# Introduction to convolutional neural network for computer vision

Lijing Wang, DSSG tutorial session, 07/23/2021

# Icebreaker discussion 💻

- Name one exciting 😆 computer vision application you have used/heard of

# What is computer vision?

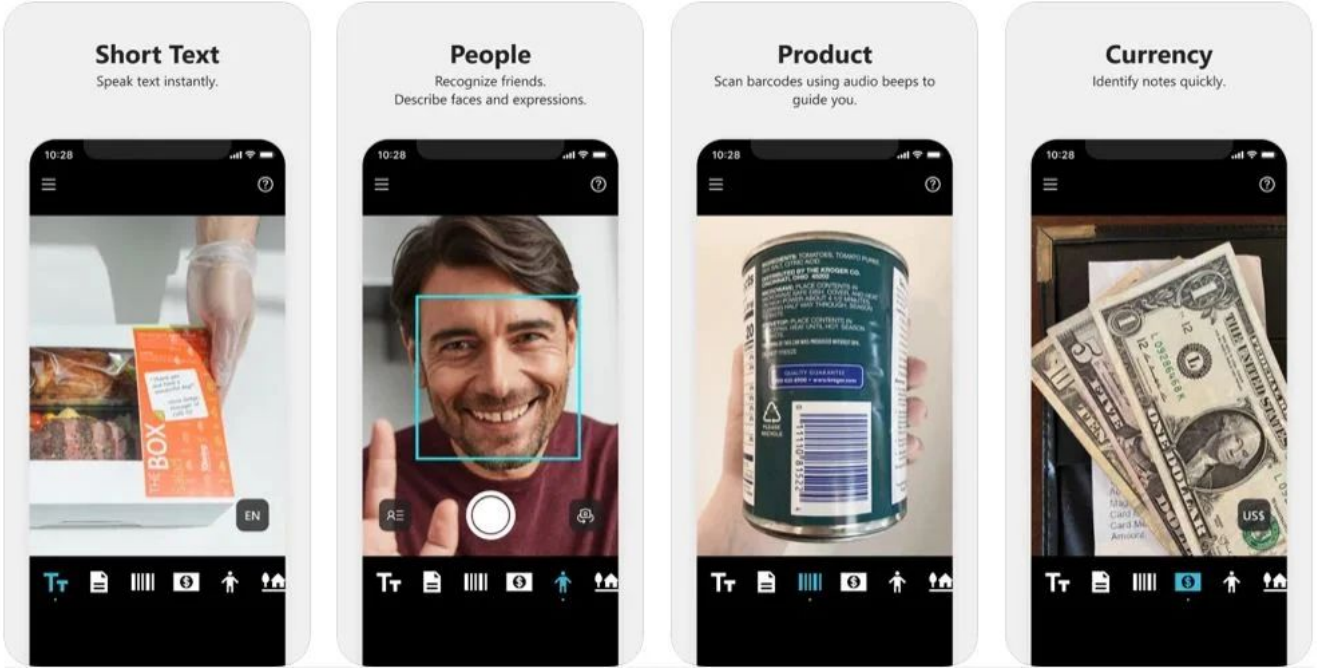
From wikipedia:

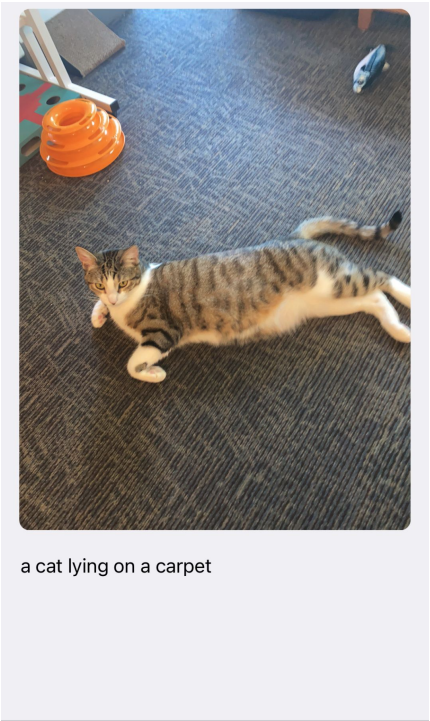
- how computers can gain high-level understanding from digital images or videos
- understand and automate tasks that the human visual system can do

# AI helps return Rembrandt’s The Night Watch (1642) to original size

# Seeing AI by Microsoft

## Seeing AI: Talking carema for the blind and low vision community



# Other exciting applications

- Self-driving car
- AI-assisted elderly care solution
- Crop yield and crop type mapping in remote area
- Face recognition
- Retailing: Amazon Go
- Image search
- …

# Basic computer vision tasks

- Image classification ← A Core Task in Computer Vision
- Object detection
- Image captioning
- Semantic segmentation

Object detection
car



Action recognition
bicycling

Time



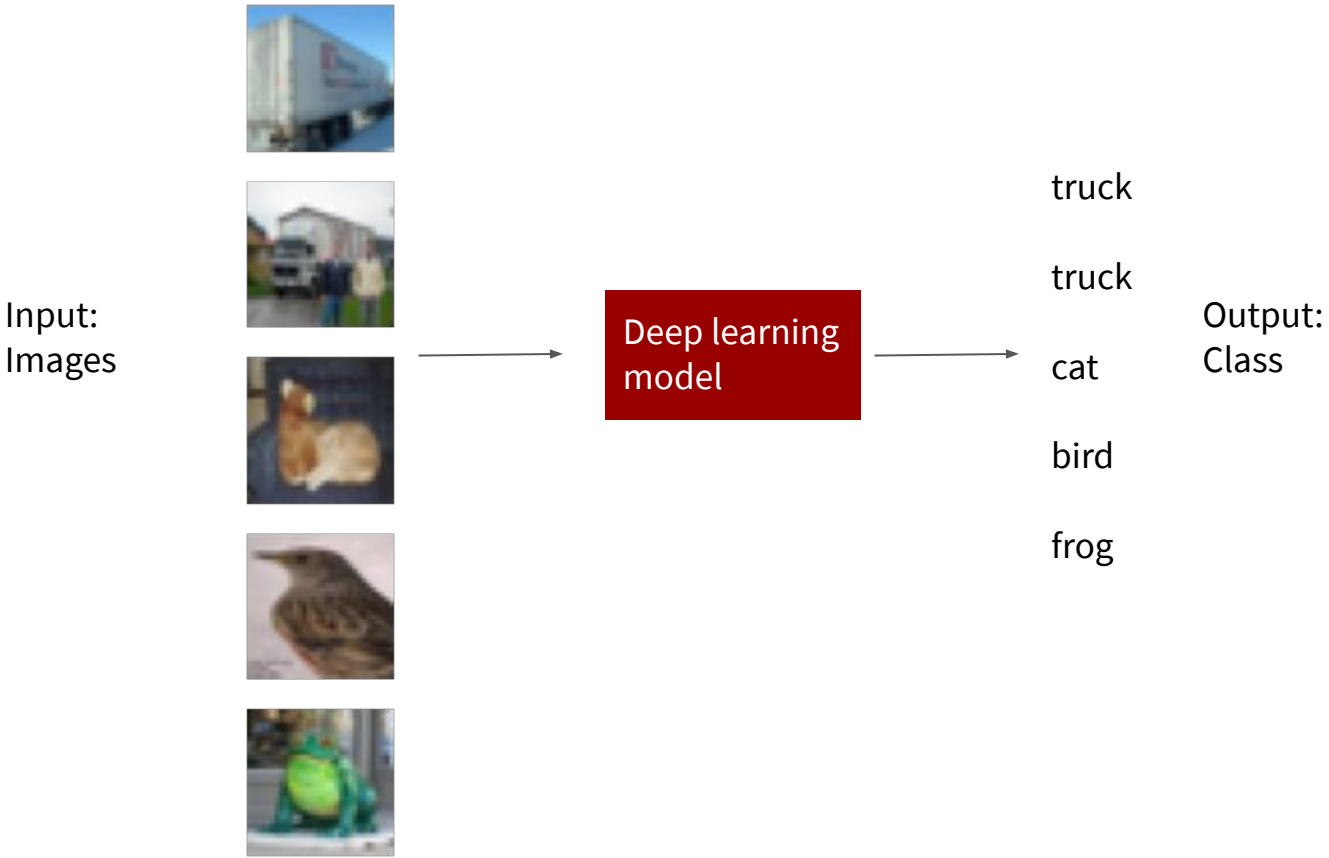
Scene graph prediction
<person - holding - hammer>

Captioning:
*a person holding a hammer*



Credit: http://cs231n.stanford.edu/slides/2021/lecture_1.pdf

# Image classification: CIFAR-10



# Today’s learning goals

1. Understand how the convolutional neural networks (CNN) work
2. Implement CNN using Tensorflow and Keras
3. Train your CNN on CIFAR-10
4. Visualize the performance of your trained CNN
5. [Optional] Transfer learning

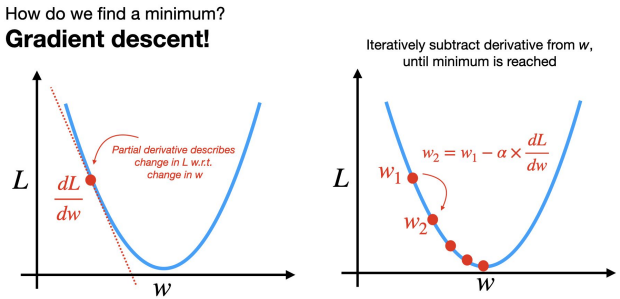
# Recap: deep learning basics by Armin

Perceptron:

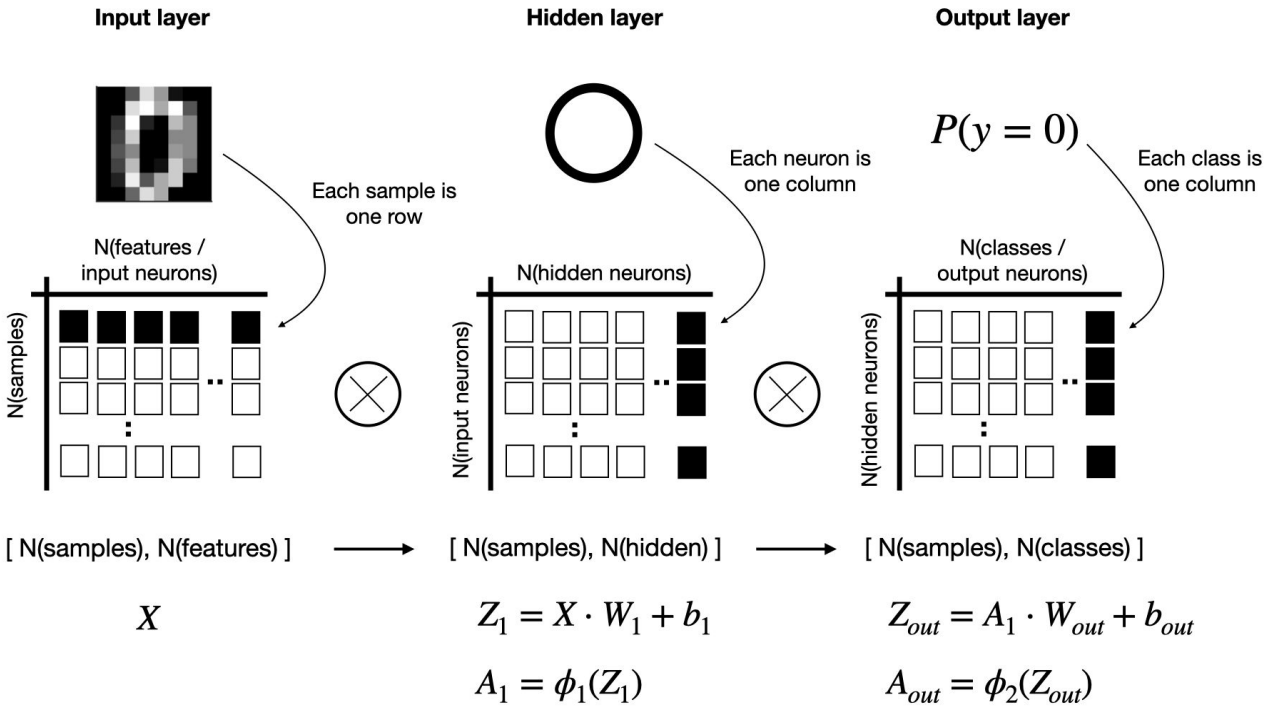
- Loss function: binary cross-entropy
- Gradient descent: learning rate, batch size

Artificial neural networks (ANN): a sequence of multiple perceptrons

- Activation functions: sigmoid, tanh, relu ...
- Backpropagation: enable the gradient descent in ANN



# Artificial Neural Network




Credit: Armin's dssg-dl-tutorial notebooks

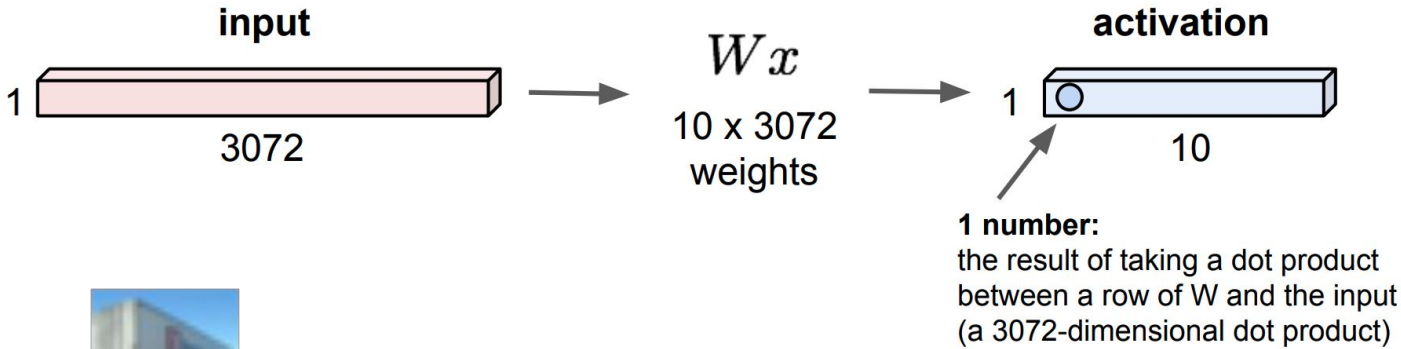
# Why do we need CNN in computer vision?

Using the ANN, without convolutional neural networks:

How many parameters do we need?

10 x 3072 ~ 30k

32x32x3 image -> stretch to 3072 x 1




Input: 32 x 32 x 3

**Each neuron looks at the full input volume**

Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# Why do we need CNN in computer vision?

Using ANN (**fully connected neural network**):

- Many parameters to be trained
- Ignore the spatial structures of digital images or videos

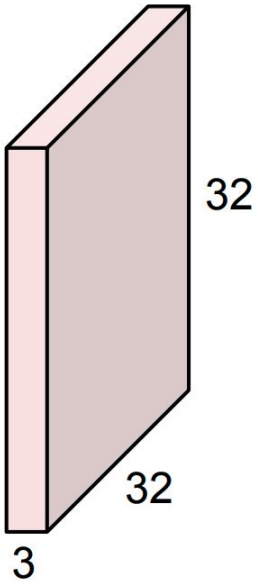
Input: 32 x 32 x 3

Using convolutional neural networks:

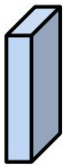
- Few parameters than ANN
- Preserve the spatial structure and take advantage of local spatial coherence

# Convolution layer

32x32x3 image




5x5x3 filter



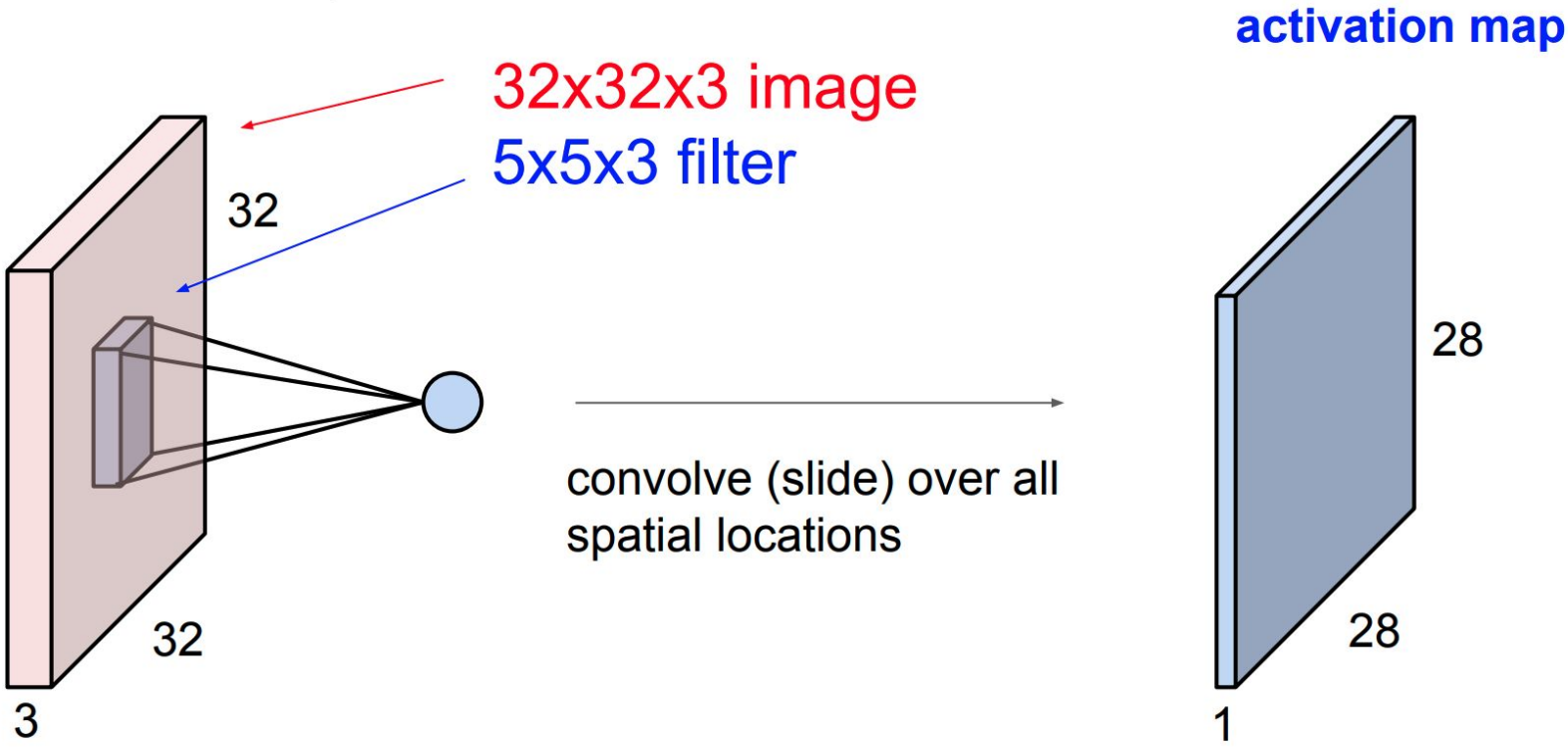
**Convolve** the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

How many parameters do we need?

5 x 5 x 3 = 75

# Apply the convolution kernels/filters




Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# Comparison between ANN and CNN

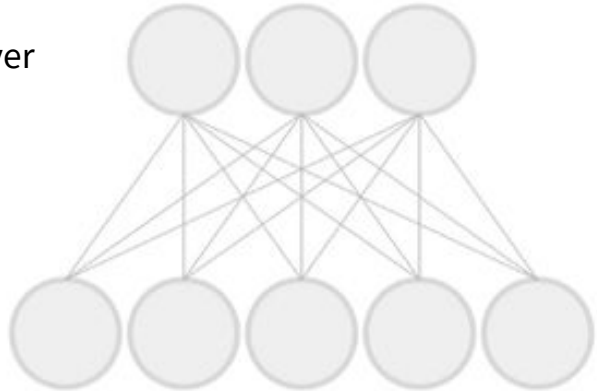
Input: 32 x 32 x 3

Every connection is one new parameter

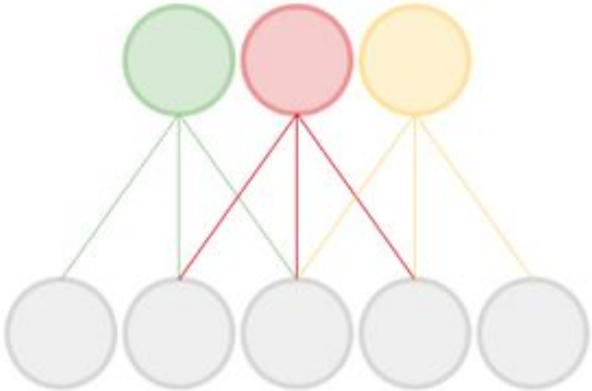
Shared filter parameters

Hidden layer

Input



Fully connected layer

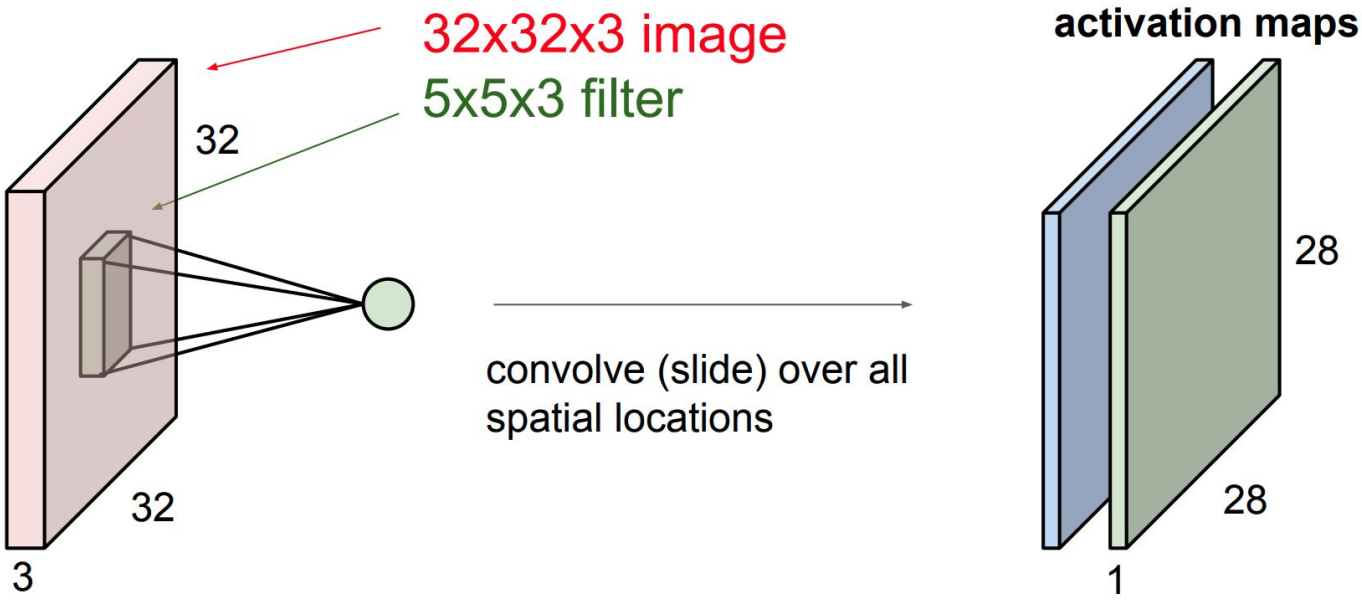


Convolutional layer

Credit: https://www.quora.com/What-are-the-advantages-of-a-convolutional-neural-network-CNN-compared-to-a-simple-neural-network-from-the-theoretical-and-practical-perspective

# Multiple convolution filters

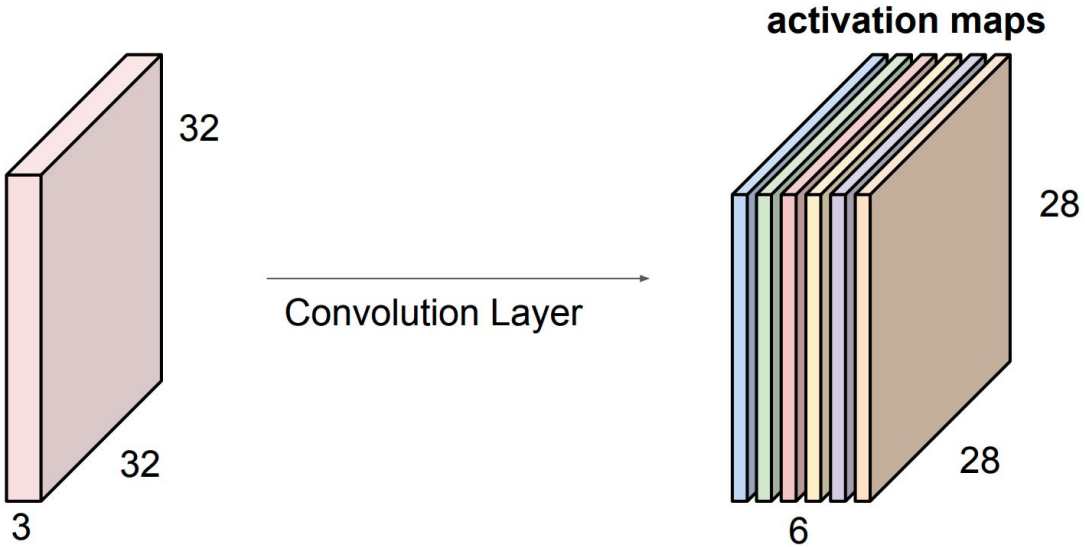
Another green filter




Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# Multiple convolution filters

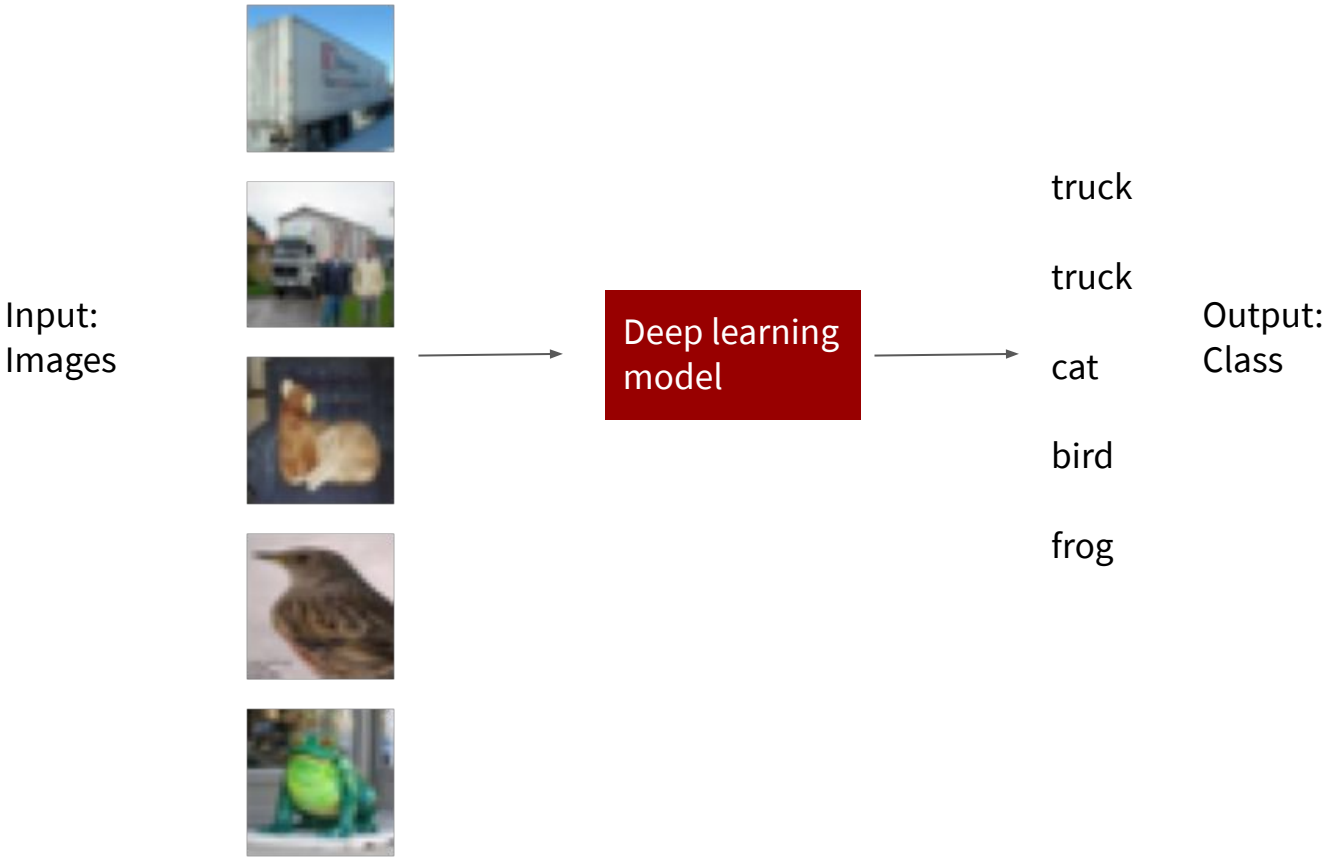
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:




We stack these up to get a "new image" of size 28x28x6!

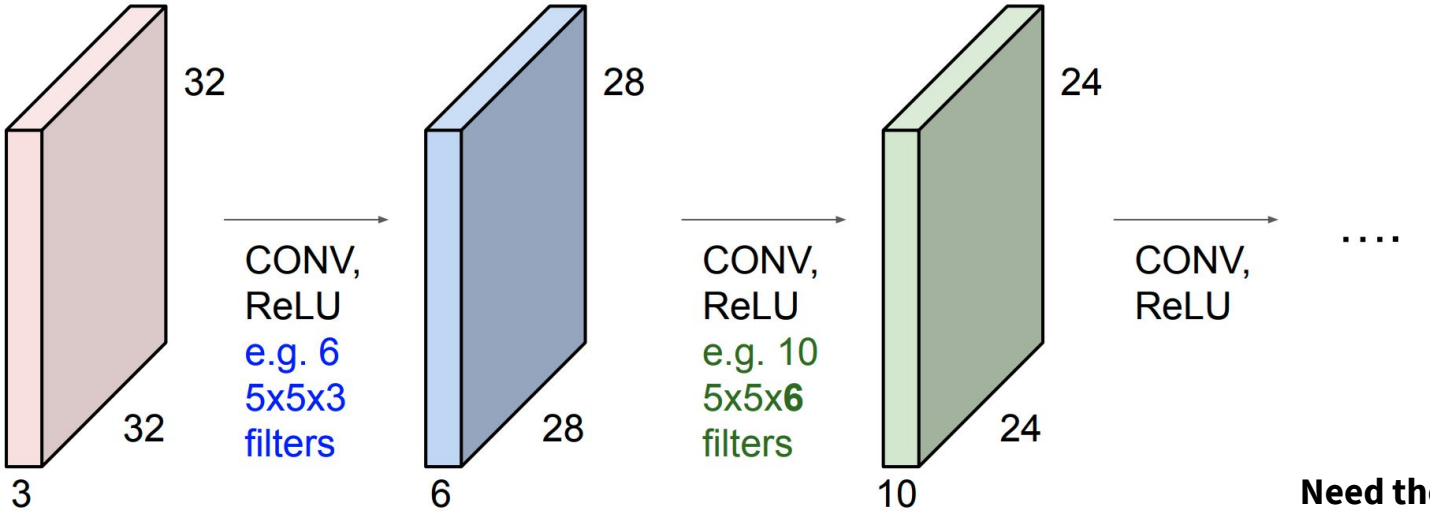
Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# Recall: Image classification, CIFAR-10



# ConvNets: a sequence of convolutional layer

- Stack 1) convolutional layer, 2) activation function, 3) **pooling layer**
- Without using the pooling layer



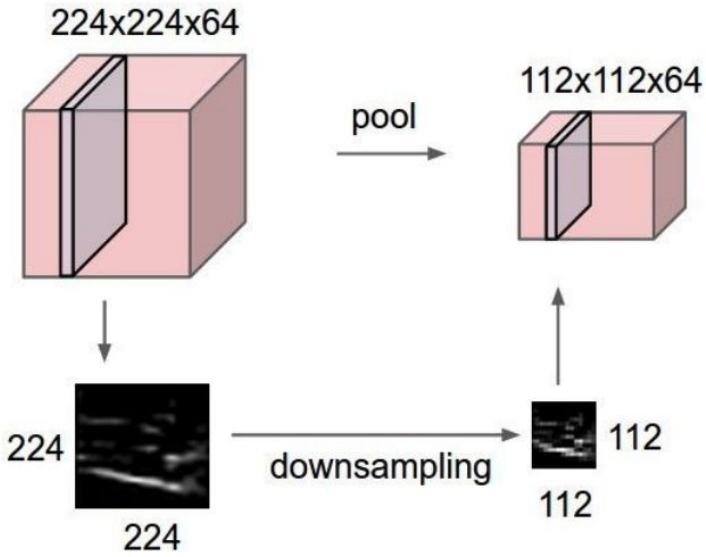

Output: the score for being each class

Dimension: 10

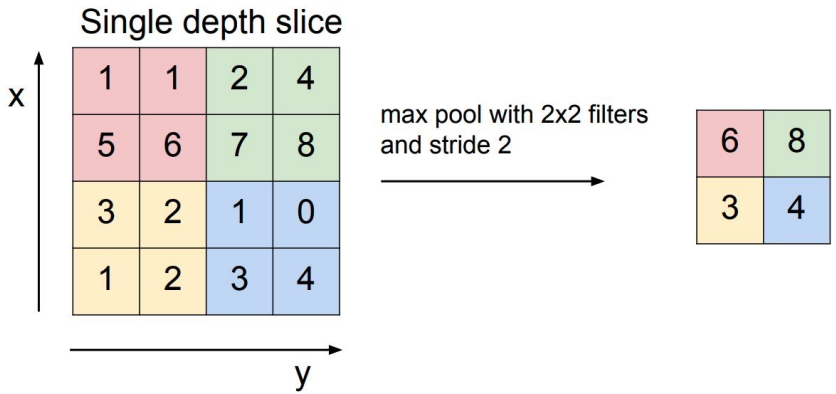
**Need the downsampling!**

Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# Pooling layer: downsampling




Max pooling

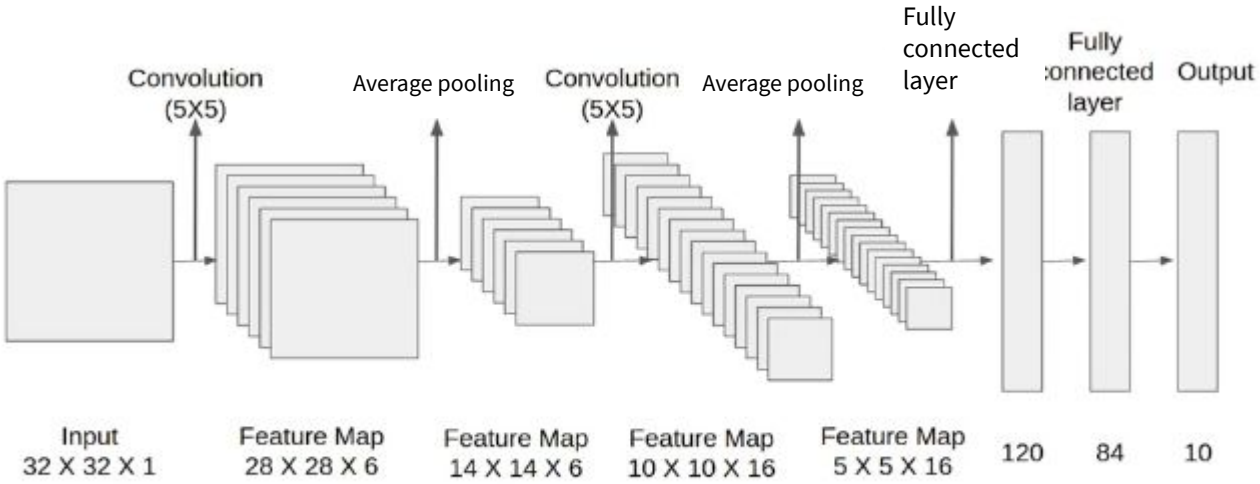



Stride: the amount of movement for that filter

Credit: http://cs231n.stanford.edu/slides/2021/lecture_5.pdf

# One simple ConvNets architecture: LeNet

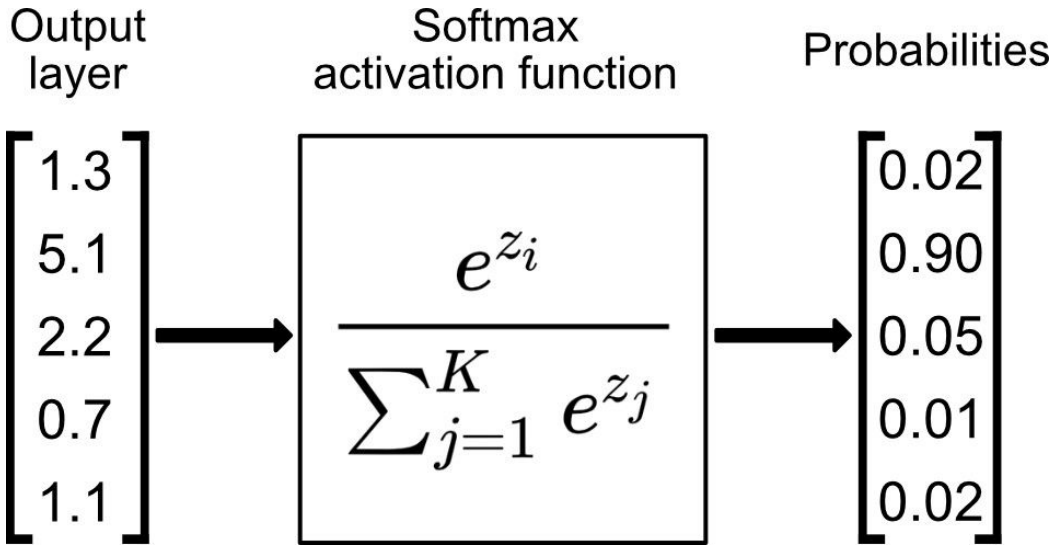
Yann LeCun et al. 1989




Credit: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/

# Softmax activation function:

- the last activation for the classification task: Exponential + Normalize

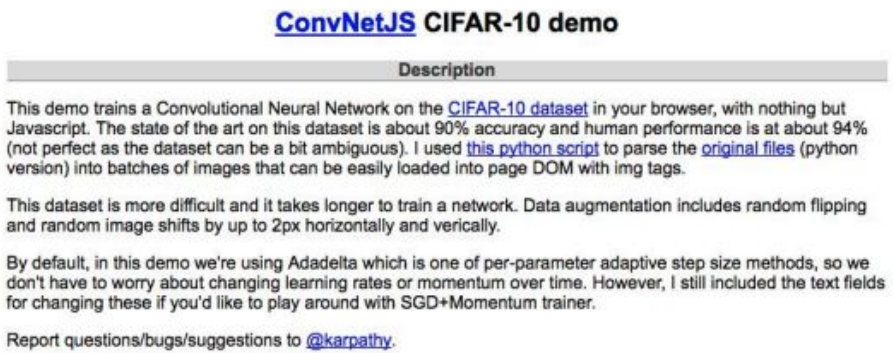


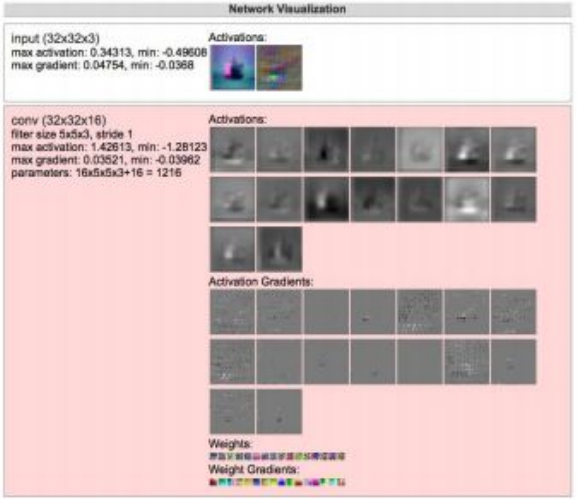

Probability for each class

Credit: https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60

# ConvNetJS CIFAR-10 demo

https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html



# Start with our first CNN!

Colab notebook: https://colab.research.google.com/drive/1ZtAebRezPJ00LGavIpgCK7L7Vo5FjHVJ?usp=sharing

- How do the convolution kernels/filters work?
- How does the pooling layer work?
- The classification task on CIFAR-10!

# Transfer learning

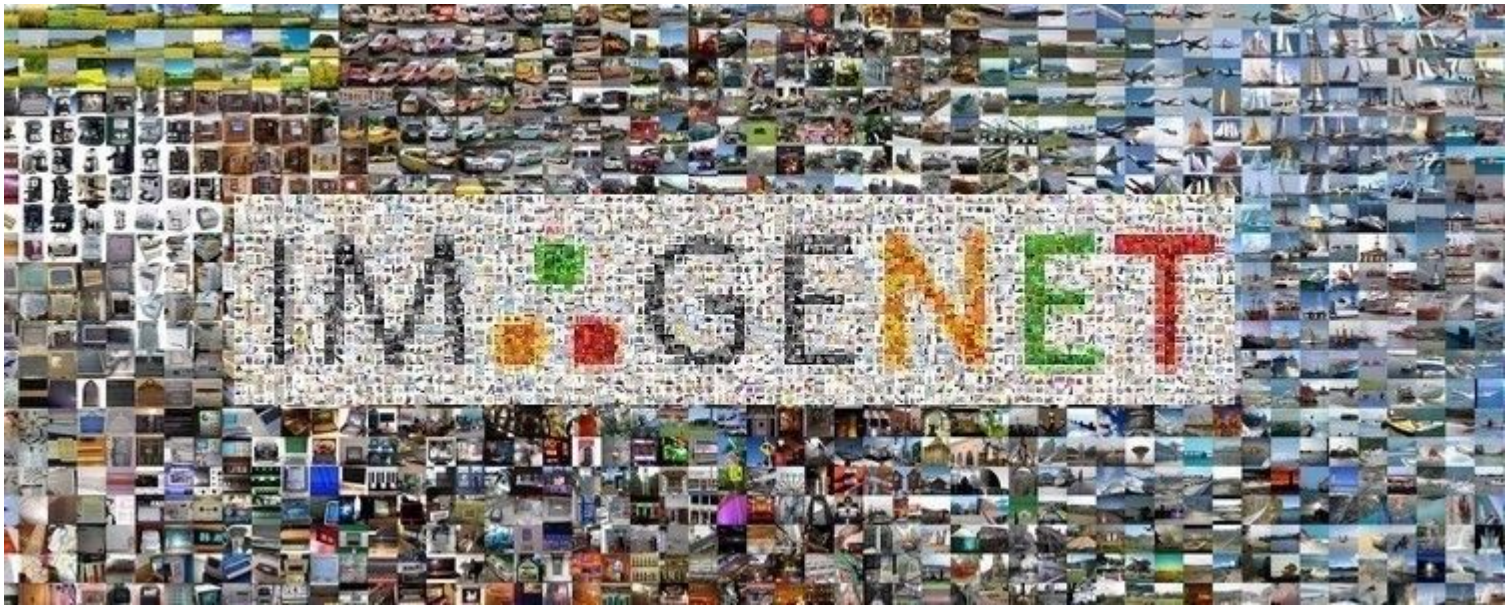
- In practice, very few people train an entire ConvNets from scratch (with random initialization), because it is relatively rare to have a dataset of sufficient size.

- In our GSV team, we don’t have enough labeled google street view images to train our ConvNets.


Source: AI & Deep Learning Memes For Back-propagated Poets

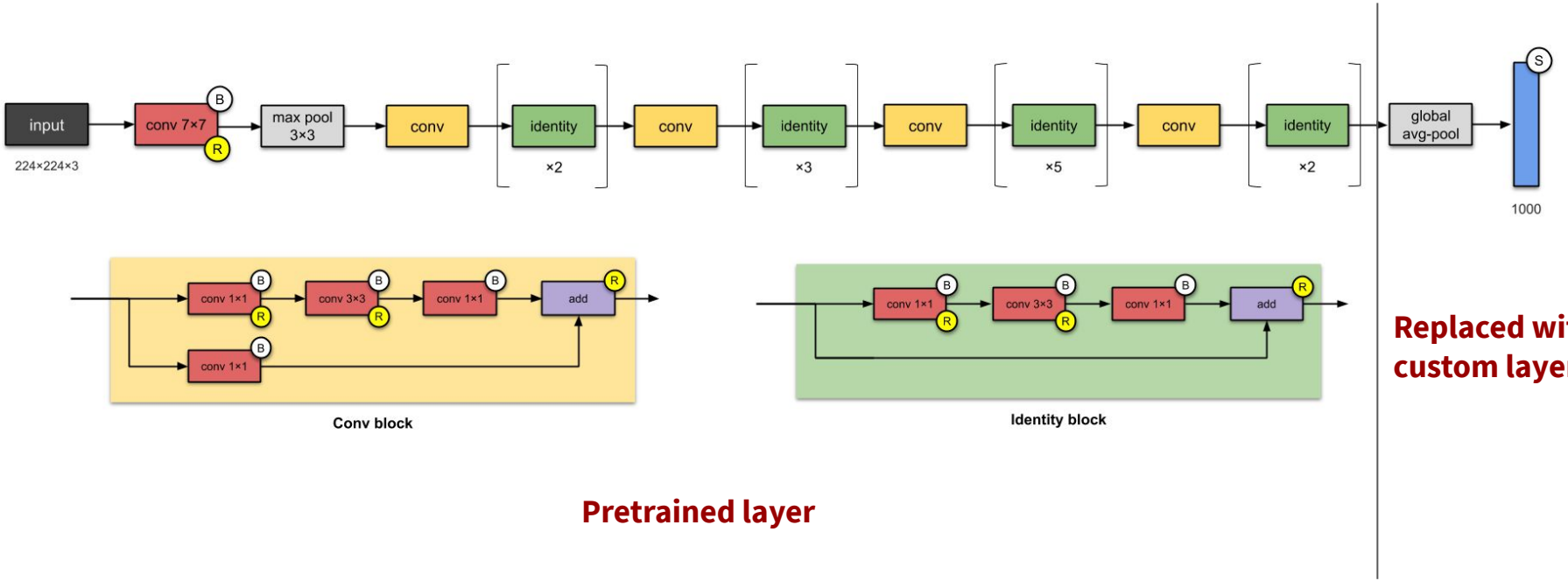
# Feature extractor: training on ImageNet

- More than 14 million images have been hand-annotated
- More than 20,000 categories




Source: https://image-net.org/

# Pretrained on ImageNet with a fancy architecture: ResNet-50

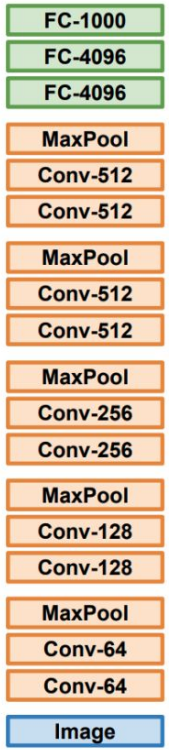



Credit: https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d

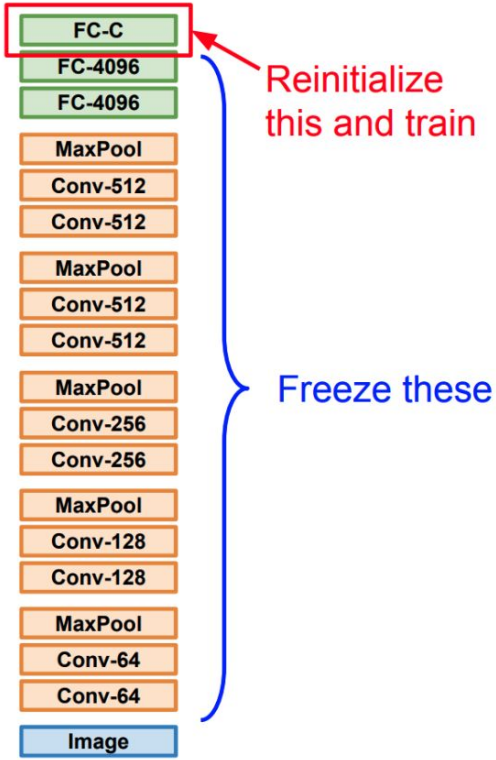
# Transfer Learning with CNNs

Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014

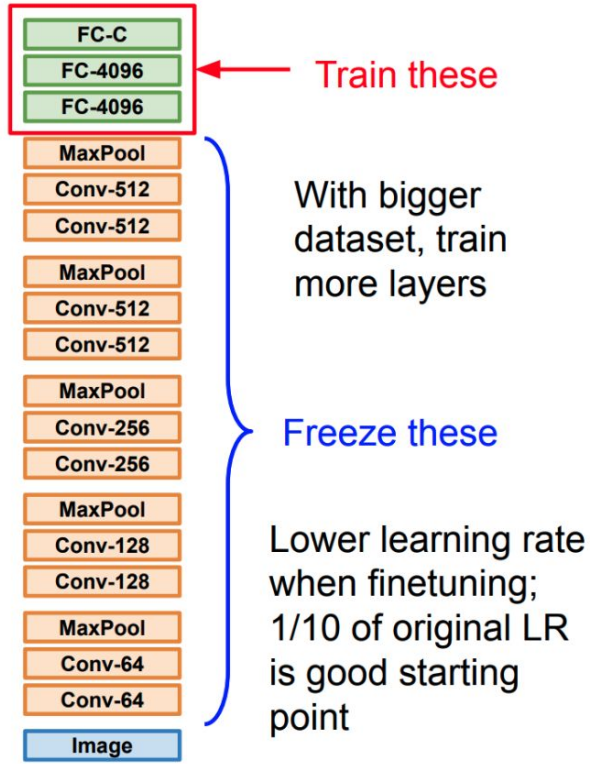
## 1. Train on Imagenet




## 2. Small Dataset (C classes)




## 3. Bigger dataset **Finetuning**

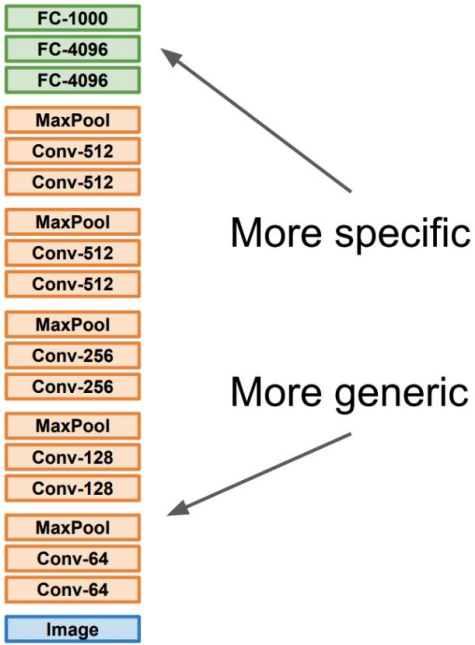



Credit: http://cs231n.stanford.edu/slides/2021/lecture_7.pdf

# Do your own transfer learning:
# stand on the shoulder of the giants

Colab notebook, transfer learning part:
https://colab.research.google.com/drive/1ZtAebRezPJ00LGavIpgCK7L7Vo5FjHVJ?usp=sharing

# Discussion on transfer learning




| | **very similar dataset** | **very different dataset** |
|---|---|---|
| **very little data** | Use Linear Classifier on top layer | You're in trouble… Try linear classifier from different stages |
| **quite a lot of data** | Finetune a few layers | Finetune a larger number of layers |

Credit: http://cs231n.stanford.edu/slides/2021/lecture_7.pdf

# Takeaways for your current/future computer vision project

1. Train a model on a large dataset (ImageNet) that similar to your task (CIFAR-10)
2. Transfer learning to your task (CIFAR-10)

# Reference:

- Stanford CS231N, Convolutional Neural Networks for Visual Recognition: http://cs231n.stanford.edu/
  - Course video 2017: https://www.youtube.com/watch?v=vT1JzLTH4G4
- Stanford Vision and Learning lab: http://svl.stanford.edu/
- Data for Sustainable Development: https://ermongroup.github.io/cs325b/