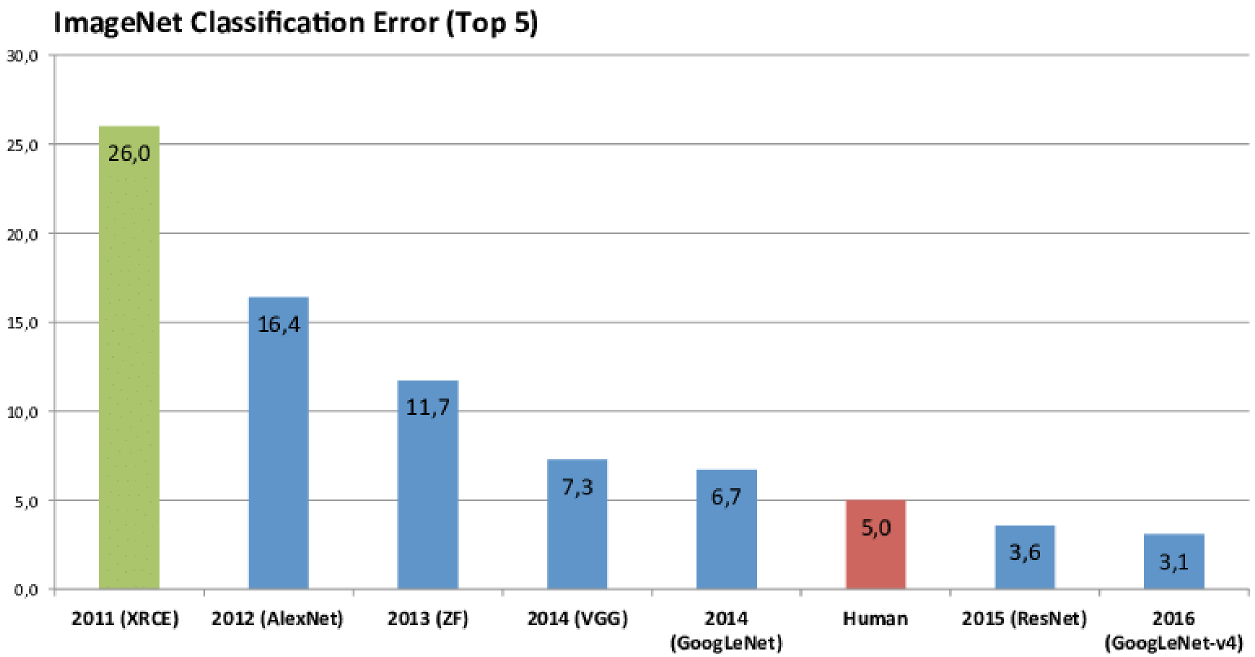
MONASH
University

Introduction to
Convolutional Neural Networks

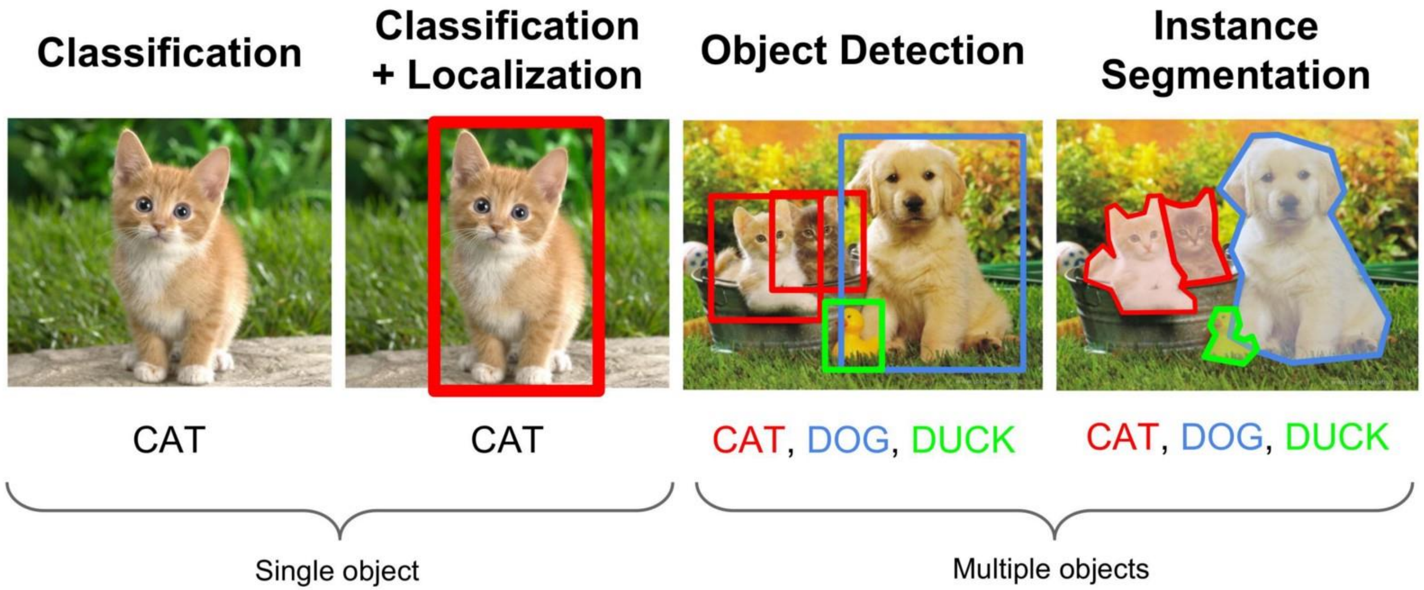
Developed by Titus Tang 2020

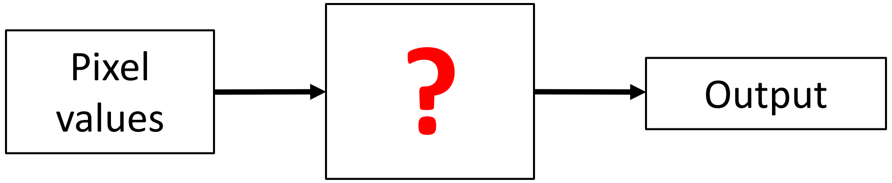
Beating humans since 2015



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
Object localisation on 1000 categories.
Test and validation set of 150,000 images.

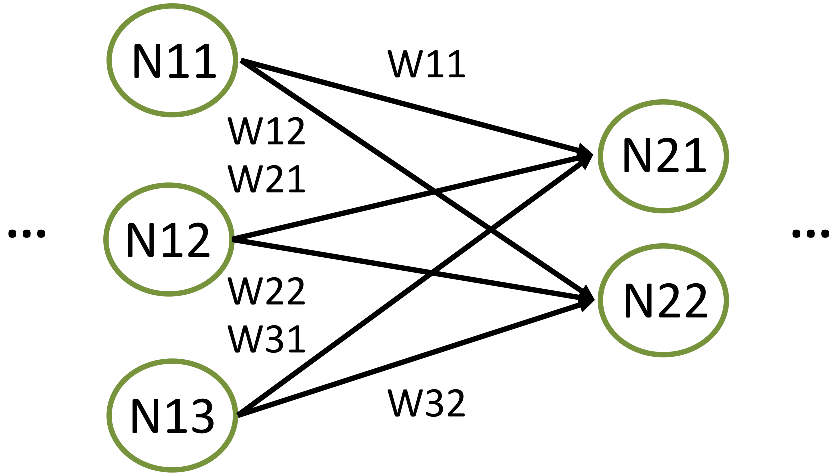
Computer Vision Tasks





How do you process visual information at the pixel level?

Can we use a Multilayer Perceptron?

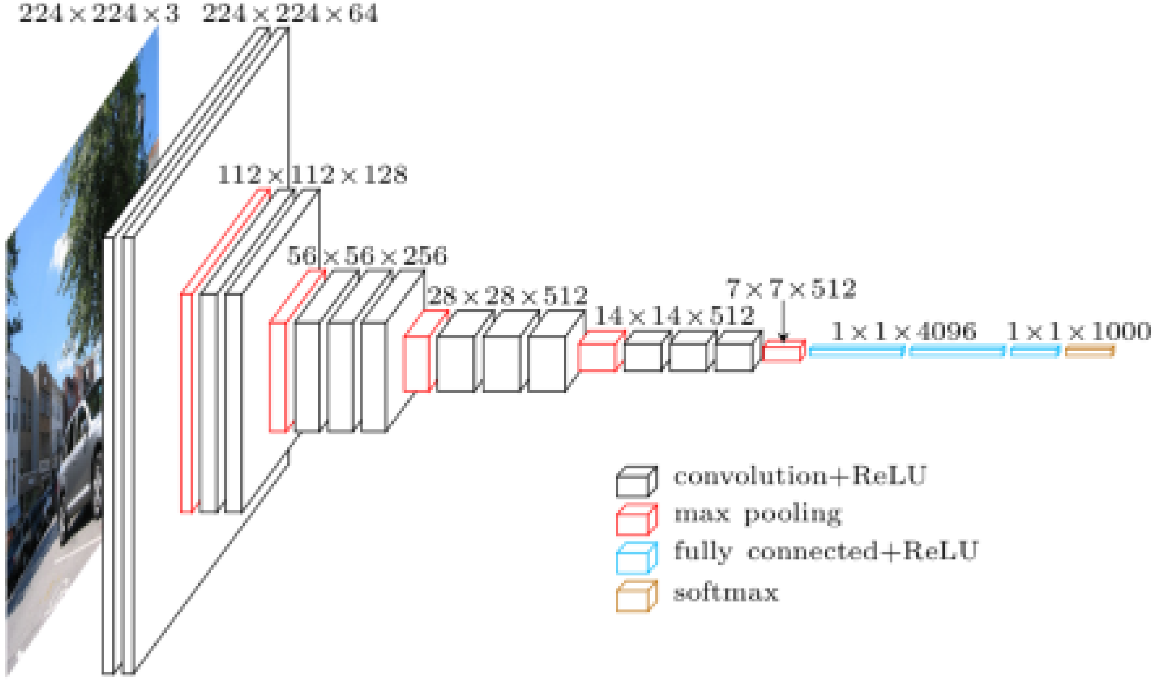


✓ Each perceptron in layer N will have #Perceptrons(Layer(N-1))+1 weights.

✓ 1000x1000 pixel image = 1M weights per perceptron.

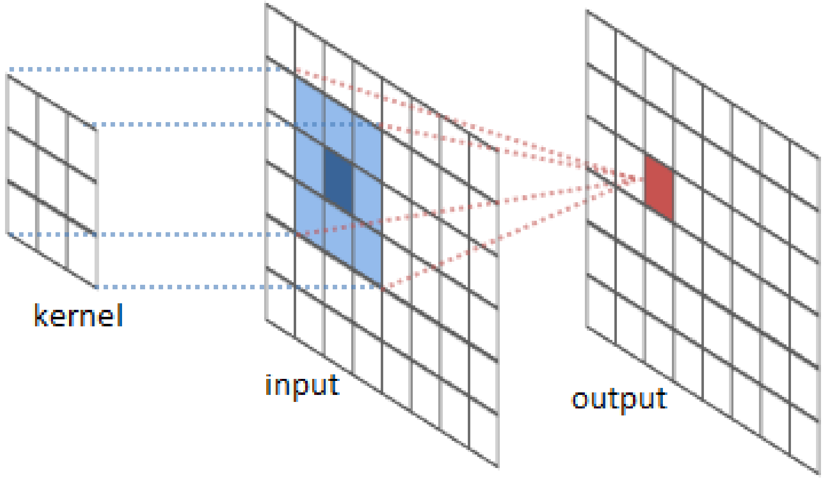
✓ If 1 perceptron per pixel, there will be ~1,000,000,000,000 weights in each of N layers. Intractable!

Overview of a Convolutional Neural Net for Classification



- ✓ The aim is to choose layers that go from a large input image to a specific desired output format.
- ✓ Repeated stacking of layers in a specific order.

Convolutions



$$Output = \sum Kernel_{i,j} * Input_{i,j}$$

✓ The kernel represents the pattern to be detected.

✓ The more the input matches the kernel, the more positive the output response would be.

✓ Convolutions are pattern detectors.

Convolution Example

Input patch Kernel Output

$$\begin{array}{|c|c|c|}\hline 10 & 6 & 6 \\ \hline 0 & 10 & 6 \\ \hline 0 & 0 & 10 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 0.5 & 0.5 \\ \hline -1 & 1 & 0.5 \\ \hline -1 & -1 & 1 \\ \hline \end{array} = \boxed{39}$$

Good match

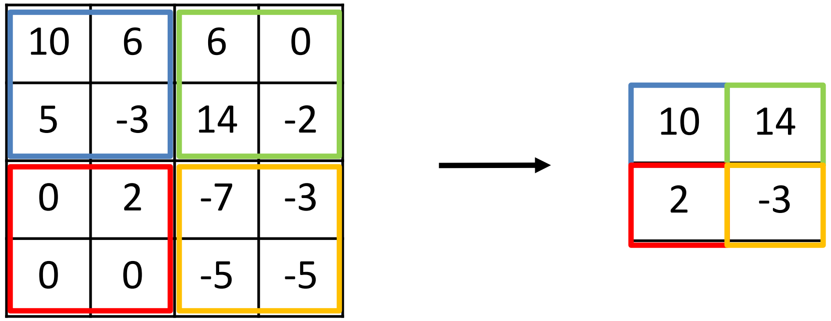
$$\begin{array}{|c|c|c|}\hline 6 & 6 & 10 \\ \hline 6 & 10 & 0 \\ \hline 10 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 0.5 & 0.5 \\ \hline -1 & 1 & 0.5 \\ \hline -1 & -1 & 1 \\ \hline \end{array} = \boxed{8}$$

Poor match

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 6 & 6 & 6 \\ \hline 10 & 10 & 10 \\ \hline \end{array} * \begin{array}{|c|c|c|}\hline 1 & 0.5 & 0.5 \\ \hline -1 & 1 & 0.5 \\ \hline -1 & -1 & 1 \\ \hline \end{array} = \boxed{-7}$$

Poor match

Pooling



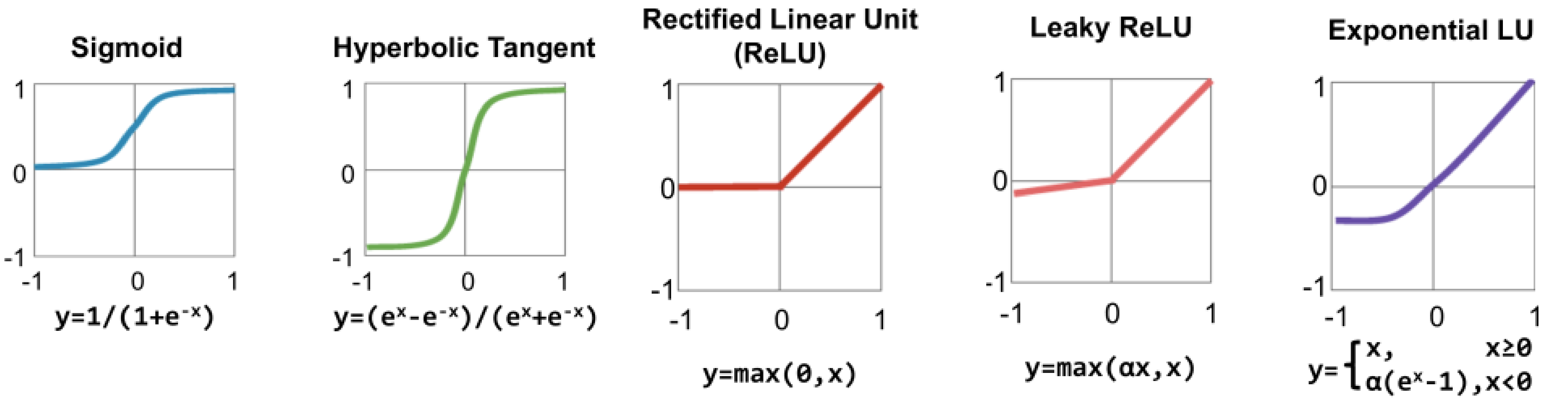
10	6	6	0
5	-3	14	-2
0	2	-7	-3
0	0	-5	-5

10	14
2	-3

Max pooling

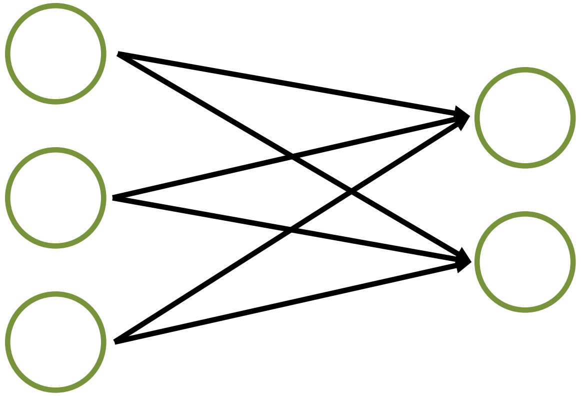
- ✓ Slide a window across the input and pick a value at every window position.
- ✓ Max pooling – take the max value.
- ✓ Average pooling – take the average value.
- ✓ Pooling layers are information filters.

Activation Functions



- ✓ Introduces nonlinearities to the network.
- ✓ Allows for the modelling of more complex non-linear functions.
- ✓ Most popular is the ReLU.

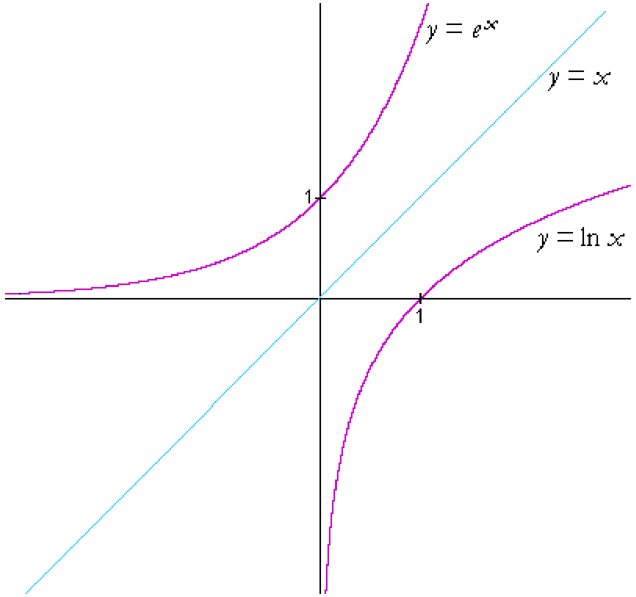
Fully Connected Layers



- ✓ Nodes in one layer are connected to every node in the next layer.
- ✓ Used to “see the big picture” of what is happening.
- ✓ An aggregator of information.

Softmax

$$Softmax(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$$



✓ Converts all input to a range between 0 and +infinity.

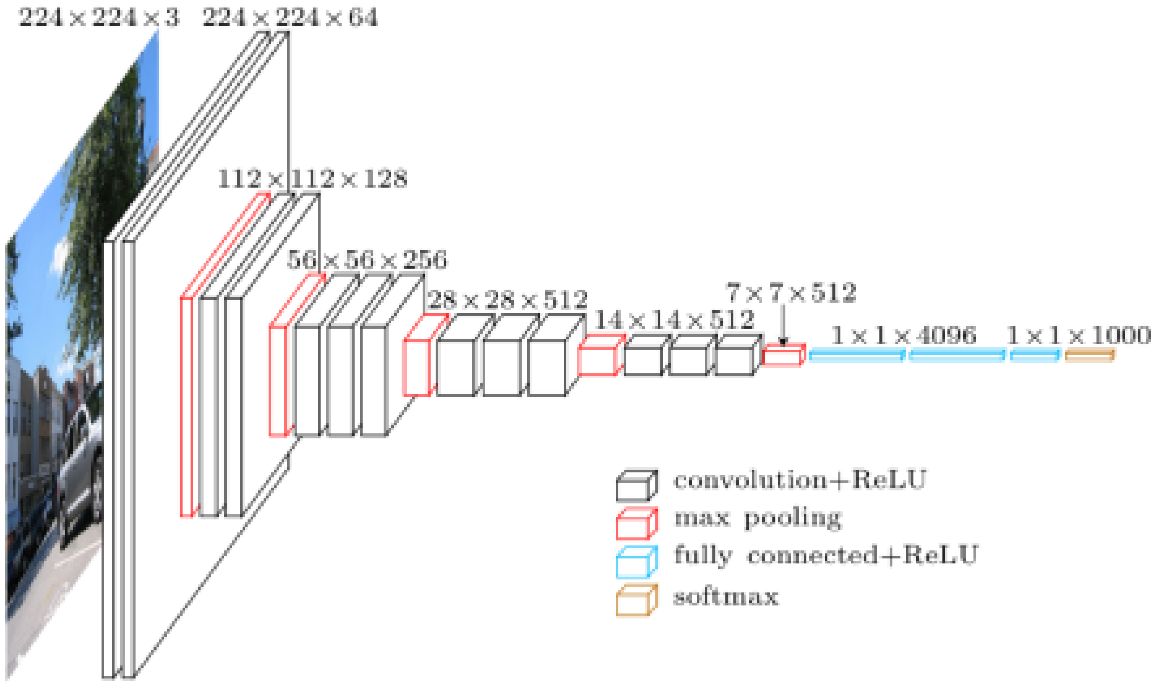
✓ Then normalises all values between 0 and 1.

✓ A convenient way to interpret network output as

probabilities.

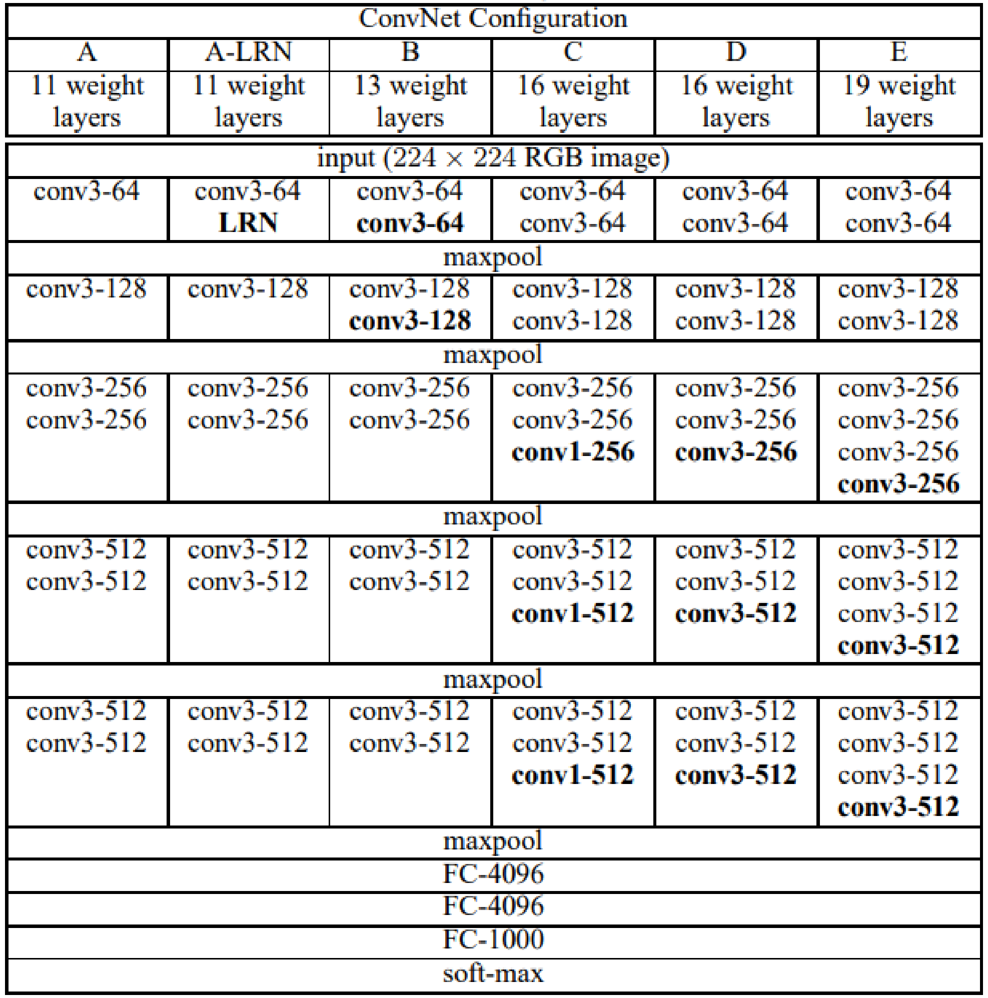
✓ e^x used because it is easily differentiable.

Building a Convolutional Neural Net



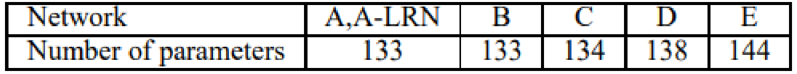
- ✓ Convolutions to find features, activations to filter good features, pooling to select the best features.

- ✓ Repeat stacking of layers to find features of features -> high level features.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

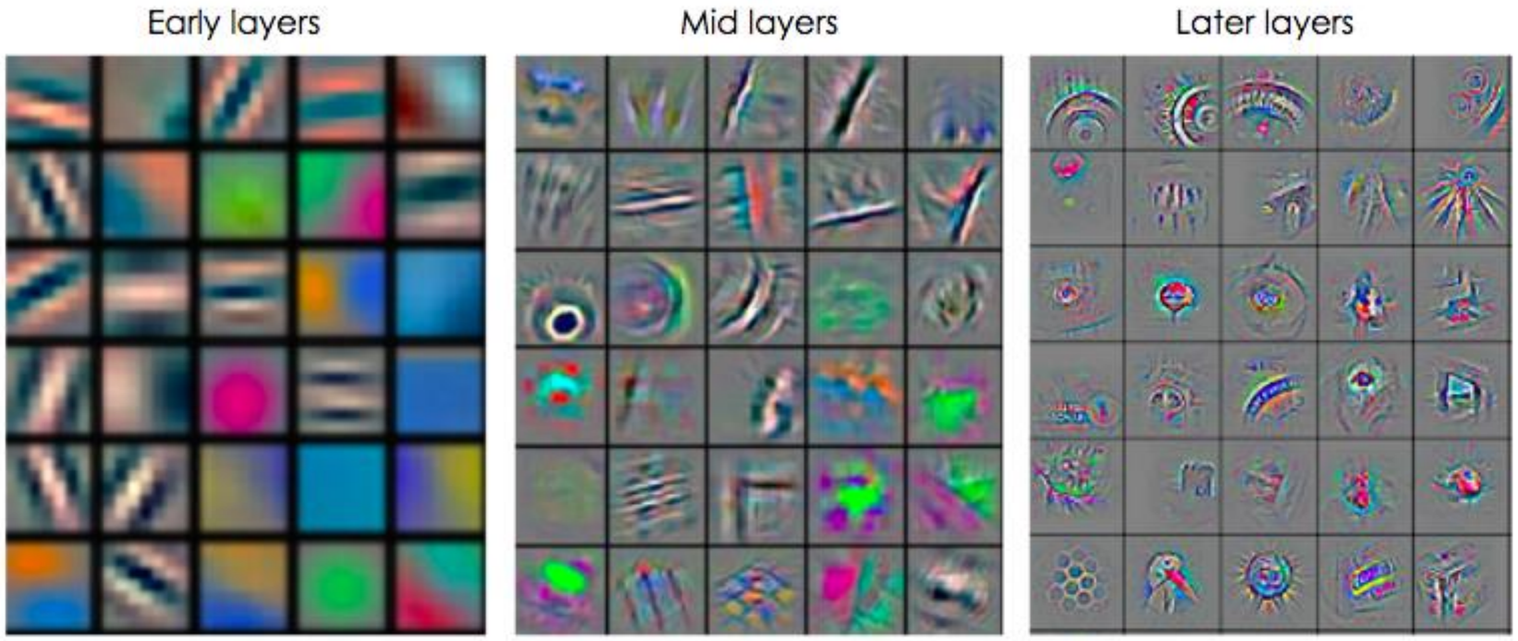
Table 2: **Number of parameters** (in millions).



Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Variants of the VGG network

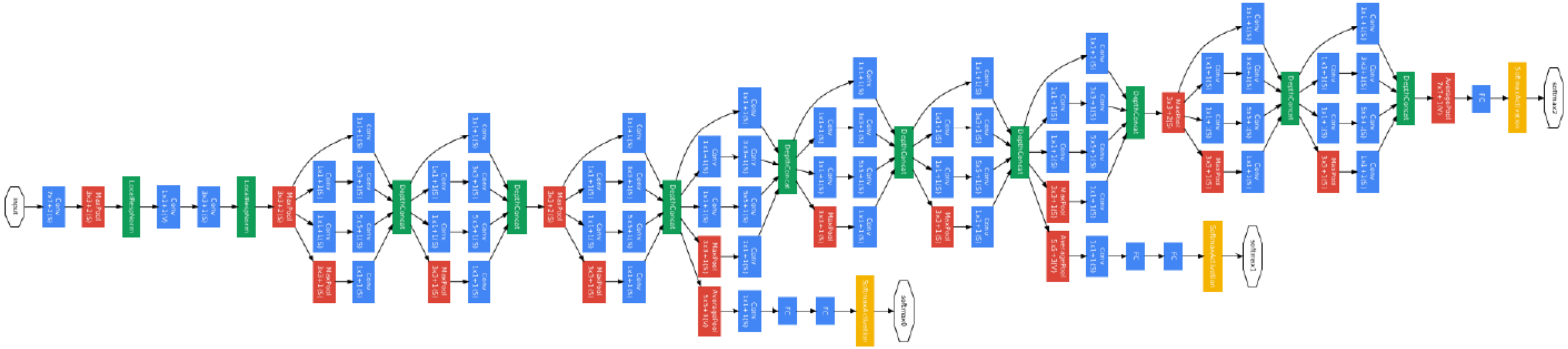
Visualising Layers of a Convolutional Neural Net

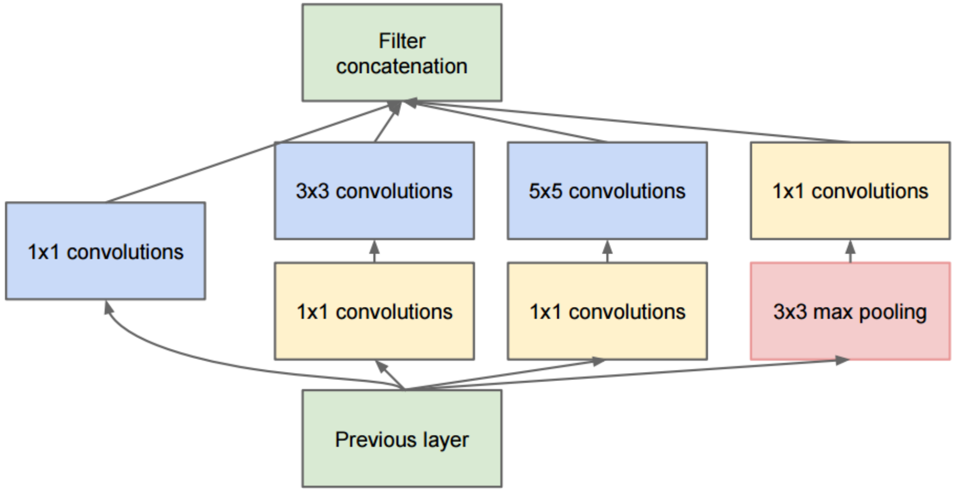


Low level features → High level features

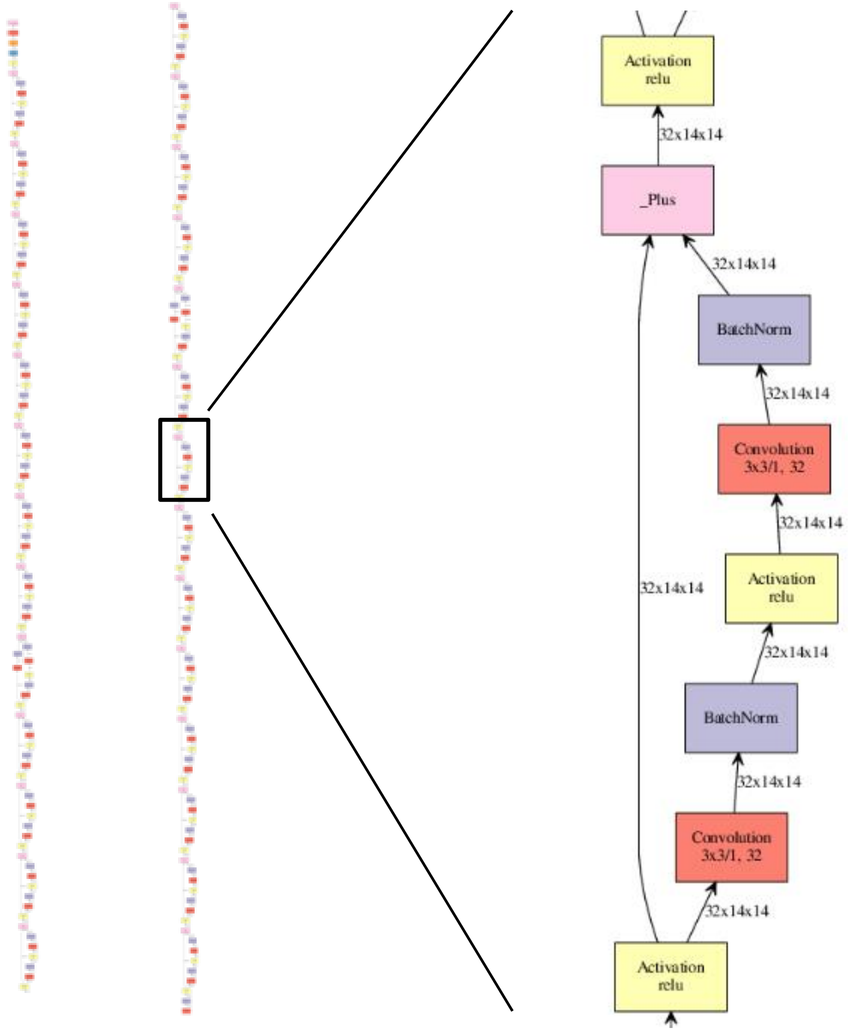
- ✓ Low level features consists of simple geometrical patterns.
- ✓ The higher you go, the more object-like patterns you can identify.

GoogleNet




Filter concatenation
1x1 convolutions
3x3 convolutions
5x5 convolutions
1x1 convolutions
1x1 convolutions
1x1 convolutions
3x3 max pooling
Previous layer

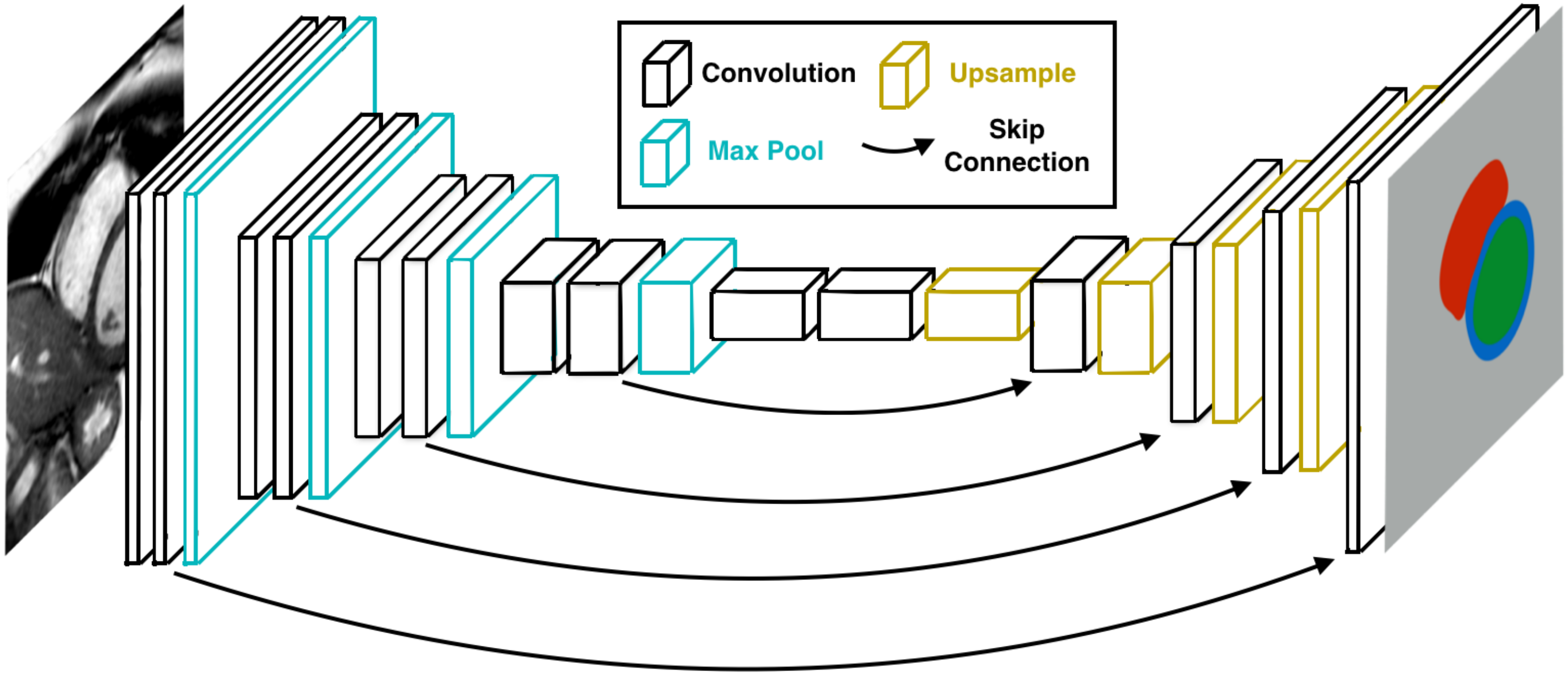
Residual Networks - Resnet



Resnet-56, 2015

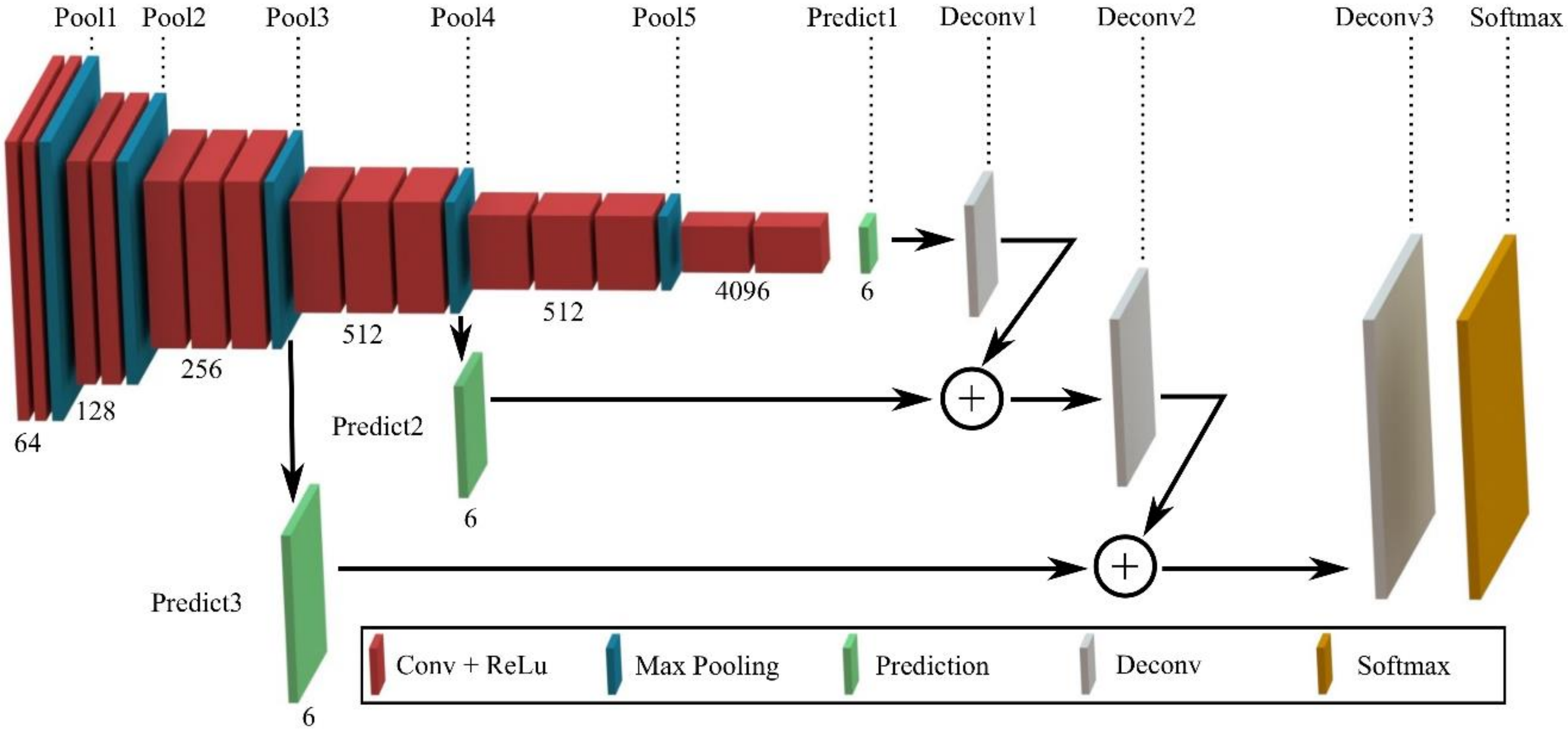
- ✓ The right branch contains the typical layers of a conv-net.
- ✓ The left branch allows the backpropagated errors to flow directly through without being diminished through the layers on the right.
- ✓ Avoids the vanishing gradient problem.

Pixel-wise Predictions for Segmentation

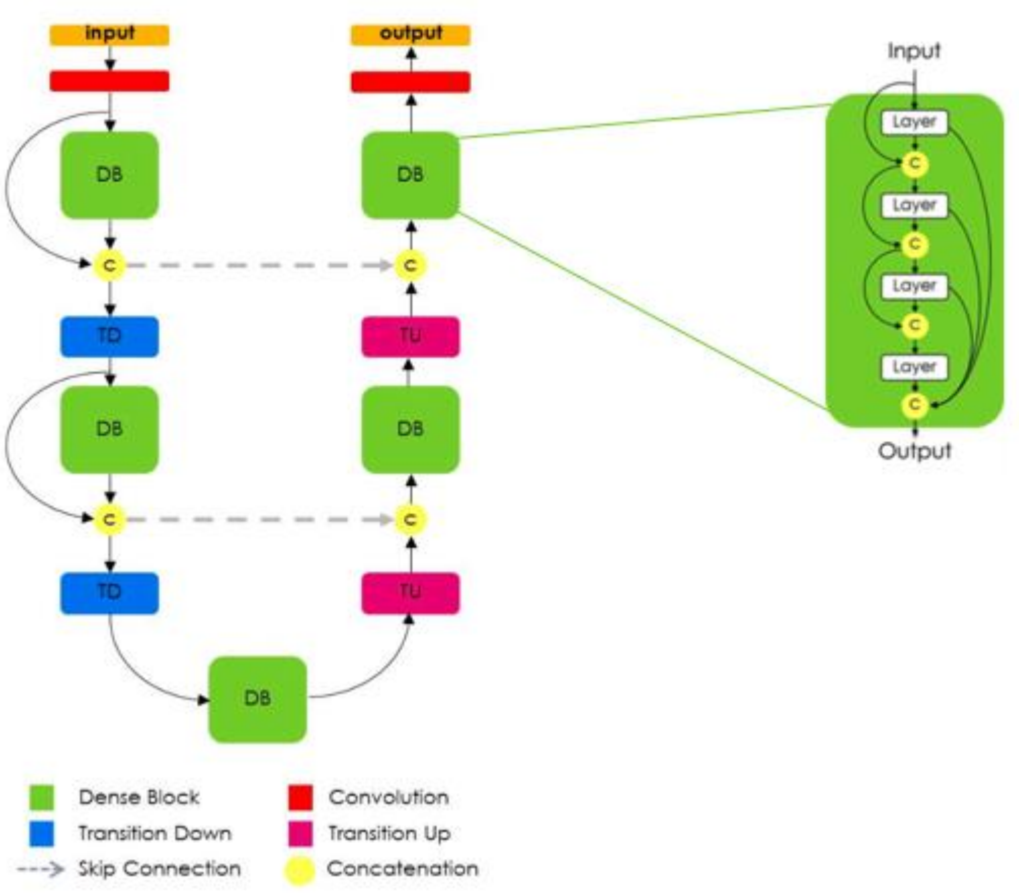


- ✓ Regenerates an output the same size as the input.
- ✓ Skip connections are vital in maintaining spatial coherency.
- ✓ Often called a U-net architecture.

Pixel-wise Predictions for Segmentation



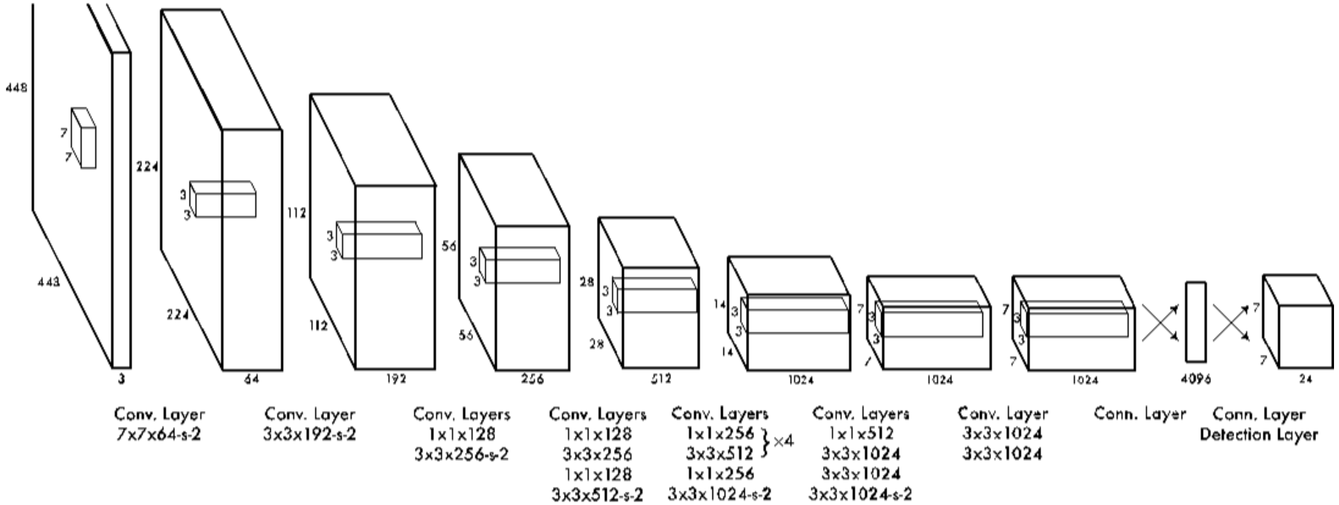
FC Densenet

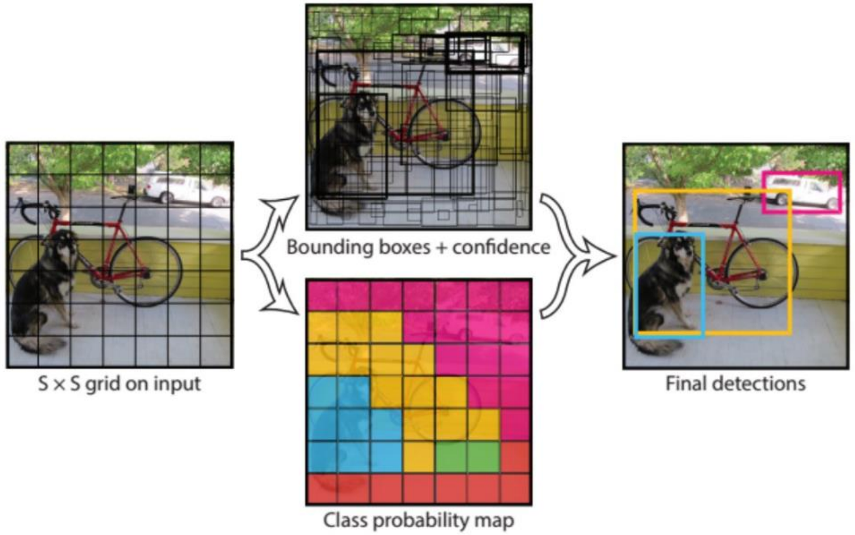


- ✓ The input to a layer is a concatenation of the outputs of all layers before it within a block.

- ✓ Allows each layer to "see" low and high level features at the same time.

Bounding Box Predictions





- ✓ Conv-net that predicts the location and size of boxes enveloping an object.
- ✓ Can represent an entire object with just 5 values: x, y, h, w, class.