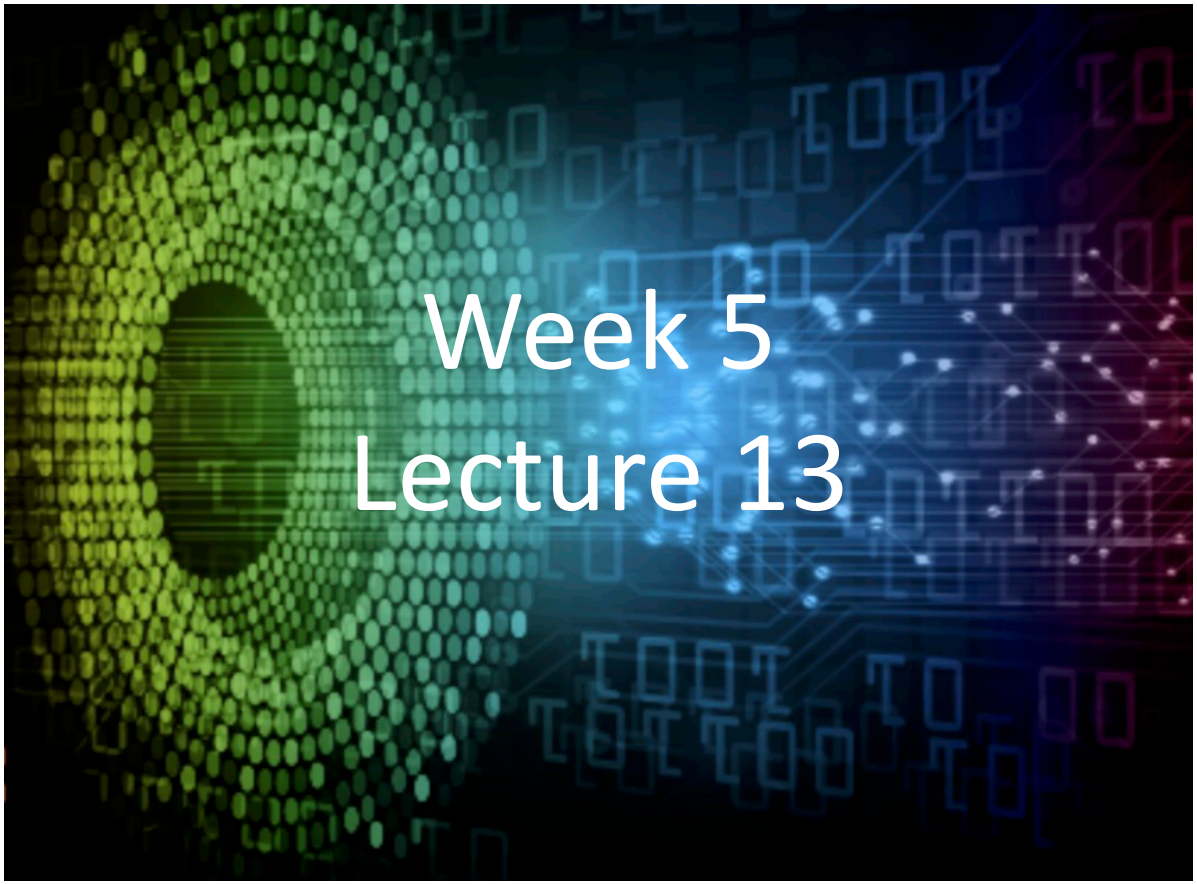
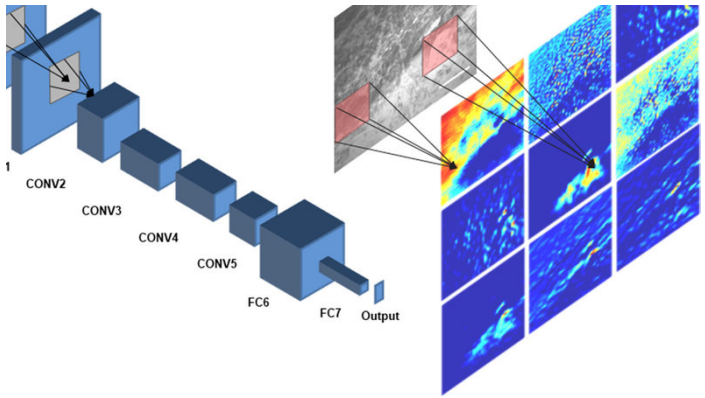
Introduction to Deep Learning
Applications and Theory



ECE 596 / AMATH 563

Previous Lecture: CNNs

- **Fundamentals of Convolutional Neural Networks**
- **Convolution and Pooling Layers**
- **Basic Networks**
  - LeNet 5
  - AlexNet 7
  - VGG 16

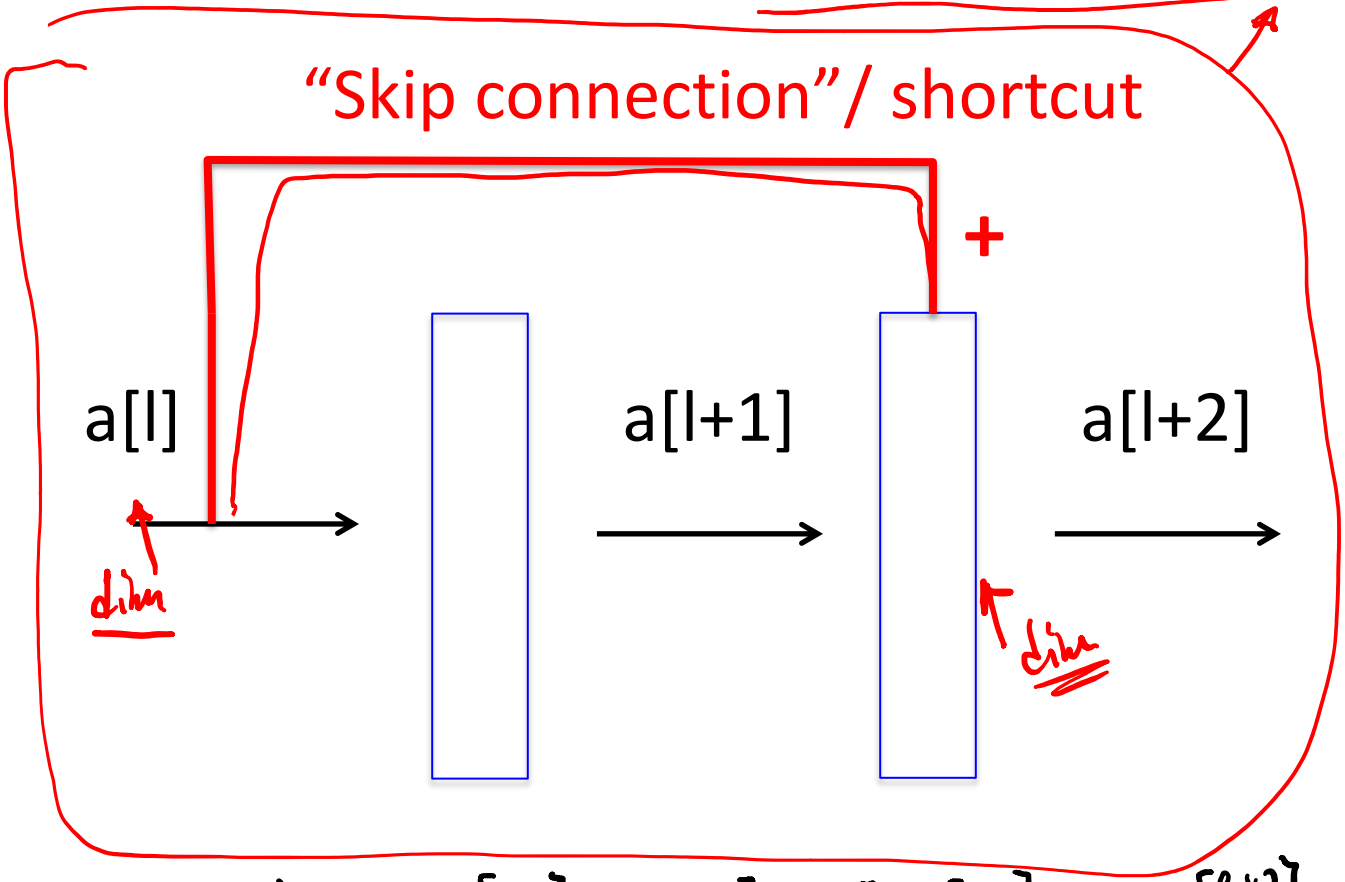


Classic Networks

- LeNet 5 –
  - Handwritten digits recognition
- AlexNet (7)
  - Object Classification
- VGG (16) / 19
  - Recognition / Classification

More Layers: Residual Block

ResNet



$$a^{[l]} = g(z^{[l]})$$

$$z^{[l+1]} = W^{[l+1]} a^{[l]} + b^{[l+1]}$$

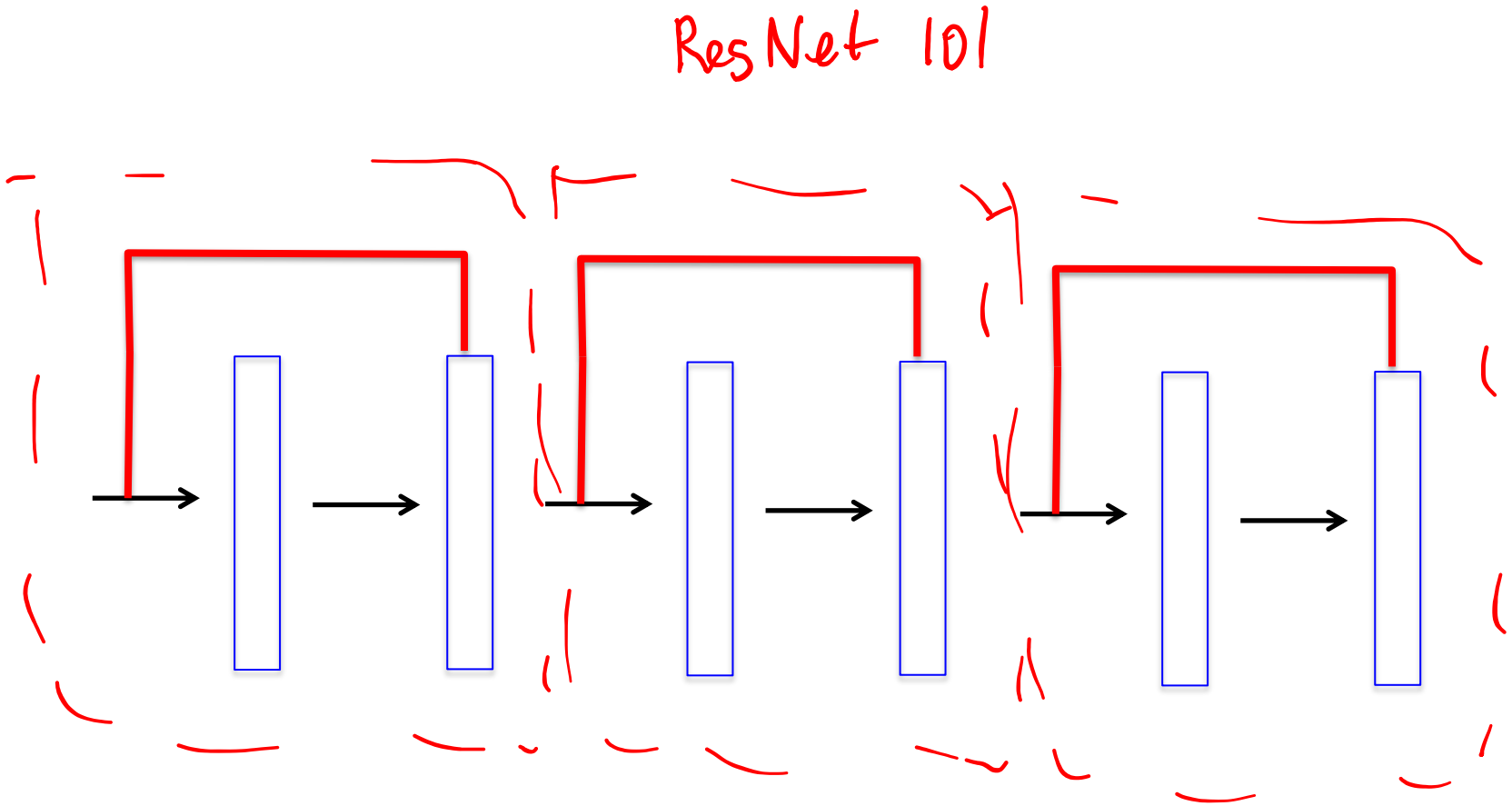
$$a^{[l+1]} = g(z^{[l+1]})$$

$$z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$$

$$a^{[l+2]} = g(z^{[l+2]})$$

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

ResNet



Kaiming He et al. "Deep residual learning for image recognition."CVPR 2016.

Motivation / Explanation

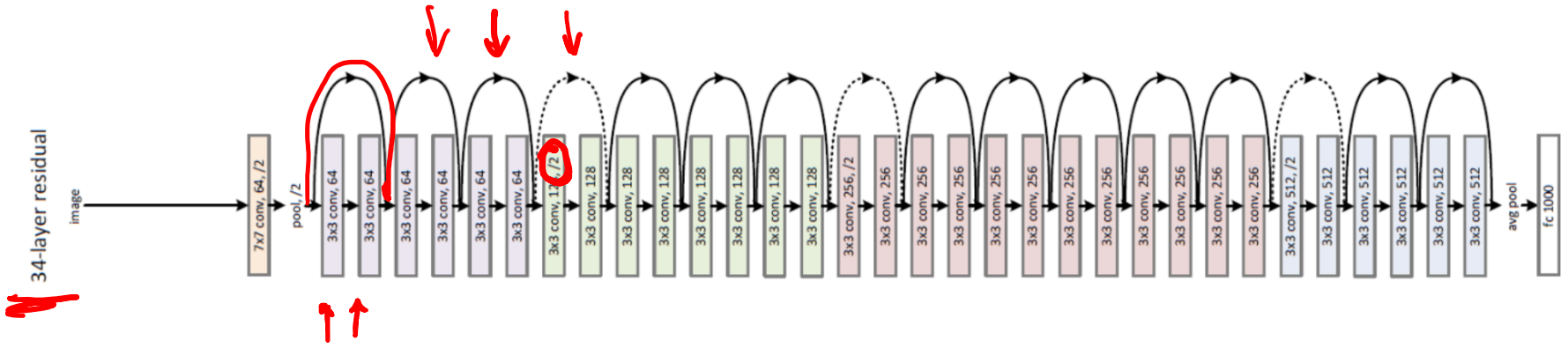
$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

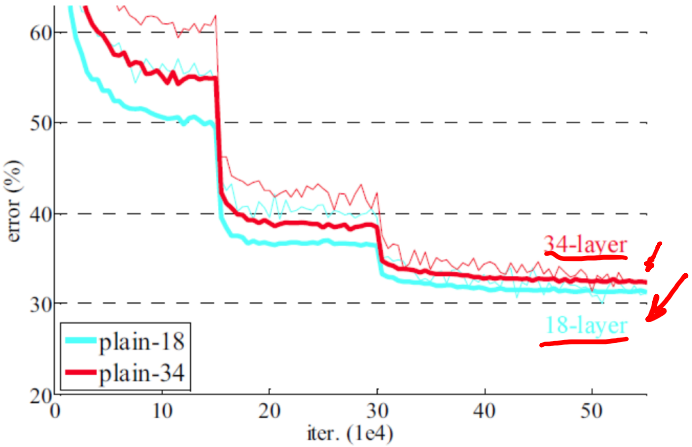
$$= g(W^{[l+2]} a^{[l+1]} + b^{[l+2]} + a^{[l]})$$

For $W^{[l+2]} \approx 0$ $b^{[l+2]} \approx 0$

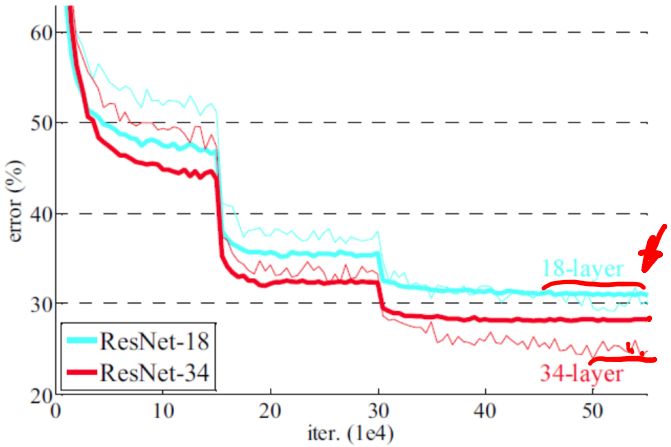
$$a^{[l+2]} = g(a^{[l]})$$

ResNet





Plain



w. skip

Alternative Explanation

$$z^{[l+2]} = W^{[l+2]} a^{[l+1]} + b^{[l+2]}$$

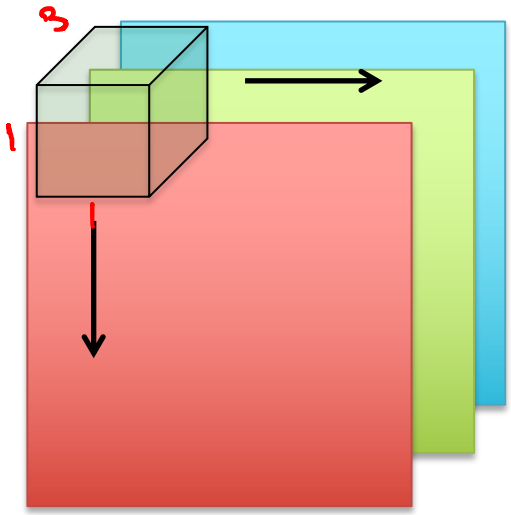
$$= W^{[l+2]} g\left(W^{[l+1]} a^{[l]} + b^{[l+1]}\right) + b^{[l+2]}$$

$$= F(a^{[l]})$$

$$a^{[l+2]} = a^{[l]} + F(a^{[l]})$$

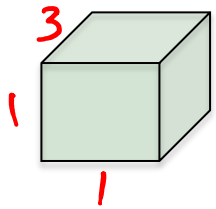
Iterative integration method!

1x1 Convolution



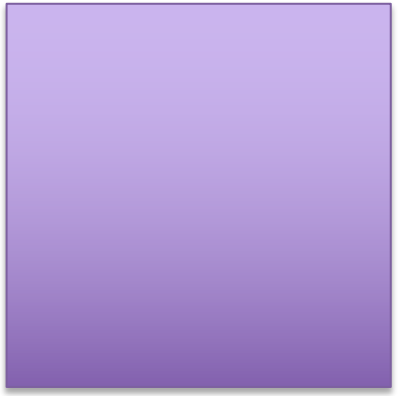
32 x 32 x 3

*



1 x 1 x 3

=



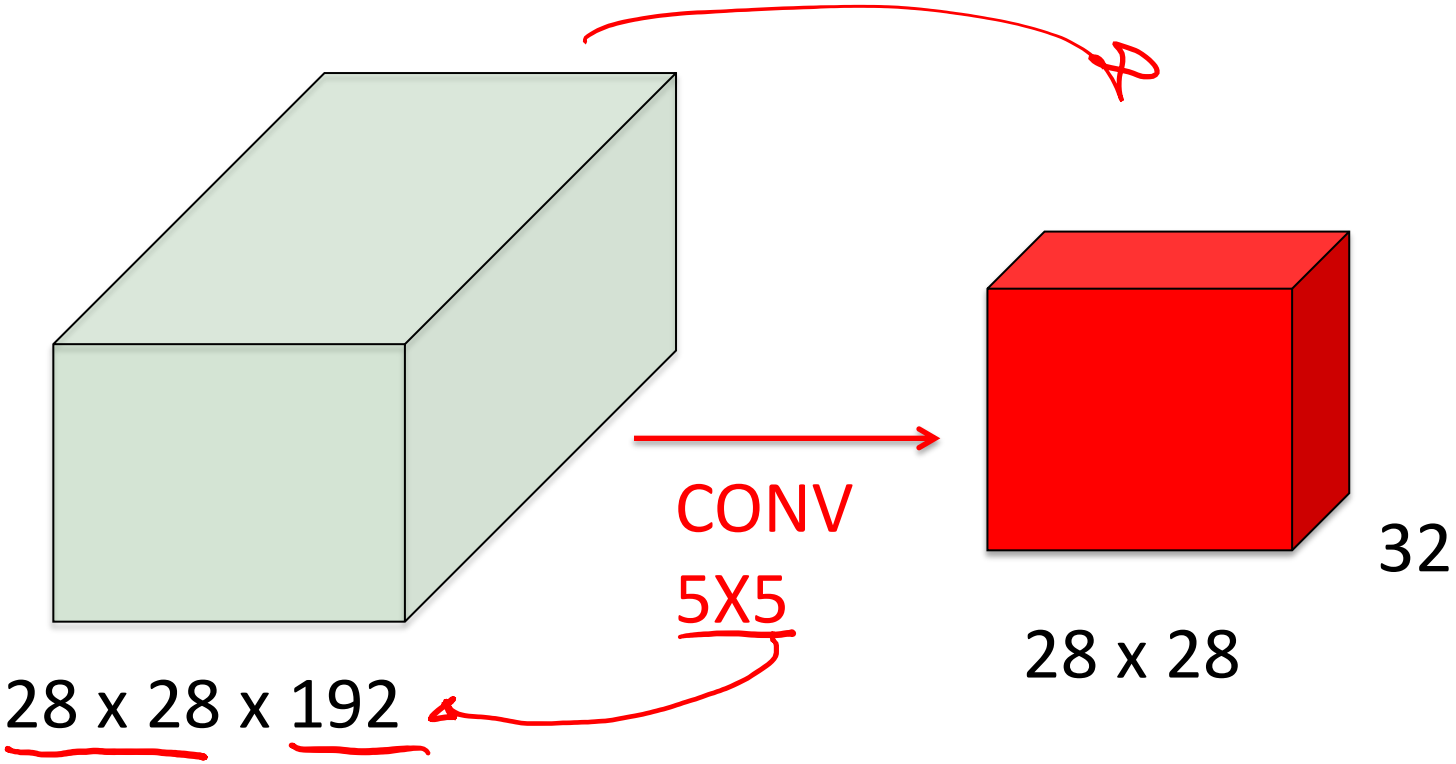
32 x 32

Readout of channels

X $n_c^{[l+1]}$

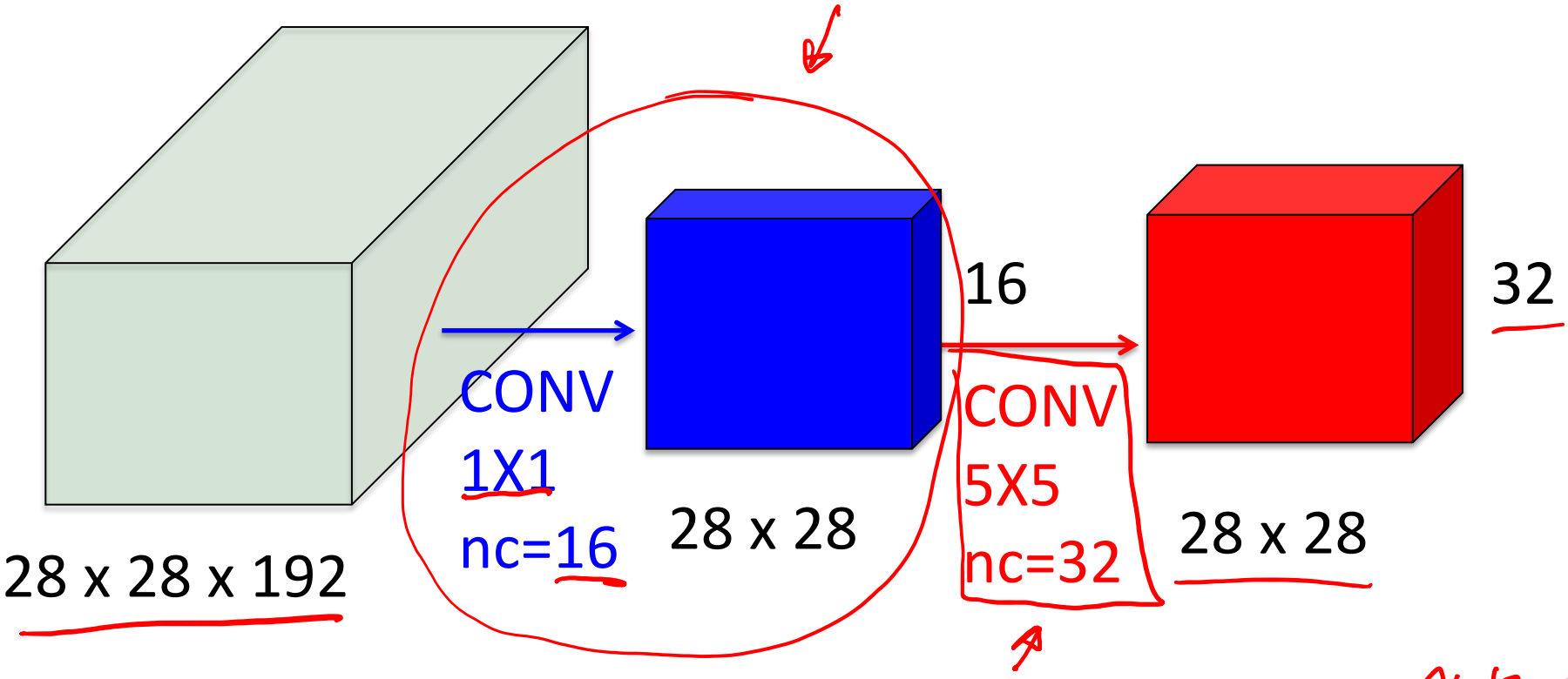
X $n_c^{[l+1]}$

More Layers: Bottleneck Layer



28 x 28 * 32 filters * 5x5x192 ~= **120M!**

Bottleneck Layer

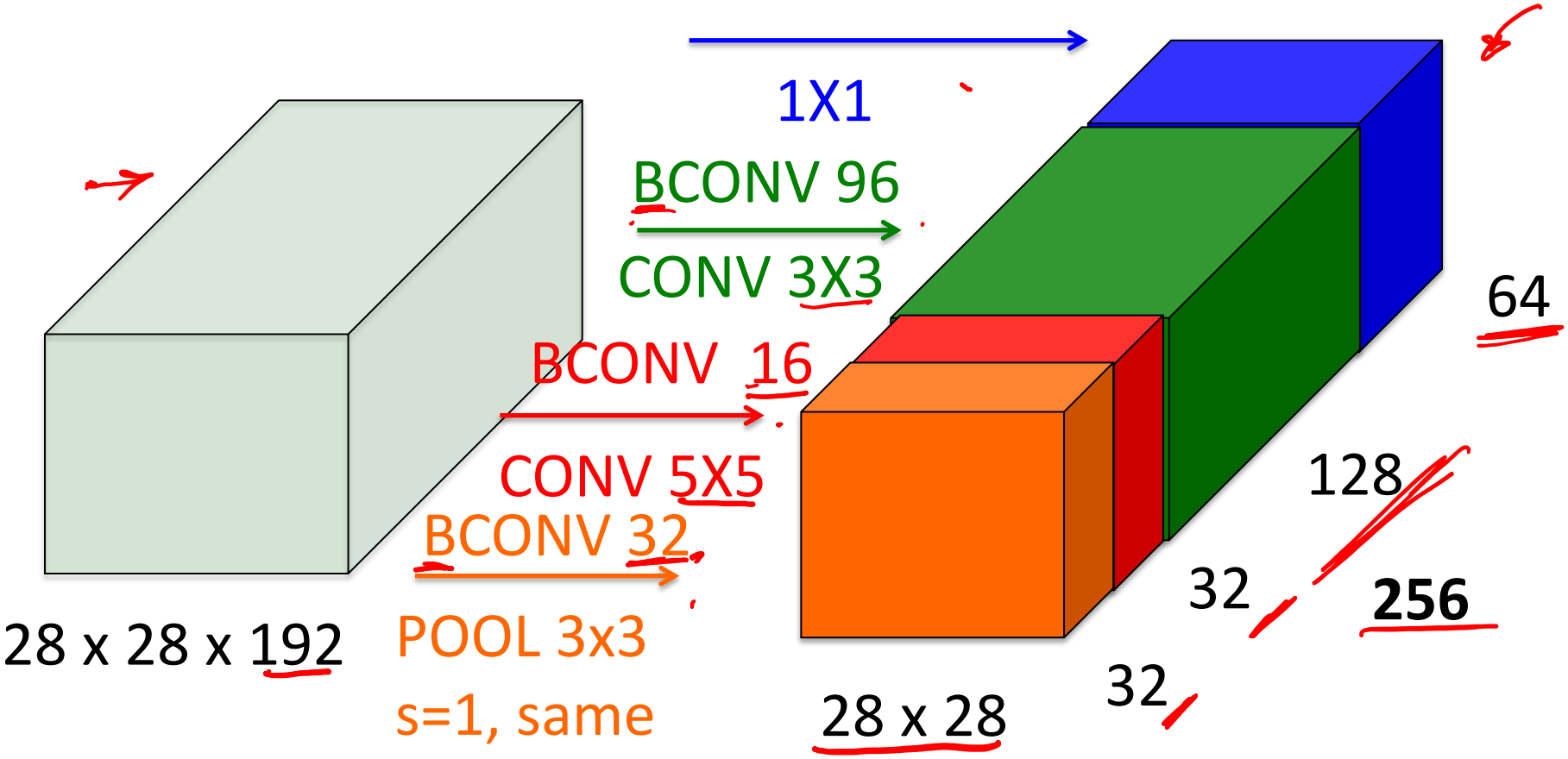


28 x 28 * 16 filters * 192 ~= **2.4M**

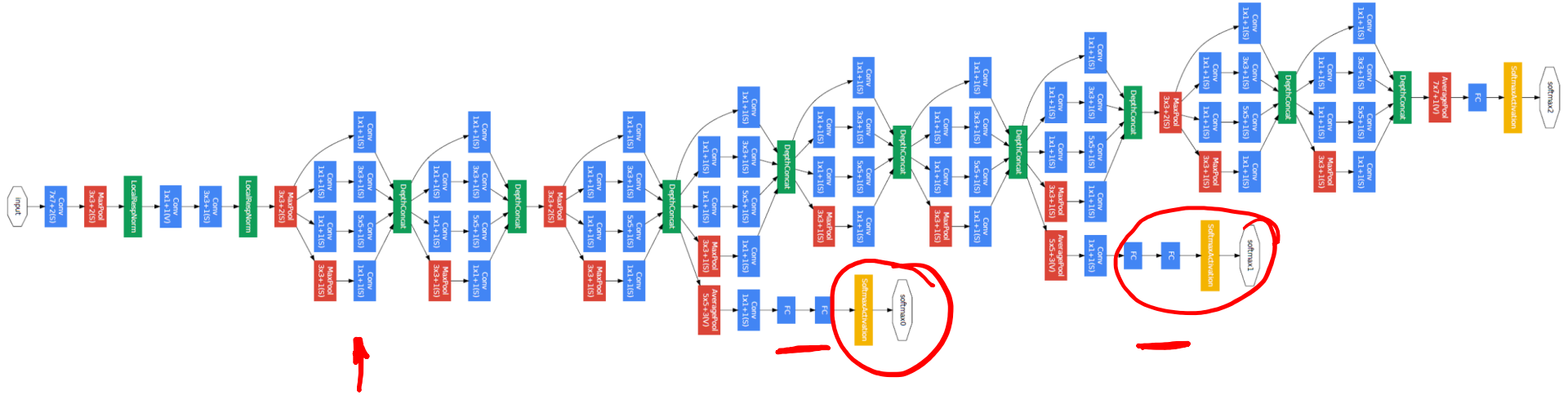
≈ 12M

28 x 28 * 32 filters x 5x5 x16 ~= **10M**

Inception Network



GoogLeNet



Szegedy et al., Going Deeper with Convolutions, CVPR 2015

Computer Vision Problems and Networks Setup

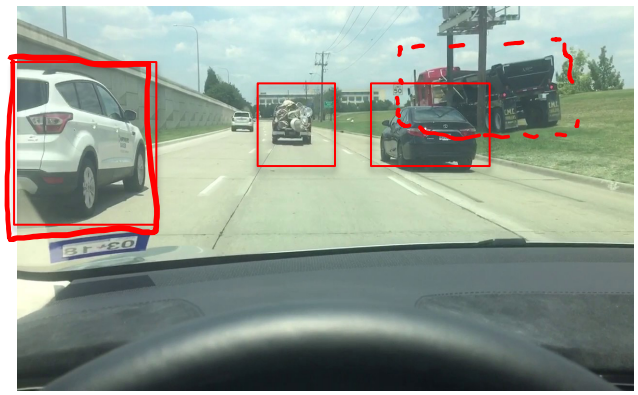
Object Classification

"Car"

Object Localization

"Car"

coord

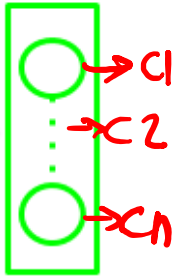


Object Detection

Object Classification

Input

CNN

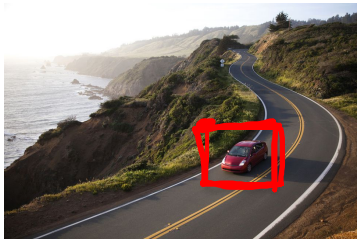


Softmax

Classes:

1. Pedestrian
2. Car
3. Dog
4. Bike

Object Localization



CNN

y = pc: car?

- bx
- by
- bw
- bh
- c1
- c2
- c3
- c4

(bx, by)

bh

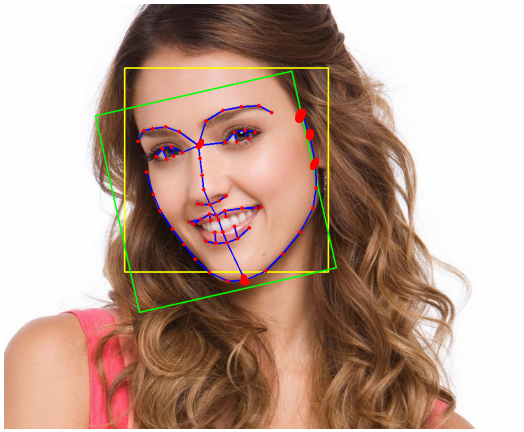
bw

If $pc = 1$ $L = (by - \hat{y}x)^2 + (by - \hat{y}y)^2 \dots$

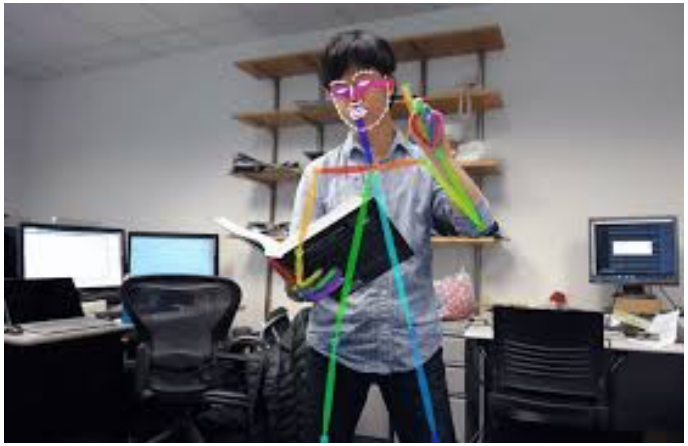
$pc = 0$ ∅

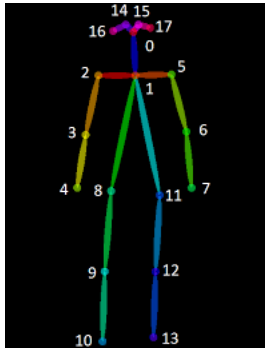
$(pc - \hat{y}pc)^2 + L$

Landmark Detection



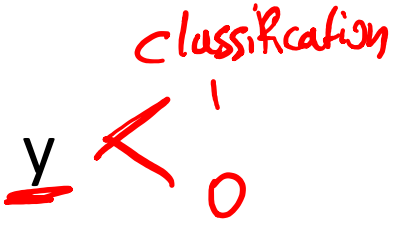
y = pf: face?
- l1x
- l1y
- l2x
- l2y
- .
- .
- .
- .

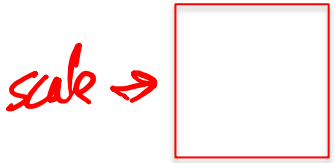




Object Detection

CNN





- Sliding window detectors