convolutional neural networks with swift for tensorflow: overview

brettkoonce.com/talks

august 7th, 2020

summary

- goal: explore image recognition w/ s4tf
- neural networks + convolutions
- mnist, cifar, vgg, resnet
- mobilenet, efficientnet
- sota, next steps

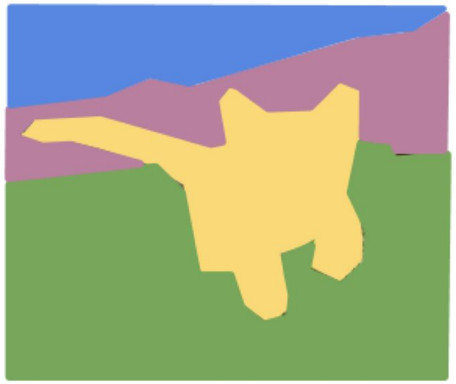
why?

- me: swift, unix, deep learning
- beginner mindset
- depth vs breadth
- cheap experimentation --> mastery
- cnn's are foundational

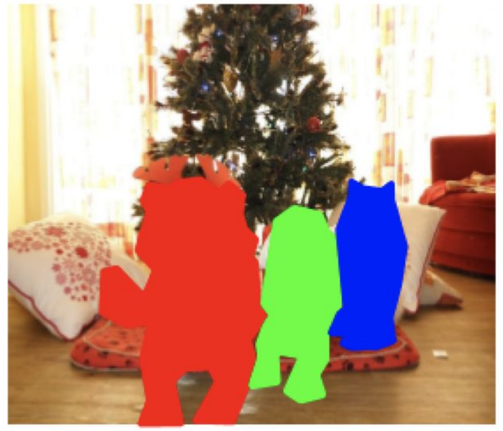
computer vision problems

- image recognition
- object detection
- image segmentation
- instance segmentation

CAT

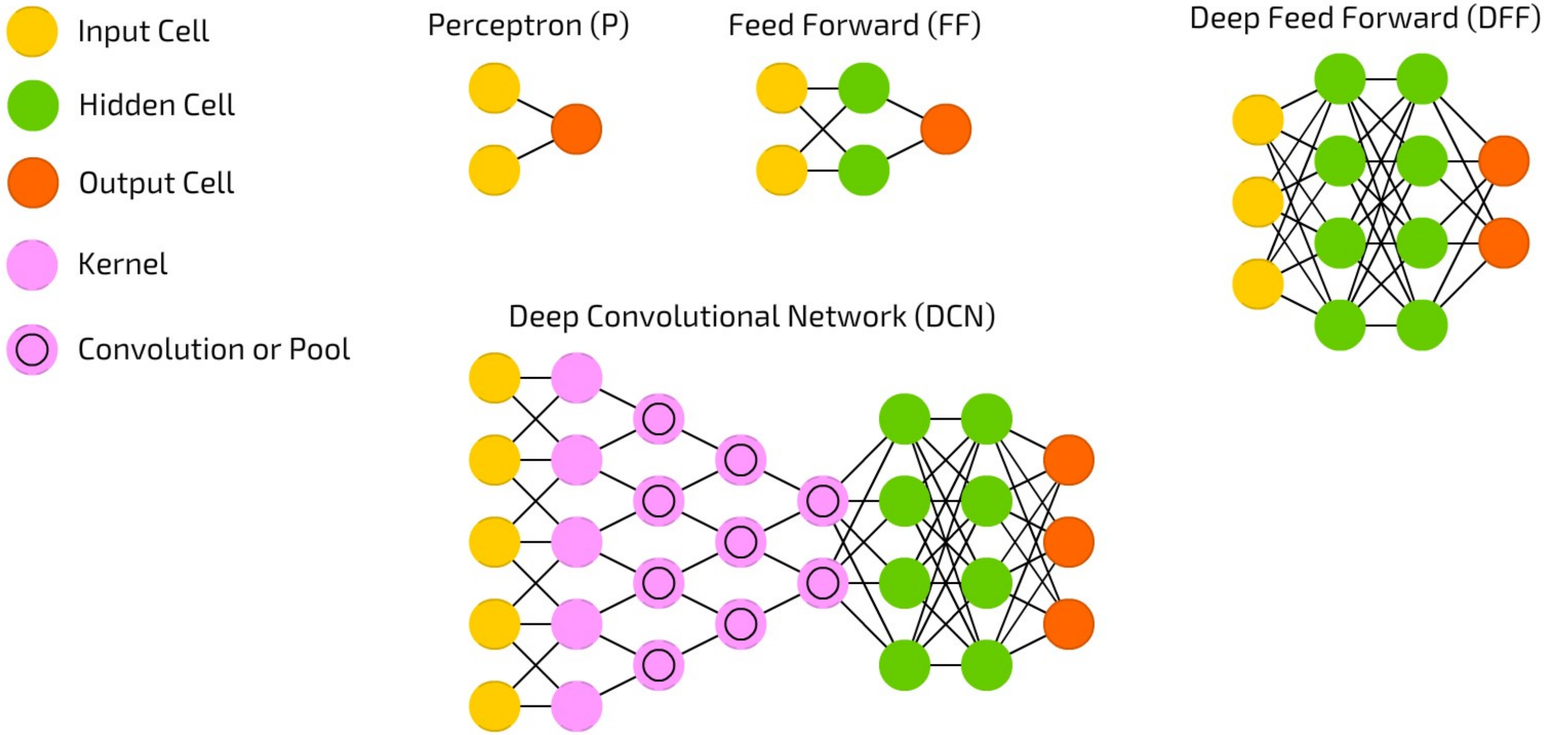


GRASS, CAT,
TREE, SKY



DOG, DOG, CAT

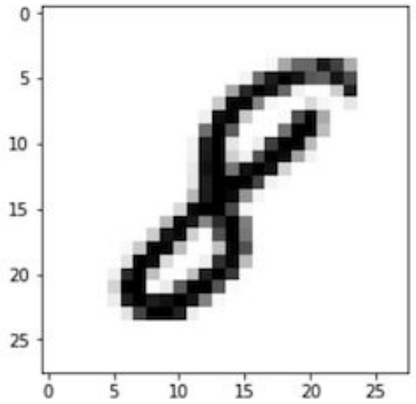
convolutional neural networks

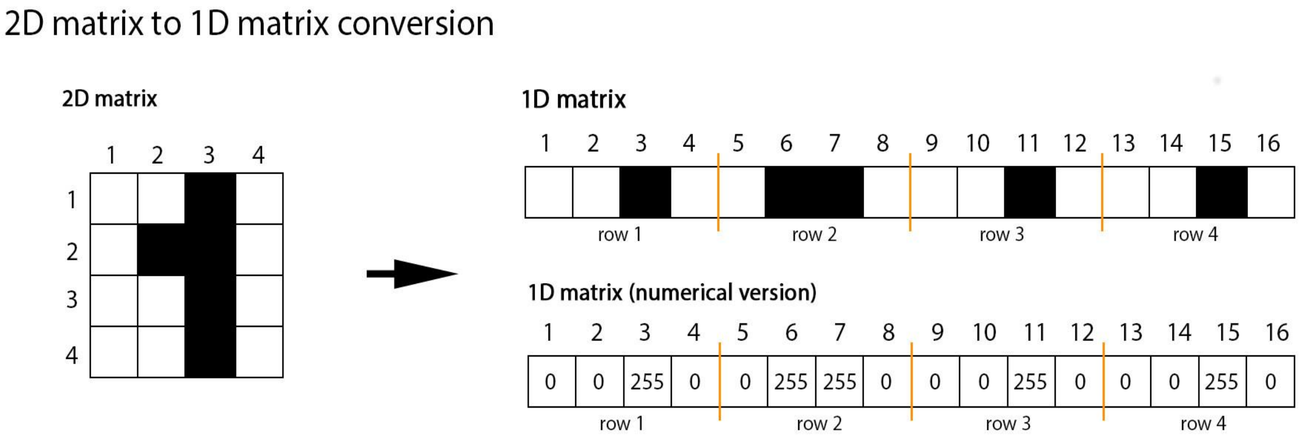


1d mnist

- mnist: 28x28x1 (h, w, 0..255 greyscale)
- Convert to 1d vector: [row1, row2, row3...row28] == (784 x 1)
- 2 * 512 fully connected layers

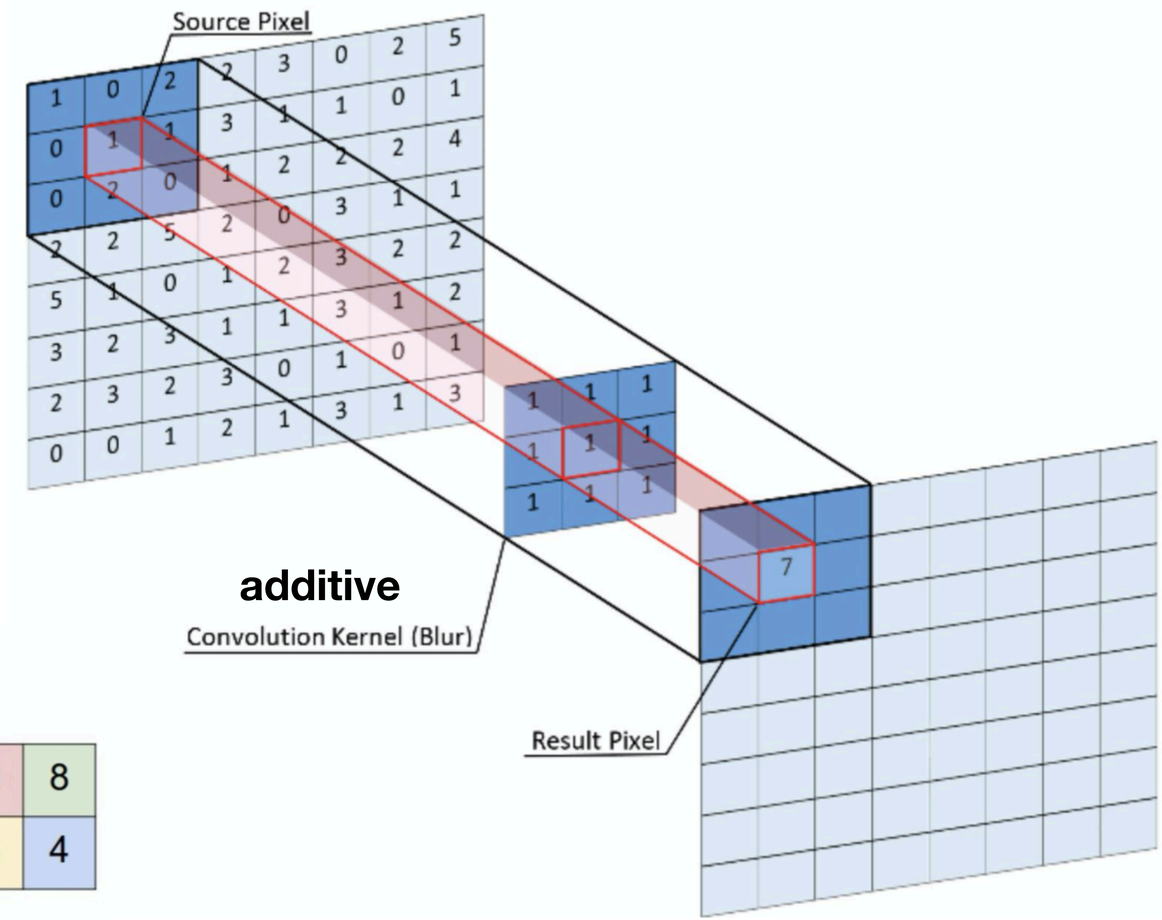
INPUT	784x1
Dense	512
Dense	512
OUTPUT	10

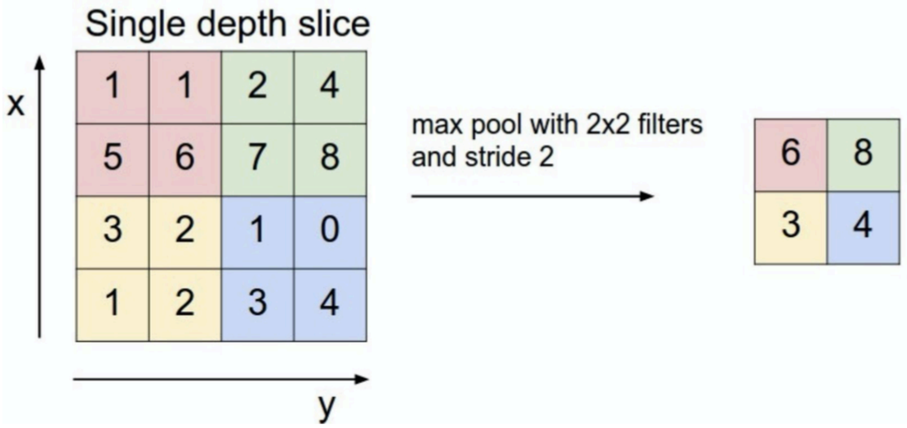




convolutions

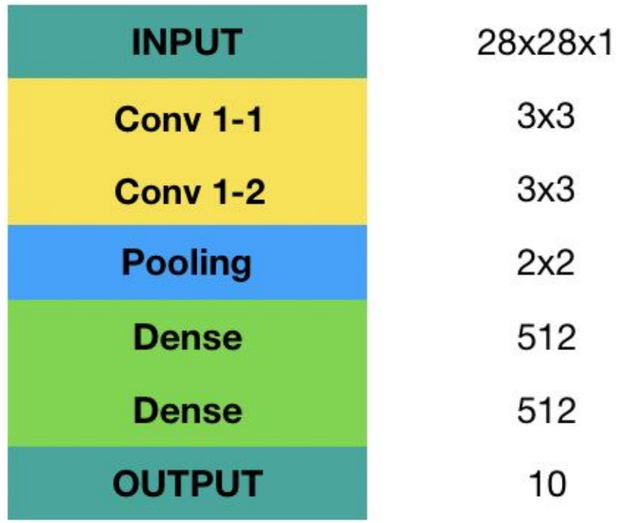
- 3x3 blur example
- 3x3 striding
- 2x2 maxpool





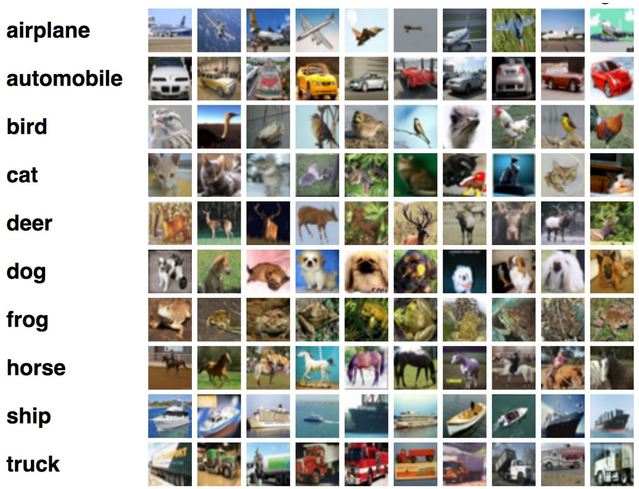
2d mnist

- 3x3 convolution
- 3x3 convolution
- 2x2 maxpool
- 2 * 512 fully connected layers



2d stacking, color, cifar

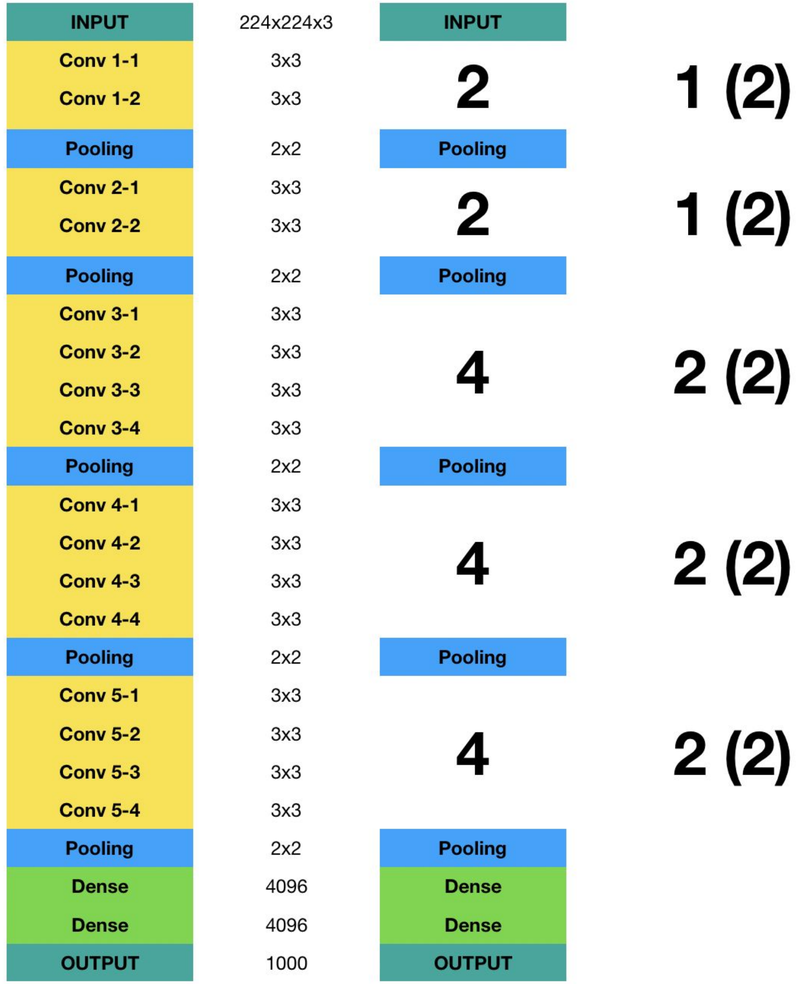
- [3x3 striding, 3x3 striding, 2x2 maxpool]
  → [block1] + [block2]
- 2 * 512 fully connected layers
- Cifar: (32x32x3, 10 categories)



INPUT	32x32x3	INPUT
Conv 1-1	3x3	2
Conv 1-2	3x3	
Pooling	2x2	Pooling
Conv 2-1	3x3	2
Conv 2-2	3x3	
Pooling	2x2	Pooling
Dense	512	Dense
Dense	512	Dense
OUTPUT	10	OUTPUT

vgg (2014)

- arxiv:1409.1556
- vgg16: [2, 2, 3, 3, 3]
- vgg19: [2, 2, 4, 4, 4]
- 2 * 4096 fully connected layers
- imagenet: (224x224x3, 1000 categories)



residual networks (2015)

- arxiv: 1512.03385
- cnn backbone + skip connections ⟶ resnet 34
- resnet 34 + bottleneck blocks (1x3x1) ⟶ resnet 50

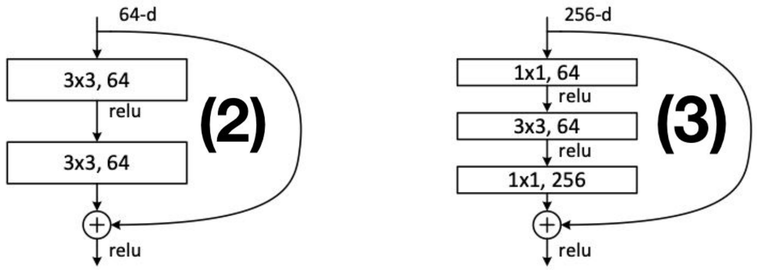
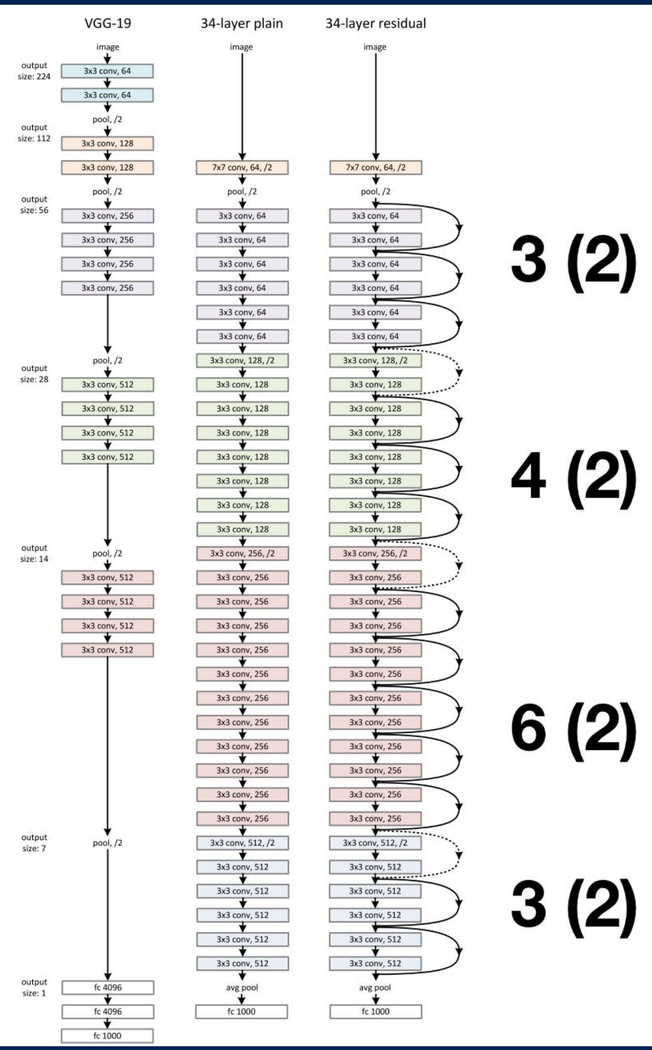


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.



mobilenet v1

- 2017 talk: depthwise, pointwise convolutions
- towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

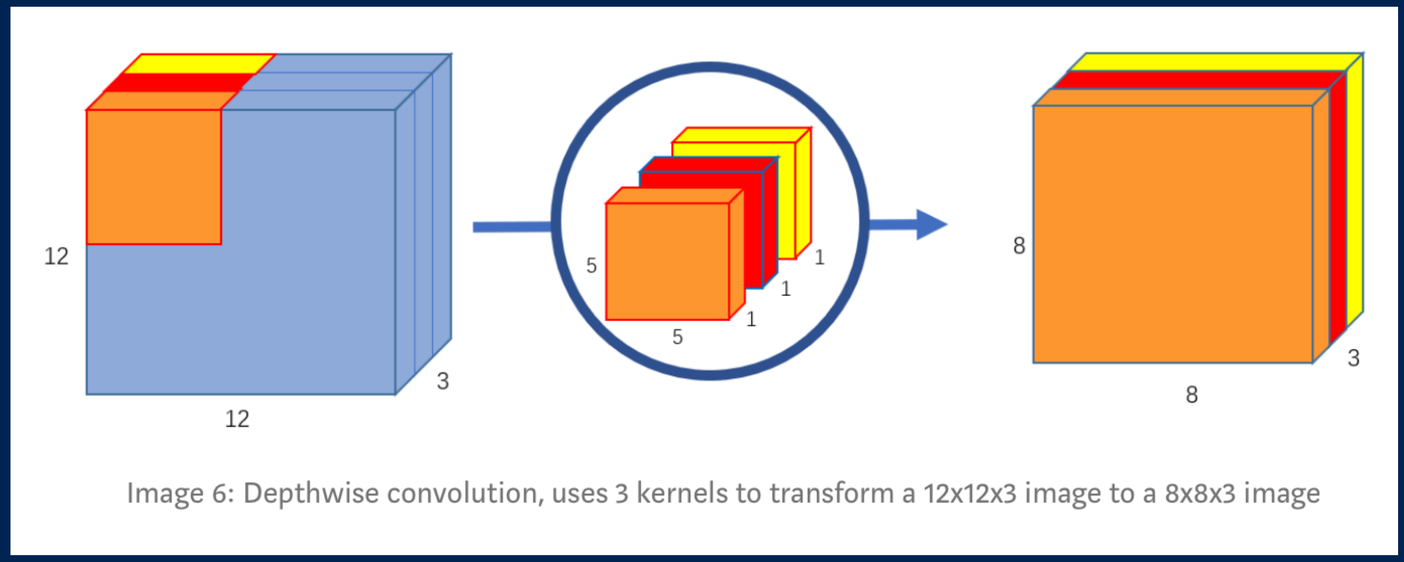
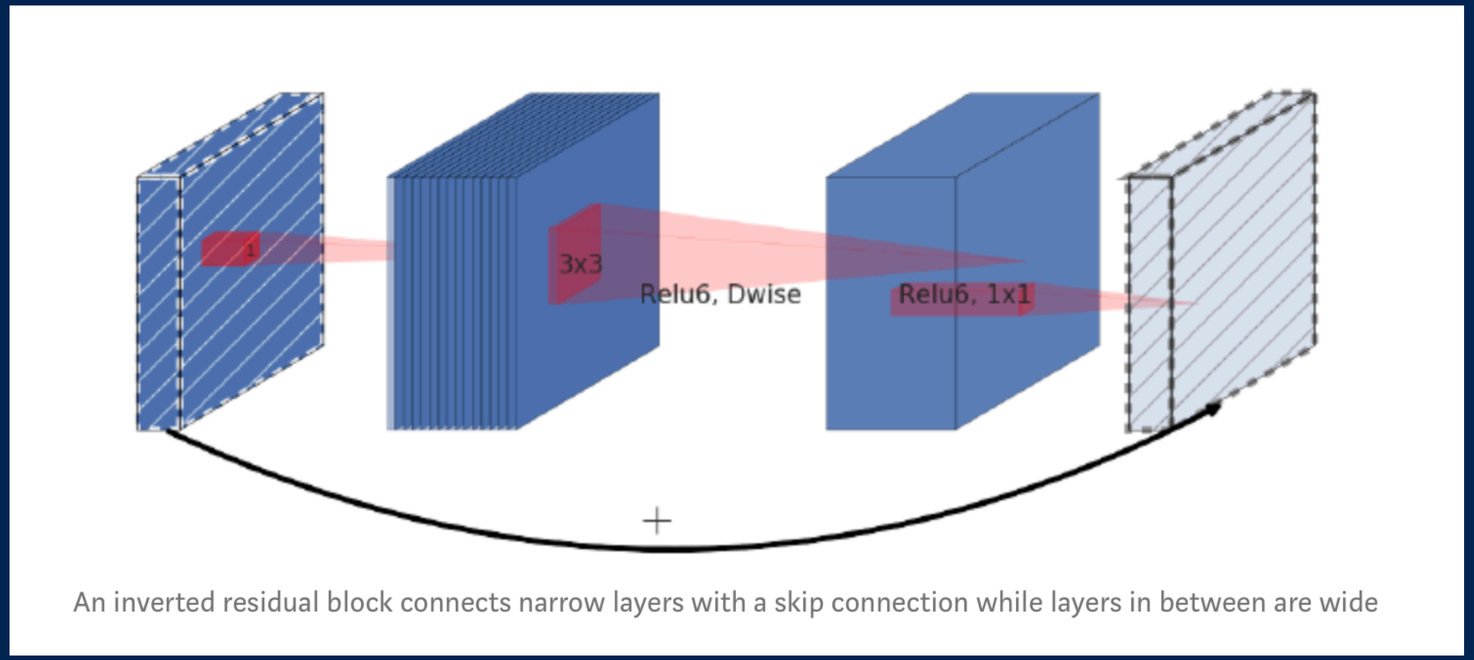


Image 6: Depthwise convolution, uses 3 kernels to transform a 12x12x3 image to a 8x8x3 image

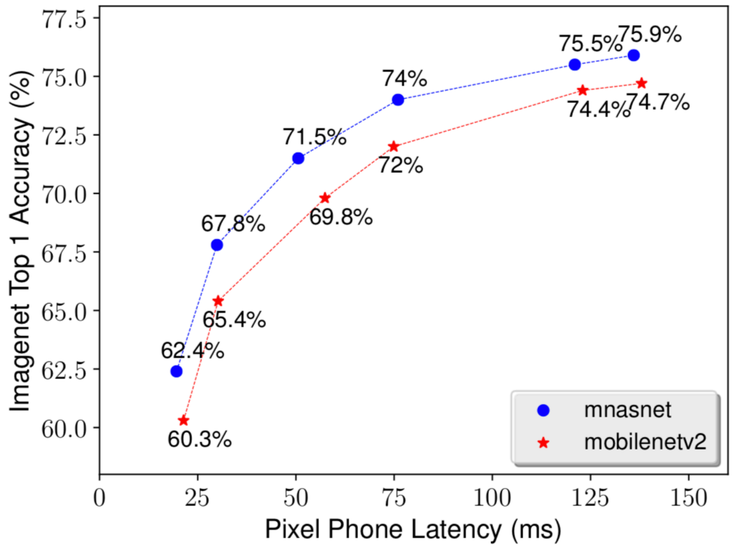
mobilenet v2

- 2018 talk: linear activations, inverted bottleneck layers
- towardsdatascience.com/mobilenetv2-inverted-residuals-and-linear-bottlenecks-8a4362f4ffd5

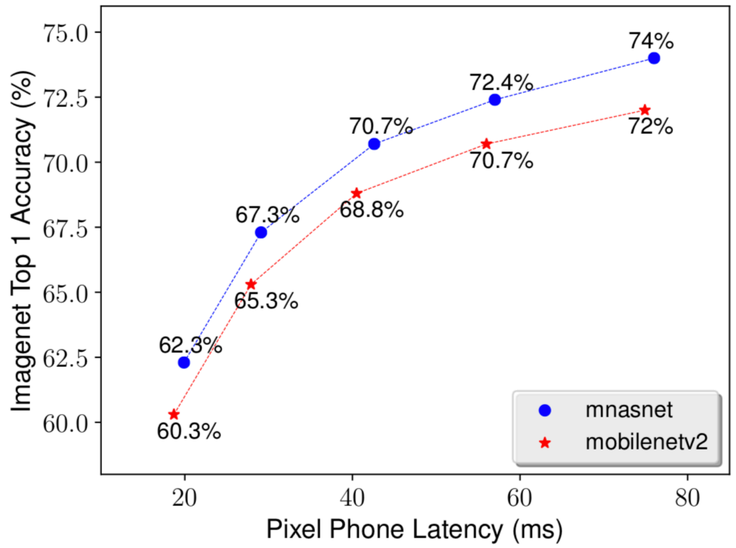


An inverted residual block connects narrow layers with a skip connection while layers in between are wide

mnasnet



(a) Depth multiplier = 0.35, 0.5, 0.75, 1.0, 1.3, 1.4, corresponding to points from left to right.



(b) Input size = 96, 128, 160, 192, 224, corresponding to points from left to right.

Figure 4: **Performance Comparison with Different Model Scaling Techniques**. MnasNet is our baseline model shown in Table 1. We scale it with the same depth multipliers and input sizes as MobileNetV2.

senet

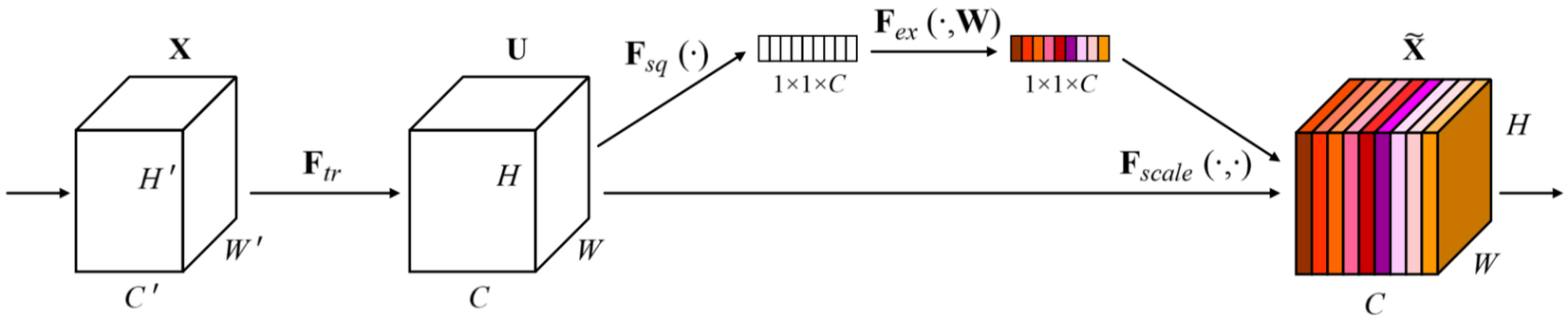


Figure 1: A Squeeze-and-Excitation block.

- squeeze + excitation networks (2018)
- https://www.robots.ox.ac.uk/~vgg/publications/2018/Hu18/presentation.pdf

efficientnet (may 2019)

- arxiv: 1905.11946

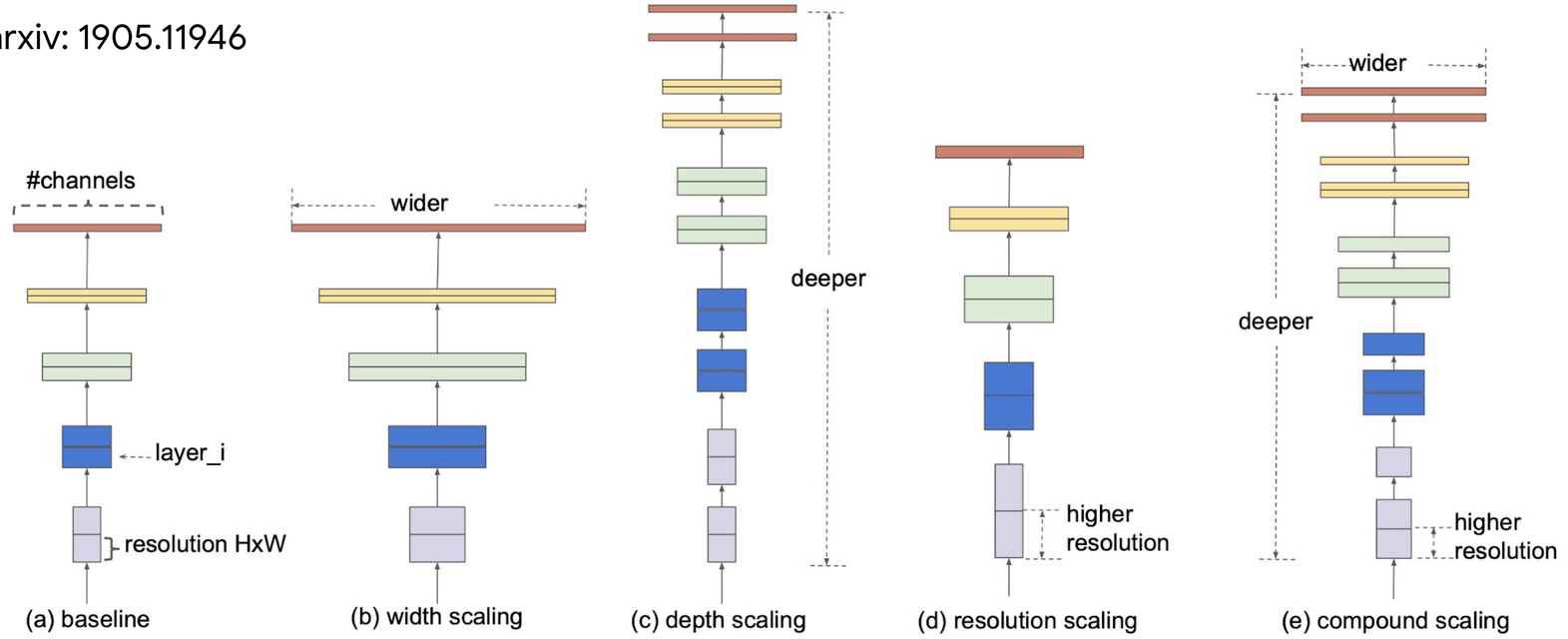
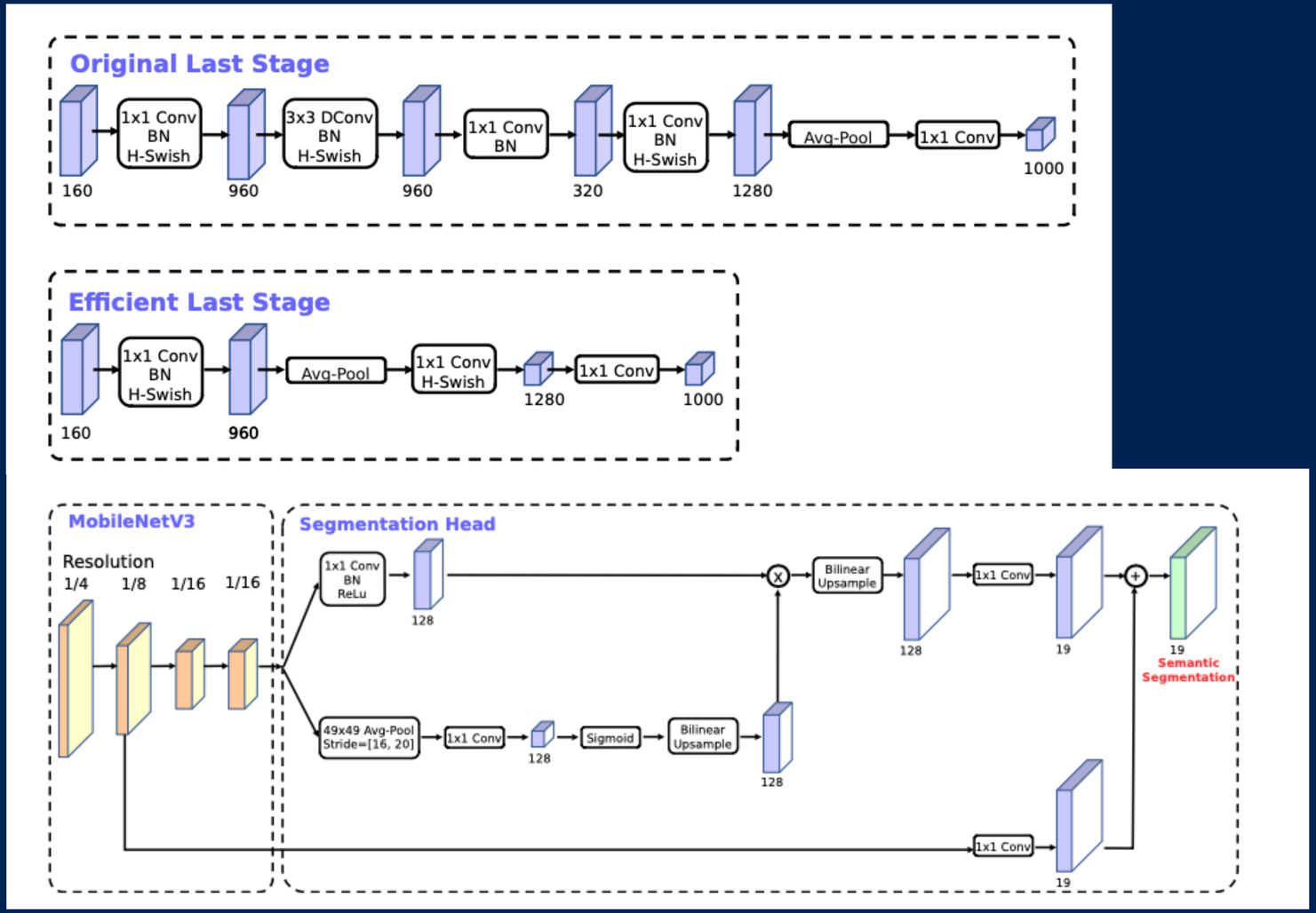


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

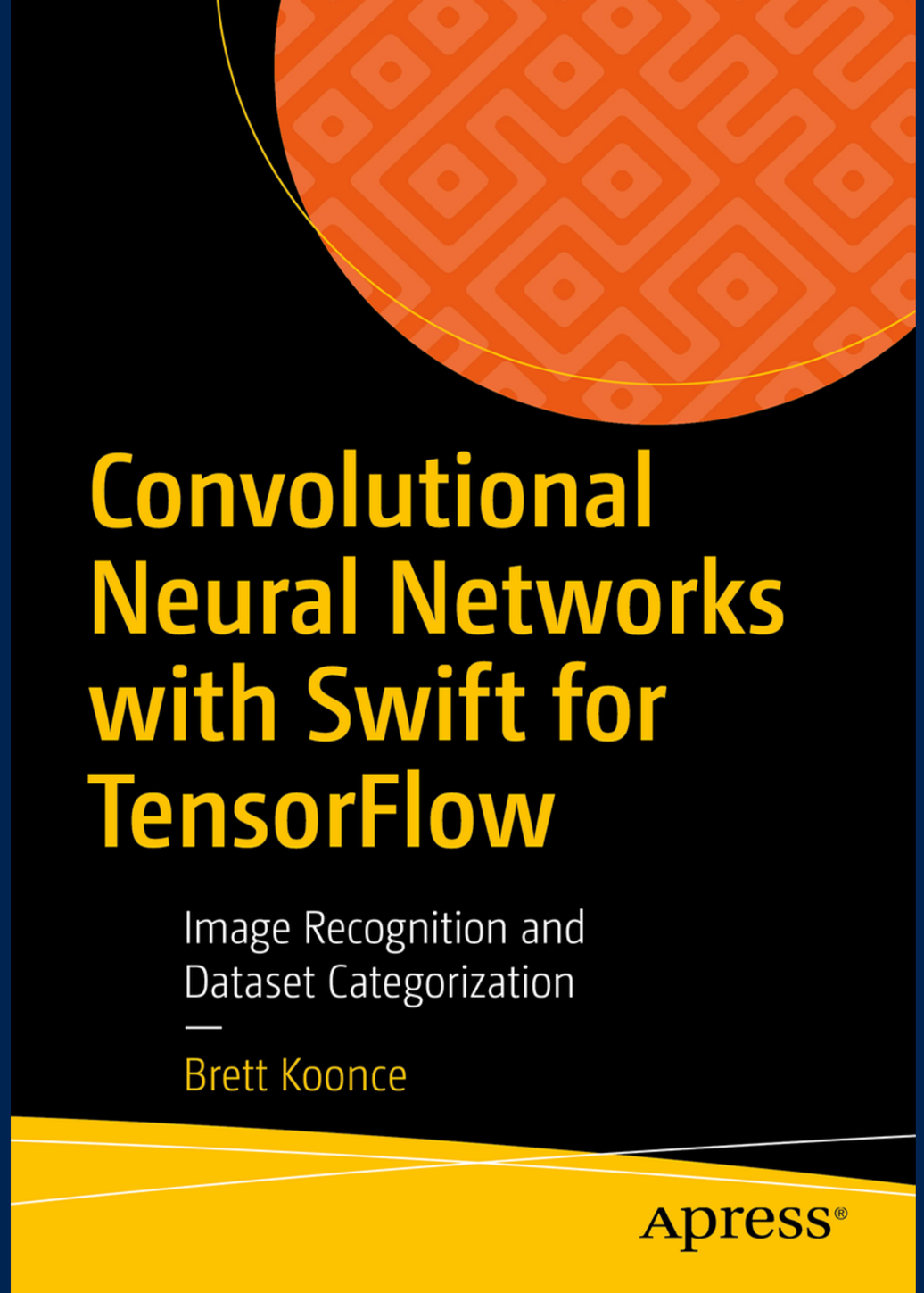
mobilenet v3



state of the art (arxiv)

- fb/instagram: 1905.00546
- randaugment: 1909.13719
- efficientdet: 1911.09070
- noisy teacher: 1911.04252


Larson
"Mr. Osborne, may I be excused?
My brain is full."



- **apress: aaron black, jessica valenti, james markham, vishwesh shrimali**
- **notes: james maki, brennan saeta**
- **parents, quarkworks**

wishlist 🎁

- checkpointing --> done!
- tpu training/scaling
- mixed precision training (bfloat16)
- data pipelines/augmentation

thanks for coming!