Convolution Neural Network(CNN)

Seungyeop Hyun

Department of Statistics
Seoul National University

Feb 11st, 2022

Outline

Convolution Neural Network

CNN Architectures

Object detection

Computer Vision

- Computer vision deals with how computers can gain high-level understanding from digital images or videos. It seeks to understand the system that the human visual system can do.
- It can be used to *Object Recognition*, *Motion Tracking*, *Image Restoration*, *Image Generation*.

Computer Vision

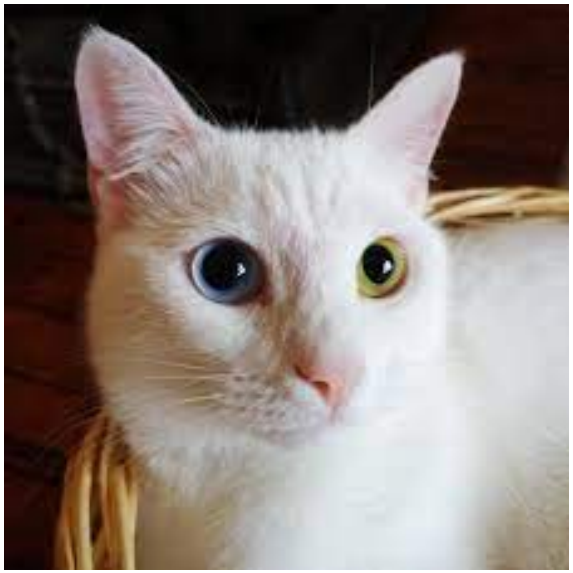
A wooden house sitting in a field.

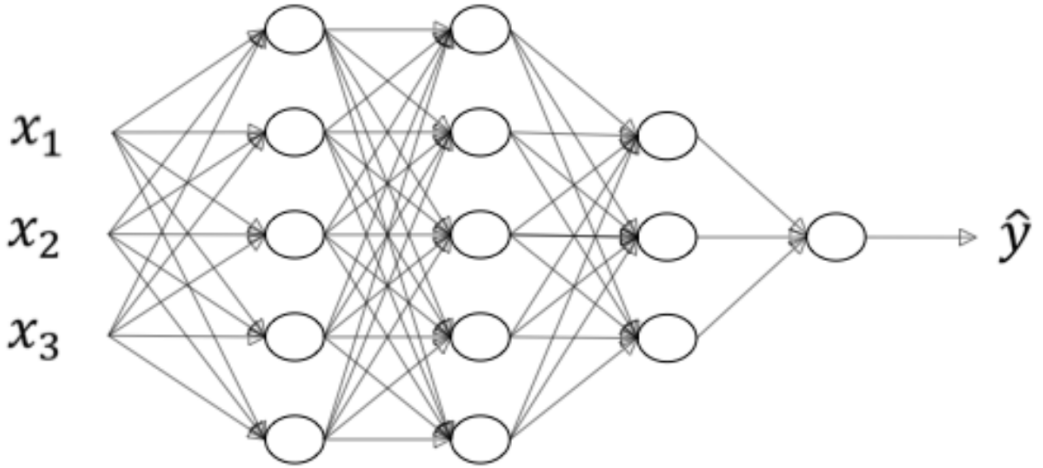
A young girl eating a very tasty looking slice of pizza.

Walnuts are being cut on a wooden cutting board.

https://github.com/microsoft/NUWA

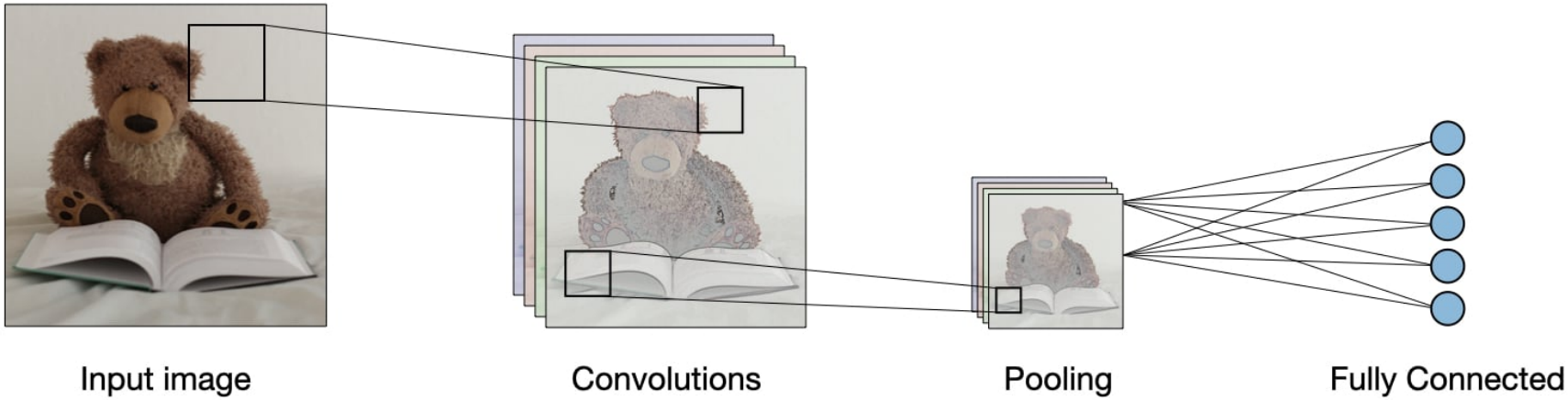
Problems of Using Standard Neural Networks.





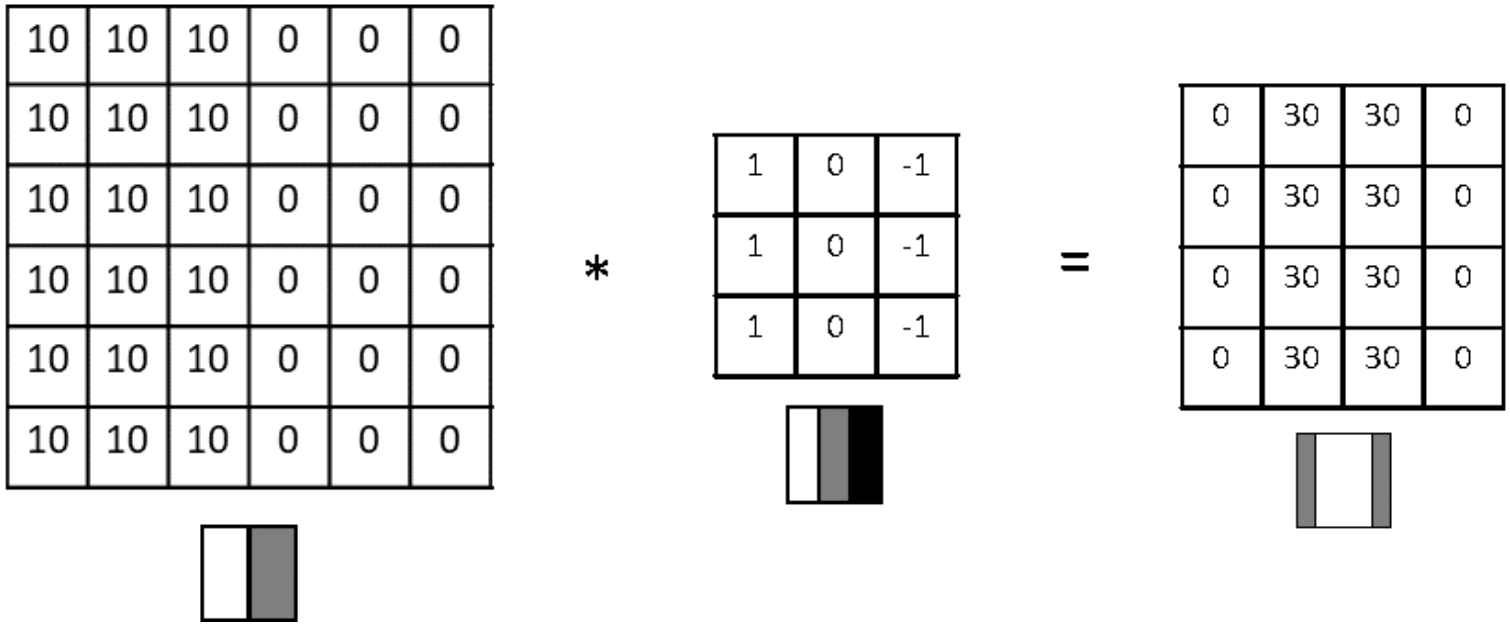
1. Image data has enormous pixels. If we use standard neural network model to image data, we have a huge amount of parameters and it is too hard to learn the model.
2. For the networks above, we have to rescale the image data to $n \times 1$. Then the data loses its spatial information.

What is CNN?



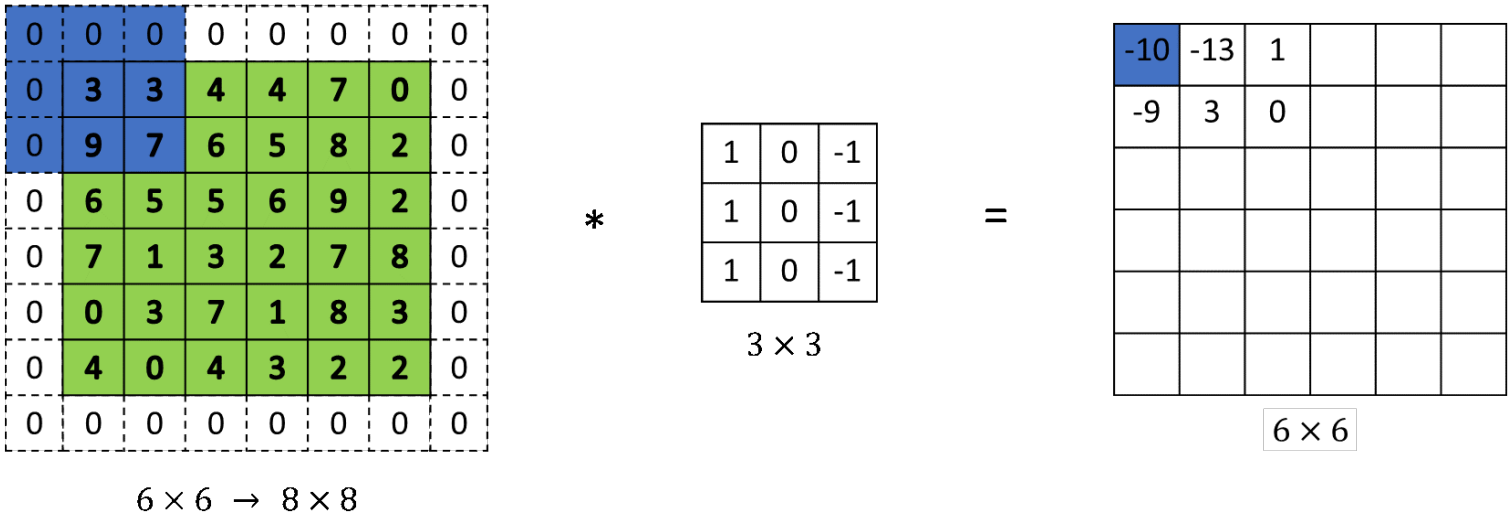
- CNN is a method that captures local spatial patterns of images.
- We use convolution operation for taking the features from image data.
- CNN architecture has 3 types of layer, **Convolution layer**, **Pooling layer**, **FC layer**.

Edge Detection



- Matrix in the middle, called filter(kernel), is used for finding vertical edges.
- If the matrix is transposed, it's the filter for horizontal edges.
- Filter slides over input data doing element-wise multiplication with the part of the input. Then we sum the results into output.

Padding



$p = 1$ padding

- Convolution operation has some problems.
  1. It reduces the dimension of data
  2. Pixels on the corners or on the edges are used less than ones in the middle.
- Using padding can keep the dimension of output same with input data.

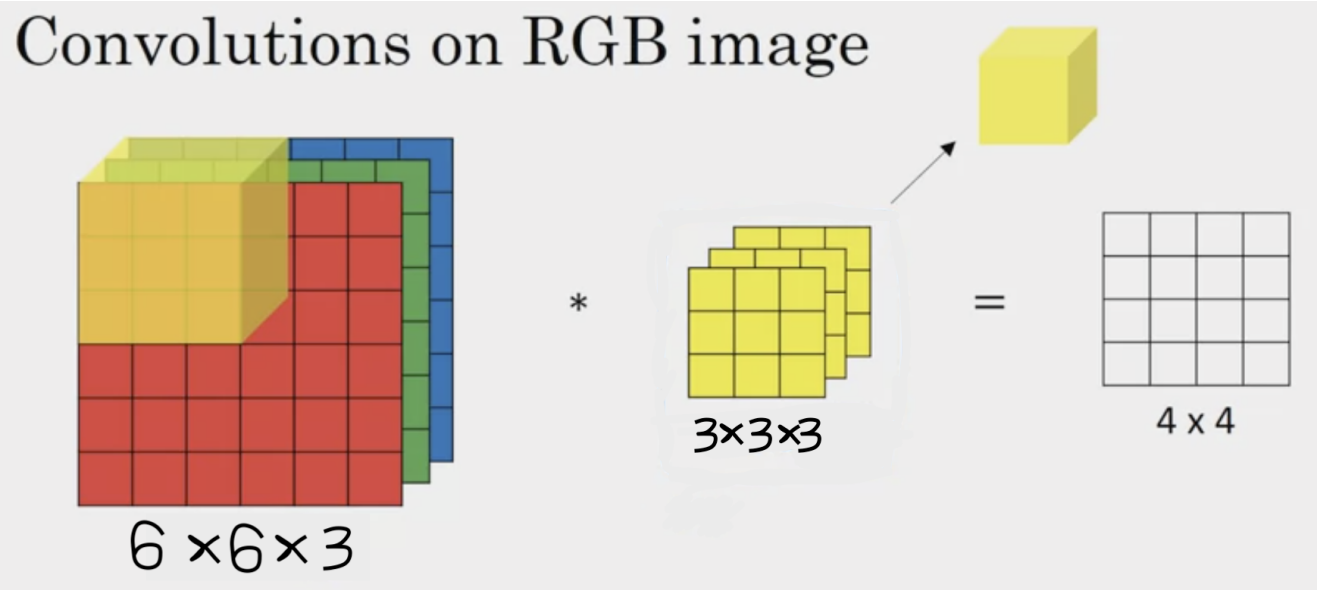
Stride

- Stride is the number of pixels shifts over the input matrix. So far, we have moved the filter pixel by pixel and it's the case when the stride is 1.
- It is another method for adjusting output size.
- With padding(p) and stride(s), the size of output is

$$\left(\left\lfloor \frac{n+2p-f}{s} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{n+2p-f}{s} \right\rfloor + 1\right)$$

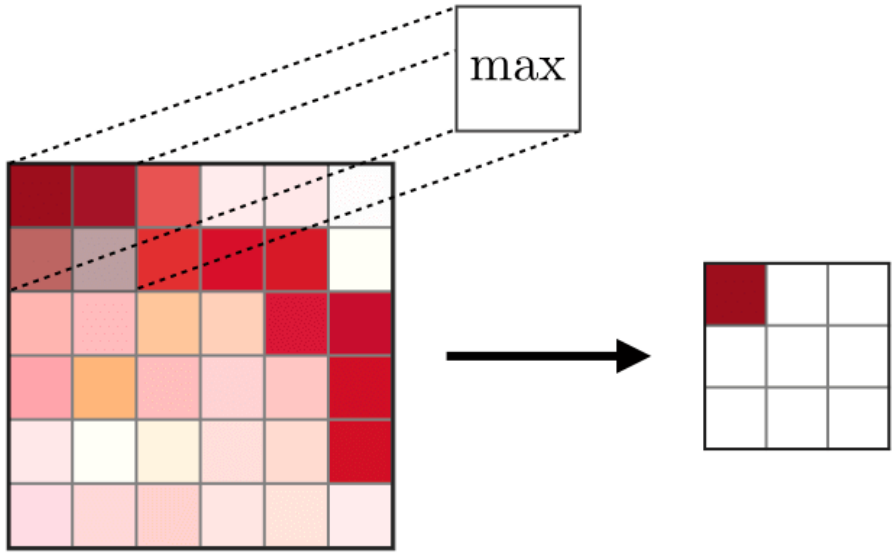
where f is the filter size.

Convolutions Over Volume



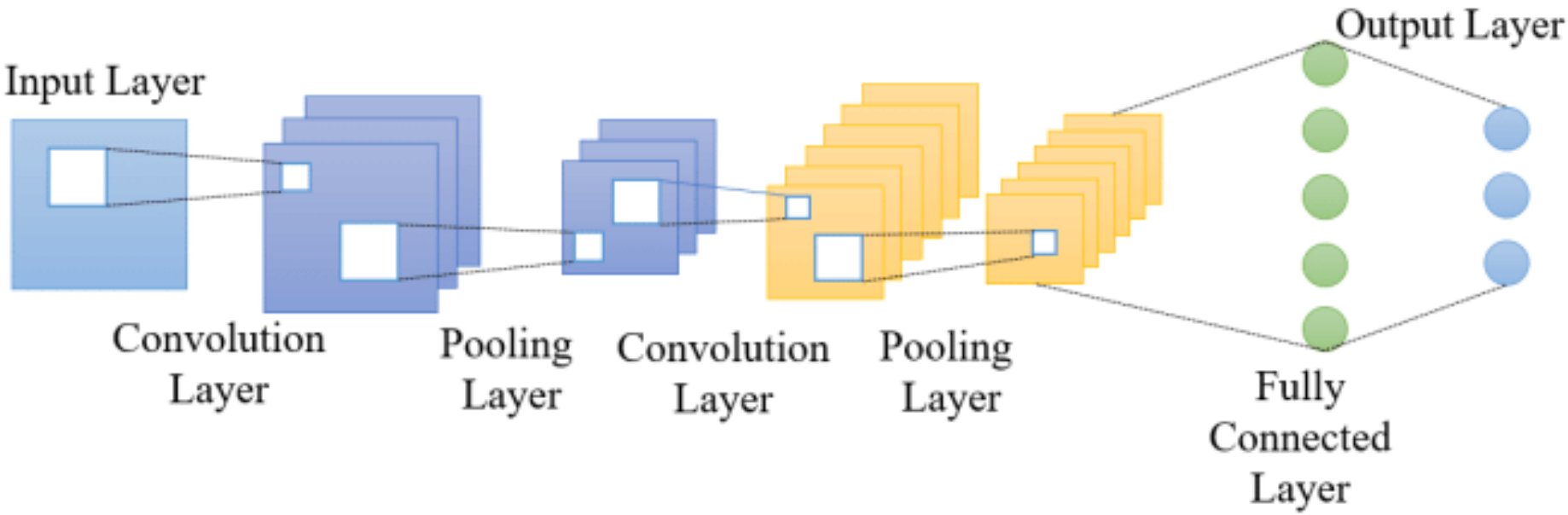
- Perform the element-wise multiplication and summation of the results in the same way as on 2D data and filters.
- The number of channel of output depends on the number of filters.

Pooling



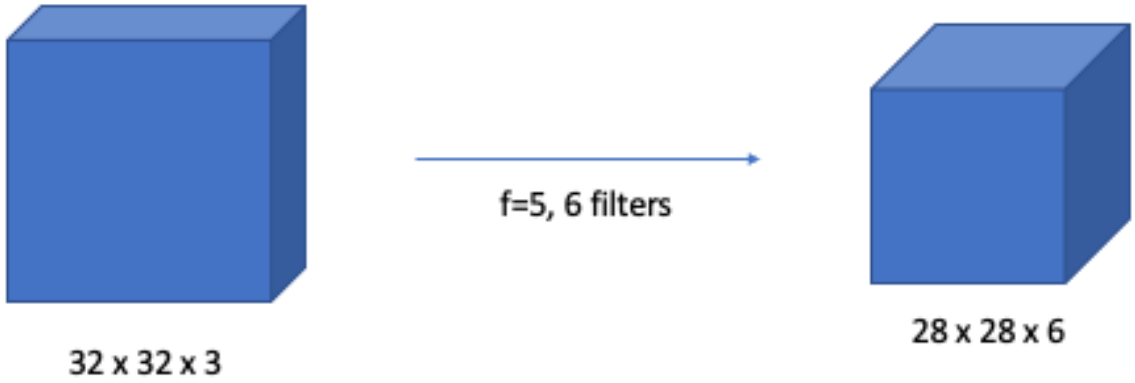
- Max pooling is commonly used pooling method.
- We can reduce the size of output by pooling layer. It summarizes the features in a region of the output generated by convolution layer.
- Pooling can cause underfitting by reducing parameters.

CNN Architecture



Basic architecture of CNN

Why Convolution?



- When using fully connected networks, we have to learn $3,072 \times 4,704 = 14,450,688$ parameters.
- However, we reduces the number of parameter to $(5 \times 5 \times 3) \times 6 = 450$ with convolution.
- How is it possible?
  - **Parameter sharing**
  - **Sparsity of connections**

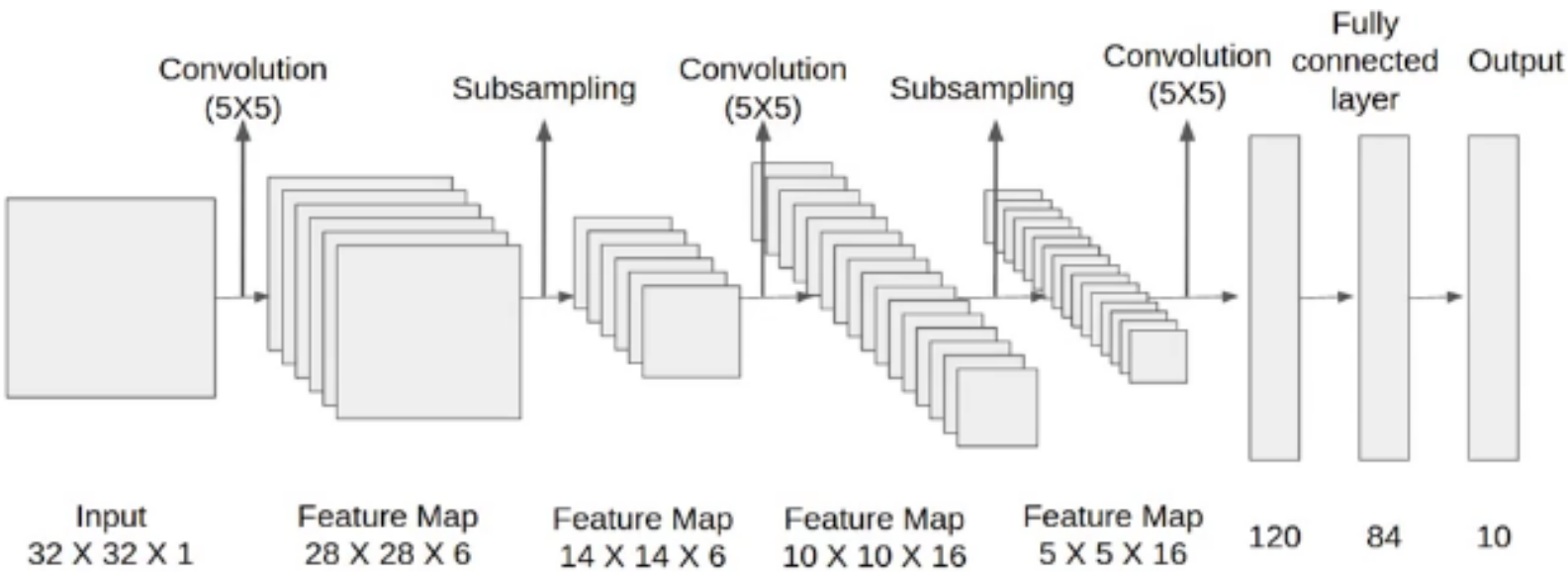
Outline

Convolution Neural Network

CNN Architectures

Object detection

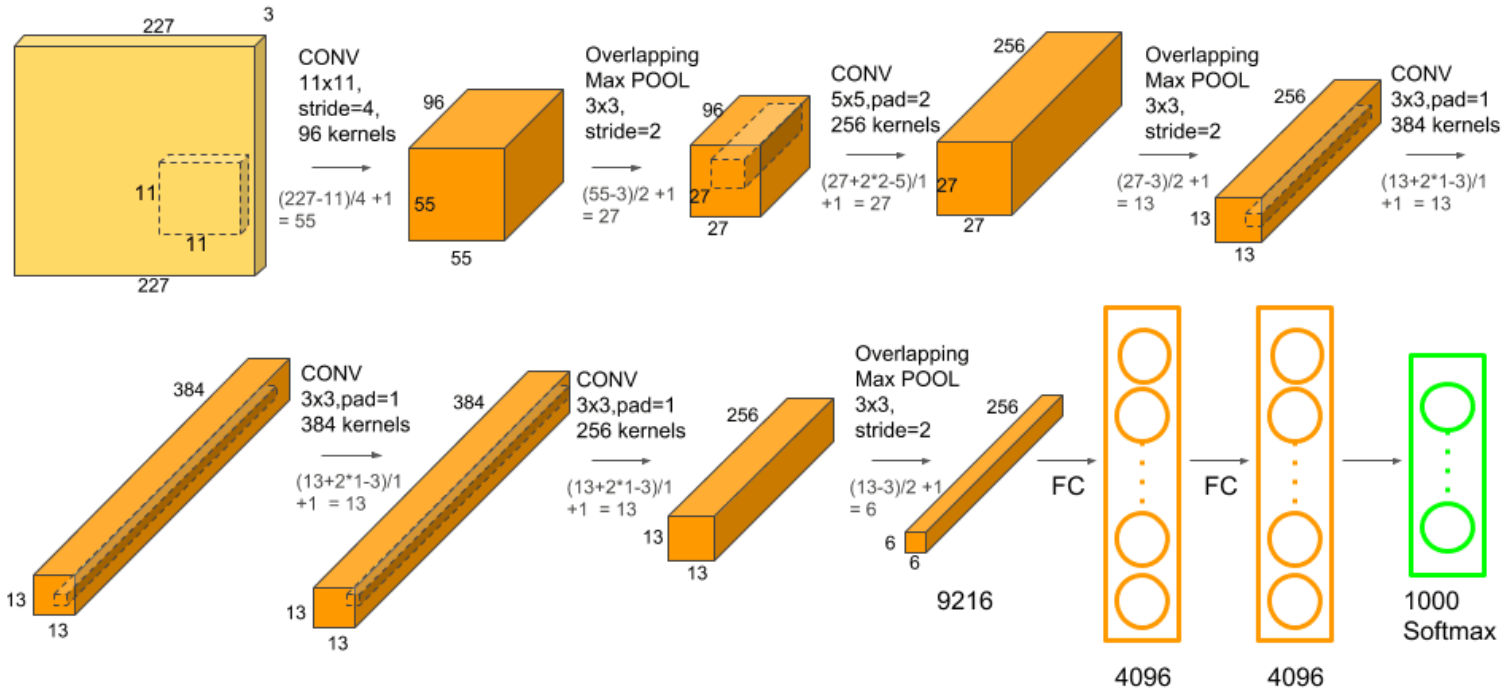
Classic Architectures - LeNet-5



LeNet-5

- LeNet-5(1998) was proposed by Yann LeCun, a founding father of CNN. There are 5 version of LeNet, 1 to 5.
- For classifying hand written number 0~9.
- 3 CONV layers, 2 FC layers with about 60K parameters.

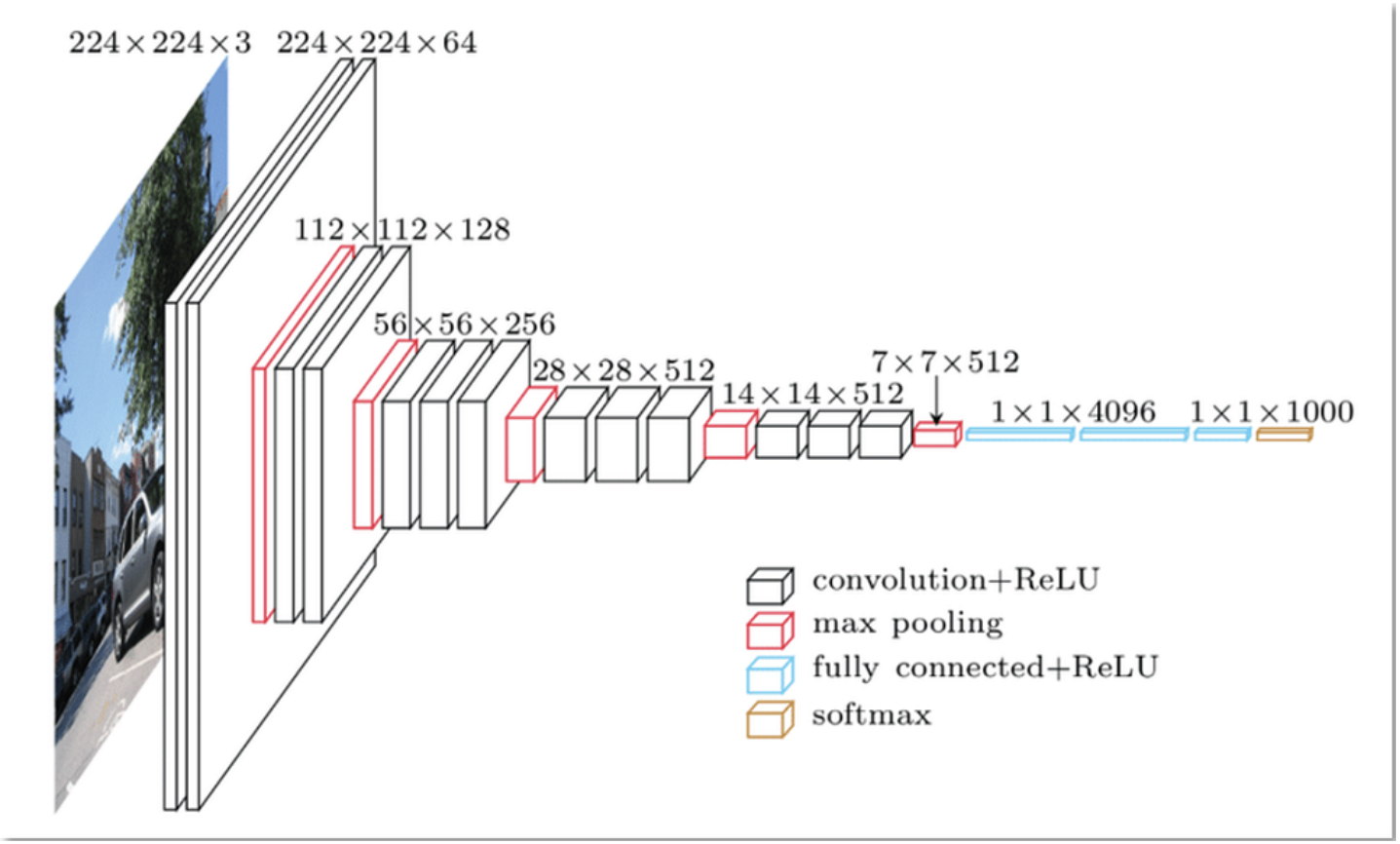
Classic Architectures - AlexNet



AlexNet

- 5 CONV layers, 2 FC layer with about 60M params
- It can classify 1000 classes.

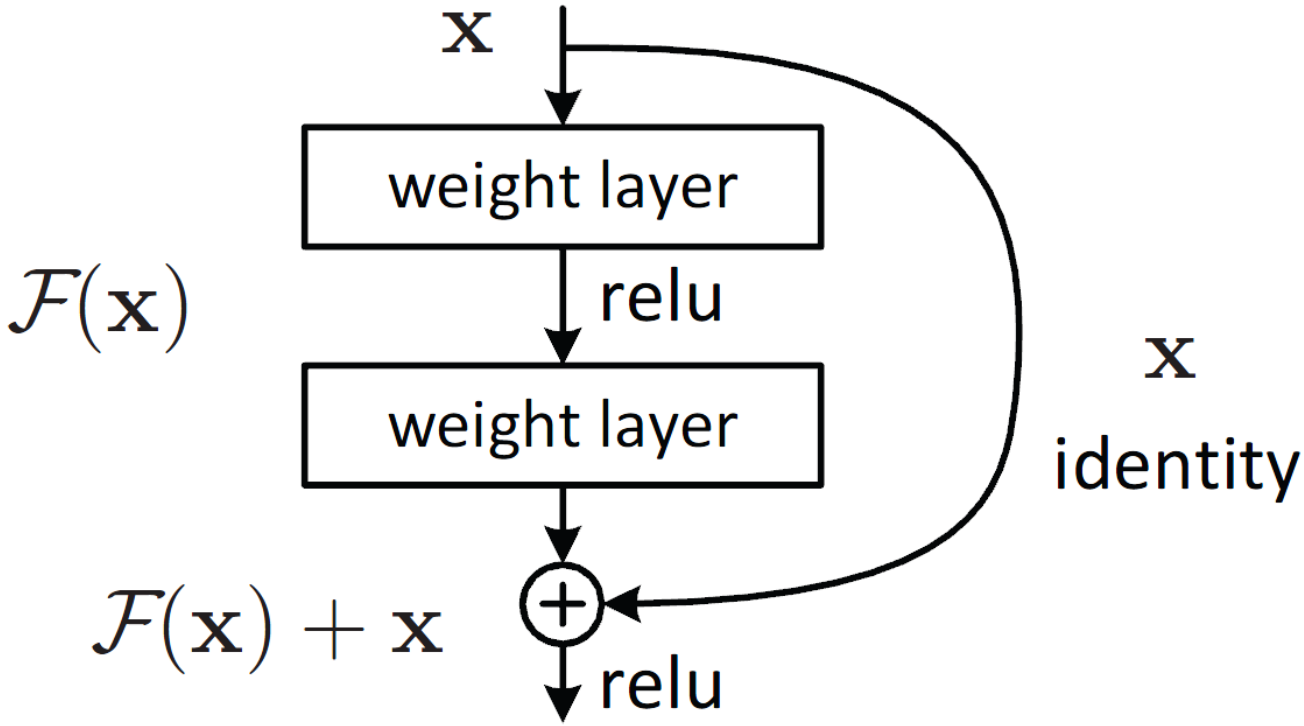
Classic Architectures - VGGNet



VGG16

- 13 CONV layers, 3 FC layers. Total 16 layers with about 140M params

Residual Block (Skip Connection)



- Operation

$$y^{[l]} = a^{[l]} + \mathcal{F}(w^{[l]}, a^{[l]})$$

$$a^{[l+2]} = g(y^{[l]}) \approx y^{[l]} = a^{[l]} + \mathcal{F}(w, a^{[l]})$$

Residual Block (Skip Connection)

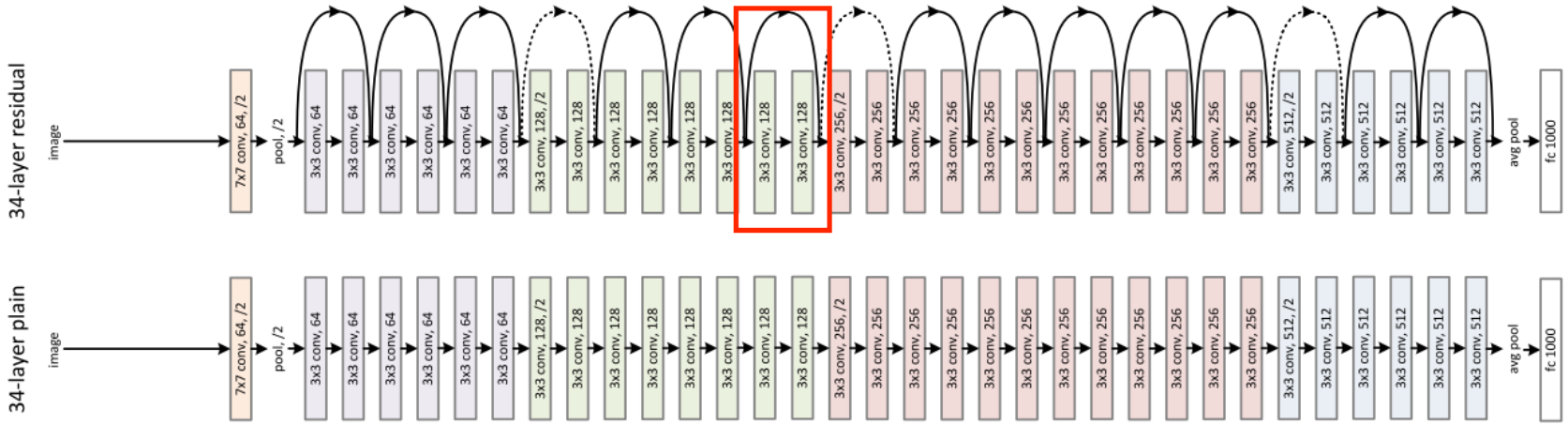
- Then the output layer can be represented by

$$a^{[L]} = a^{[l]} + \sum_{i=l} \mathcal{F}(w^{[i]}, a^{[i]})$$

  we can find that input $a^{[l]}$ is included in the last layer $a^{[L]}$

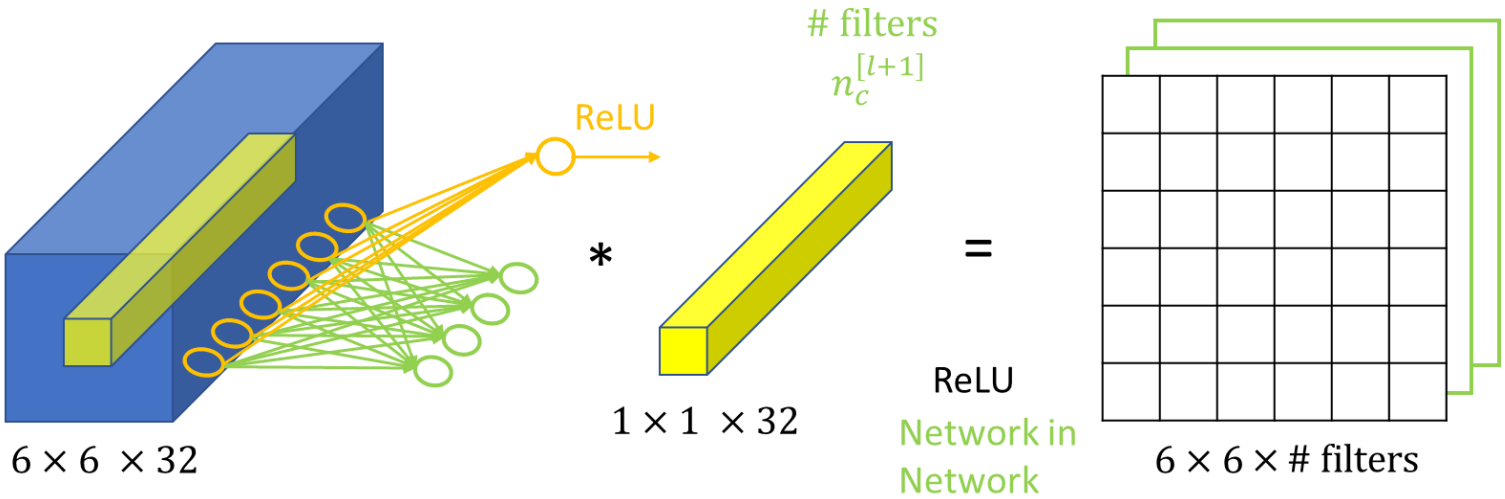
- It resolves the vanishing gradient problem, and delivers the input X to the deeper layer. In reality, it helps to propose the deeper layer model than before.

ResNets



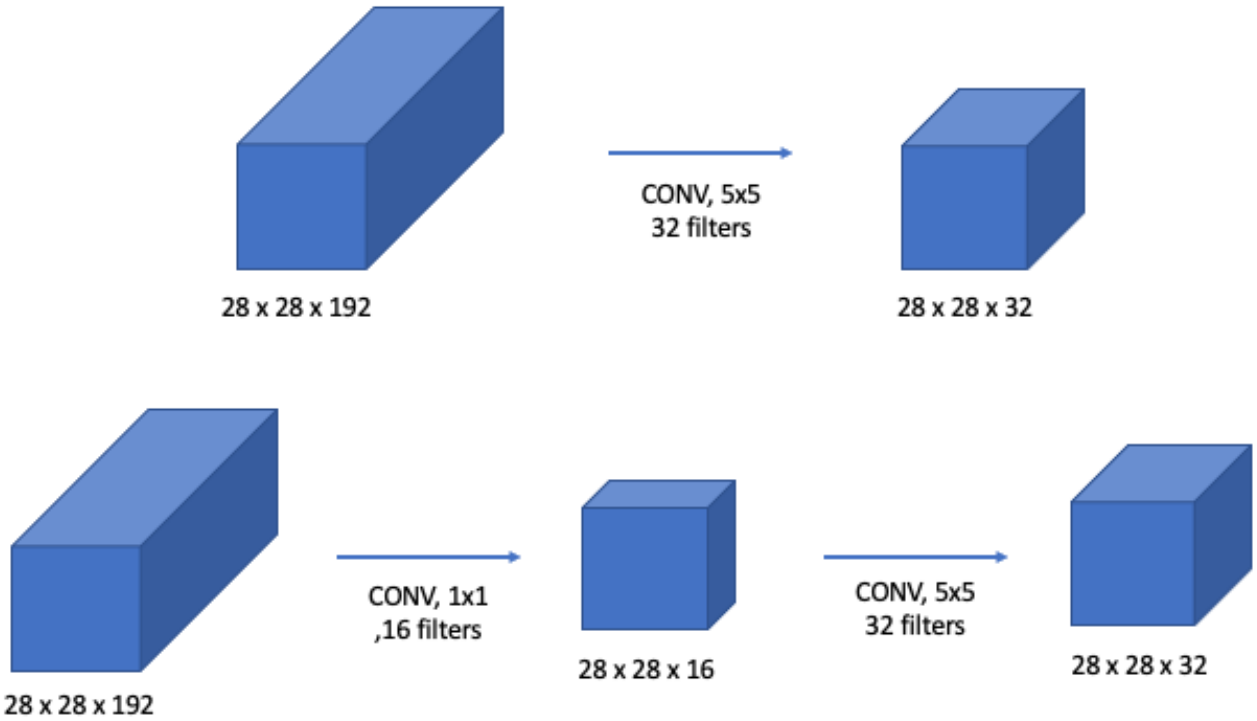
- ResNet is formed by stacking multiple residual blocks.
- It prevents the model from having the larger train error in training time.

1 x 1 Convolution



- So far, convolution operation focused on height and width of a specific channel. 1 x 1 convolution maps an input pixel with all it's channels to output pixel, not looking at anything around itself.
- 1x1 convolution is very useful for reducing the channel of output data.

1 x 1 Convolution



- 5 x 5 filters only

$$28 \times 28 \times 32 \times 5 \times 5 \times 192 \approx 120,000,000 = 120 \text{ M}$$

1 x 1 Convolution

- Using 1 x 1 convolution. In the first step,

$$28 \times 28 \times 16 \times 192 \approx 2,400,000 = 2.4 \text{ M}$$

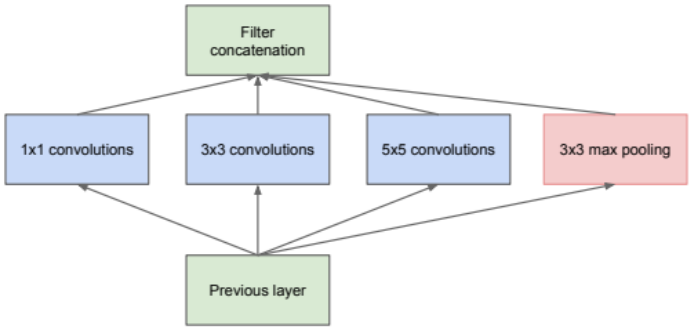
and in the second,

$$28 \times 28 \times 32 \times 5 \times 5 \times 16 \approx 10,000,000 = 10 \text{ M}$$

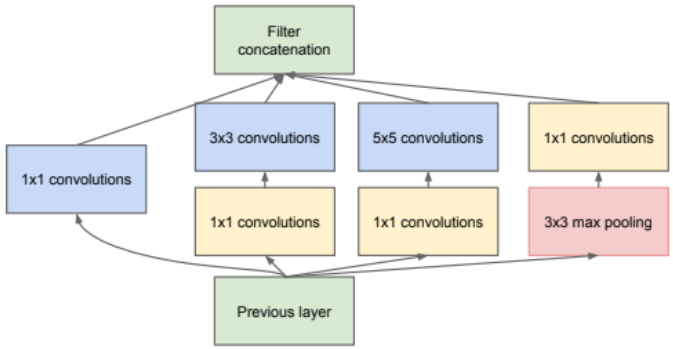
totally 12.4M computation needed.

- Case using 1 x 1 convolution reduces the amount of computation about 90%

Inception Module



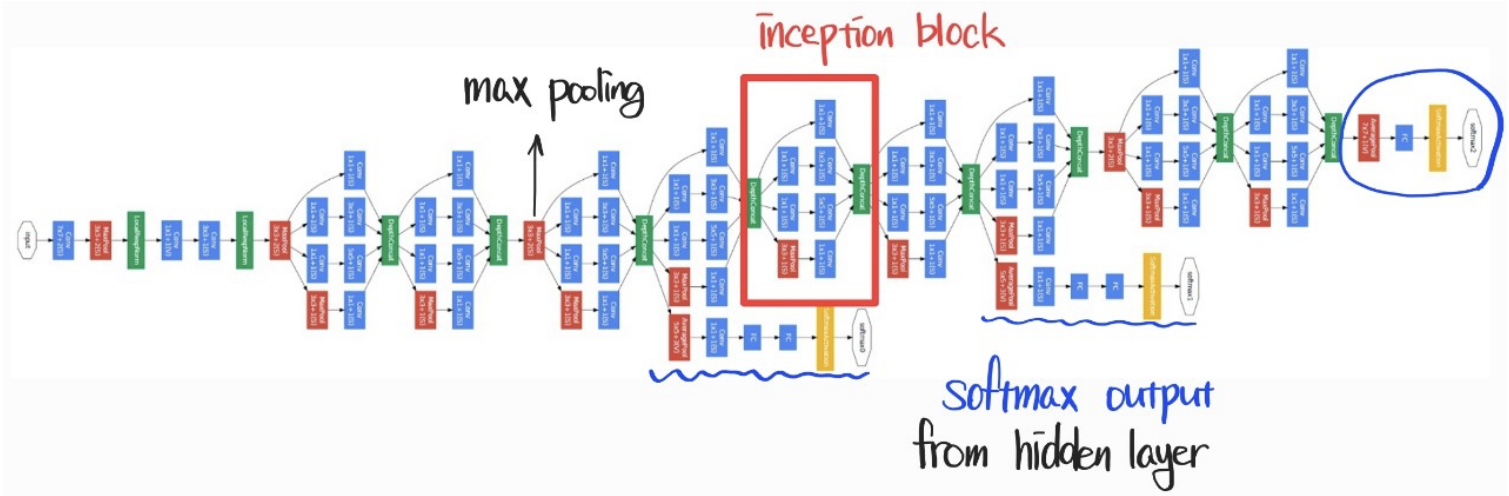
(a) Inception module, naïve version



(b) Inception module with dimension reductions

- In a single layer, use various filters in parallel. So we can catch a number of different features of images.
- 1 x 1 convolution helps to reduce the computation cost and improve the performance.

GoogLeNet (Inception Network)



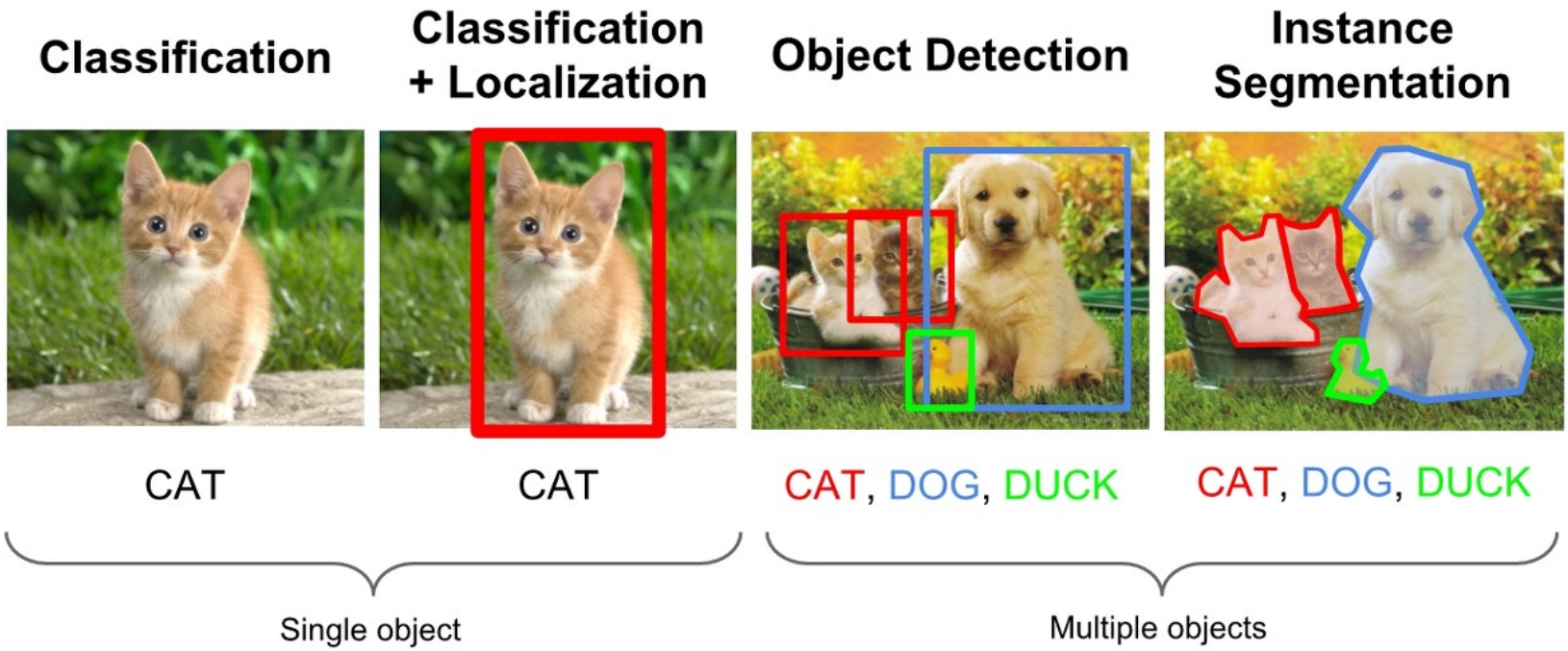
Outline

Convolution Neural Network

CNN Architectures

Object detection

Computer Vision Tasks



Fei-Fei Li & Andrej Karpathy & Justin Johnson Lecture 8 - 8 1 Feb 2016

Object localization

- Defining the target label,

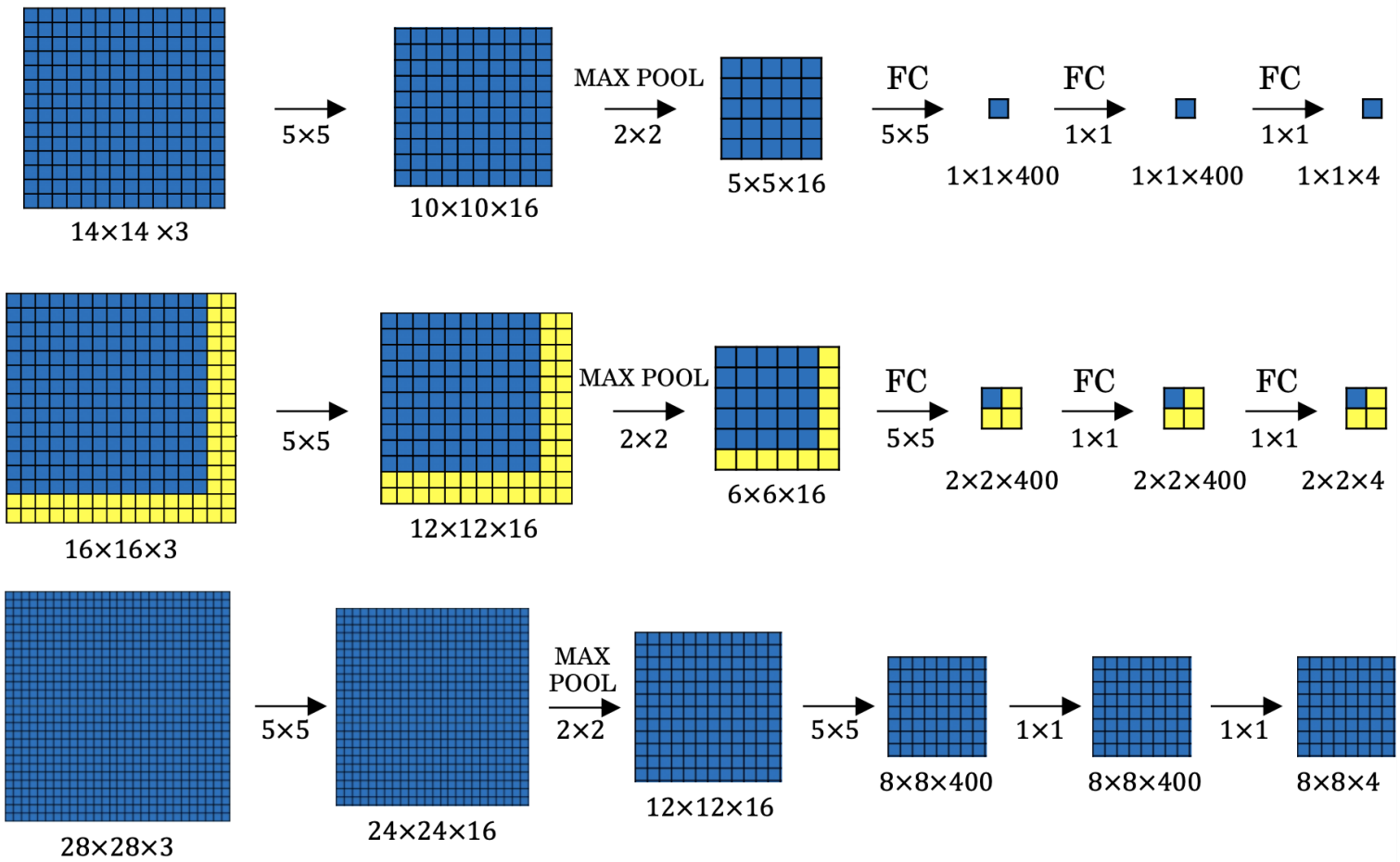
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix},$$

$$\begin{aligned} p_c &= \text{Does the obj exists? (1 or 0)} \\ b_x,\ b_y &= \text{location of bounding box} \\ b_h,\ b_w &= \text{height, width of bounding box} \\ c_i &= \text{class, all 0 means 4} \end{aligned}$$

Sliding Window

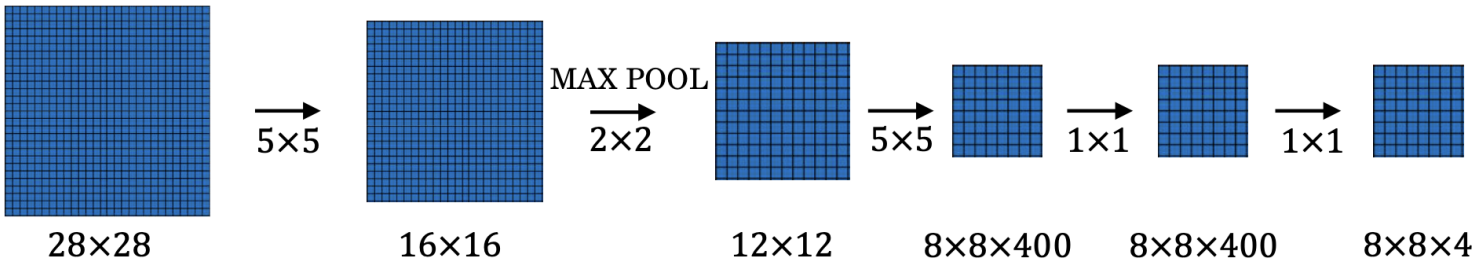
- Huge computation cost.

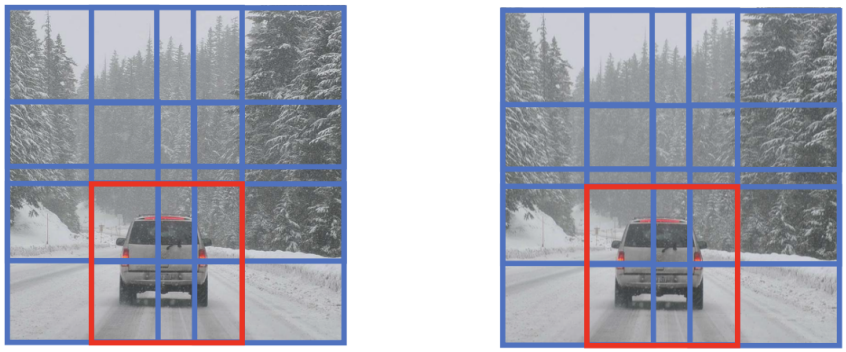
Convolutional Implementation of Sliding Windows



- Sliding window procedure is done by just one cycle of convolution network.

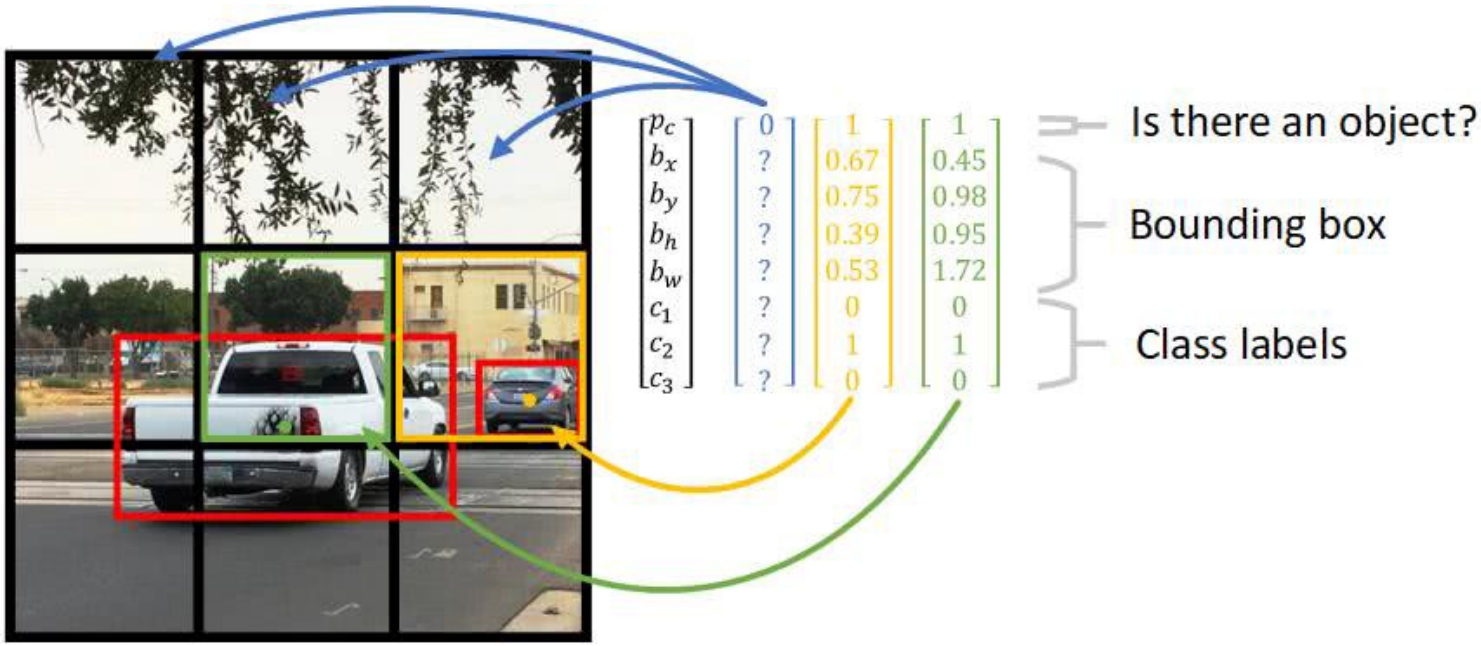
Convolutional Implementation of Sliding Windows





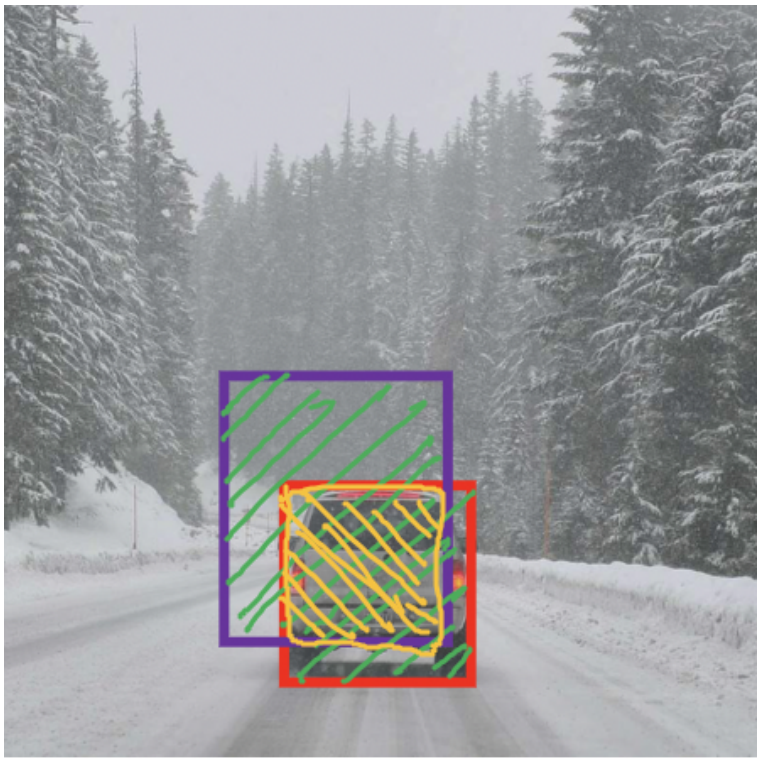
- It has a problem with having accurate bounding box.
- YOLO algorithm proposed as a solution for inaccurate bounding box.

YOLO Algorithm



- First, set a grid on the image. For example, we set the 3 x 3 grid. (19 x 19 grid usually used.)
- The labels for each grid cell are defined with the form in the above figure. We can get 9 labels in this case.

IoU: Intersection over Union

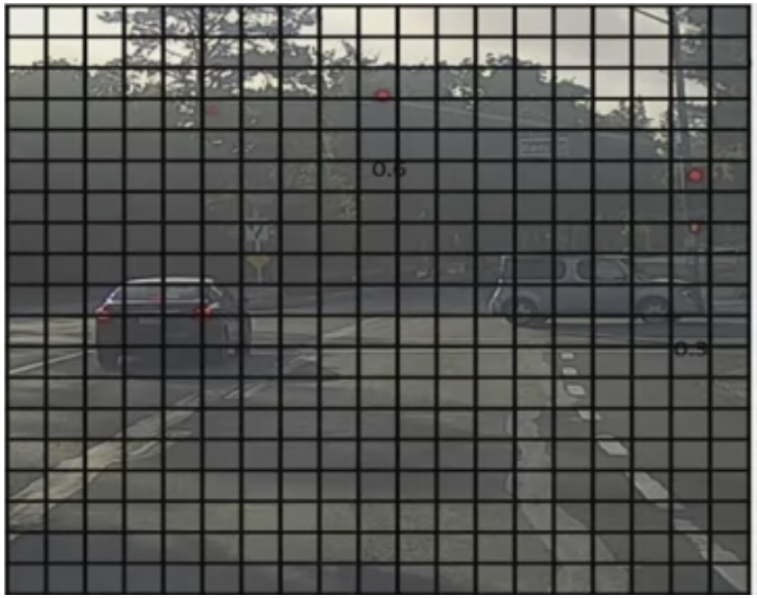


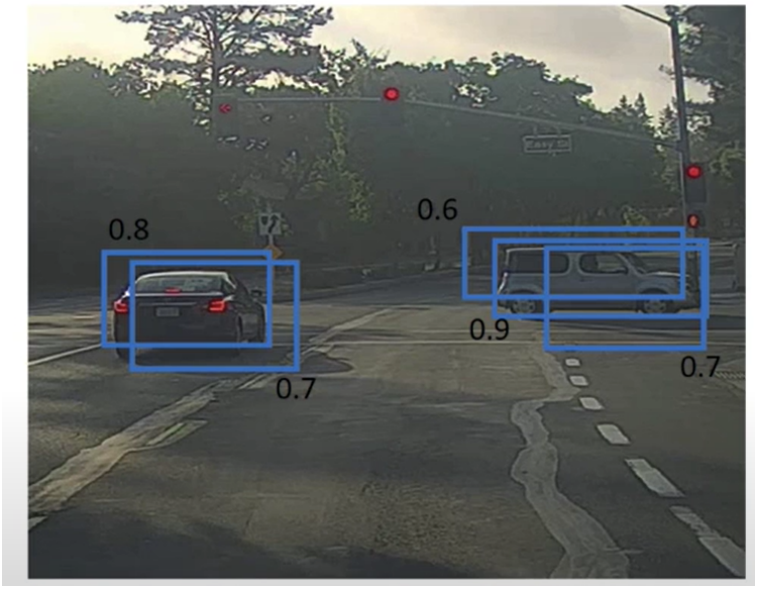
- Intersection over Union

$$\text{IoU} = \frac{\square \text{ yellow}}{\square \text{ green}}$$

- IoU is a measure of the overlap between two bounding boxes.

Non-Max Supression

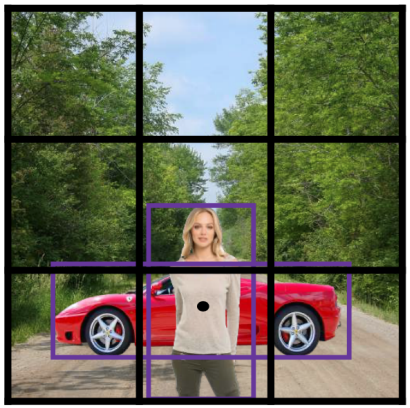


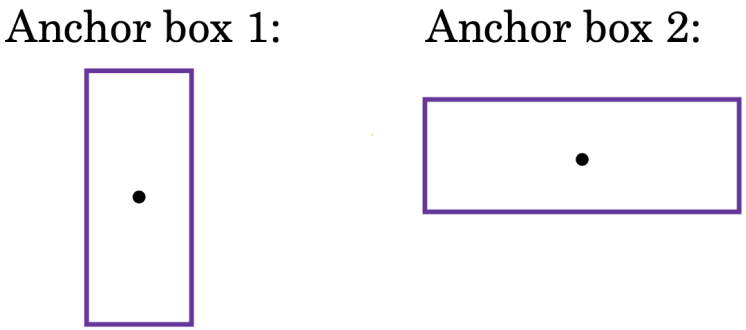


1. Discard all boxes with $p_c \leq 0.6$
2. While *there are remaining boxes* :
   - Pick the box with largest p_c
   - Discard any remaining boxes with IoU ≥ 0.5

- We have to run C times, for each class independently.
  ($C = \#$ classes)

Anchor Box





- With the fact we have learned, we can detect only one object at each grid.
- We can find multiple objects as many as the number of anchor box used.

Anchor Box

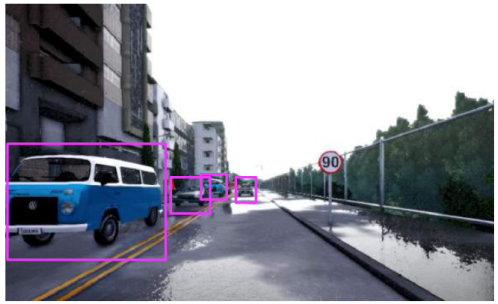
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \longrightarrow y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ \tilde{c} \\ - \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ \tilde{c} \end{bmatrix}, \text{ where } \tilde{c} = (c_1, c_2, c_3)'$$

YOLO Alogorithm

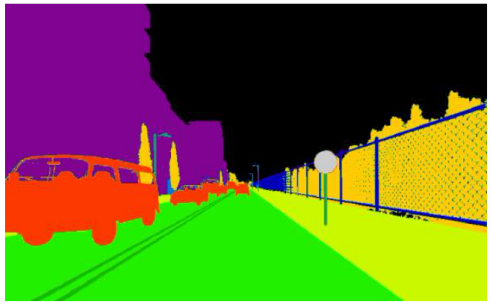
1. Train time Train the convolution network that return y.
2. Test time We get $\hat{y}$ at each grid for the input image. Also, there are the same number of bounding boxes.
3. Outputting the non-max suppression output. Finally, we can take the bounding boxes that detects what the object is, where the object is.

Semantic Segmentation

Input image

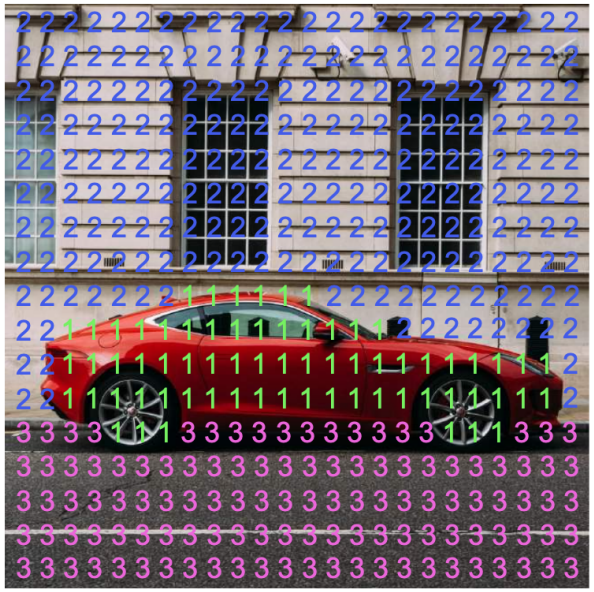


Object Detection



Semantic Segmentation

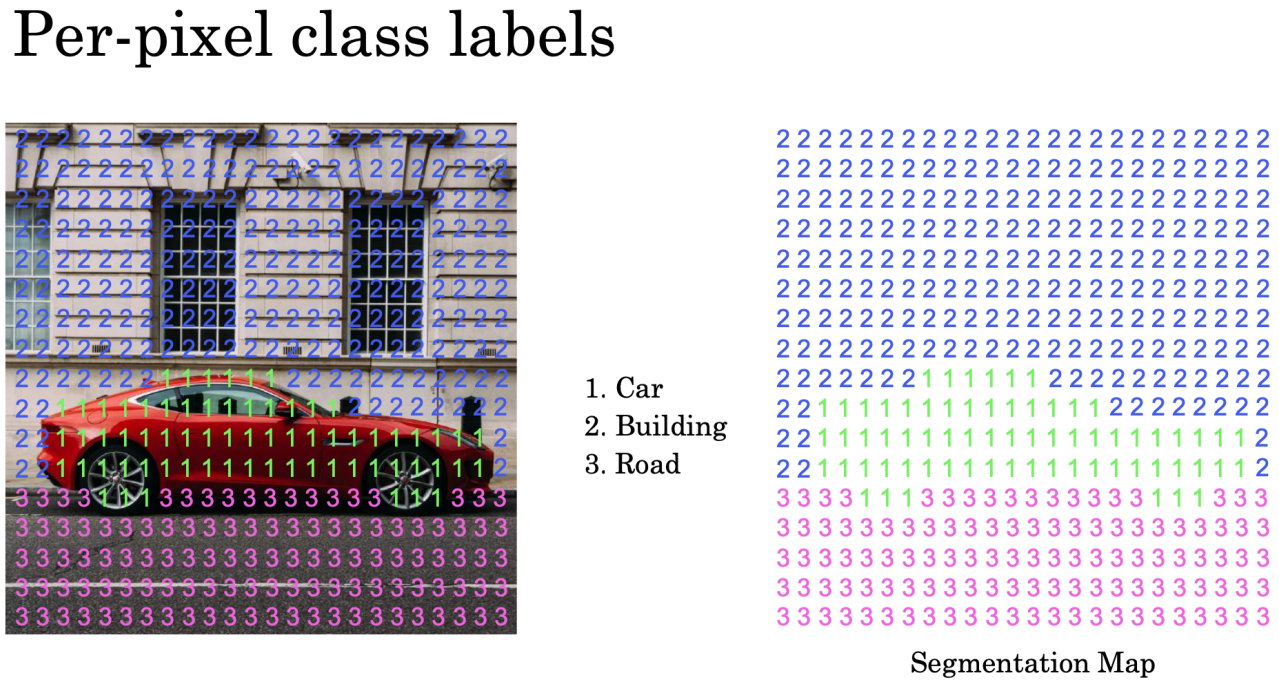
Per-pixel class labels



1. Car
2. Building
3. Road

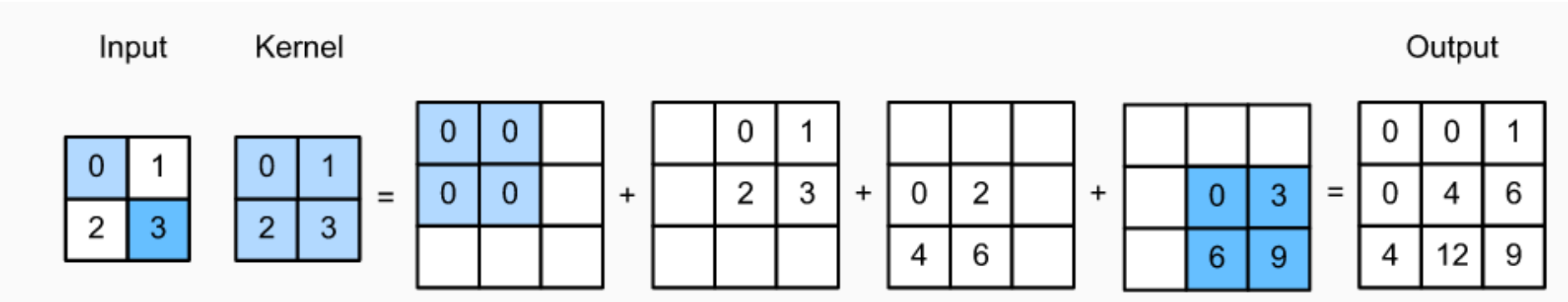
Segmentation Map

Semantic Segmentation



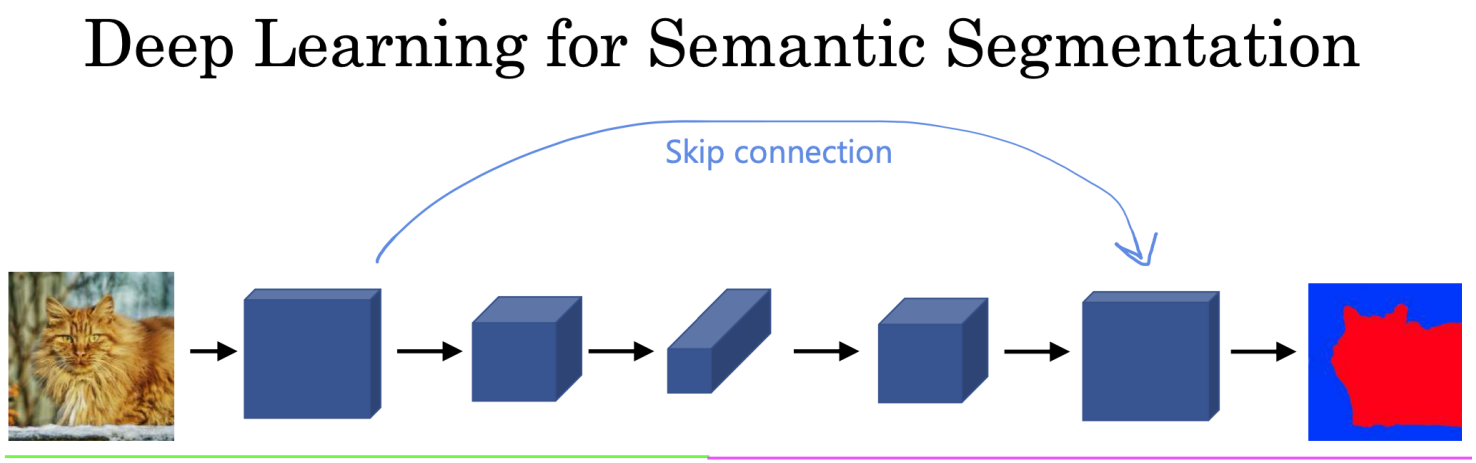
- Unlike the method such as YOLO algorithm, we want to get the output with same dimension of input image.
- The neural network we've learned tend to lower dimension output than the input. How to upsize the convolution layer?

Transpose Convolution



- For this operation, we move the filter on the expected output dimension.
- We can also use padding and stride.
- There are various ways for upsize the layer, it is known that transpose convolution return the good output.

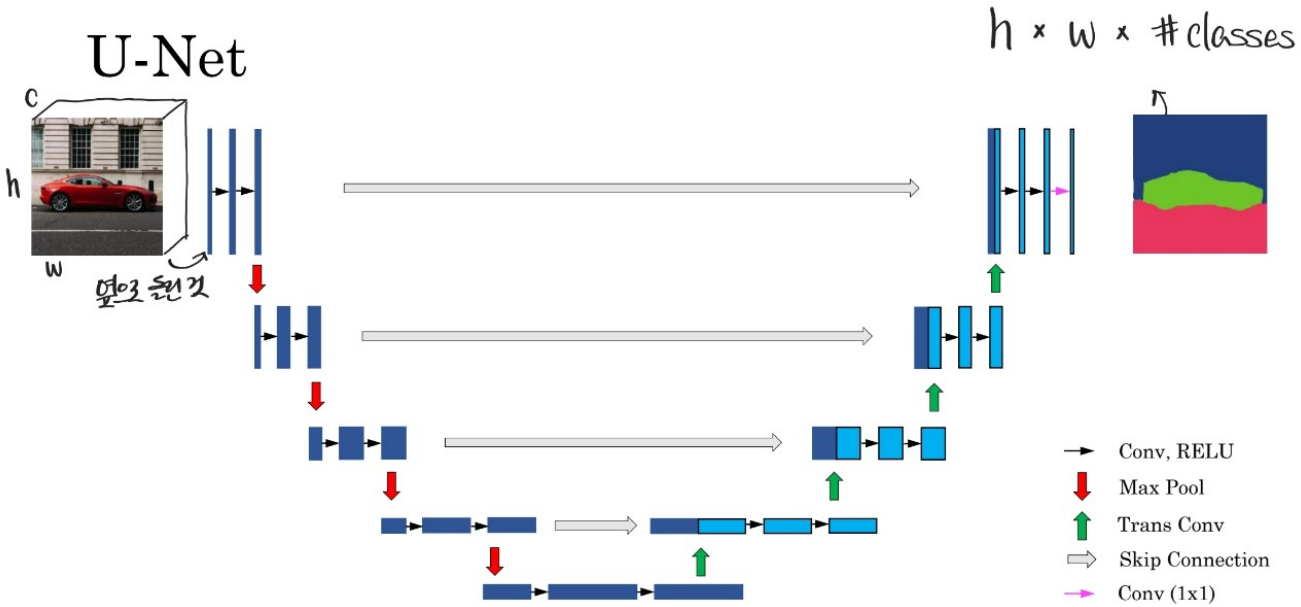
U-Net Intuition



U-Net

- Green highlighted part is the procedure of encoding the data. It loses spatial information because of reduced dimension.
- Decoding the compressed data in the purple part. We restore the data to the original size of image. Skip connection throw the data including spatial information.

U-Net Architecture



- Downsize the layer using Max pooling. Transpose convolution upsize the layer.
- Dimension of output is $h \times w \times (classes)$. For each class, the elements of $h \times w$ matrix represent the probability of how likely is a pixel belong to that class.
- Finally, apply arg max to each pixels for all channels.

Reference

1. https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks
2. https://cs230.stanford.edu
3. phil-baek.tistory.com/entry/3-GoogLeNet-Going-deeper-with-convolutions-논문-리뷰