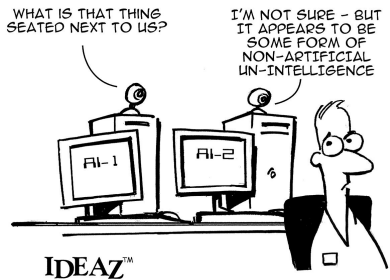
Deep Learning

Krystian Mikolajczyk & Chen Qin & Seyed Moosavi

Department of Electrical and Electronic Engineering

Imperial College London

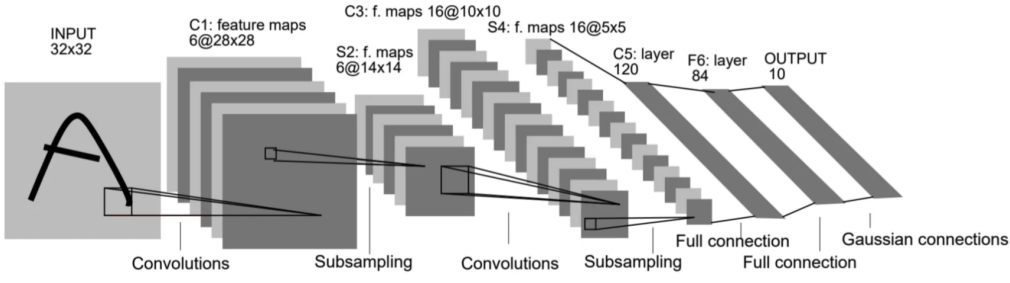


CNN Architectures

- 1998 LeNet
- 2012 AlexNet
- 2014 VGG Net
- 2014 GoogLeNet
- 2015 ResNet
- 2016 ResNeXt
- 2017 DenseNet

- Encoder-Decoder Architecture
  - Fully Convolutional Networks
  - U-Net
- Siamese Architecture
- Multi-task networks

CNN LeNet

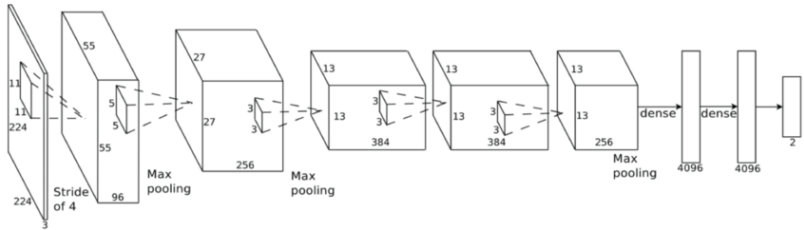


- 1989-98 Yann LeCun et al.
  - *INPUT => CONV => SIGMOID => POOL => CONV => SIGMOID => POOL => FC => SIGMOID => FC*
  - 2 convolutional layers
  - 2 pooling Layers
  - 2 FC layers
  - 60k parameters
  - 99% accuracy on MNIST dataset

CNN Data

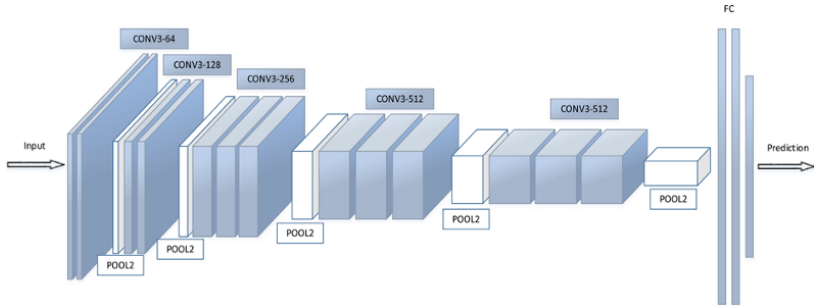
- **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**
  - Standard benchmark for comparing recognition systems
  - 1.2 million images with 1000 classes for recognition
  - Driving the progress in CNN architectures
  - Annual competition since 2010

CNN AlexNet



- 2012 AlexNet by Alex Krizhevsky et al.
  - LeNet + lots of data + GPUs
  - 5 conv and 3 FC layers, each followed by ReLU (instead of Tanh $\rightarrow$ 6 times faster)
  - Dropout of 0.5 for FC layers (instead of L2 regularisation, doubled training time)
  - 2 pooling layers to reduce network size
  - 62.3 million parameters, and 1.1 billion computation units in a forward pass
  - Conv layers (6% of all the parameters) take 95% of the total computation.
- Training with SGD with learning rate 0.01, momentum 0.9 and weight decay 0.0005.
  - $W_{t+1} = W_t + Z_{t+1}, \qquad Z_{t+1} = 0.9Z_t - 0.01(0.0005W_t - \nabla_W \mathcal{L}(W_t))$
  - Learning rate is divided by 10 when the accuracy plateaus, 3 times during the training process.
- The network takes 90 epochs in $\sim$ 100 hours to train on two GTX 580 GPUs
- Achieved the top-5 error rate of 15.3% on ImageNet

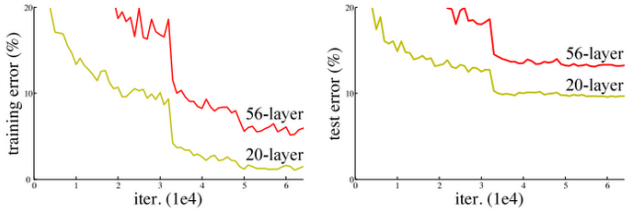
CNN VGGNet



- 2014 VGG by Karen Simonyan et al. follows simple rules:
  - Improves AlexNet by replacing the large filters in the first two layers with 3x3 filters
  - Multiple stacked smaller size kernels allows the network to learn more complex features
  - Increases the representational capacity of the ConvNet at the same number of parameters.
  - Double the number of convolutional filters (channels) every time we halve the spatial resolution of the input to prevent information bottlenecks.
  - Use Rectified Linear Units (ReLU) as an activation function to avoid the problem of saturation.
  - Convolutional layers are followed by 3 fully connected layers
  - VGG-16 has 138M parameters
  - Achieved the top-5 error rate of 7.3% on ImageNet

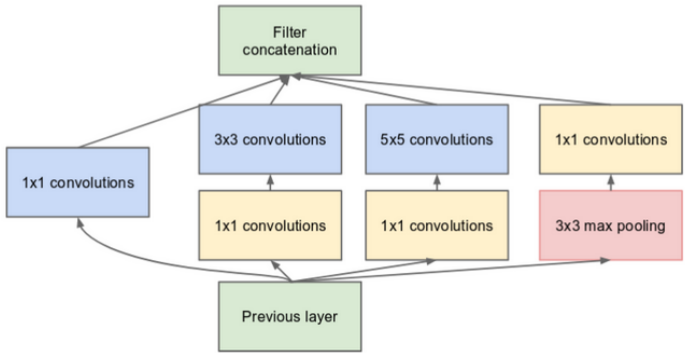
CNN Deep Networks

- Salient parts in the image can have extremely large variation in size and location.
  - Difficult to set to right kernel size, larger for global and smaller for local information
- Applying many filters lead to very deep networks
  - Vanishing and exploding gradients
  - Stacking large convolution operations is computationally expensive
  - Many activations are zero or correlated
  - State-of-the-art CNN architecture is going deeper and deeper, AlexNet with 5 convolutional layers, the VGG network (19 layers) and GoogleNet (22 layers).
  - As the network goes deeper, its performance gets saturated or even starts degrading rapidly

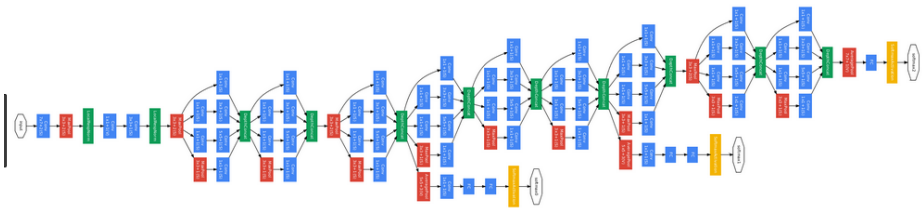


CNN GoogLeNet

- 2014 GoogLeNet by Christian Szegedy et al.
  - Inception block - filters with multiple sizes that operate on the same level to capture details at various scales
  - Wider rather than deeper
  - Max pooling followed by 1x1 convolution to reduce depth before the 3x3 and 5x5 convolutions
  - Global average pooling instead of FC



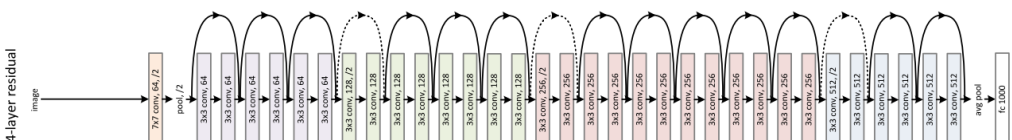
CNN GoogLeNet



- 2014 GoogLeNet by Christian Szegedy et al.
  - Auxiliary classifiers to help with vanishing gradients, i.e. as extra supervision to prevent the middle part of the network from "dying out"
  - The total loss function is a weighted sum of the auxiliary loss and the real loss.
- Version 2 factorised filters to improve speed
- It achieved a top-5 error rate of 6.67% on ImageNet, v4 achieved 5%

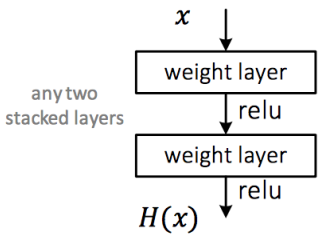
CNN ResNet

- 2015 Residual NN (ResNet) by Kaiming He et al.
  - Stacking layers should not degrade the network performance, because we could simply stack identity mappings (layer that does not do anything) in any network, and the resulting architecture would perform the same.



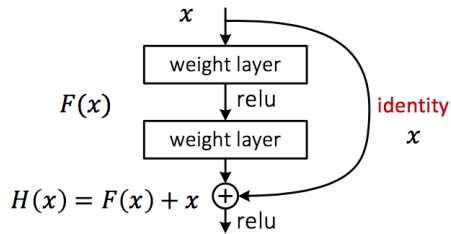
CNN ResNet

- 2015 Residual NN (ResNet) by Kaiming He et al.
  - The core idea of ResNet is introducing a so-called "identity shortcut connection" (skip connections) that skips one or more layers
- Plain net mapping $H(x)$



- 2 layers must fit $H(x)$

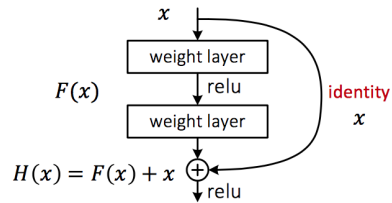
- ResNet mapping $H(x) = F(x) + x$



- 2 layers must fit $F(x)$

CNN ResNet

- 2015 Residual NN (ResNet) by Kaiming He et al.
  - Letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping.
  - At least two layers inside block to have enough nonlinearity

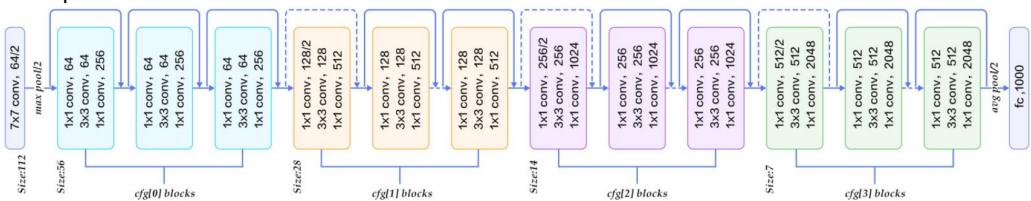


$$F(\mathbf{x}_l) = W_2\Theta(W_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2$$

$$\mathbf{x}_{l+1} = \Theta(F(\mathbf{x}_l) + \mathbf{x}_l)$$

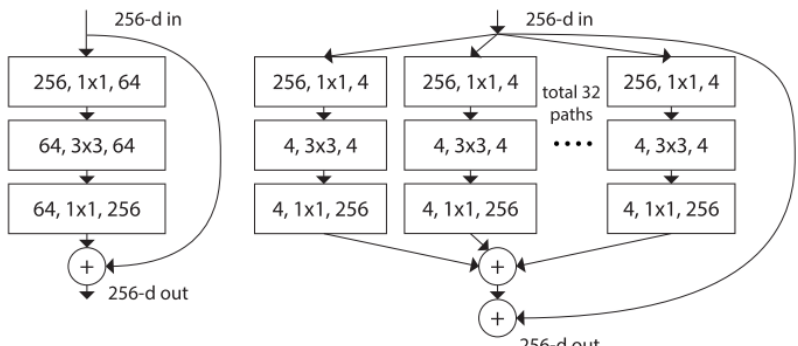
CNN ResNet

- 2016 ResNet by Kaiming He et al.
  - 18, 50, 101, 152, 200 layer variants
  - The filters are mostly $3x3$
  - Also uses a global average pooling followed by the classification layer
  - ResNet-200 achieved a top-5 error rate of 4.8% on ImageNet, which beats human performance



CNN ResNeXt

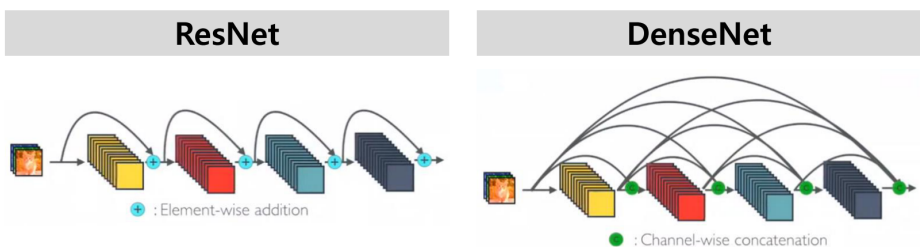
- Improvements to ResNet and GoogLeNet
  - ResNeXt - combination of Inception module and skip connections
  - ResNeXt-101 achieved a top-5 error rate of 4.4% on ImageNet



CNN DenseNet

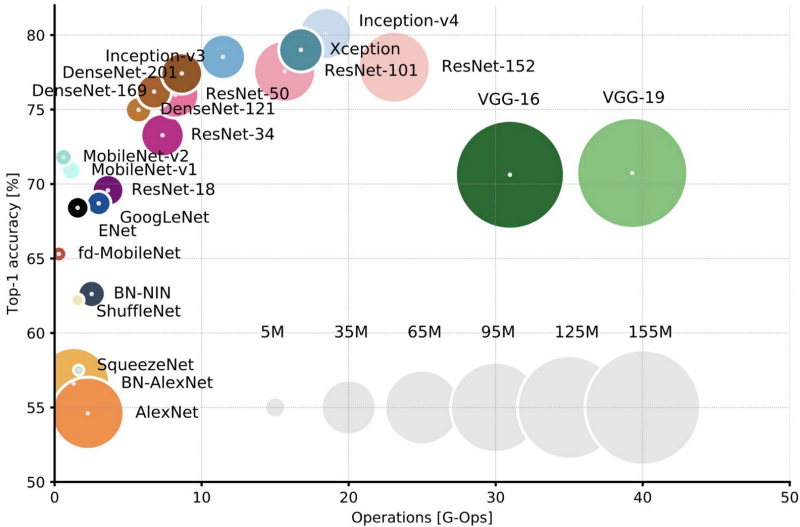
- 2017 DenseNet by Huang Gao et al.
  - ResNet: $\mathbf{x}_l = F_l(\mathbf{x}_{l-1}) + \mathbf{x}_{l-1}$; DenseNet: $\mathbf{x}_l = F_l([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}])$
  - Advantages: Strong gradient flow; Parameter and computational efficiency; Maintains low complexity features

ResNet vs. **DenseNet**



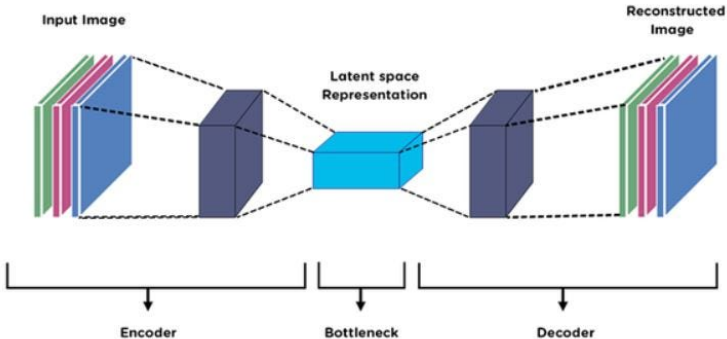
CNN Performance

- Performance comparison of various architectures



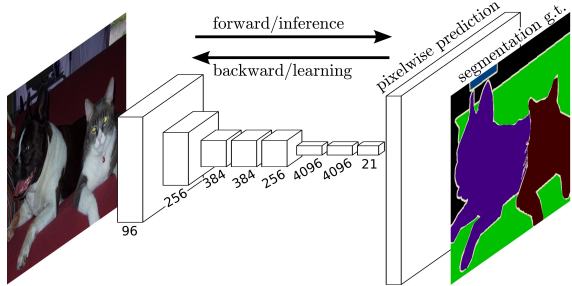
CNN Encoder-Decoder Architecture

- The encoder-decoder architectures are commonly used for dense prediction tasks, e.g., image segmentation, image enhancement etc.
- Typically, the encoder is for extracting feature maps and decoder for recovering feature map resolution.



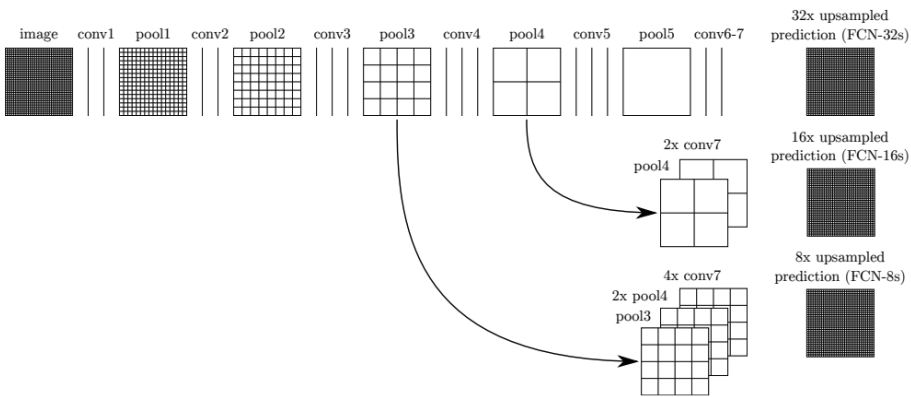
CNN Encoder-Decoder Architecture

- Fully Convolutional Networks (FCN) by Long et al. (2015)
  - build "fully convolutional" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning
  - adapt classifiers for dense prediction: fully connected layers are viewed as convolutions with kernels that cover their entire input regions



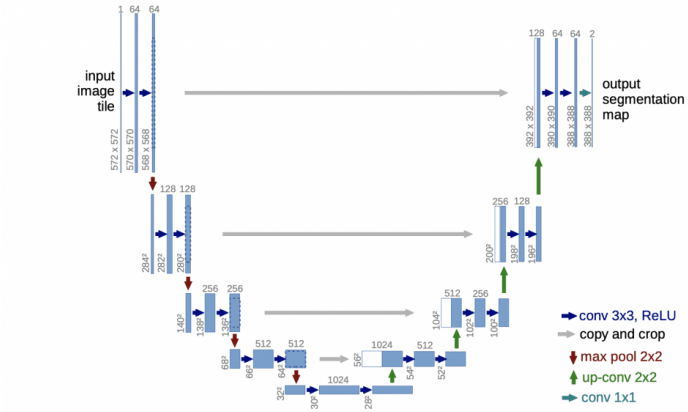
CNN Encoder-Decoder Architecture

- Fully Convolutional Networks (FCN) by Long et al. (2015)
  - Upsampling is backwards strided convolution
  - Pixel-wise loss



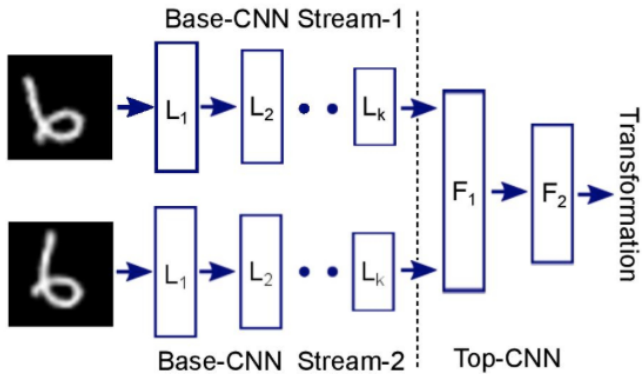
CNN Encoder-Decoder Architecture

- U-Net by Ronneberger et al. (2015)
  - U-Shaped architecture
  - Skip connections
  - Multi-scale feature maps



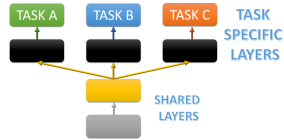
CNN Siamese networks

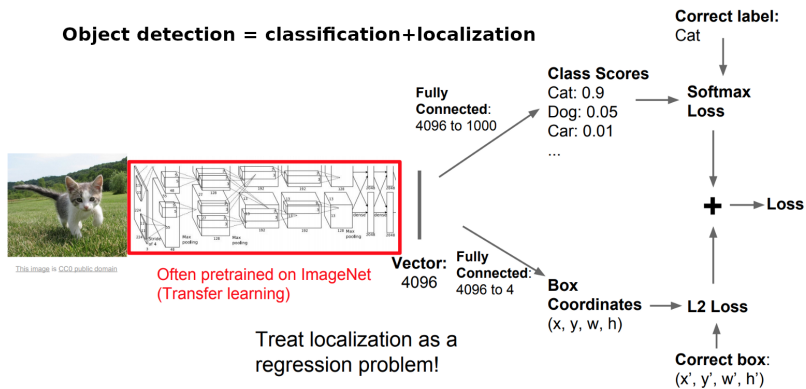
- Two networks in parallel
  - L2 loss comparing output features
- Possible merging of two streams by fully connected layers
  - Metric learning in FC layers
  - Classification loss e.g. positive or negative for similar or dissimilar pairs of input examples.



CNN Multitask learning

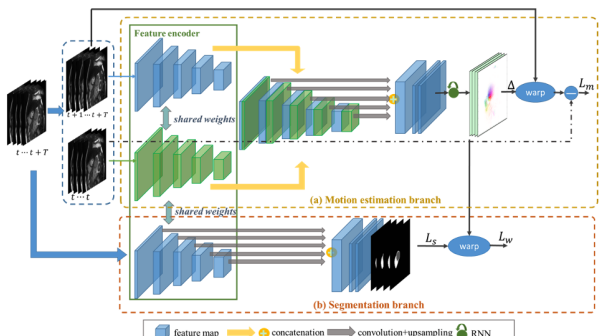
- Multiple tasks
- Multiple FC layers
- Multiple loss functions
- Task A: Classification - is an object present ? Softmax loss
- Task B: Localization(regression) - Where is the object ? L2 (or smooth L1)

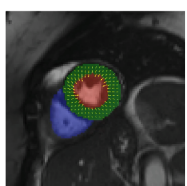




CNN Application Example

- Joint Image Segmentation and Motion Tracking in Cardiac MR Sequences (Qin et al. 2018)
  - Multitask learning: Top: motion estimation branch; Bottom: Segmentation branch
  - Siamese networks: Triplet VGG-based feature encoder; FCN-type decoder
  - Composite loss: $L = L_m + \lambda_1 L_s + \lambda_2 L_w$





CNN Architectures

- 1998 LeNet
- 2012 AlexNet
- 2014 VGG Net
- 2014 GoogLeNet (Inception Net)
- 2015 ResNet
- 2016 ResNeXt
- 2017 DenseNet

- Encoder-Decoder Architecture
  - Fully Convolutional Networks
  - U-Net
- Siamese Architecture
- Multi-task networks
- Many other types of networks
  - Autoencoders, CapsuleNet, MobileNet, ShuffleNet ...