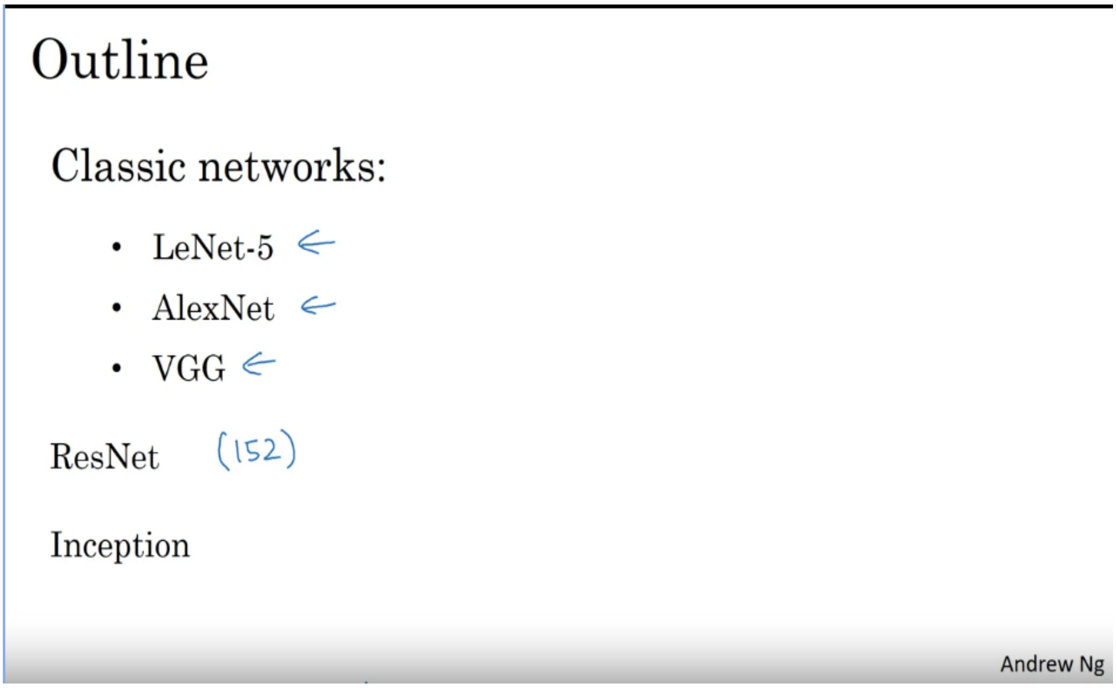
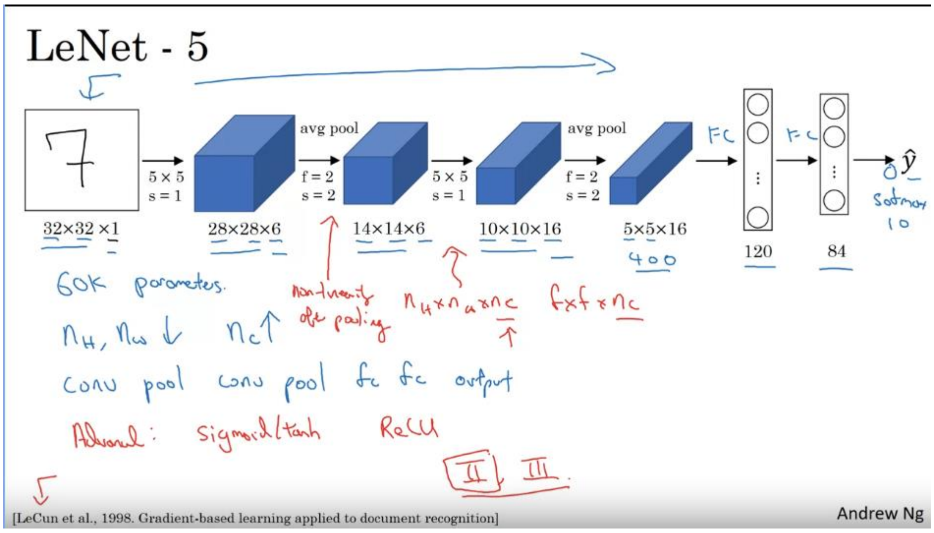
WEEK 2 CNN COURSE

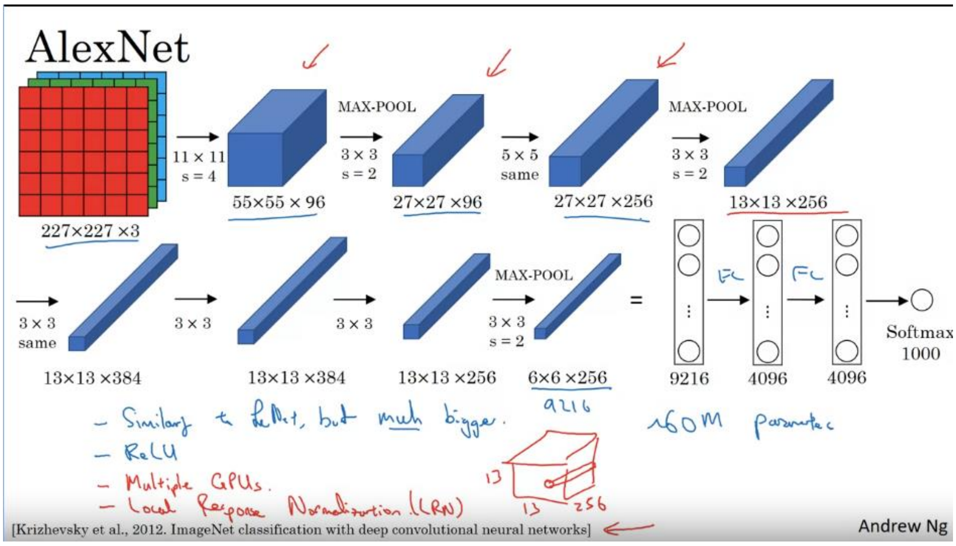
WHY LOOK AT CASE STUDIES?

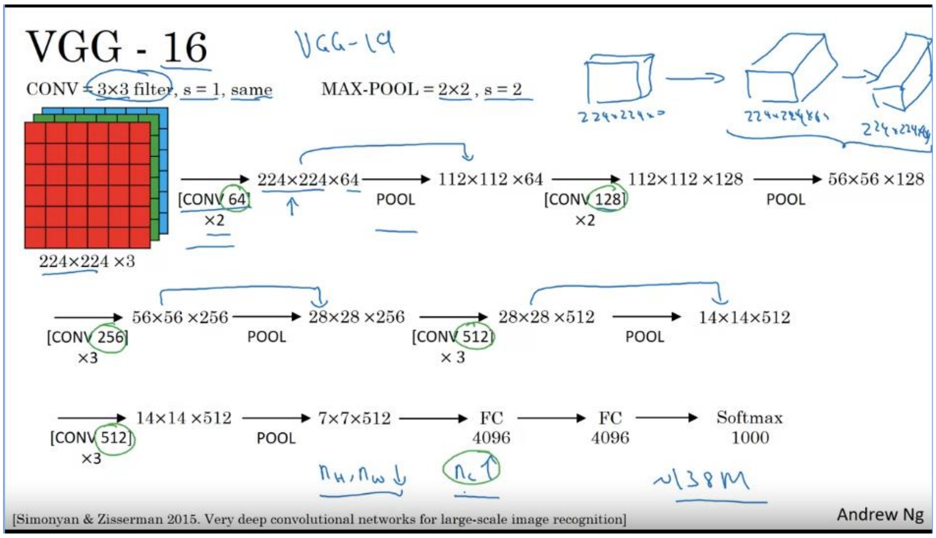


CLASSIC NETWORKS:

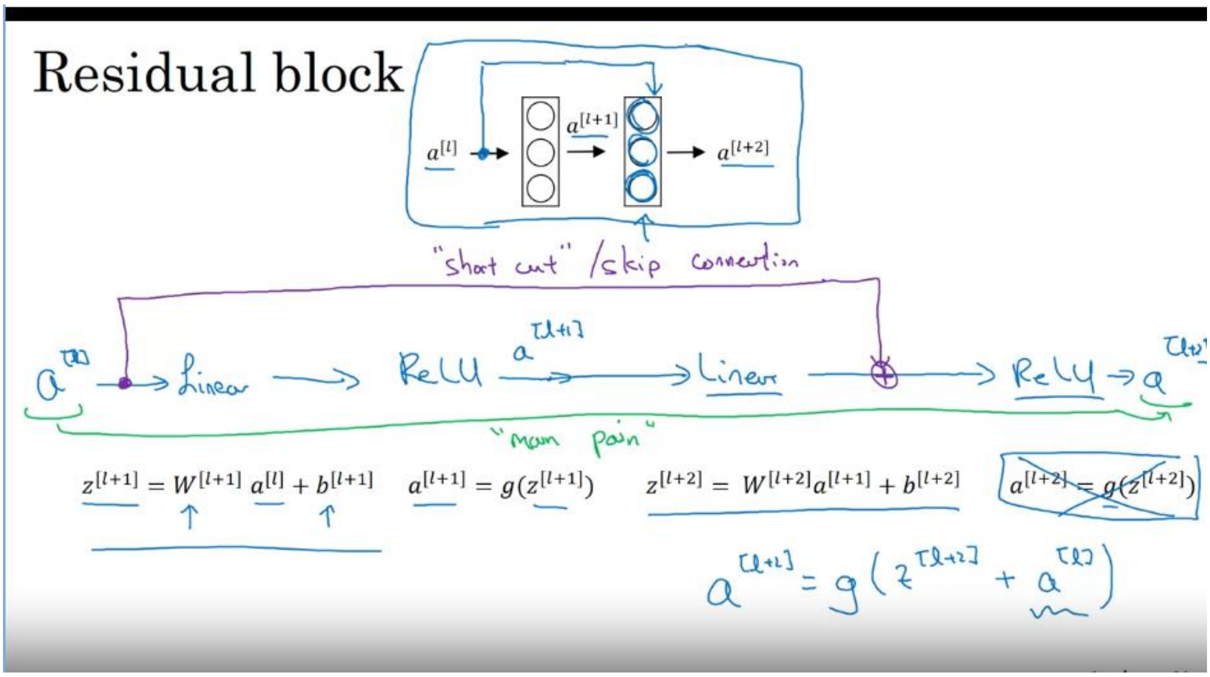


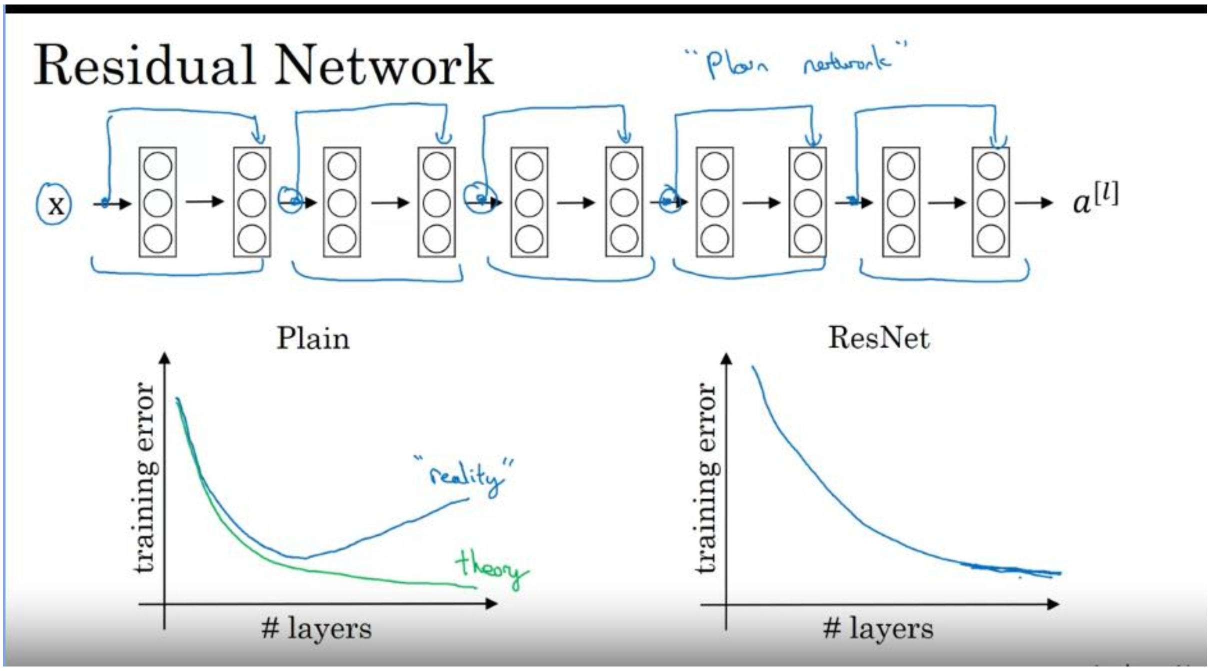
Red part can be skipped its optional. ⇑⇑⇑



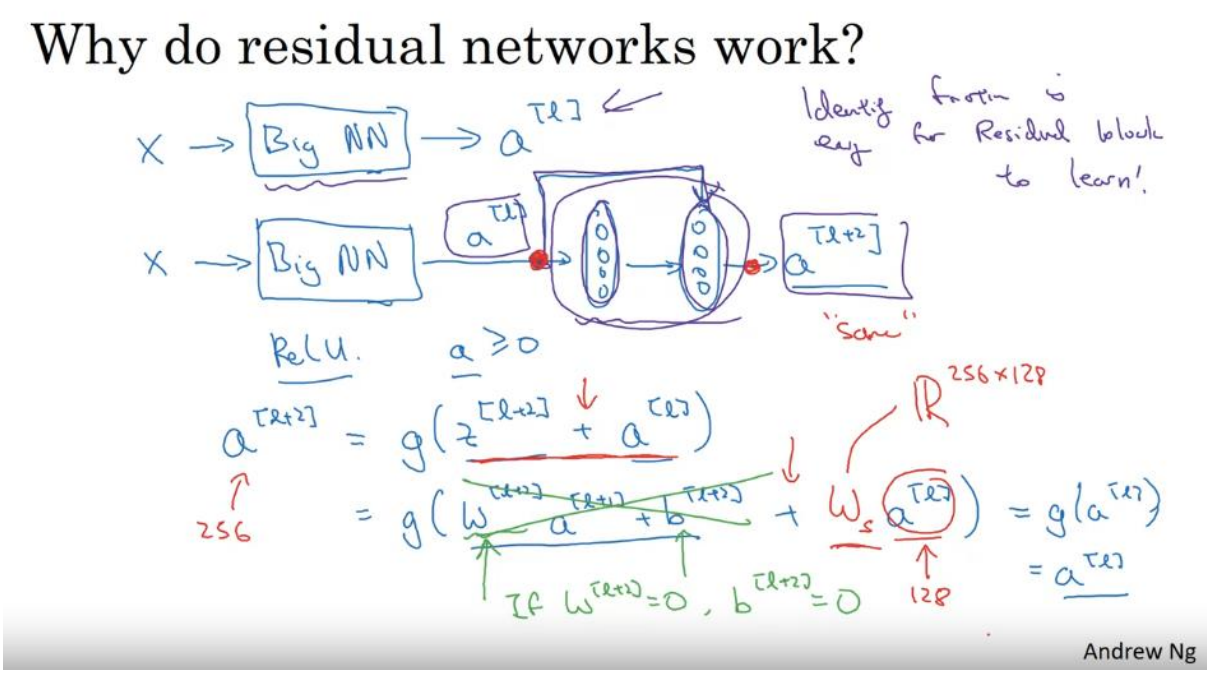


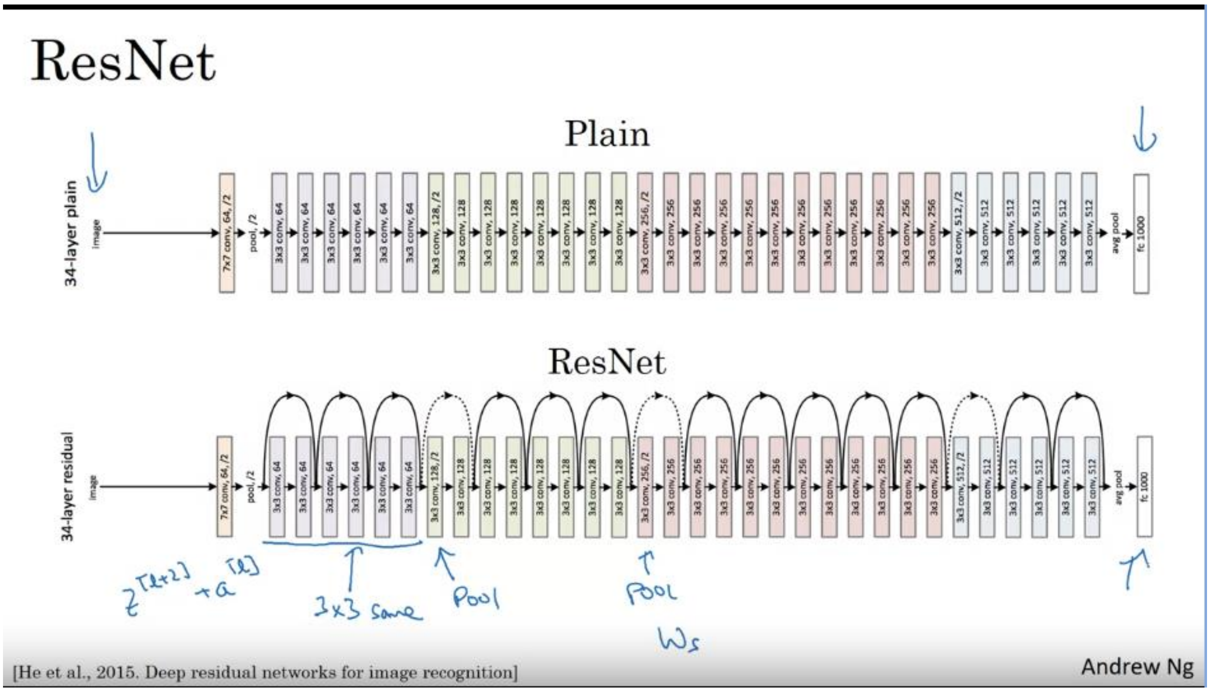
RESNETS (RESIDUAL NETWORKS)



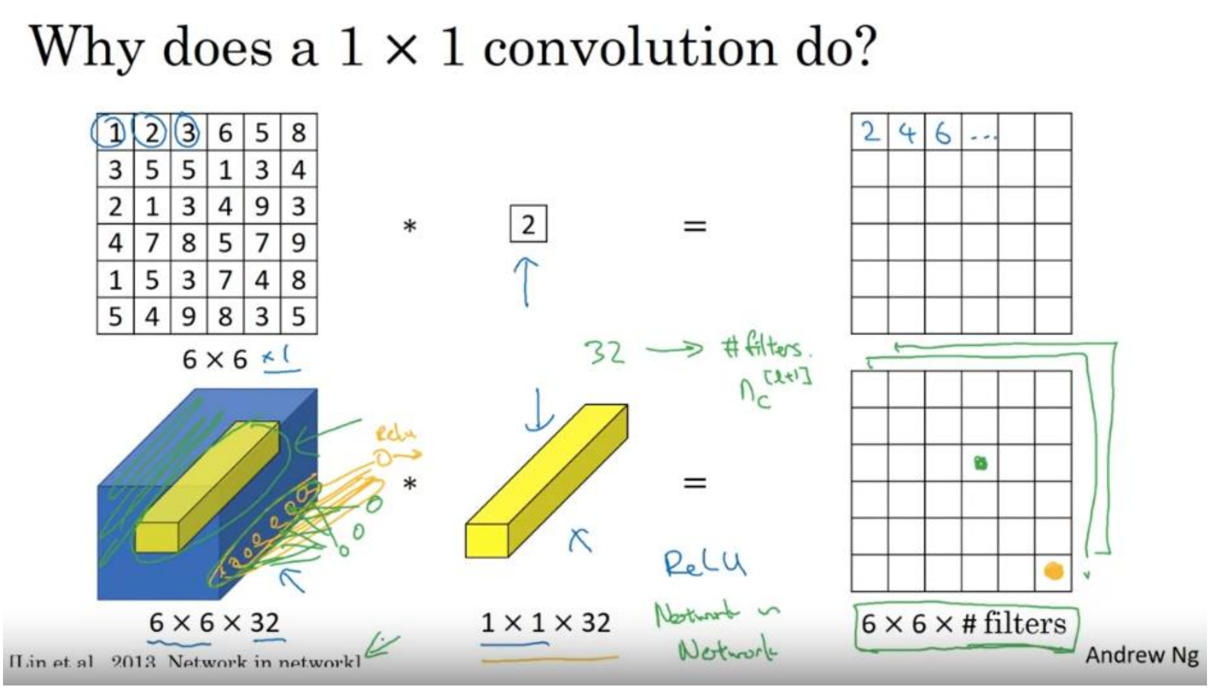


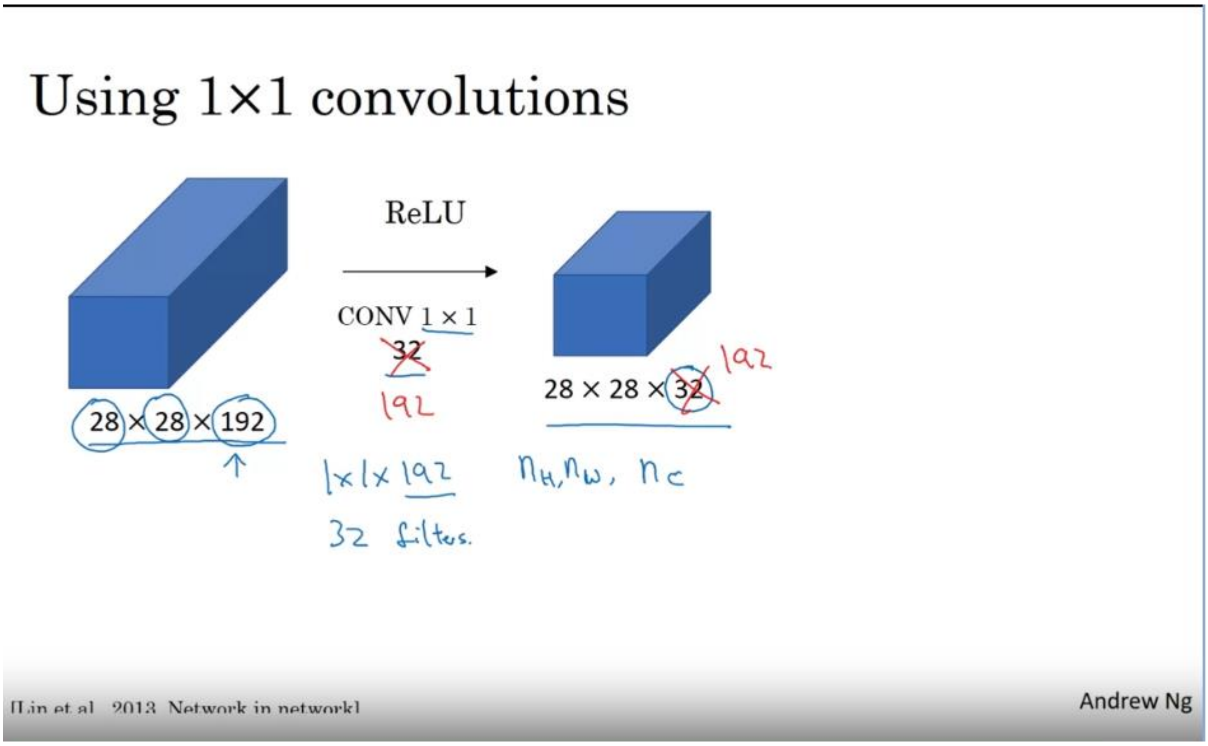
WHY DO RESNETS WORK?



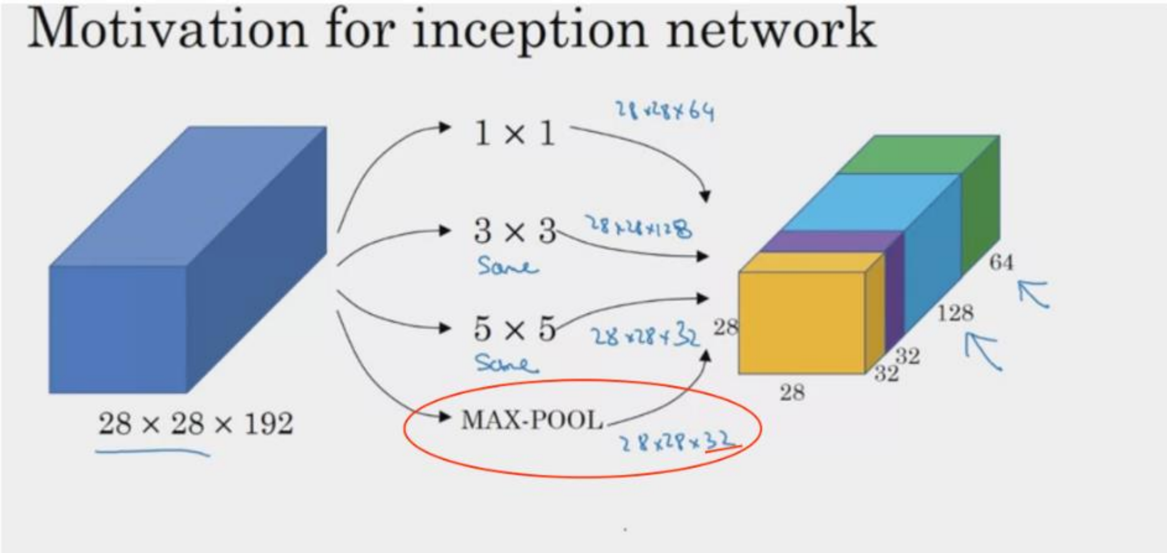


NETWORKS IN NETWORKS AND 1x1 CONVOLUTIONS:





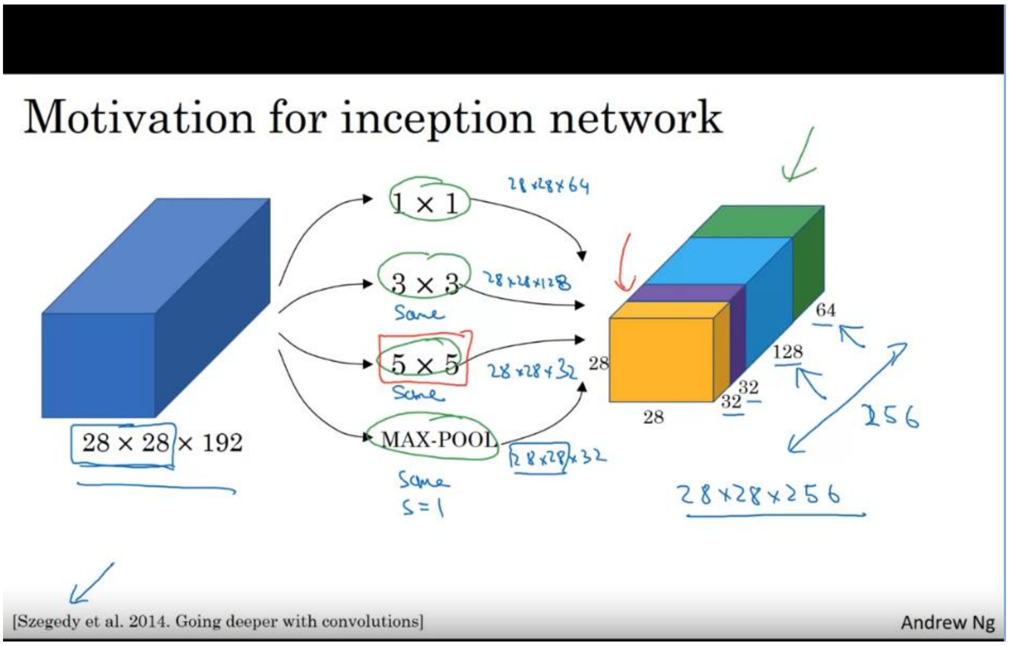
Clarifications about Upcoming Inception Network Motivation Video

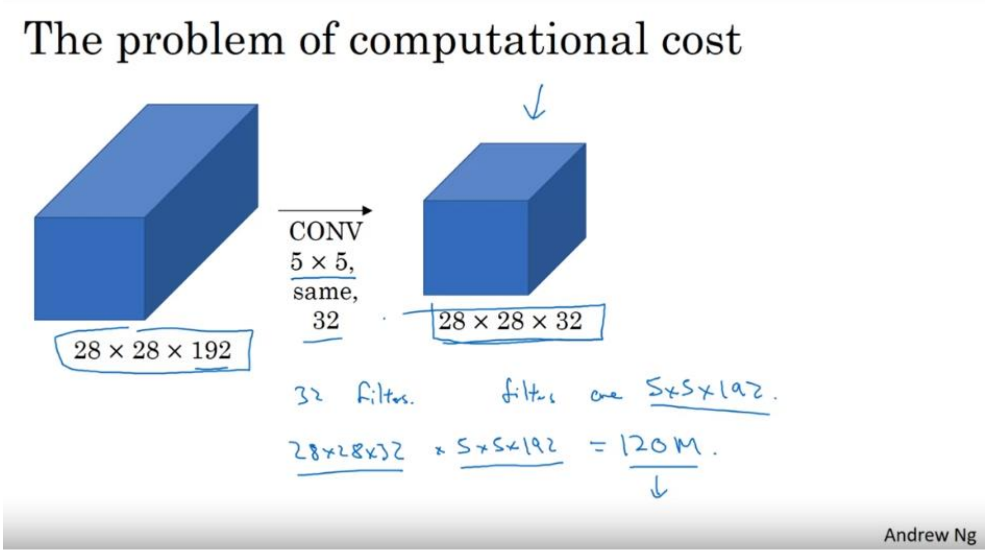


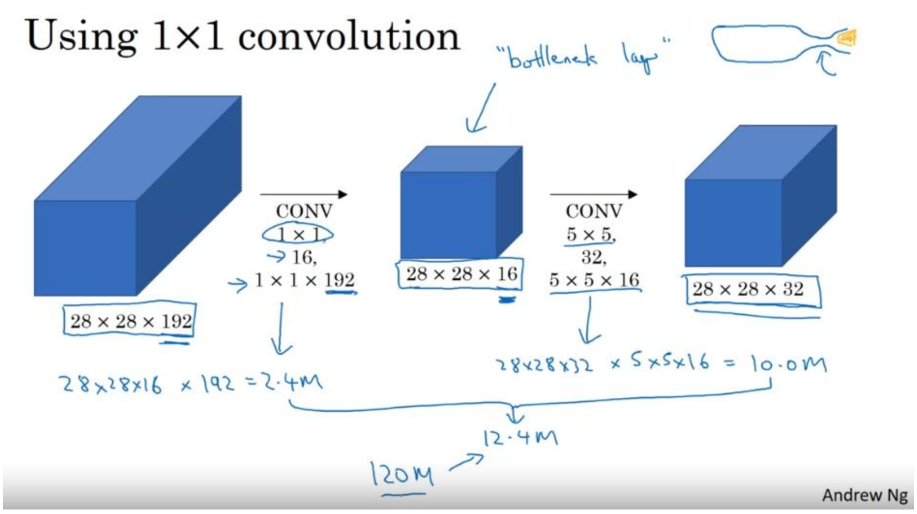
Note 1: If at 2:03 you get confused why after applying max pooling you get 28 x 28 x 32 output, please know Andrew mentions to talk about this in detail later, which he does in the following lecture video, [Inception Network](#) and explains why it got shrunk down from 192 to 32.

Note 2: At 3:00, Andrew should have said 28 x 28 x 192 instead of 28 x 28 x 129. The subtitles have been corrected.

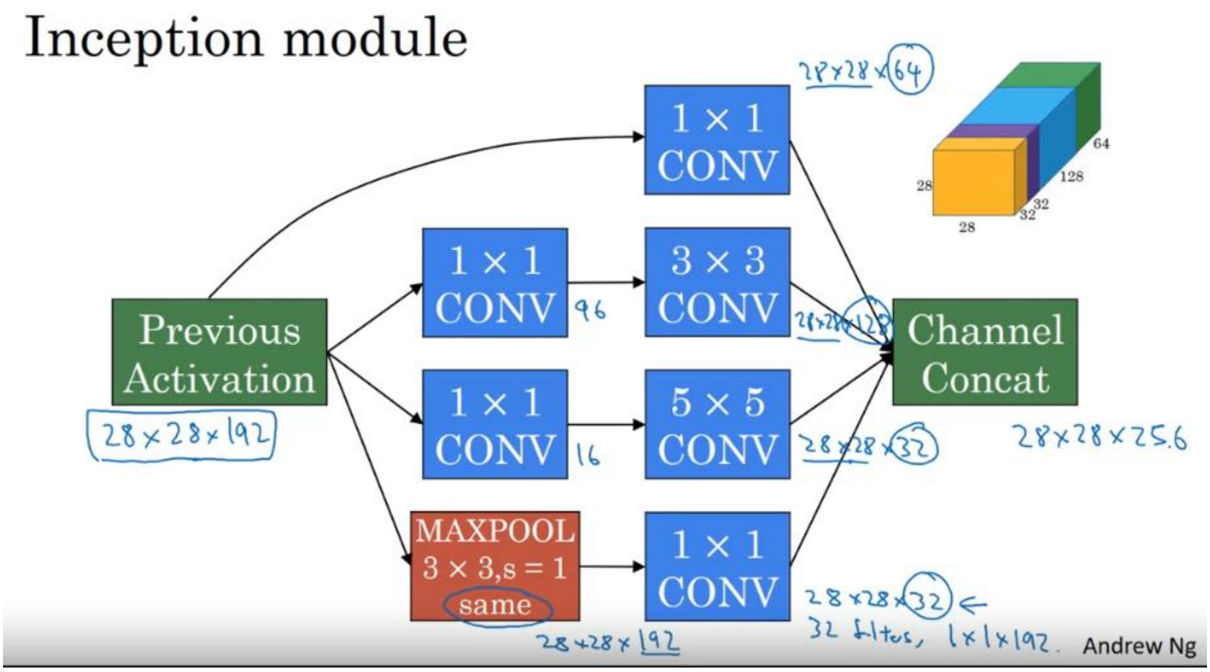
INCEPTION NETWORK MOTIVATION

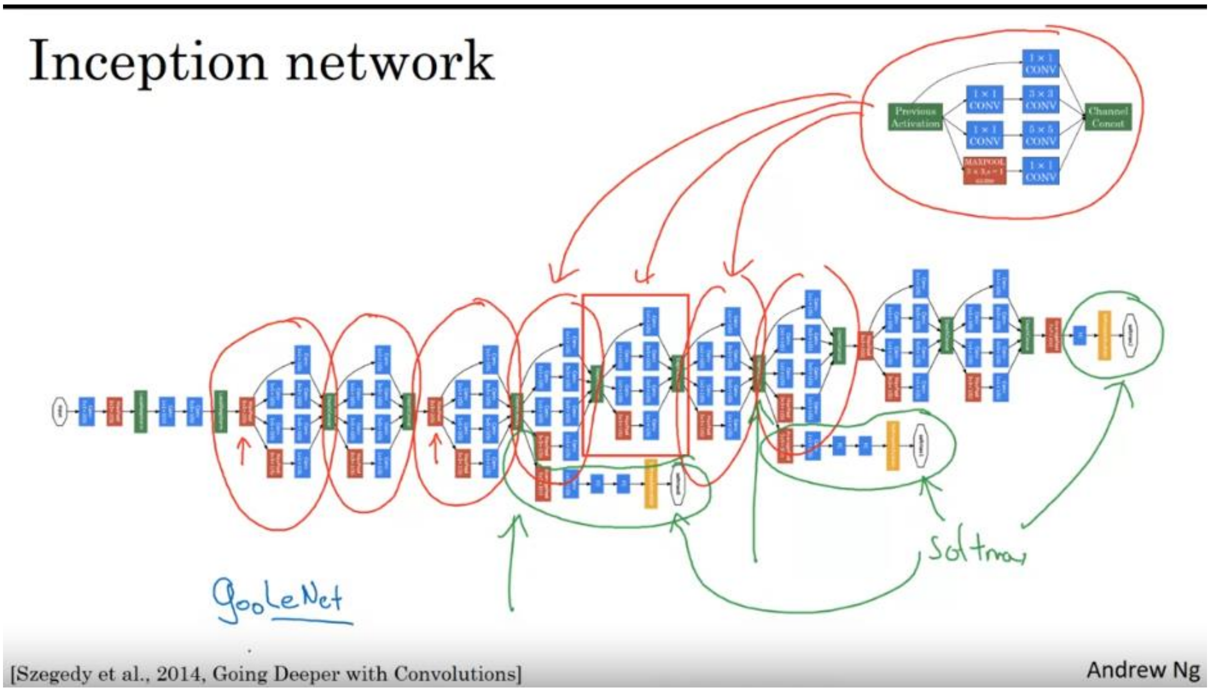




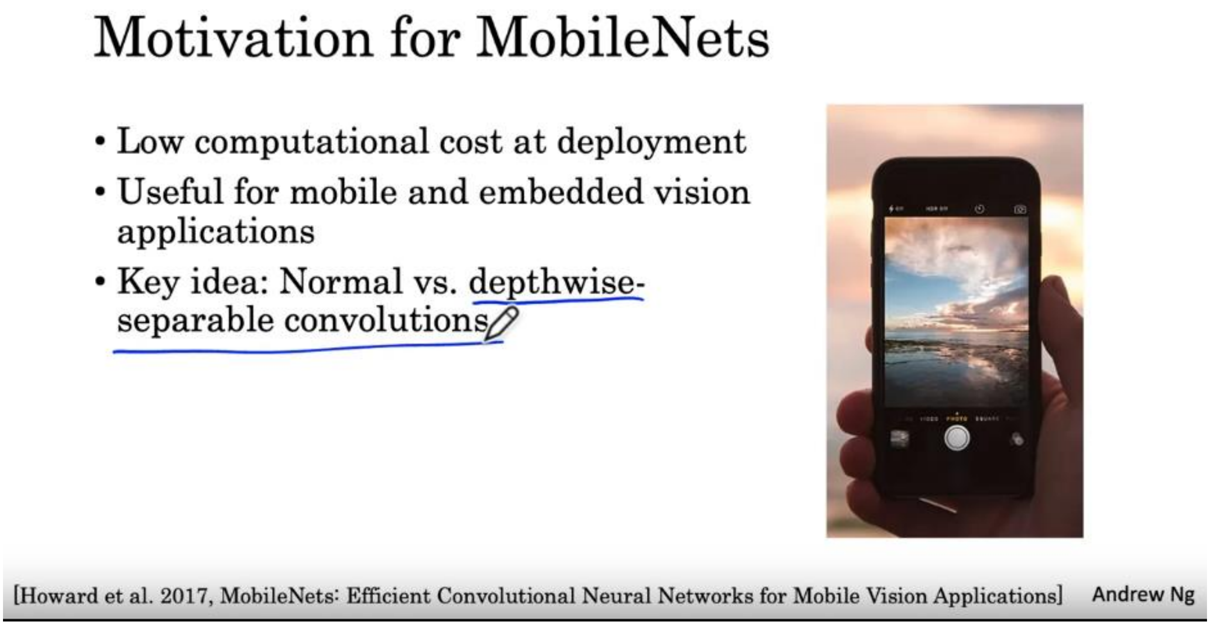


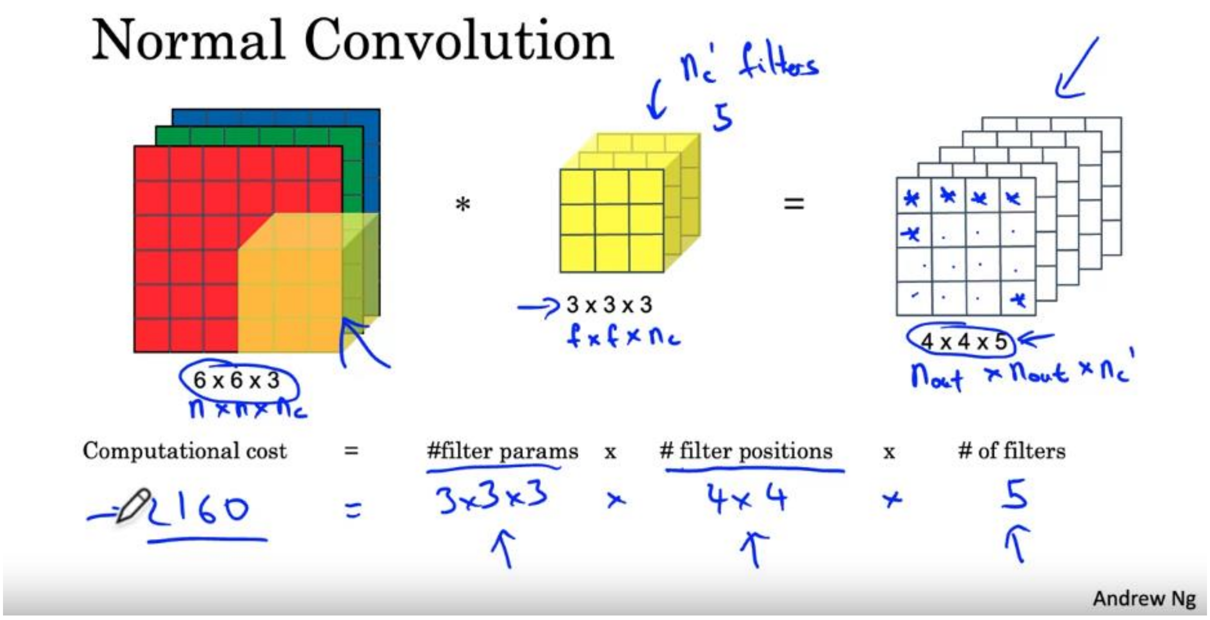
INCEPTION NETWORK





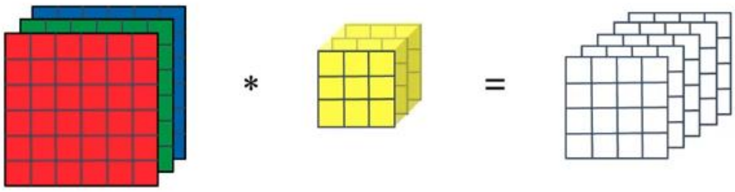
MOBILE NET


Motivation for MobileNets
• Low computational cost at deployment
• Useful for mobile and embedded vision applications
• Key idea: Normal vs. depthwise-separable convolutions
[Howard et al. 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications]
Andrew Ng

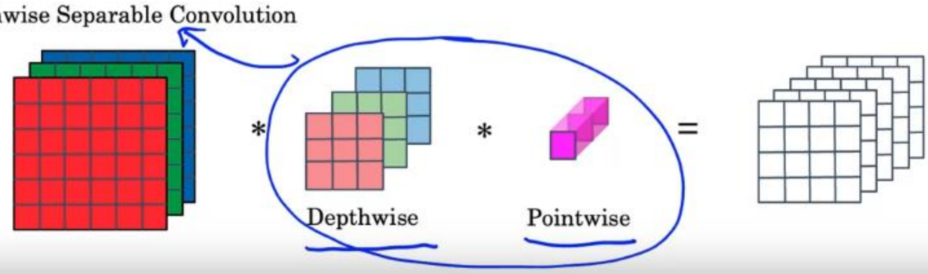

Normal Convolution
n_c' filters
5
*
=
6 x 6 x 3
$n \times n \times n_c$
3 x 3 x 3
$f \times f \times n_c$
4 x 4 x 5
$n_{out} \times n_{out} \times n_c'$
Computational cost = #filter params x # filter positions x # of filters
2160 = 3x3x3 x 4x4 x 5
Andrew Ng

Depthwise Separable Convolution

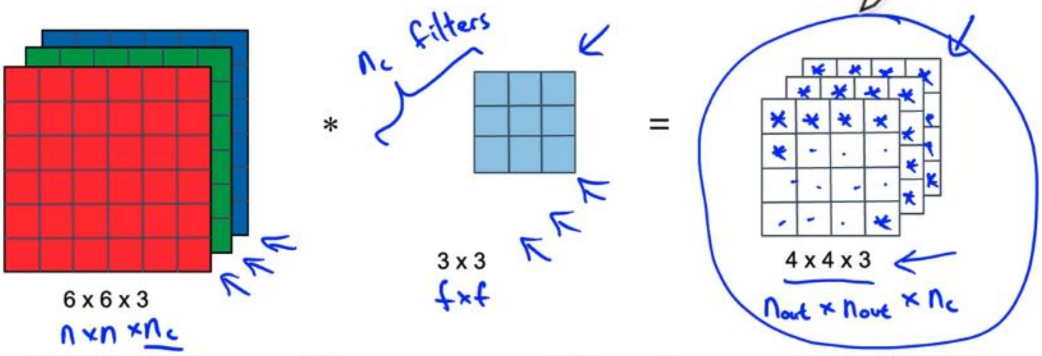
Normal Convolution



Depthwise Separable Convolution



Depthwise Convolution

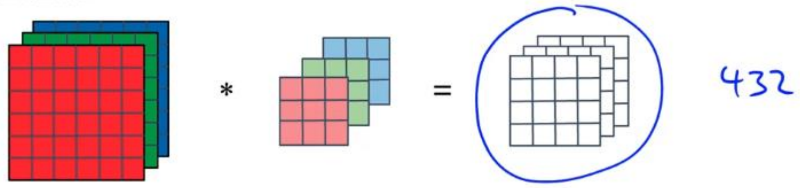


Computational cost = #filter params x # filter positions x # of filters

$432 = 3 \times 3 \times 4 \times 4 \times 3$

Depthwise Separable Convolution

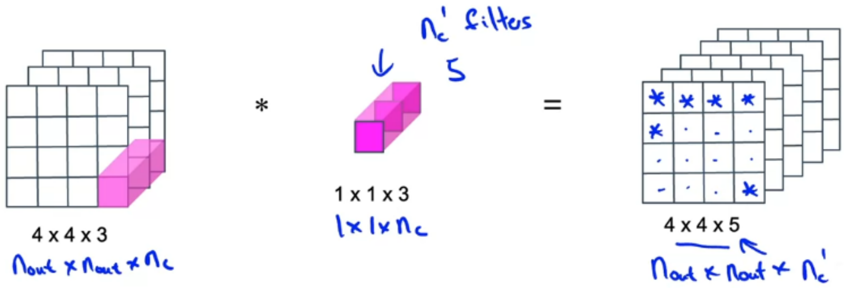
Depthwise Convolution



Pointwise Convolution

Andrew Ng

Pointwise Convolution



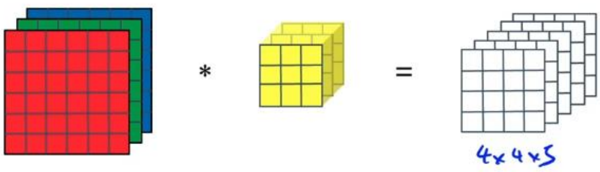
Computational cost = #filter params x # filter positions x # of filters

$1 \times 1 \times 3 \quad \times \quad 4 \times 4 \quad \times \quad 5$

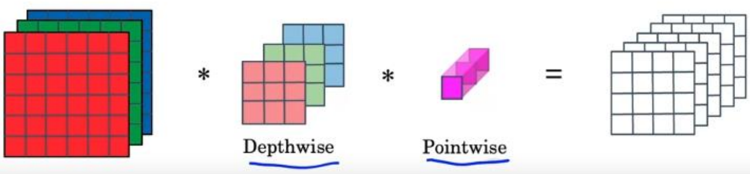
Andrew Ng

Depthwise Separable Convolution

Normal Convolution



Depthwise Separable Convolution



Andrew Ng

Cost Summary

Cost of normal convolution 2160

Cost of depthwise separable convolution

depthwise + pointwise

$432 + 240 = 672$

$$\frac{672}{2160} = 0.31$$

$$= \frac{1}{n_c'} + \frac{1}{f^2}$$

$$\frac{1}{5} + \frac{1}{9}$$

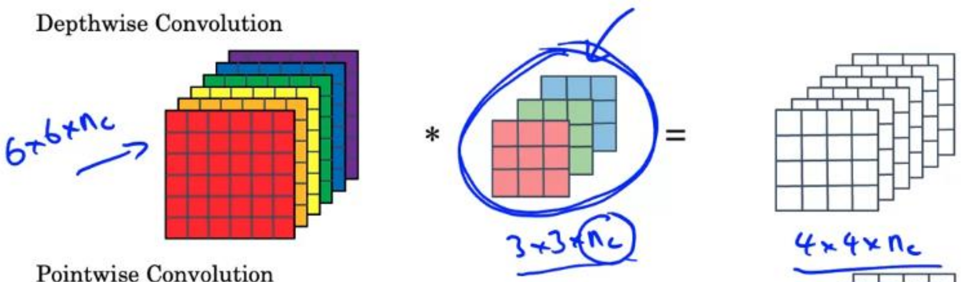
$$= \frac{1}{512} + \frac{1}{3^2}$$

~ 10 times cheaper

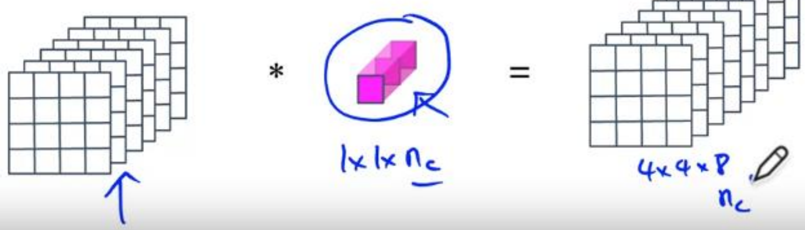
[Howard et al. 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications] Andrew Ng

Depthwise Separable Convolution

Depthwise Convolution

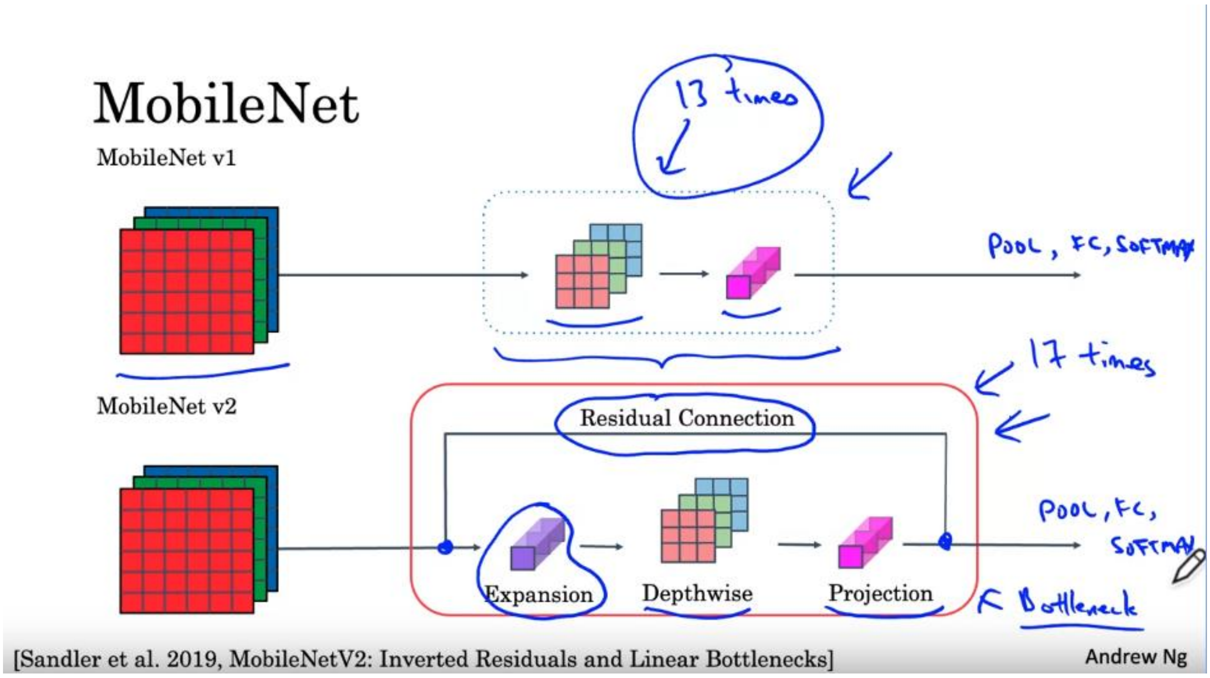


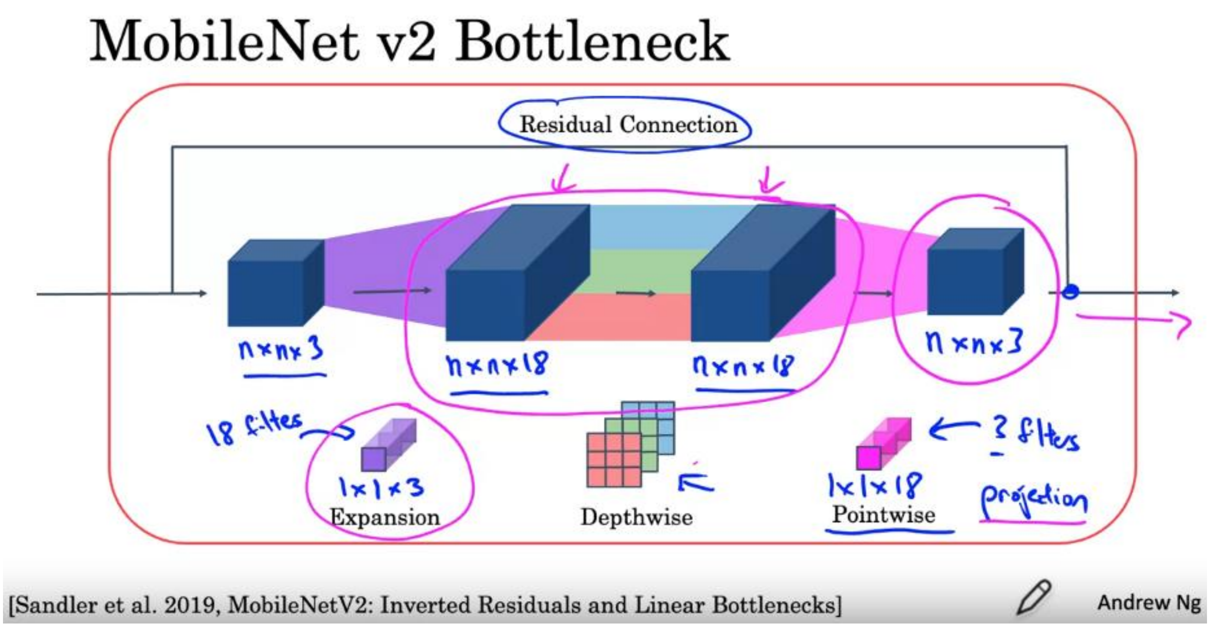
Pointwise Convolution



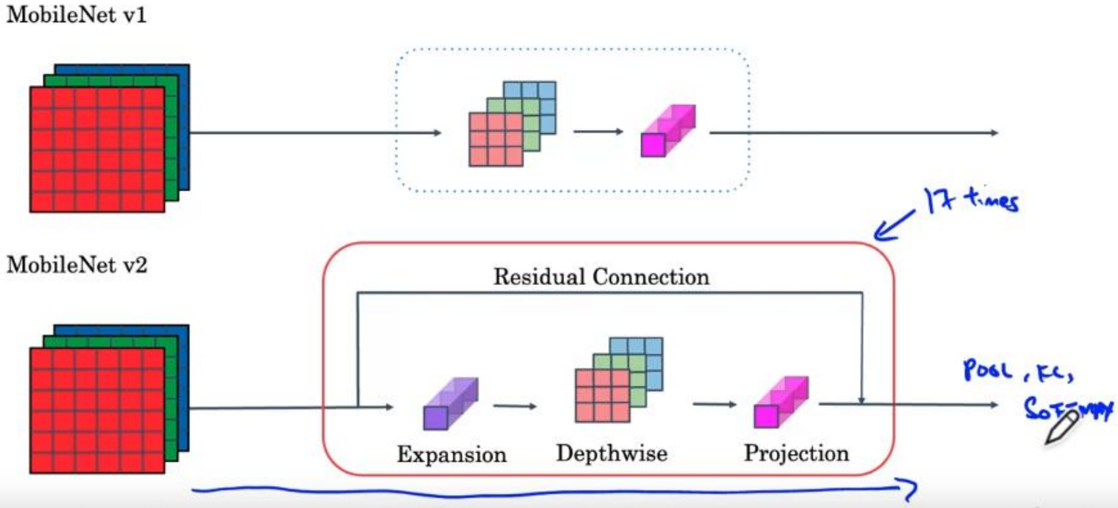
Andrew Ng

MOBILE NET ARCHITECTURE





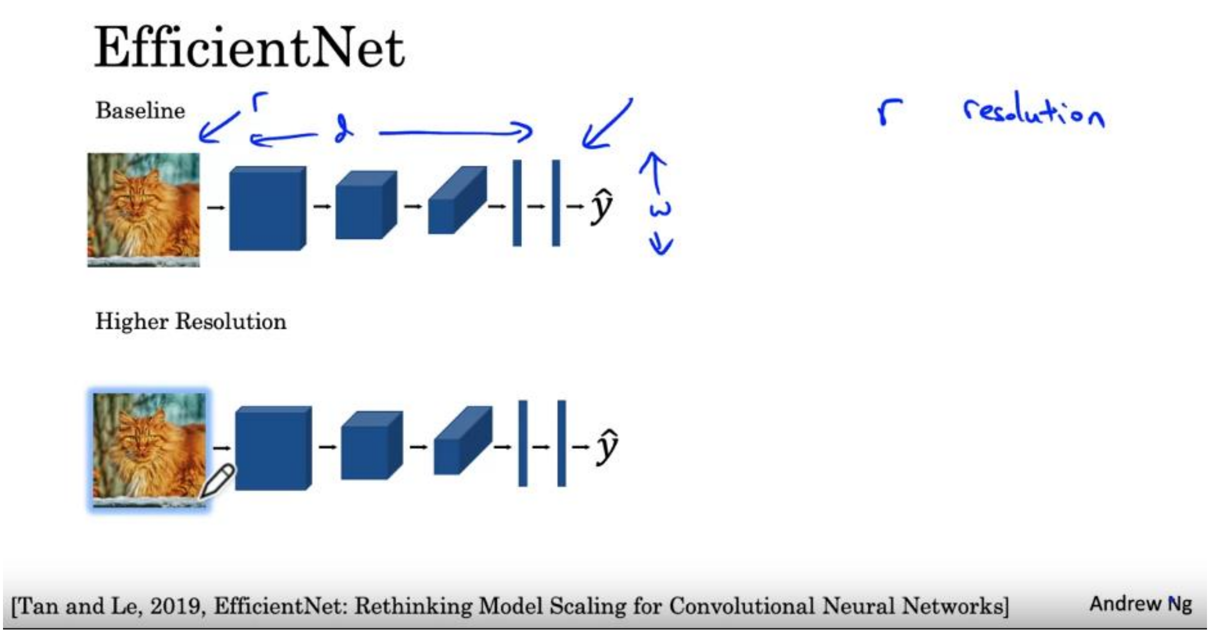
MobileNet

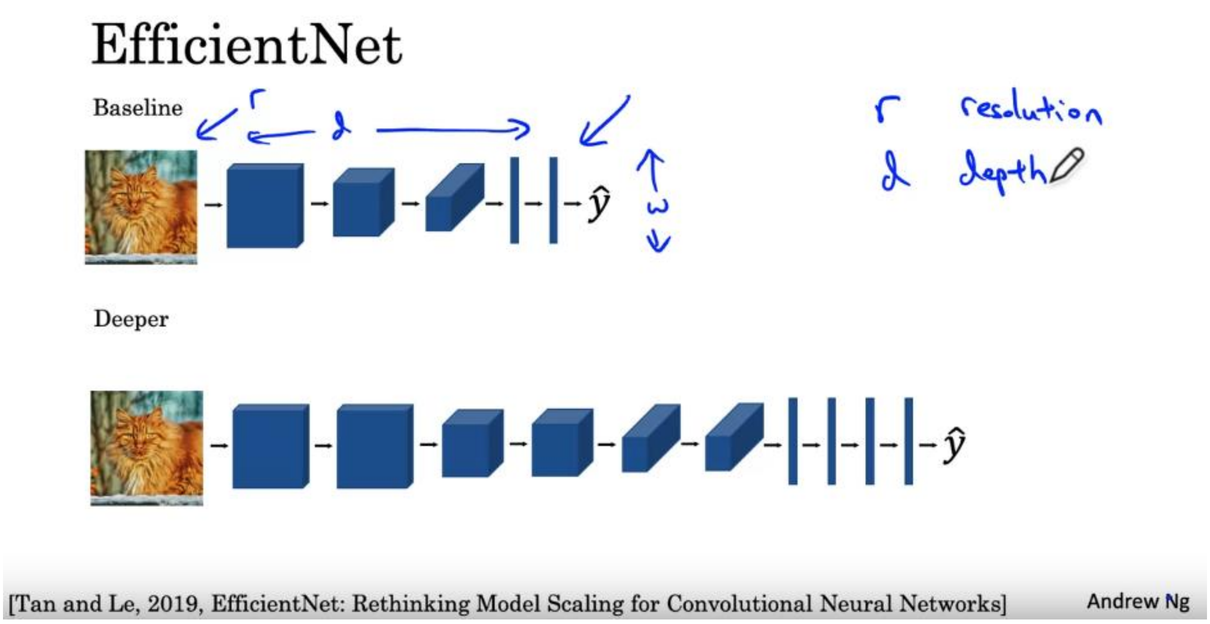


[Sandler et al. 2019, MobileNetV2: Inverted Residuals and Linear Bottlenecks]

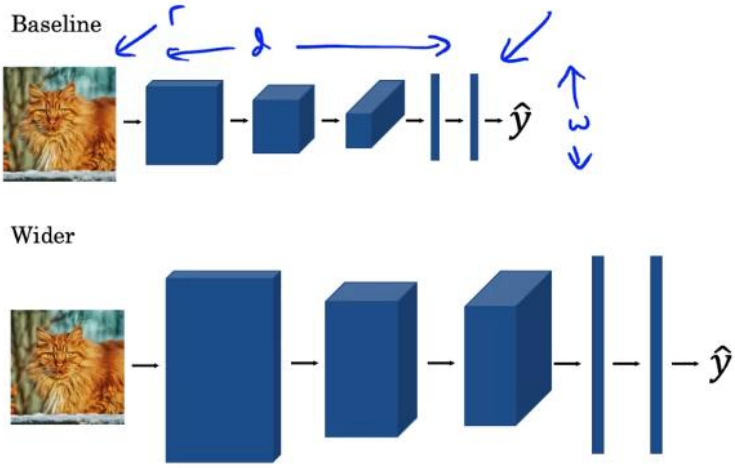
Andrew Ng

EFFICIENT NET





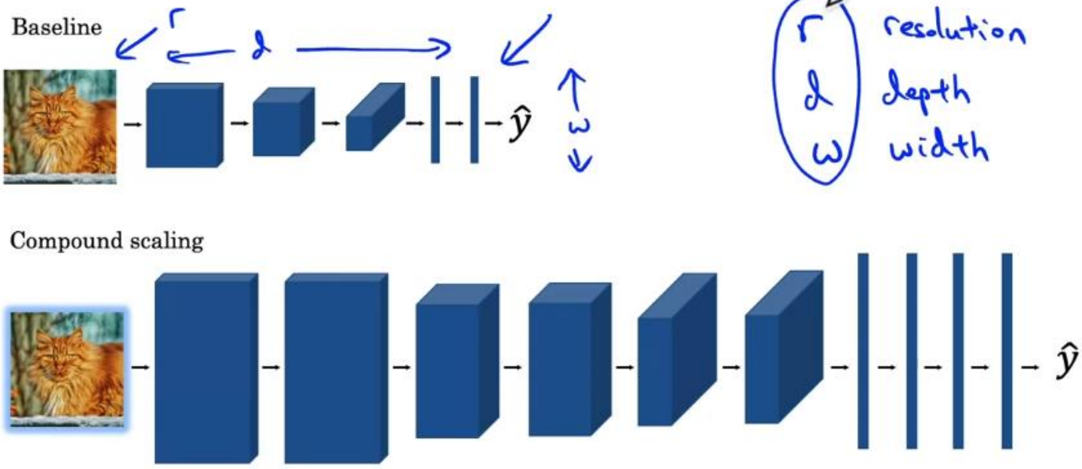
EfficientNet



r resolution
d depth
ω width

[Tan and Le, 2019, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks] Andrew Ng

EfficientNet



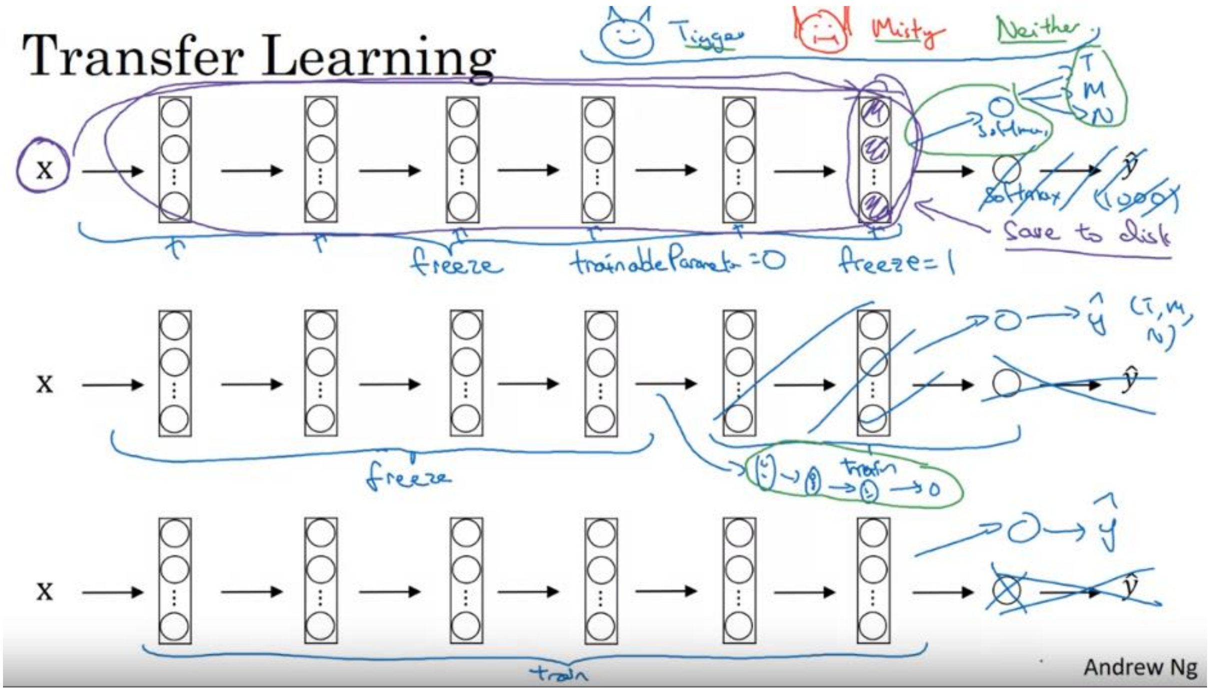
r resolution
d depth
ω width

[Tan and Le, 2019, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks] Andrew Ng

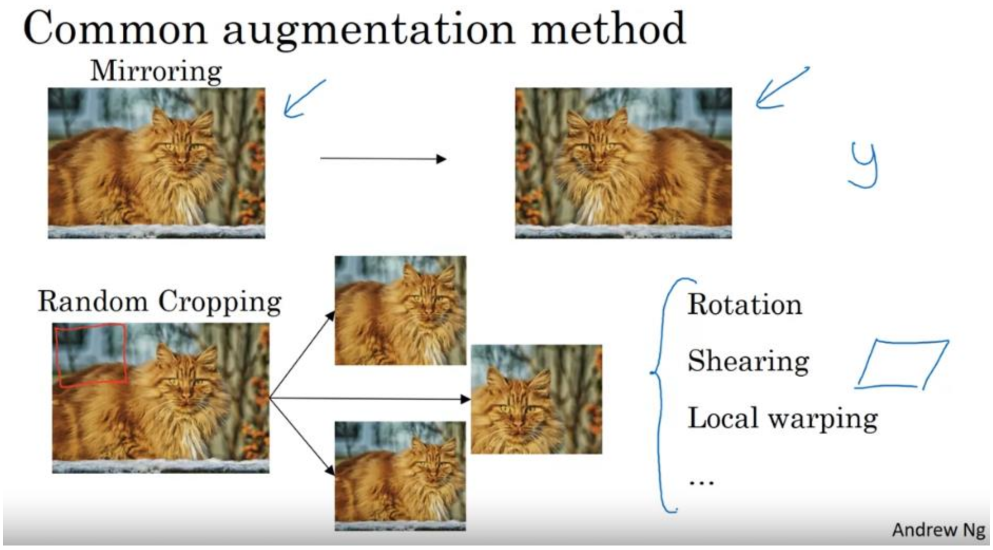
USING OPEN SOURCE IMPLEMENTATIONS

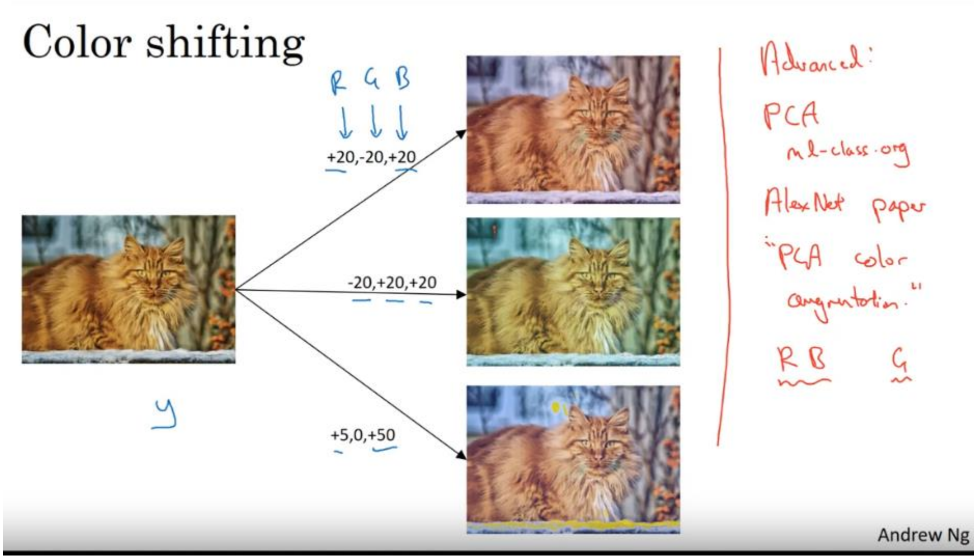
- About git giithub

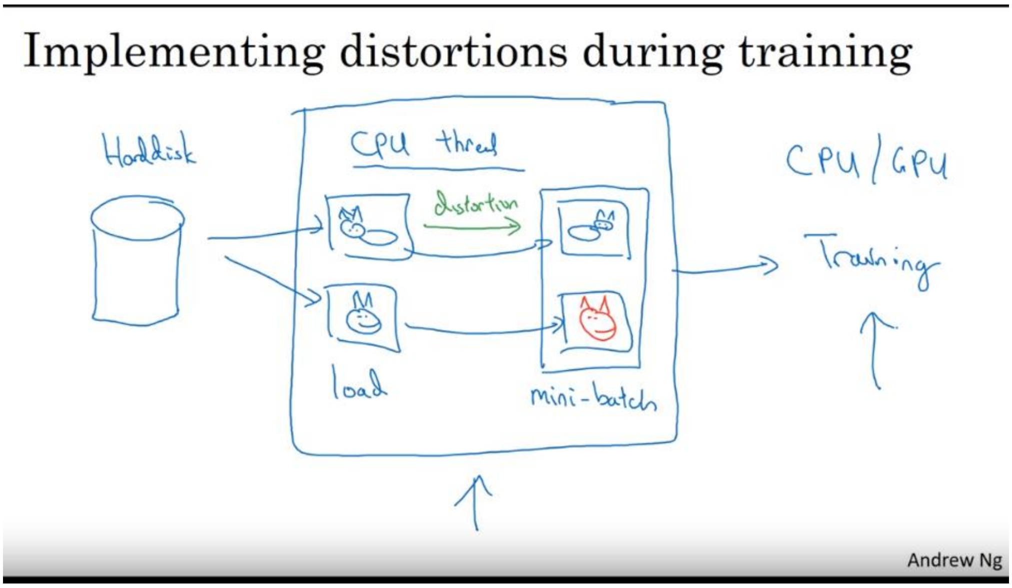
TRANSFER LEARNING



DATA AUGMENTATION







STATE OF COMPUTER VISION

Data vs. hand-engineering

Little data — More hand-engineering ("hacks")

Transfer learning

Object detection

Image recognition

Speech recognition

Lots of data — Simpler algorithms, less hand-engineering

Tigger/Misty/neither.

Two sources of knowledge

- Labeled data (x,y)
- Hand engineered features/network architecture/other components

Andrew Ng

Tips for doing well on benchmarks/winning competitions

Ensembling — 3-15 networks → $\hat{y}$

- Train several networks independently and average their outputs

Multi-crop at test time

- Run classifier on multiple versions of test images and average results

10-crop

1 + 4 + 1 + 4

Andrew Ng

Use open source code

- Use architectures of networks published in the literature
- Use open source implementations if possible
- Use pretrained models and fine-tune on your dataset

Andrew Ng