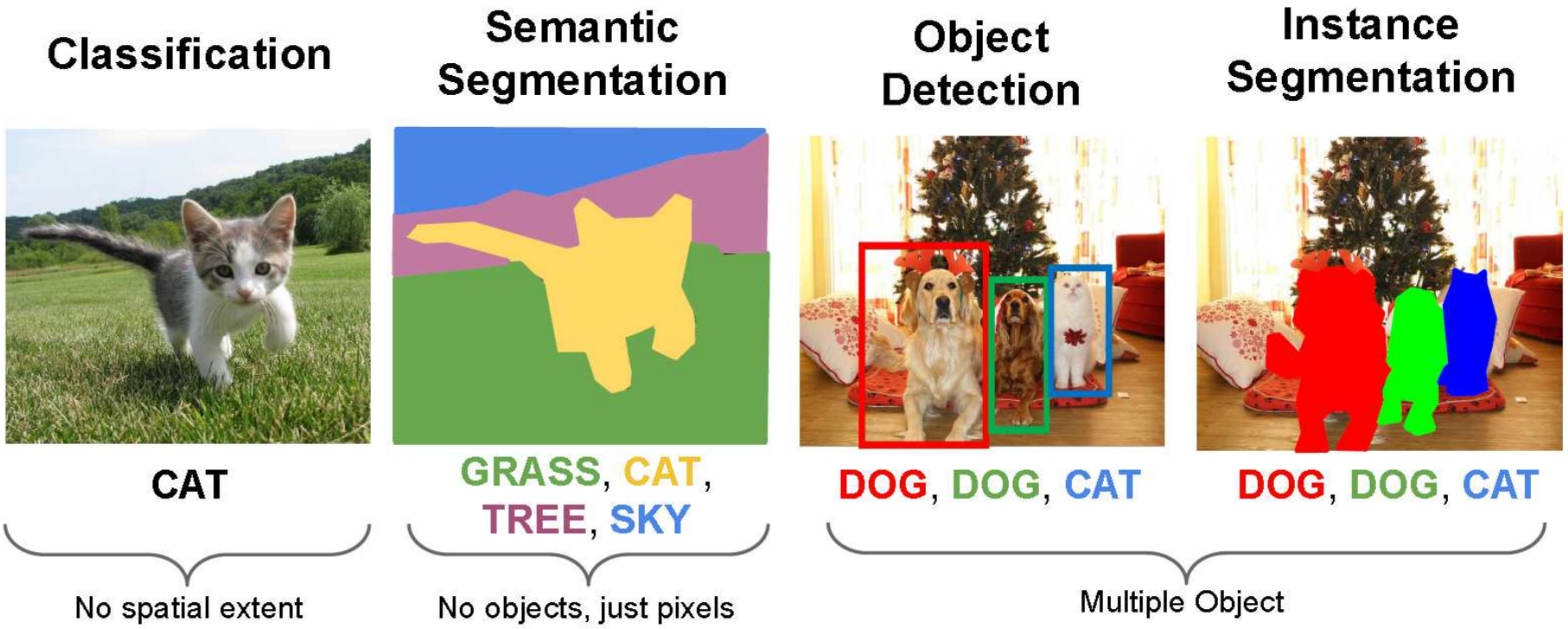
ImageNet

- **Large Scale Visual Recognition Challenge (ILSVRC)**
  - In 2012 – 10 000 000 images in 10 000+ categories
  - Label is provided but location in the image is not
  - 150 00 images for testing
  - 3 tasks:
    - Classification
    - Classification with localisation
    - Fine-grained classification

ImageNet Examples



Task Definition



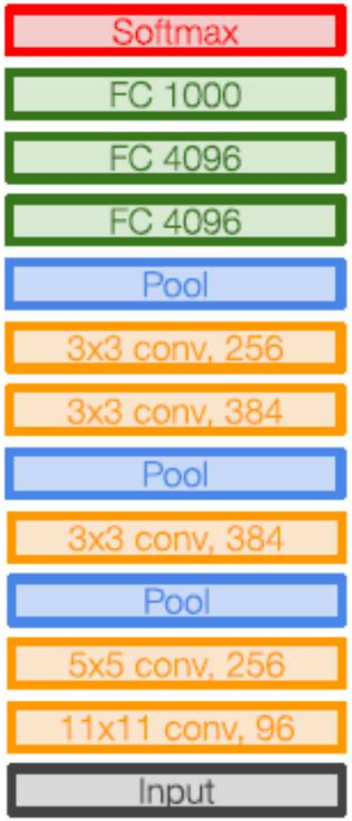
Task 1: Classification

AlexNet

- This network started the massive use of convolutional neural networks in 2012
- Utilizes ReLU activation function for the first time
- Originally written with CUDA
- First winner of ImageNet based on CNN

AlexNet Structure

- Basic structure idea comes from LeNet-5(1998)
  - CONV-POOL-CONV-POOL-FC-FC
- Normalisation Layers used between blocks
- Input size – 227x227x3
- Total of 8 layers
- 62M Parameters



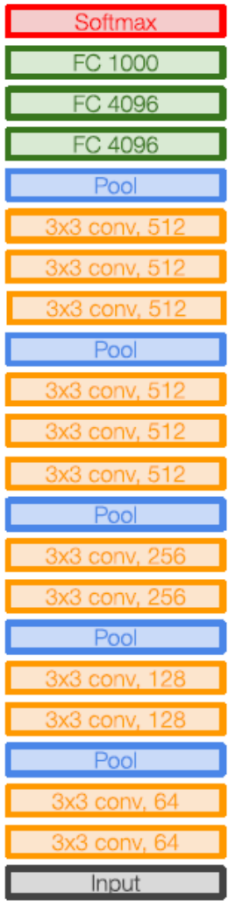
AlexNet

VGG

- Evolution of AlexNet from 2014
- Introduces deeper structure of convolutions
  - Different number of stacks of convolutional layers followed by maxpooling
- Utilizing growing structure of convolution filters

VGG Structure

- Stack of Convolutions followed by Pooling layer
- Implements growing number of Convolutional filters
- All of the filters are 3x3
- No normalisation layers
- Two versions
  - VGG16 - 138M Parameters
  - VGG19 - 144M Parameters



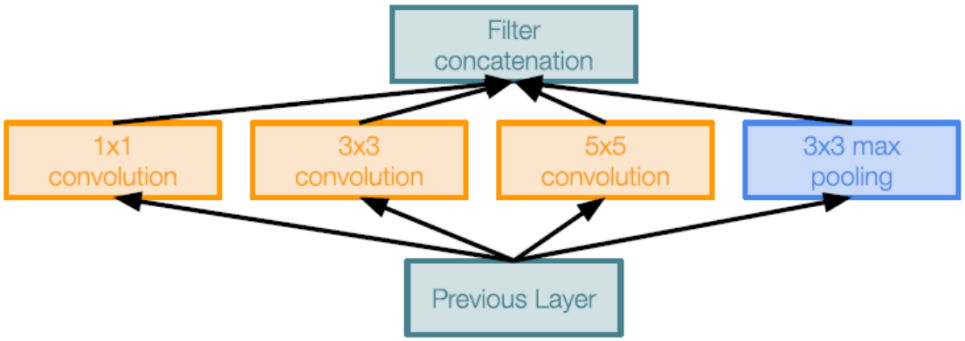
VGG16

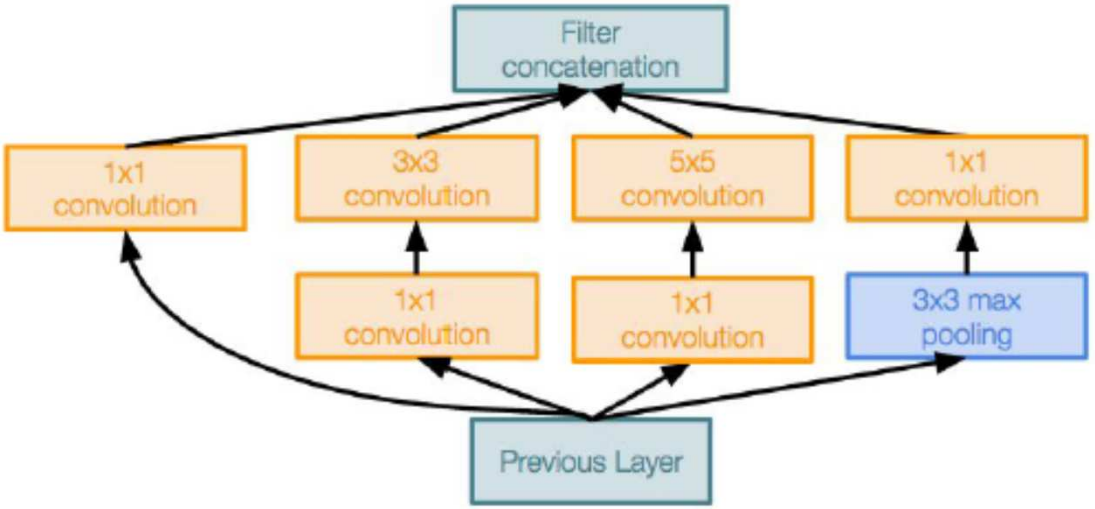
Receptive field

- The area visible to the network
  - The consecutive convolutional layers with kernels 3x3 and stride 1 has the same receptive field as one convolutional layer with kernel 7x7
  - The difference is in the number of parameters needed to train these layers
  - 7x7 -> 7x7xCxK (C – number of channels, K – number of kernels)
    - For input with 3 channels and 64 kernels we get 9408 parameters
  - 3x3 -> 3x3xCxK + 3x3x64 + 3x3x64 = 1728 + 576 + 576 = 2880
  - There are also two more activation functions in the 3x3 stack

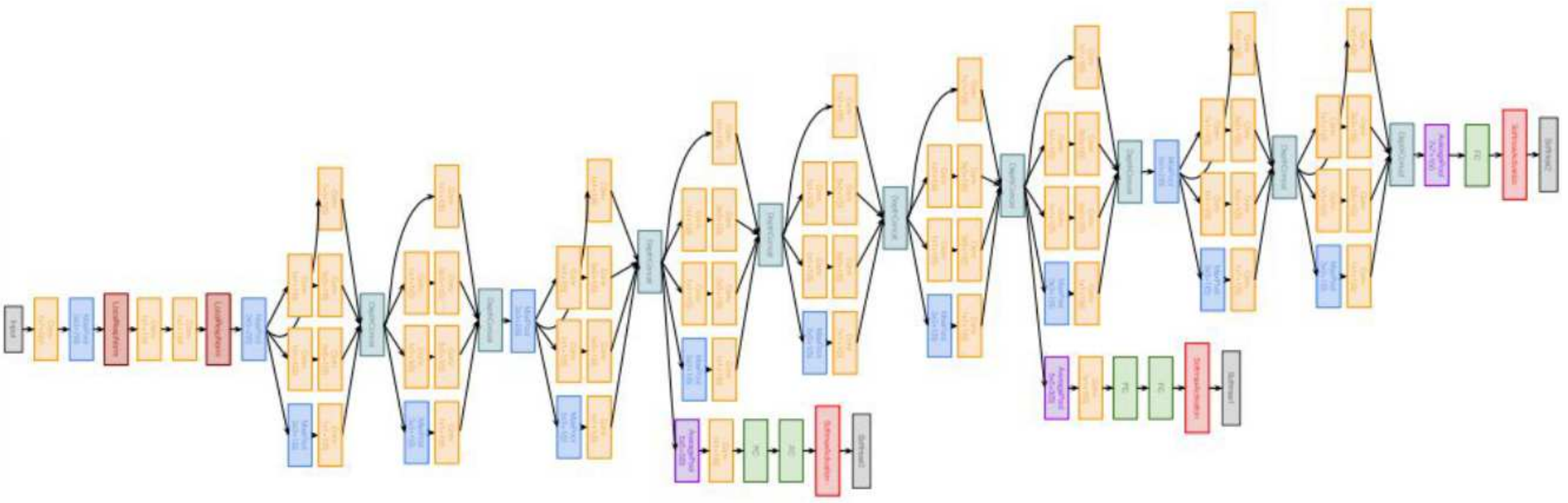
Inception

- Inception module represents an approach called NiN (Network in Network)
- Processing of data in different reception fields
- Outputs concatenated and filtered with 1x1 convolutional layer





GoogLeNet



GoogLeNet

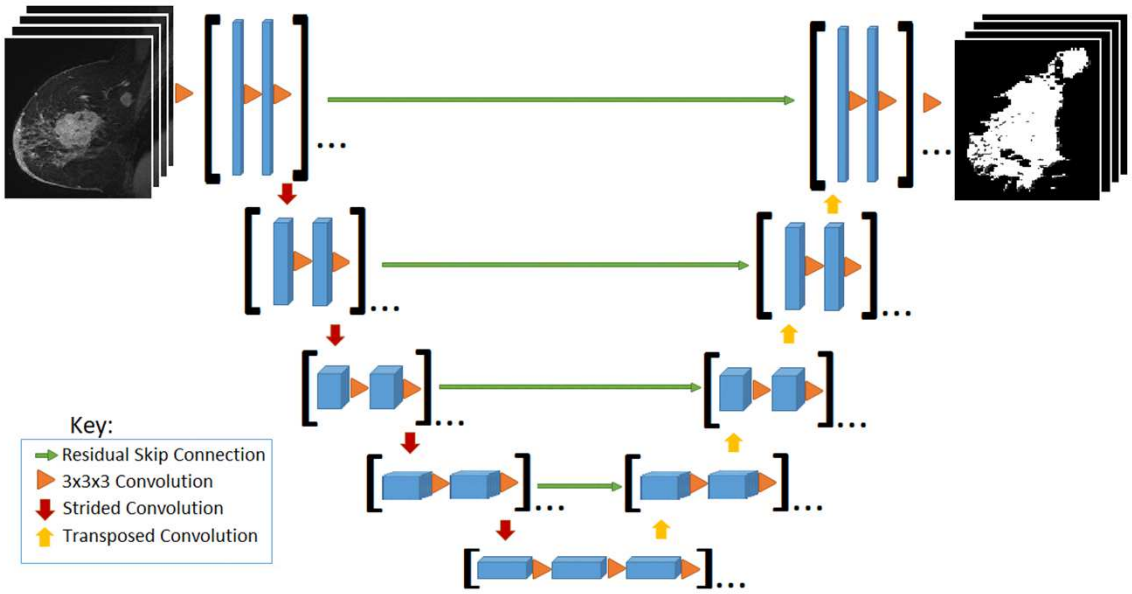
- Deeper network consisting of 22 layers
- Fully convolutional – 0 Fully connected layers
- “Only” 5 million parameters in total
- Winner of 2014 ImageNet competition

Residual Learning

- The problem with Very Deep Neural networks is the diminishing of gradients during backpropagation
- One of the solutions is to introduce so called skip connections that transfer unprocessed information from previous steps to next layers in the forward pass and also serve as a channel for unweighted gradient backpropagation during the backward pass

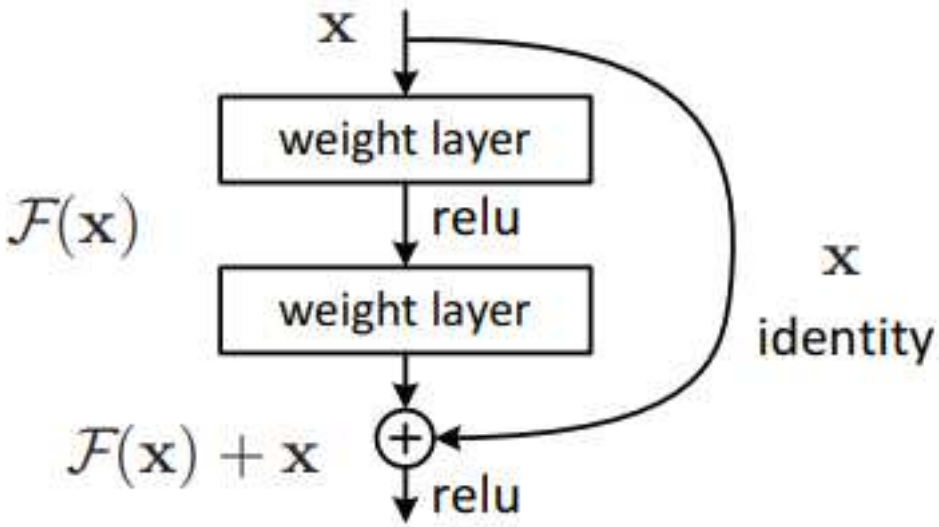
U-Net

- Special type of network implementing bottle neck with residual skip connections
- Convolutions for down-sampling of the input
- Deconvolutions for up-sampling
  - Deconvolution can be interpreted as convolution with transposed kernel



ResNet – very deep neural network

- First network to introduce skip connections – short-cuts in 2015
- Inspired by pyramidal cells in human brain
- Residual learning block:
- This approach improves the training process for very deep networks and allows successful learning with basic SGD algorithms



Resnet

- First Very Deep Neural Network
- Test with hundreds of layers
- Showed that more layers can improve results but only to a certain point
  - Trade of between speed a computing resources over accuracy

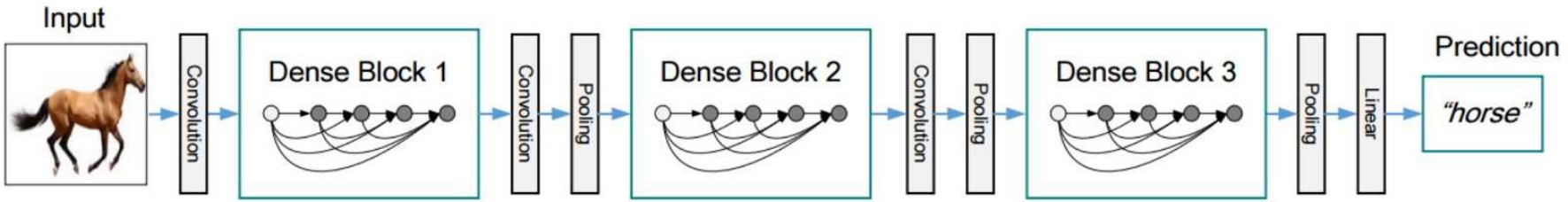
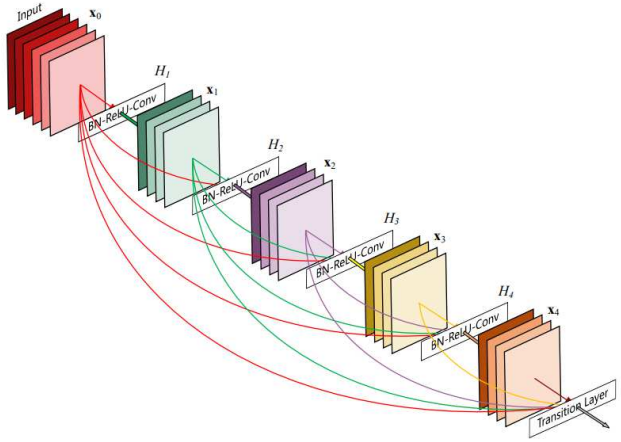


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

Dense Net

- Inspired by skip connections from ResNet and other networks
- Implements several parallel skip connections – dense blocks



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

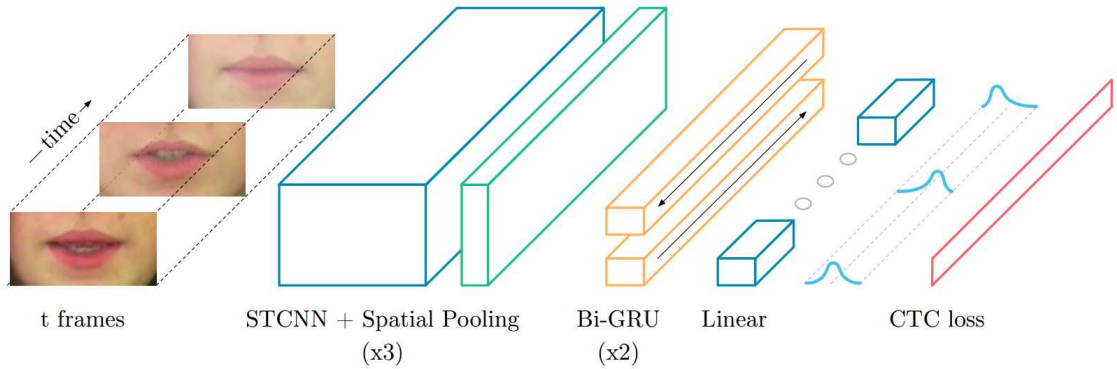
Dense block

- Each layer takes all the preceding feature maps as input
- Different topologies are described in the table
- This network actually uses less trainable parameters thanks to the skip connections

Visual Speech Recognition

LipNet – end-to-end lipreading

- Recurrent network for lipreading
- Implements 3D convolutions, Bidirectional Gated Recurrent Units(GRU), and Connectionist temporal classification

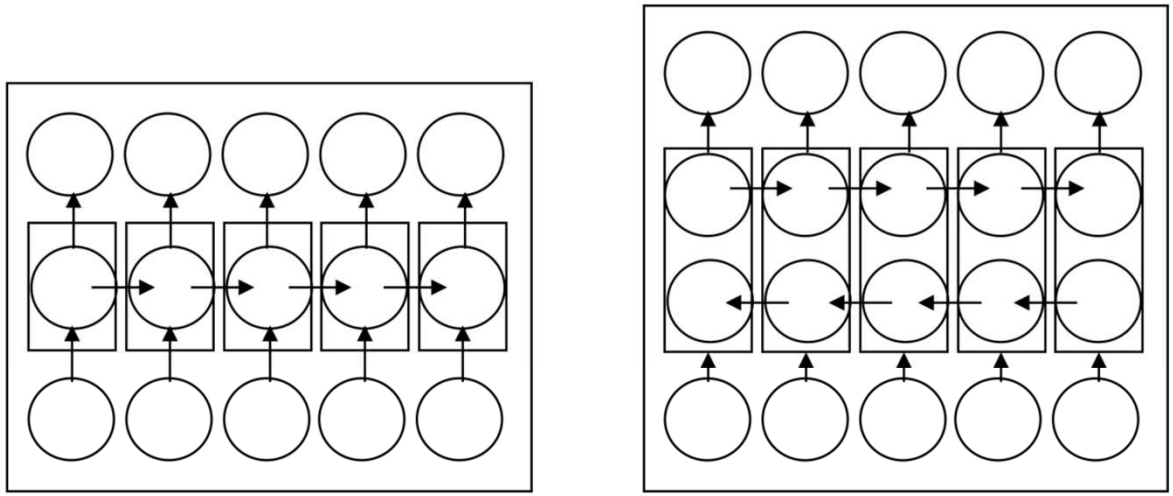


Spatio-temporal Convolution(3D Convolution)

- Special type of convolution with three dimensional kernel
  - Typically cube 3x3x3, but can be any size
- Enables analysis of dynamic(temporal) properties of sequences
  - In a video sequence – analysis of changes occurring between frames
- Can be used in different configurations
  - Depth axis can span across the whole sequence

Bi-Directional GRU

Provides the ability to analyse the input sequence from both directions



(a) (b)

Structure overview

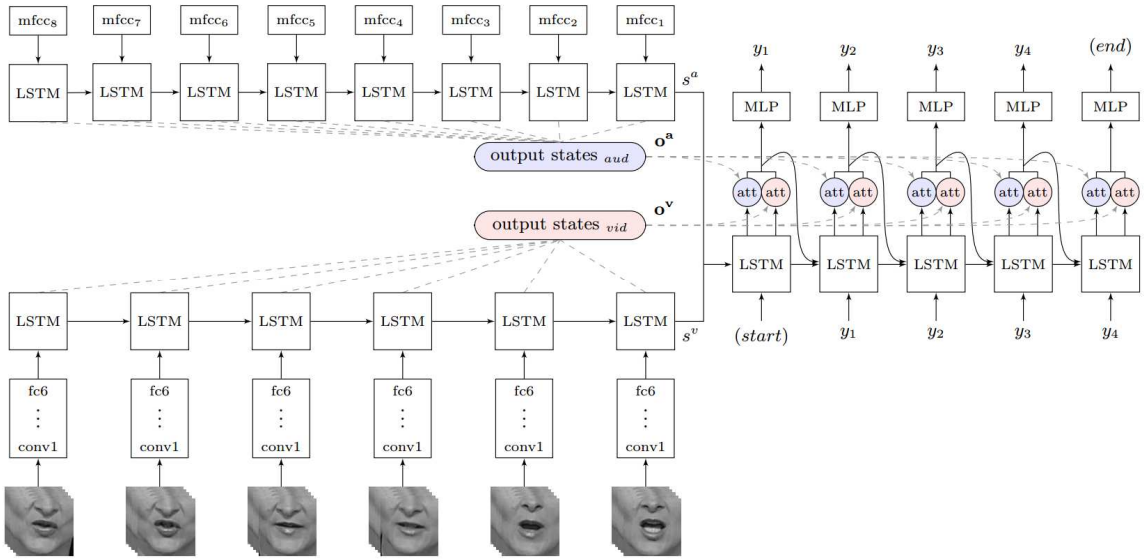
(a) unidirectional RNN
(b) bidirectional RNN

Connectionist Temporal Classification

- Special type of scoring function that looks at the output sequence as a whole
- Does not try to find the alignment between the input sequence and the output sequence
- Incorporates a blank symbol

WLAS – Watch, Listen, Attend and Spell

- Takes images and MFCC features as input to provide character-based output
- Audio and video parts can work independently
- Implements attention system – keeps track of synchronization between audio and video



Attention Mechanism

Provides a way to analyse the hidden states of a sequence as a whole