
Computer vision

Computer Vision

Lecture 6: Object Recognition II

Dr. Dina Khattab
dina.khattab@cis.asu.edu.eg
Scientific Computing Department

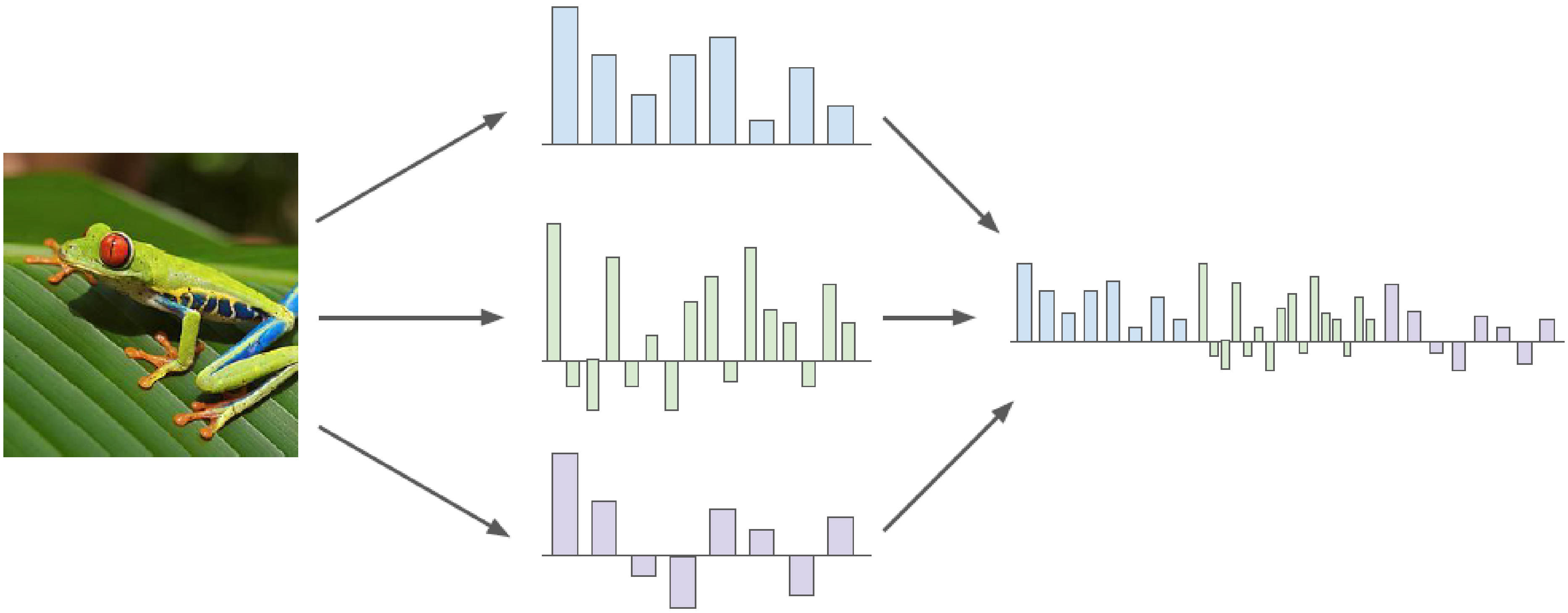
Instructor:	Dr. Dina Khattab
Email:	dina.khattab@cis.asu.edu.eg
Office:	Main Building – 4th floor – Room 302
Office Hours:	Monday 12:00 - 2:00 PM Thursday 11:00 AM to 12:00 PM

Agenda

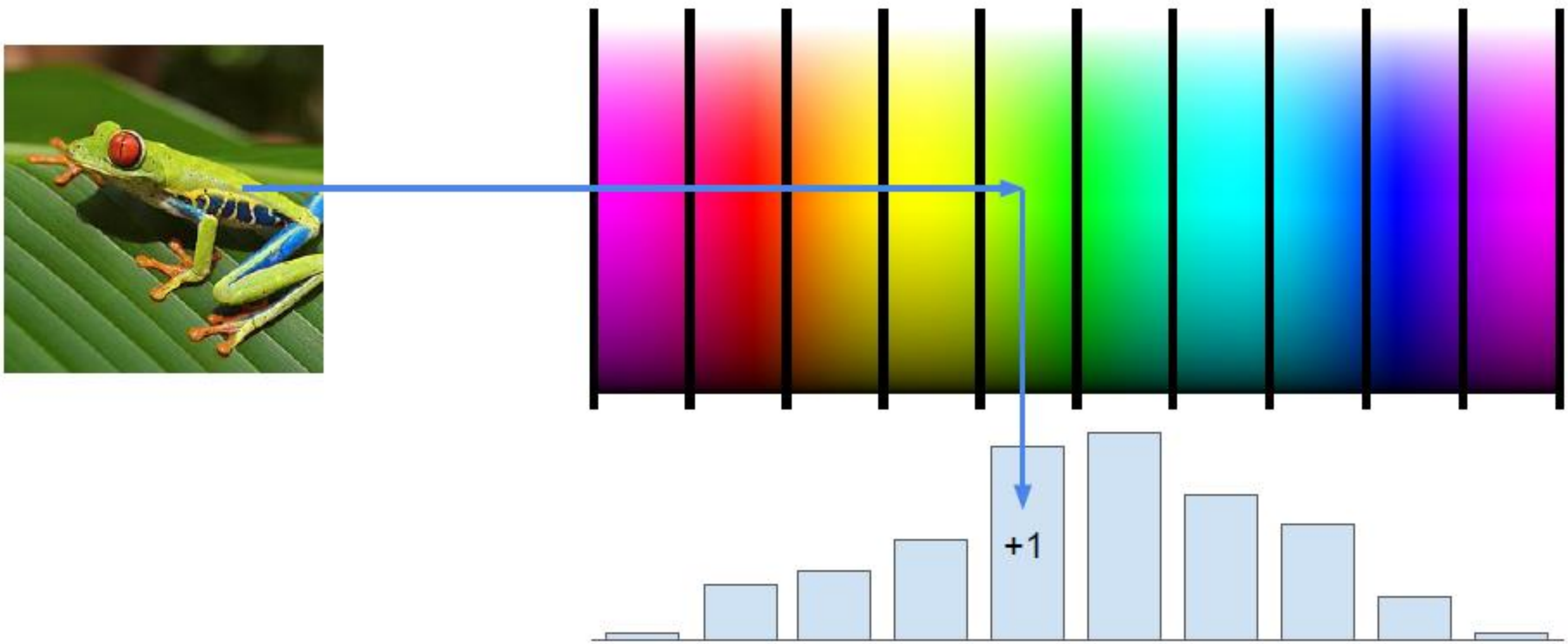
- Object Recognition
  - Feature extraction Vs. ConvNets
  - Recap on CNNs
  - CNN architectures

- Applications:
  - Content Based Image Retrieval (CBIR)
  - Face Verification

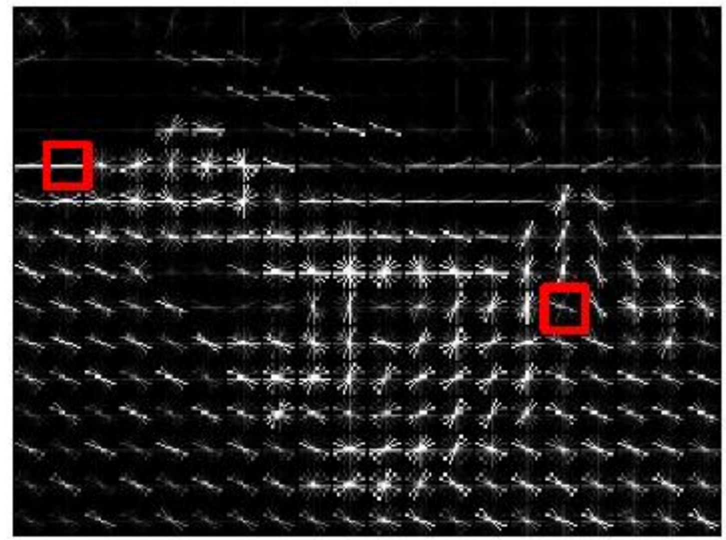
Image Recognition



Example: Color Histogram



Example: Histogram of Oriented Gradients (HoG)



Divide image into 8x8 pixel regions Within each region quantize edge direction into 9 bins

Dalal and Triggs, "Histograms of oriented gradients for human detection," CVPR 2005

Example: Bag of Words

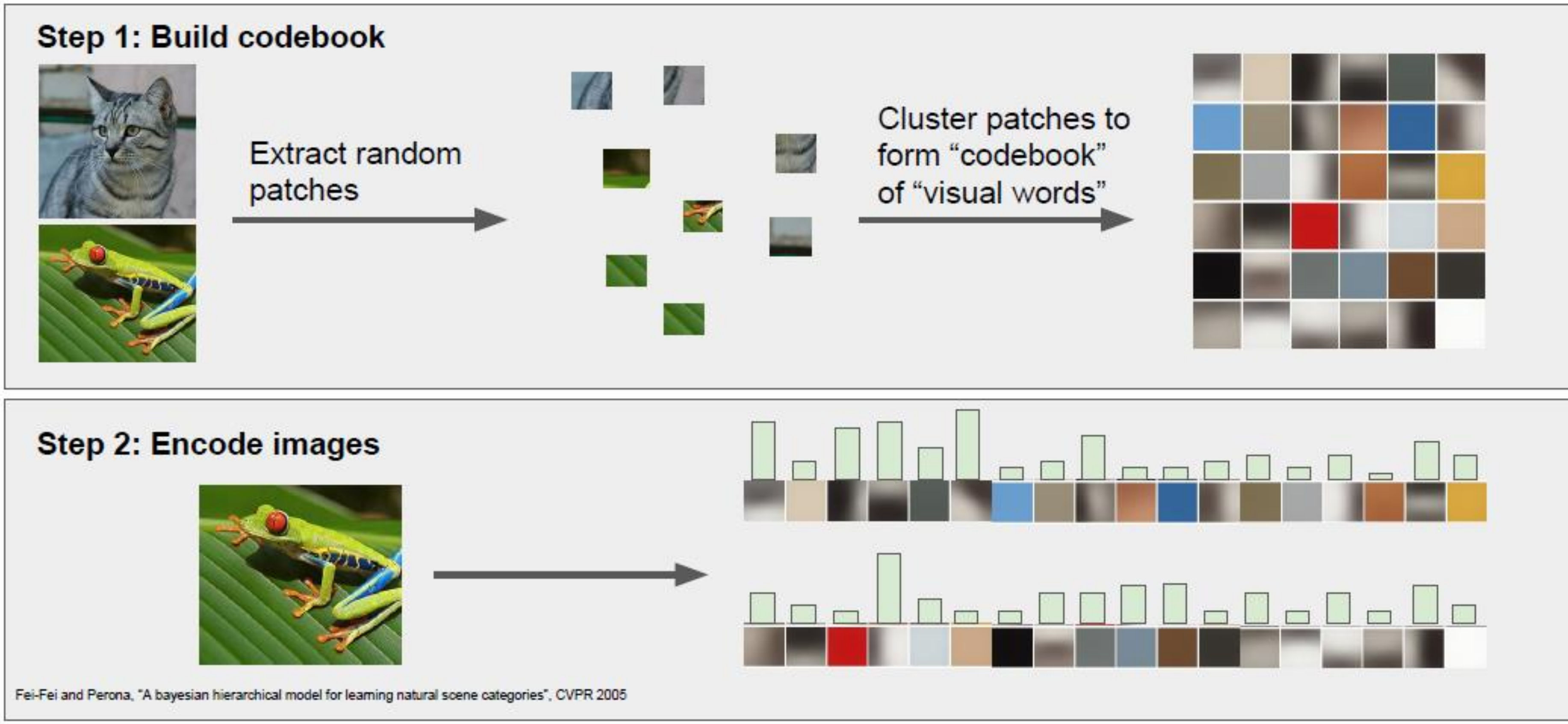
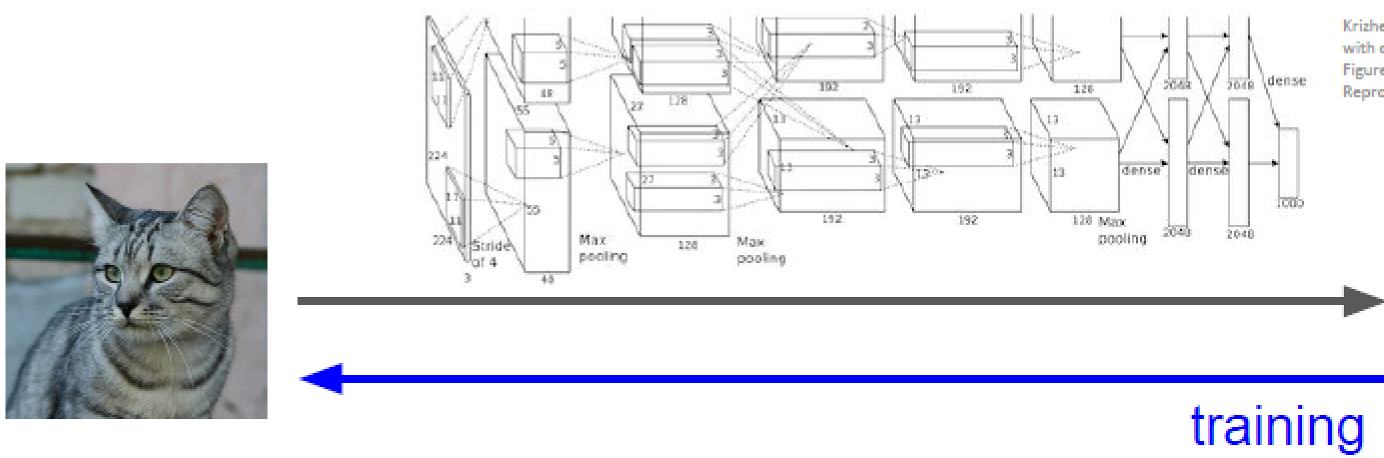


Image features vs ConvNets

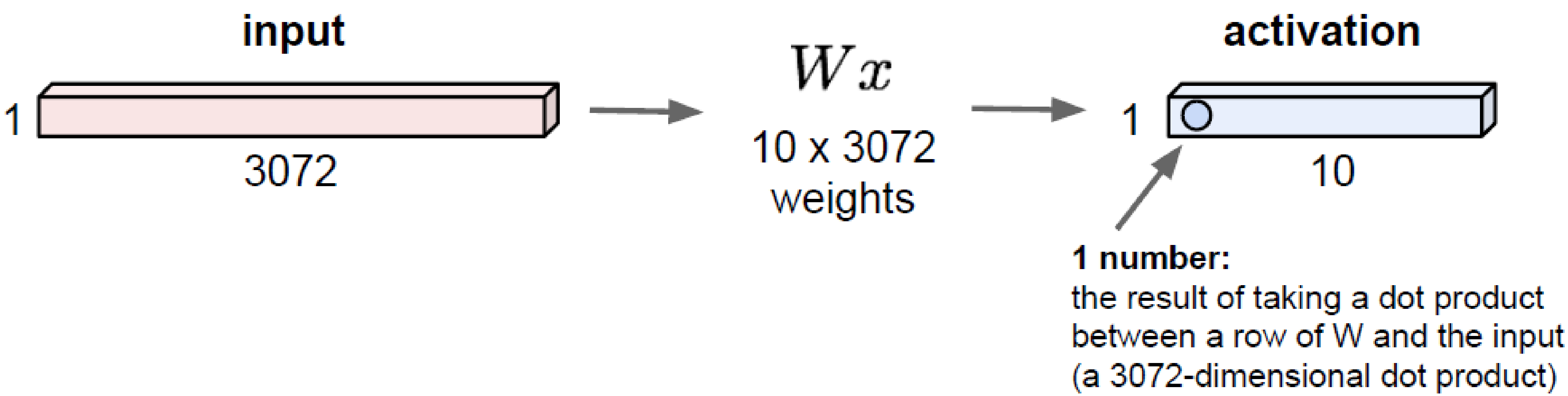


Krizhevsky, Sutskever, and Hinton, "Imagenet classification with deep convolutional neural networks", NIPS 2012.

RECAP ON CNNS

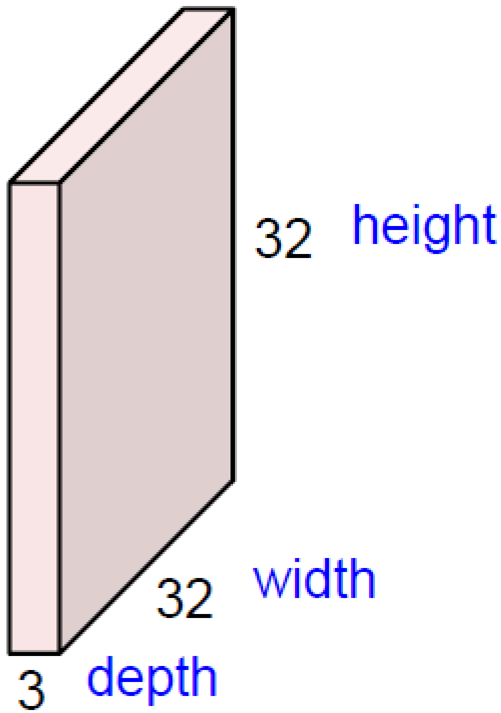
Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1

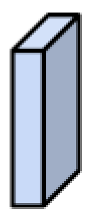


Convolution Layer

32x32x3 image -> preserve spatial structure



5x5x3 filter

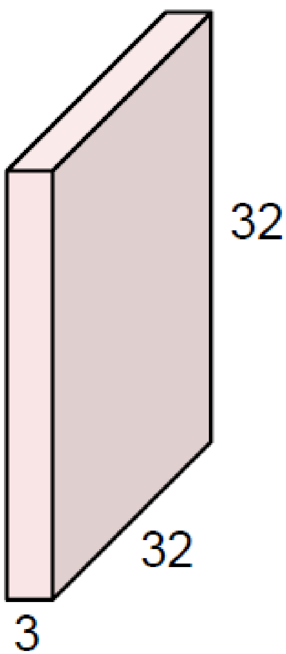


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

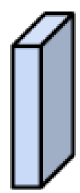
Convolution Layer

Filters always extend the full depth of the input volume

32x32x3 image



5x5x3 filter

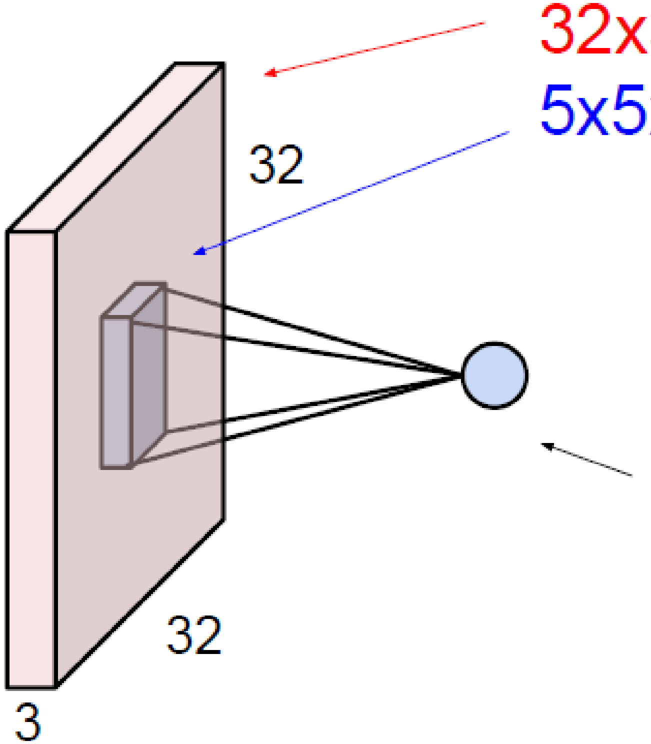


Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

Convolution Layer

$32 \times 32 \times 3$ image

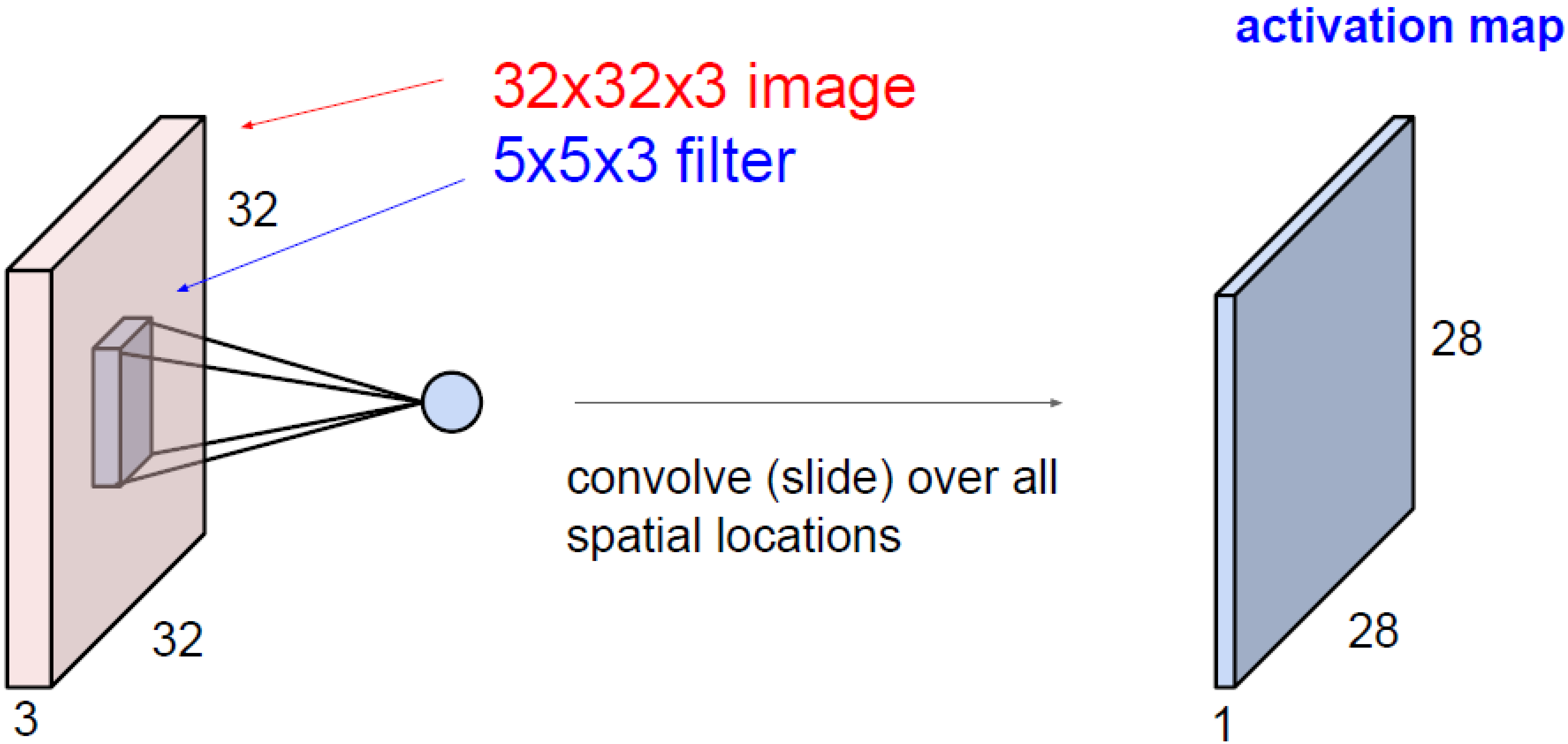
$5 \times 5 \times 3$ filter w



1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

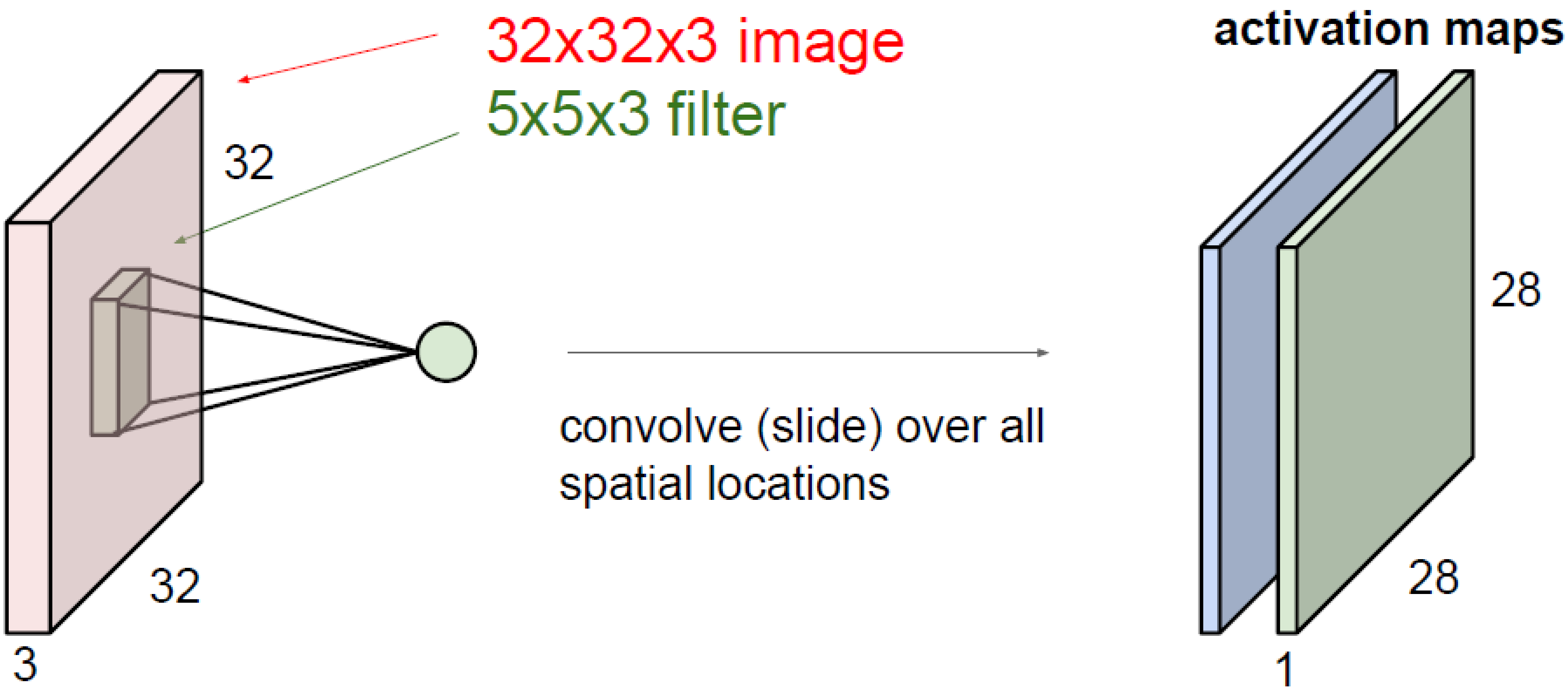
$$w^T x + b$$

Convolution Layer



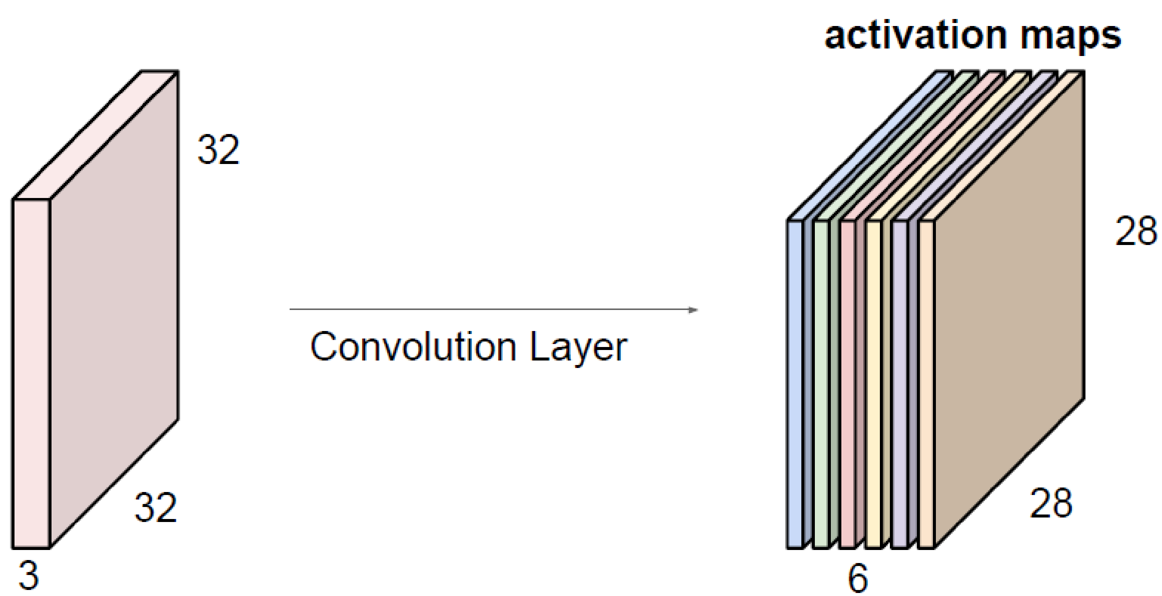
Convolution Layer

consider a second, green filter



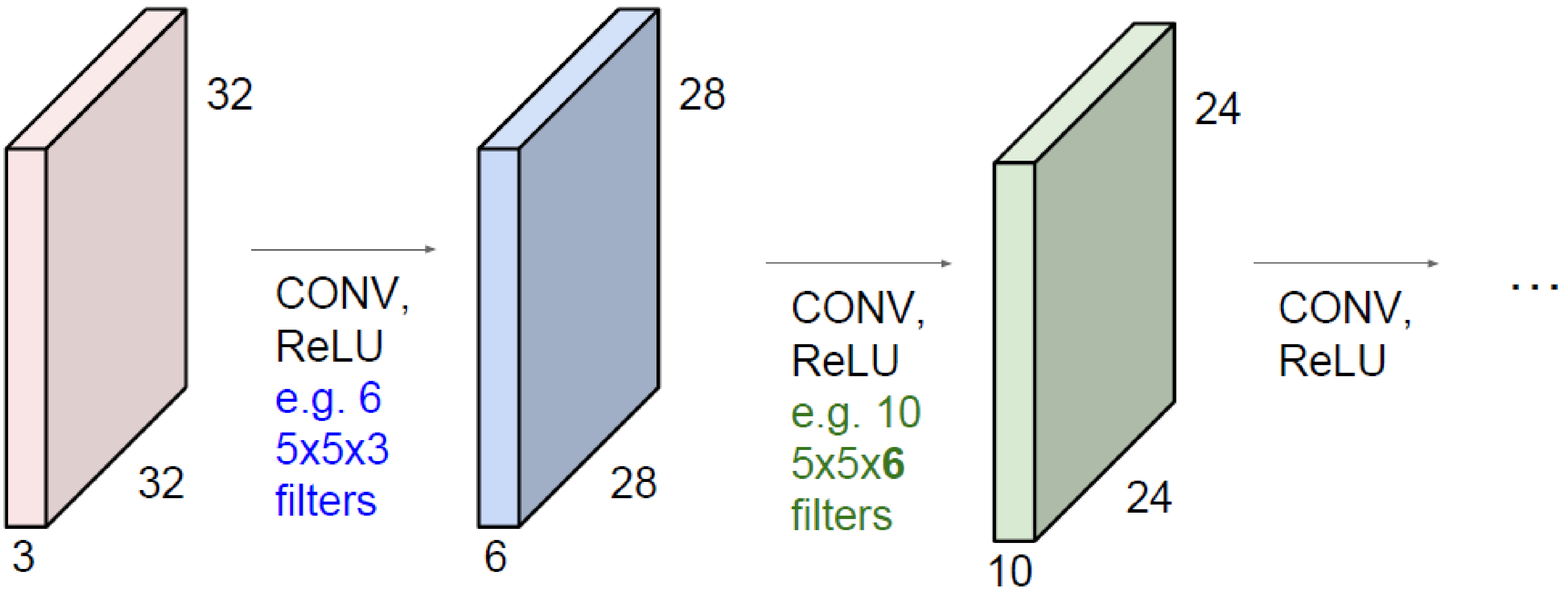
Convolution Layer

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

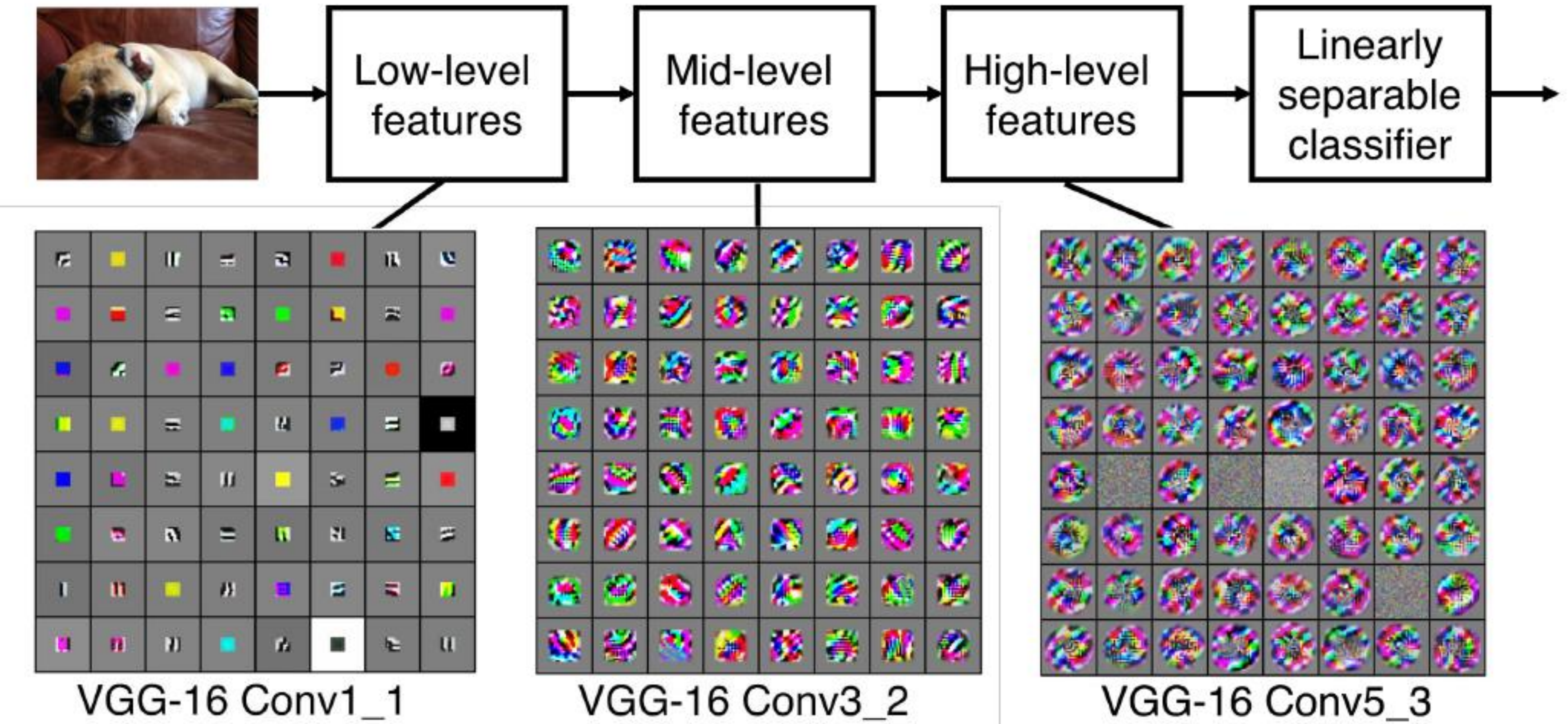


We stack these up to get a "new image" of size 28x28x6!

- ConvNet is a sequence of Convolution Layers, interspersed with activation functions



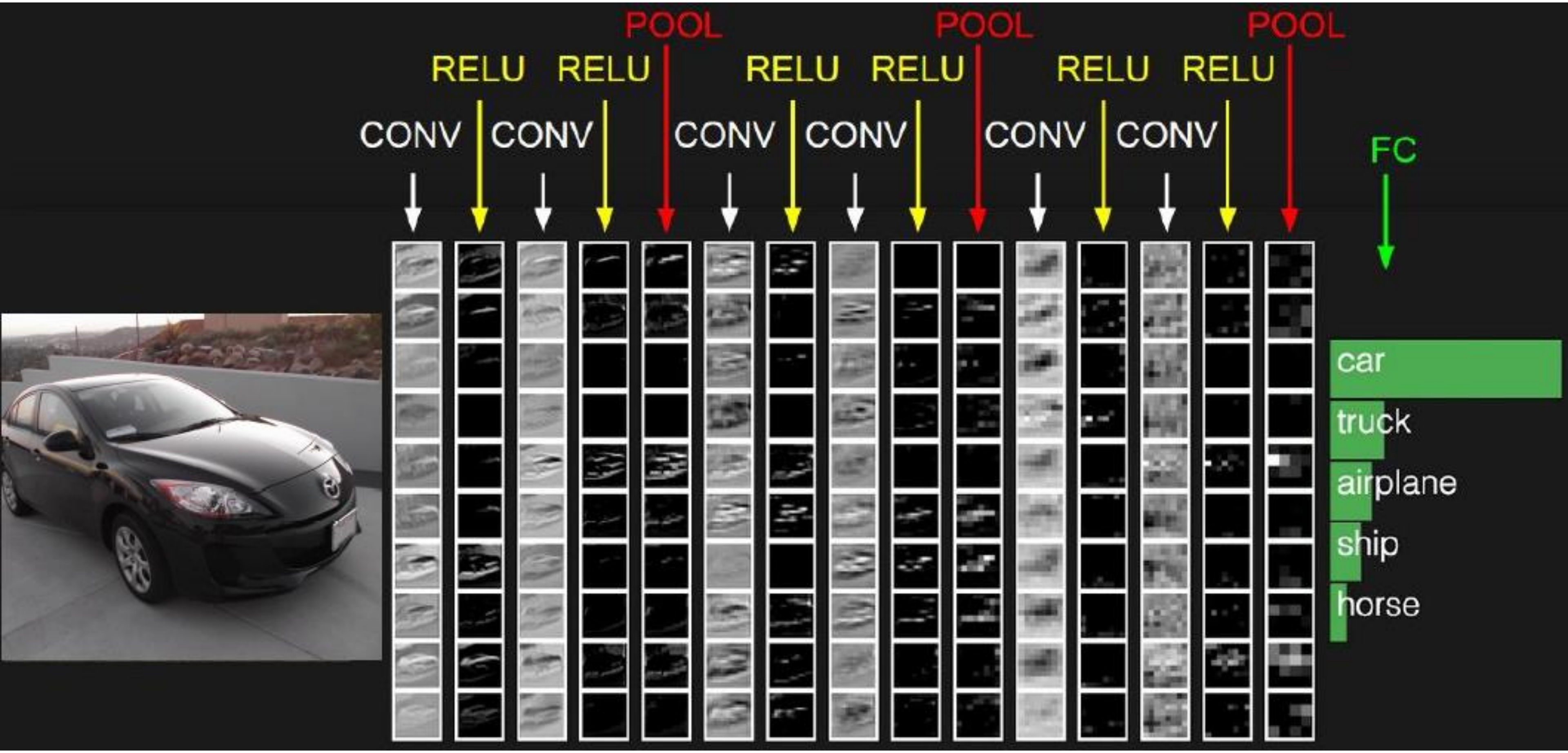
Visualization of activation maps



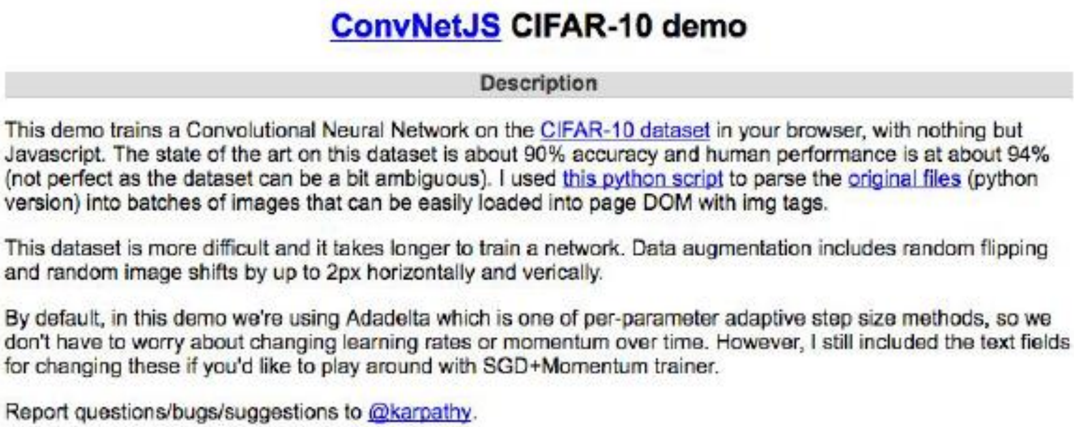
[Zeiler and Fergus 2013]

Visualization of VGG-16 by Lane McIntosh. VGG-16 architecture from [Simonyan and Zisserman 2014].

ConvNet for Classification



[ConvNetJS demo: training on CIFAR-10]



ConvNetJS CIFAR-10 demo

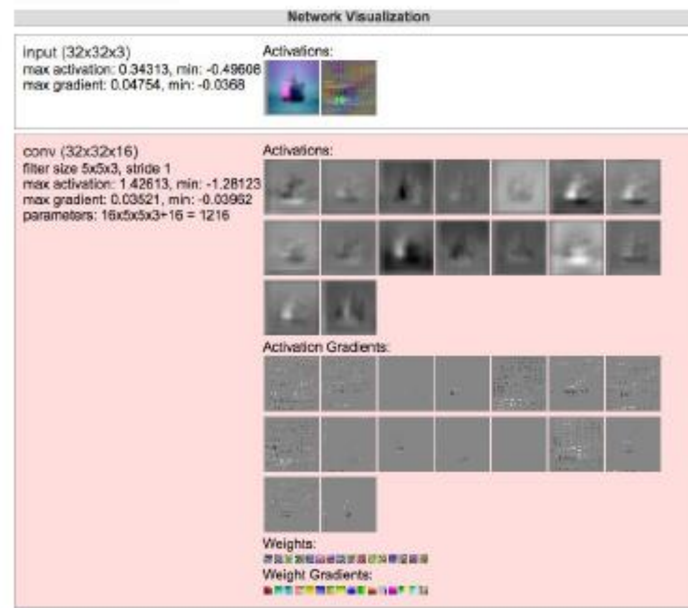
Description

This demo trains a Convolutional Neural Network on the CIFAR-10 dataset in your browser, with nothing but Javascript. The state of the art on this dataset is about 90% accuracy and human performance is at about 94% (not perfect as the dataset can be a bit ambiguous). I used this python script to parse the original files (python version) into batches of images that can be easily loaded into page DOM with img tags.

This dataset is more difficult and it takes longer to train a network. Data augmentation includes random flipping and random image shifts by up to 2px horizontally and verically.

By default, in this demo we're using Adadelta which is one of per-parameter adaptive step size methods, so we don't have to worry about changing learning rates or momentum over time. However, I still included the text fields for changing these if you'd like to play around with SGD+Momentum trainer.

Report questions/bugs/suggestions to @karpathy.



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

Self reading

- **Training Neural Networks: Part 1** **(CS231n Stanford Uni. – Lecture 6)**
  https://www.youtube.com/watch?v=wEoyxE0GP2M&list=PLC1qU-LWwrF64f4QKQT-Vg5Wr4qEE1Zxk&index=6
  - Activation Functions
  - Data Preprocessing
  - Weight Initialization
  - Batch Normalization
  - Hyperparameter Optimization

- **Training Neural Networks: Part 2** **(CS231n Stanford Uni. – Lecture 7)**
  https://www.youtube.com/watch?v=_JB0AO7QxSA&list=PLC1qU-LWwrF64f4QKQT-Vg5Wr4qEE1Zxk&index=7
  - Improve your training error:
    - Optimizers (SGD + Momentum, AdaGrad, RMSProp, Adam)
    - Learning rate schedules
  - Improve your test error:
    - Regularization (Dropout, Batch Normalization, Data Augmentation, DropConnect, Fractional Max Pooling, Stochastic Depth, Cutout, Mixup)
    - Choosing Hyper-parameters

Classification Datasets

80 million tiny images (2008)
https://groups.csail.mit.edu/visio
n/TinyImages/

Caltec 101 (2004)
http://www.vision.caltech.edu/Im
age_Datasets/Caltech101/

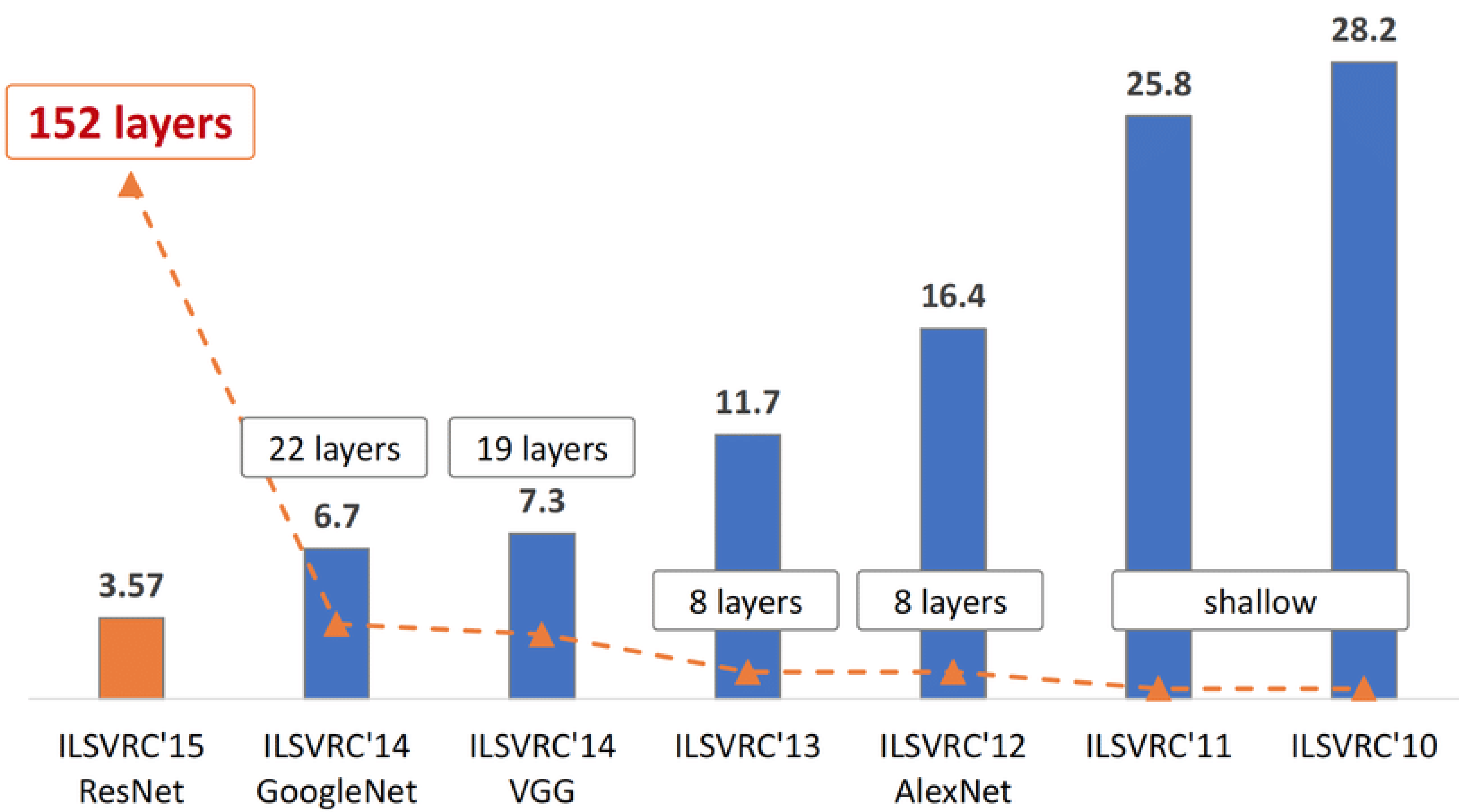
PASCAL2
Pattern Analysis, Statistical Modelling and
Computational Learning

Pascal VOC (2005-2012)
http://host.robots.ox.ac.uk/pasca
l/VOC/

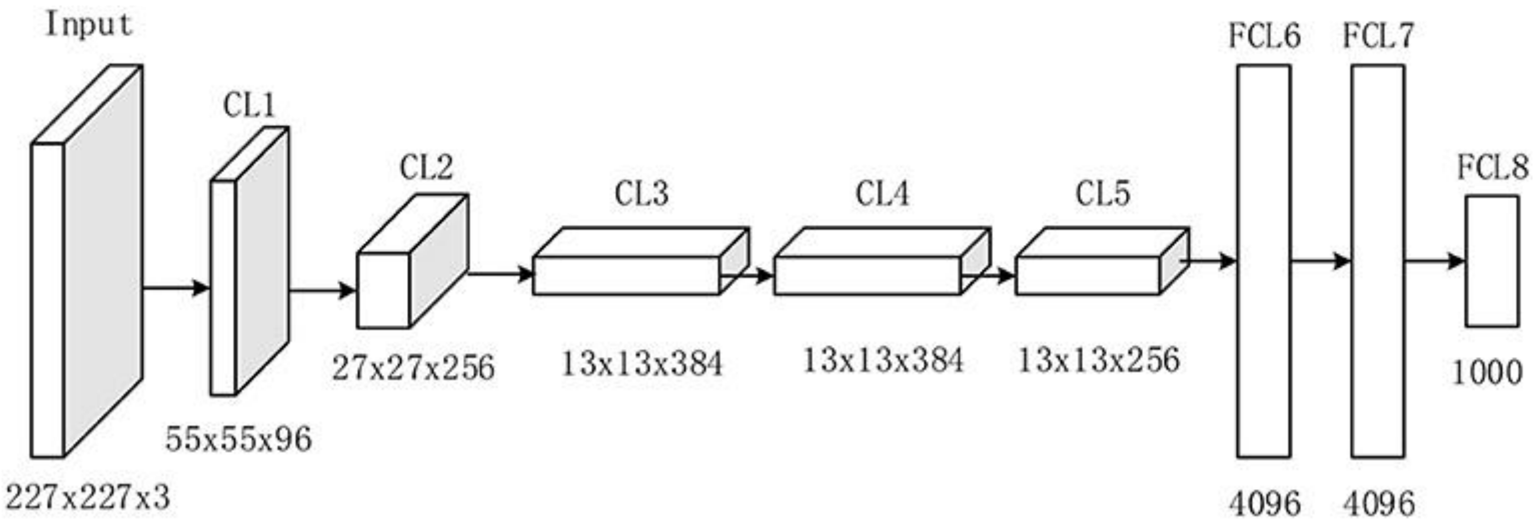
IMAGENET

ImageNet (2005-2012)
http://www.image-net.org/

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



AlexNet *[Krizhevsky et al. 2012]*

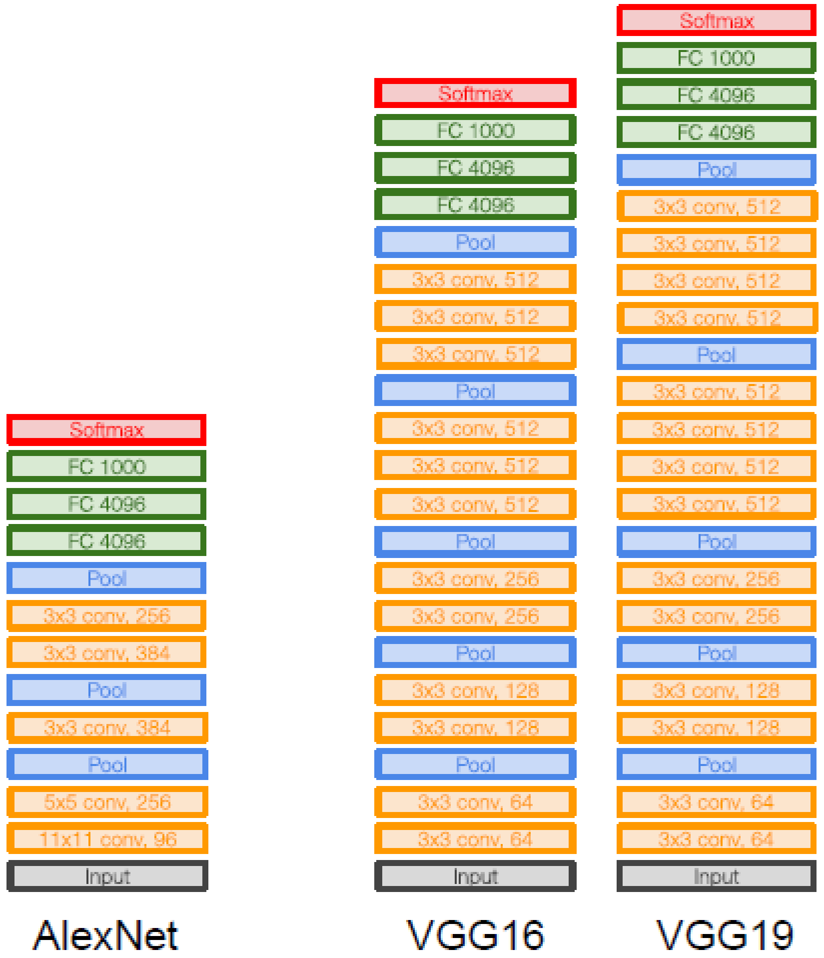


- 60M Parameter

[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

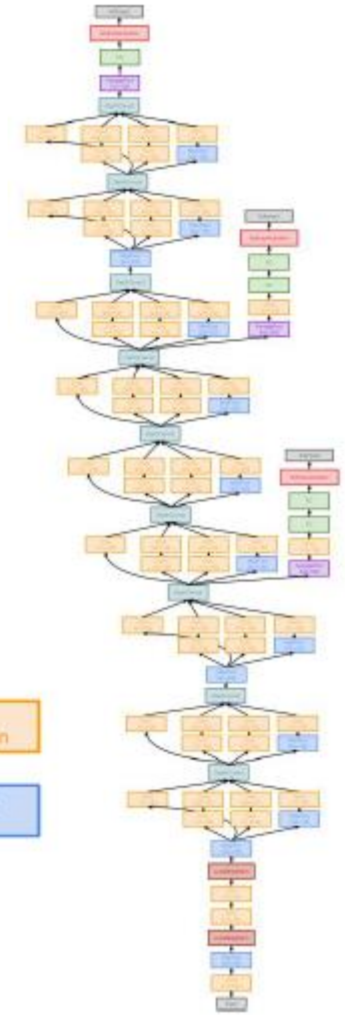
VGGNet *[Simonyan and Zisserman, 2014]*

- Small filters, Deeper networks
- 16 - 19 layers
- Only 3x3 CONV stride 1, pad 1 and 2x2 MAX POOL stride 2
- Stack of three 3x3 conv (stride 1) layers has same **effective receptive field** as one 7x7 conv layer
- But deeper, more non-linearities
- TOTAL params: 138M parameters (VGG16)

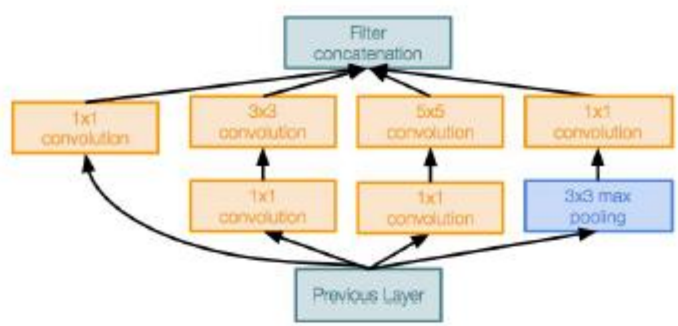


GoogLeNet *[Szegedy et al., 2014]*

- Deeper network, with computational efficiency
- 22 layers
- Efficient “Inception” module
- No FC layers
- Only 5 million parameters! 12x less than AlexNet
- ILSVRC’14 classification winner (6.7% top 5 error)

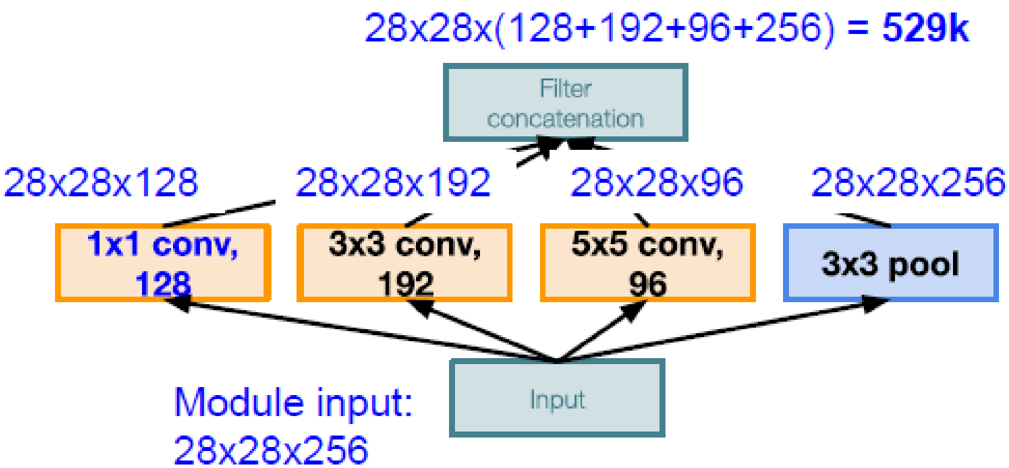


Inception module: design a good local network topology (network within a network) and then stack these modules on top of each other



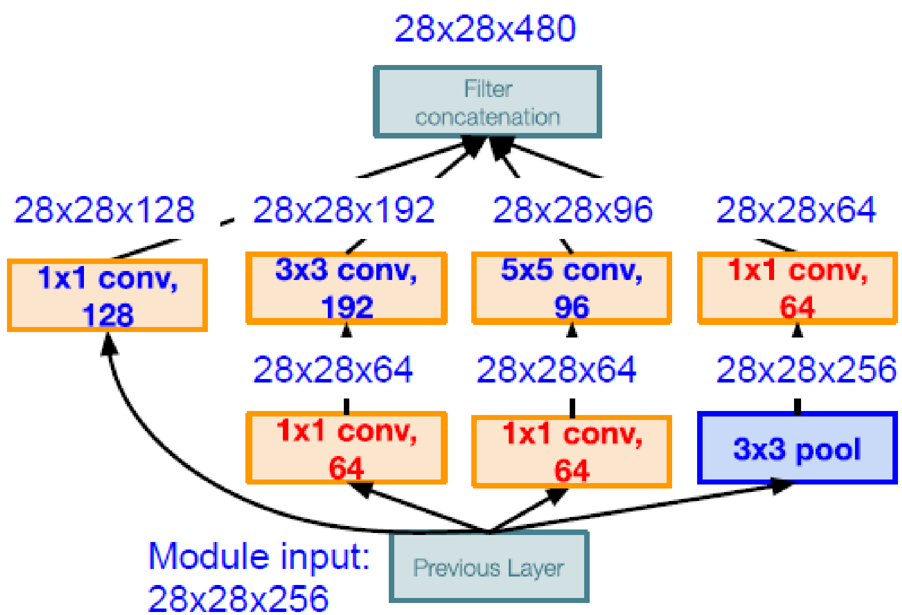
Inception module

GoogLeNet *[Szegedy et al., 2014]*



Naive Inception module

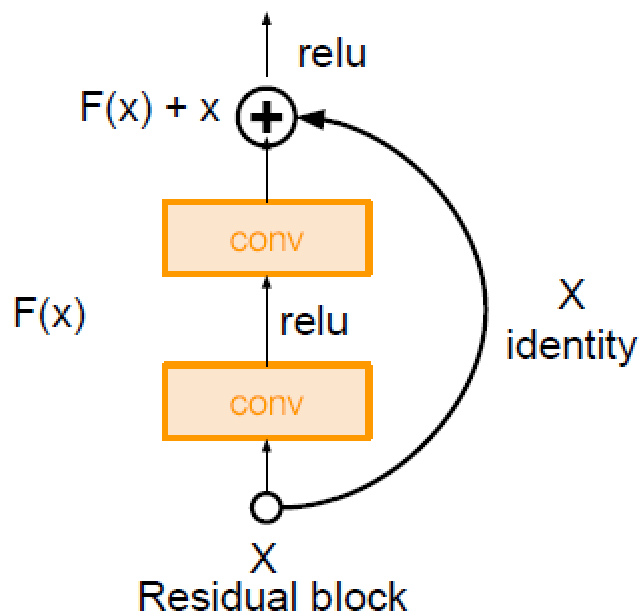
Problem: computational complexity
Solution: use 1x1 convolutions to reduce feature depth

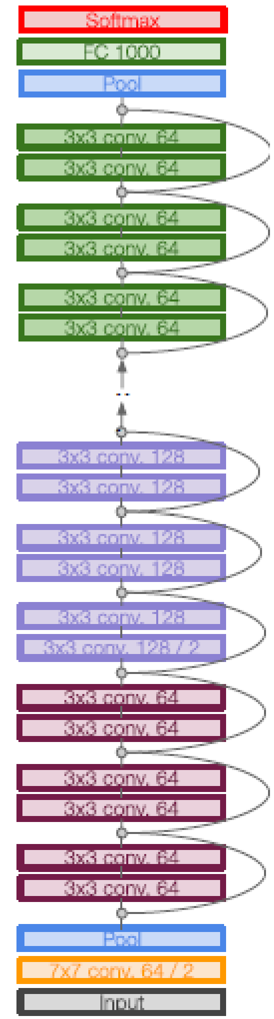


Inception module with dimension reduction

ResNet *[He et al., 2015]*

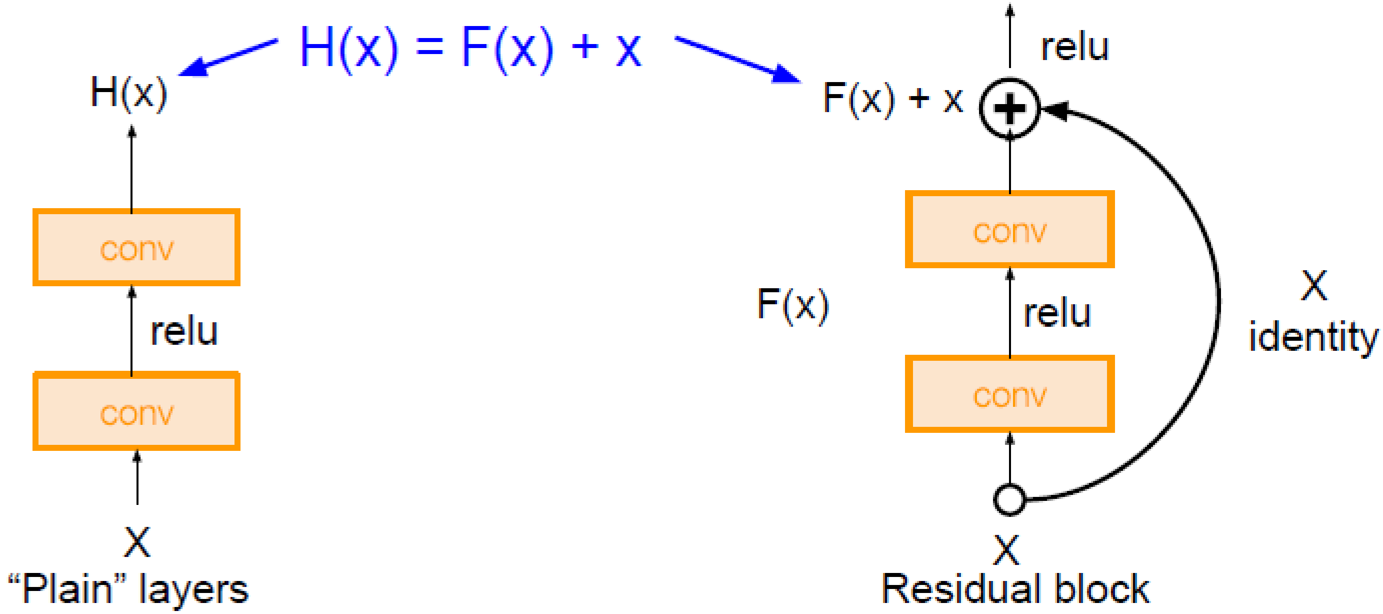
- Very deep networks using residual connections
- 152-layer model for ImageNet
- ILSVRC'15 classification winner (3.57% top 5 error)
- Swept all classification and detection competitions in ILSVRC'15 and COCO'15!





ResNet *[He et al., 2015]*

- Stacking deeper layers on a “plain” CNN cause optimization problem
- The deeper model should be able to perform at least as well as the shallower model.
- A solution is copying the learned layers from the shallower model and setting additional layers to identity mapping.
- It is easier to fit a residual mapping instead of directly trying to fit a desired underlying mapping



ResNet *[He et al., 2015]*

Experimental Results:

- Able to train very deep networks without degrading (152 layers on ImageNet, 1202 on Cifar)
- Deeper networks now achieve lowing training error as expected

- **1st places in all five main tracks**
  - ImageNet Classification: *"Ultra-deep"* (quote Yann) 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

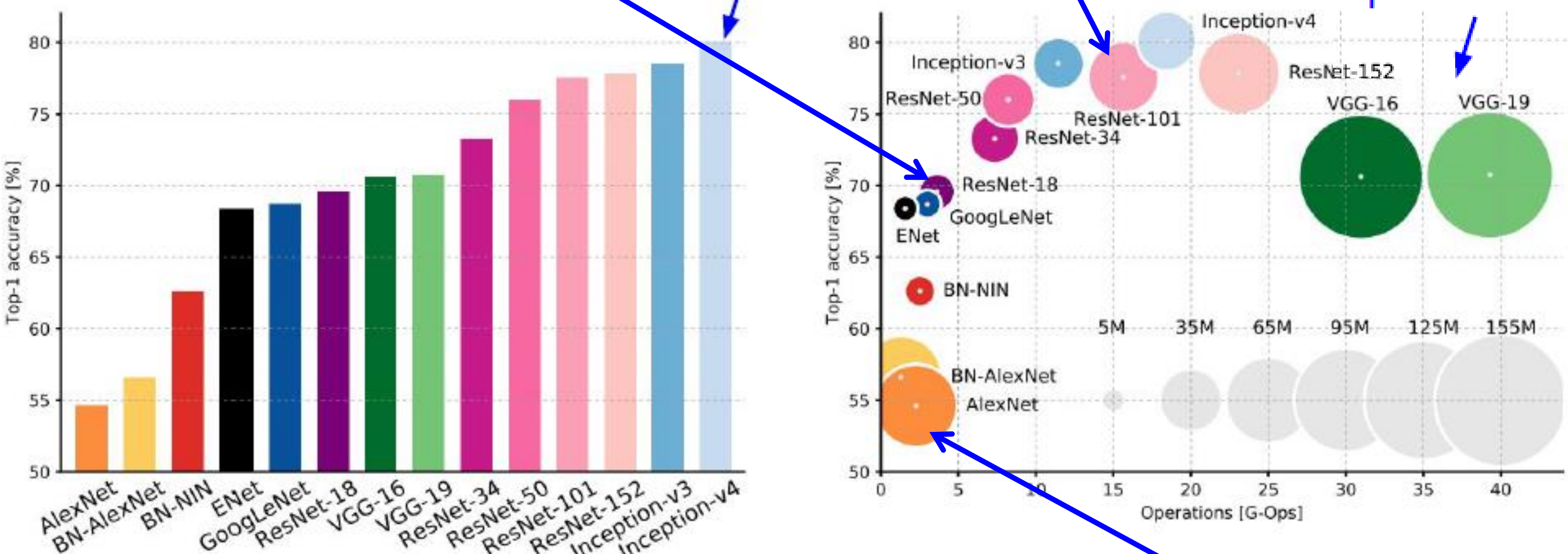
Comparing complexity...

ResNet:
Moderate efficiency depending on model, highest accuracy

GoogLeNet:
most efficient

VGG: Highest memory, most operations

Inception-v4: Resnet + Inception!



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

AlexNet:
Smaller compute, still memory heavy, lower accuracy

CONTENT BASED IMAGE RETRIEVAL (CBIR)

Content Based Image Retrieval (CBIR)

- Retrieve images from a large database using query by image content.
- The retrieval is based on content of the image rather than meta-data such as keywords, tags or descriptions associated with the image.
- Content might refer to color, shape, texture.. Etc.

Definition of Image similarity

- Semantic gab: is the difference between user intent and what could be extracted from the image
- Content feature similarity (e.g. color similarity)

- Near-duplicates: Same image changed in color or compressed

Definition of Image similarity

- Object Retrieval: output the same object or scene which may include large viewpoint and background variations compared to near duplicates search

- Geometric similarity: (distinct objects)

Definition of Image similarity

- Category-level classification: retrieve images of the same scene but with high visual distinctiveness

An example: banquet hall

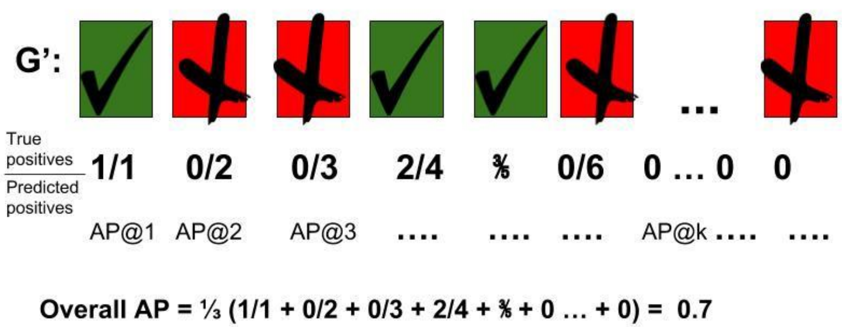
Visual similarity Vs. semantic similarity

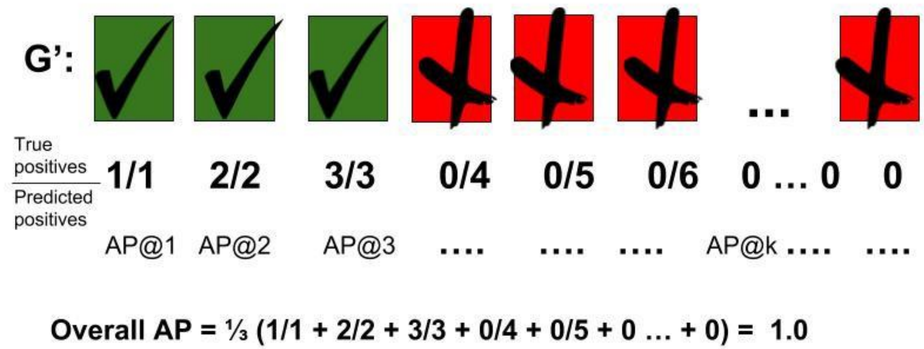
- Different methods for different problem formulations

Image Retrieval Evaluation

- Average Precision (AP): is the **mean** of the **precision scores** after each relevant image is retrieved.

$$AP_k = \frac{1}{GTP} \sum_{i=1}^{K} \frac{TP_{seen}}{i}$$

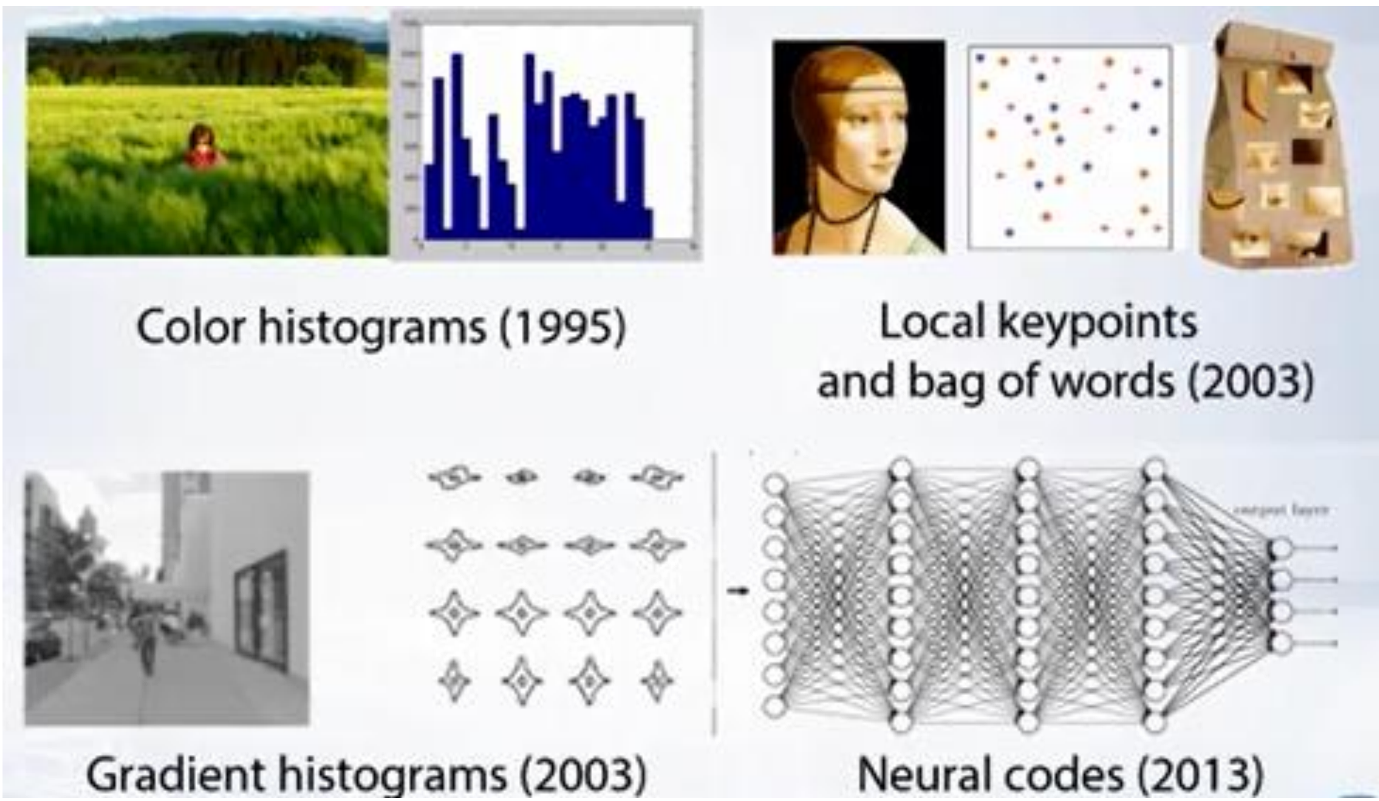




- Mean Average Precision (mAP): the mean over number of queries of the same object

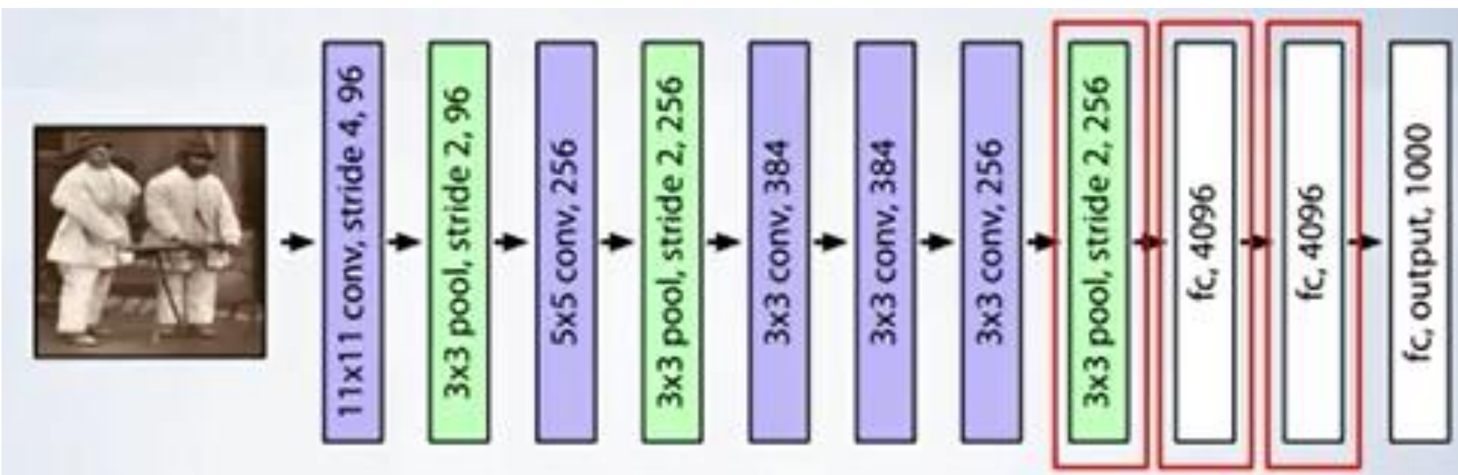
$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

Visual Features for Image Retrieval



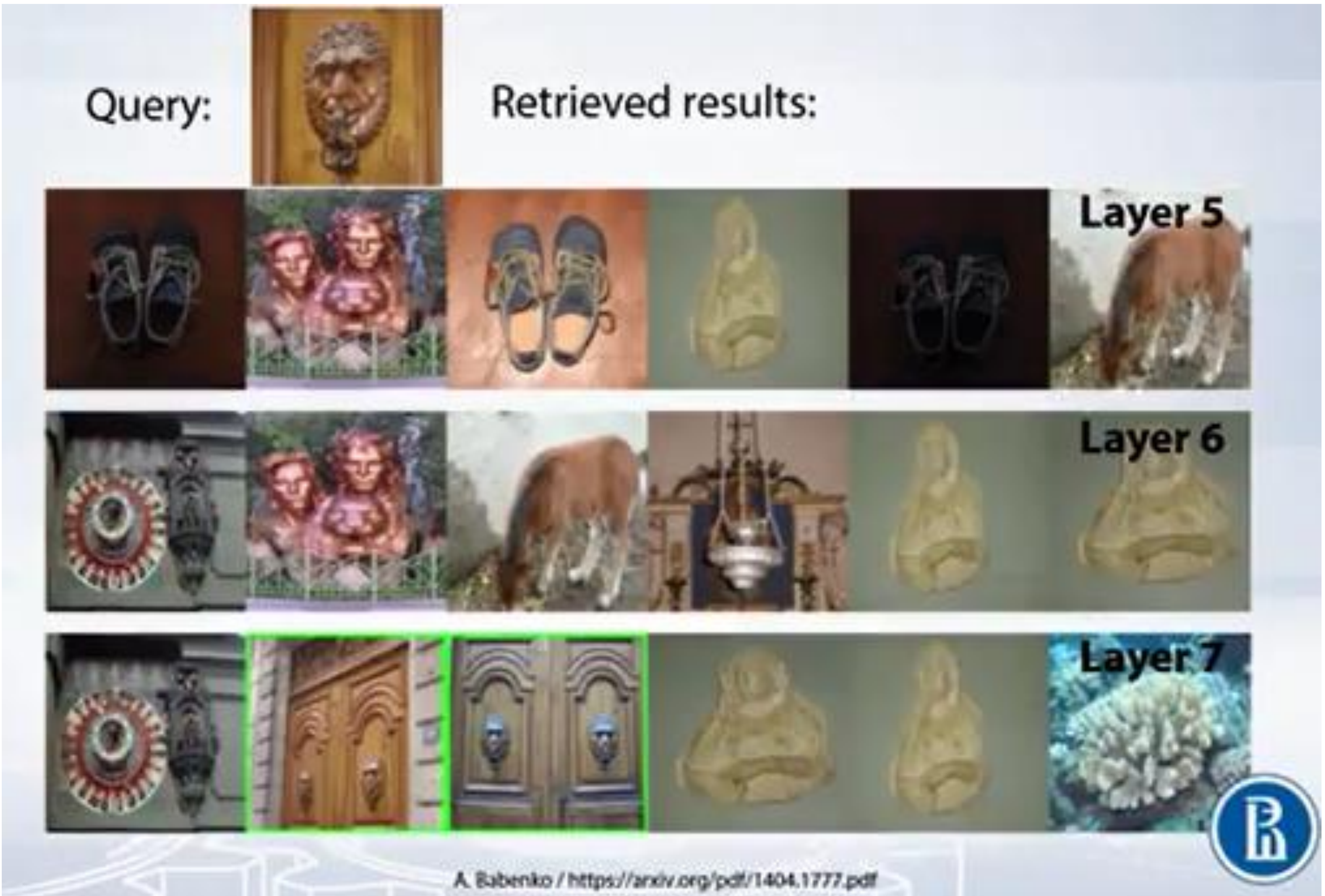
Neural codes for Image Retrieval (2003)

- Activations induced in upper layers of the network are used as semantic image features.
- Compute CNN features for each sub-patch in query image
- Compute distance between every pair of sub-patches in query and reference image.
- Average over sup-patches to compute distance between query and reference images.



Retrieval of same objects: low-level texture features

Retrieval of same objects: high-level concepts

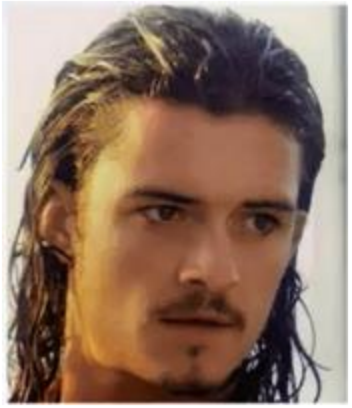


FACE VERIFICATION

Face Verification Vs. Face Recognition

- Face Recognition:
  - Has a database of K persons
  - Get an input image
  - Output ID if the image is one of the K persons or “not recognized”

- Face Verification (building block of face recognition in biometry applications)
  - Input Image & ID / Two images
  - Output if the image is that of the claimed person.



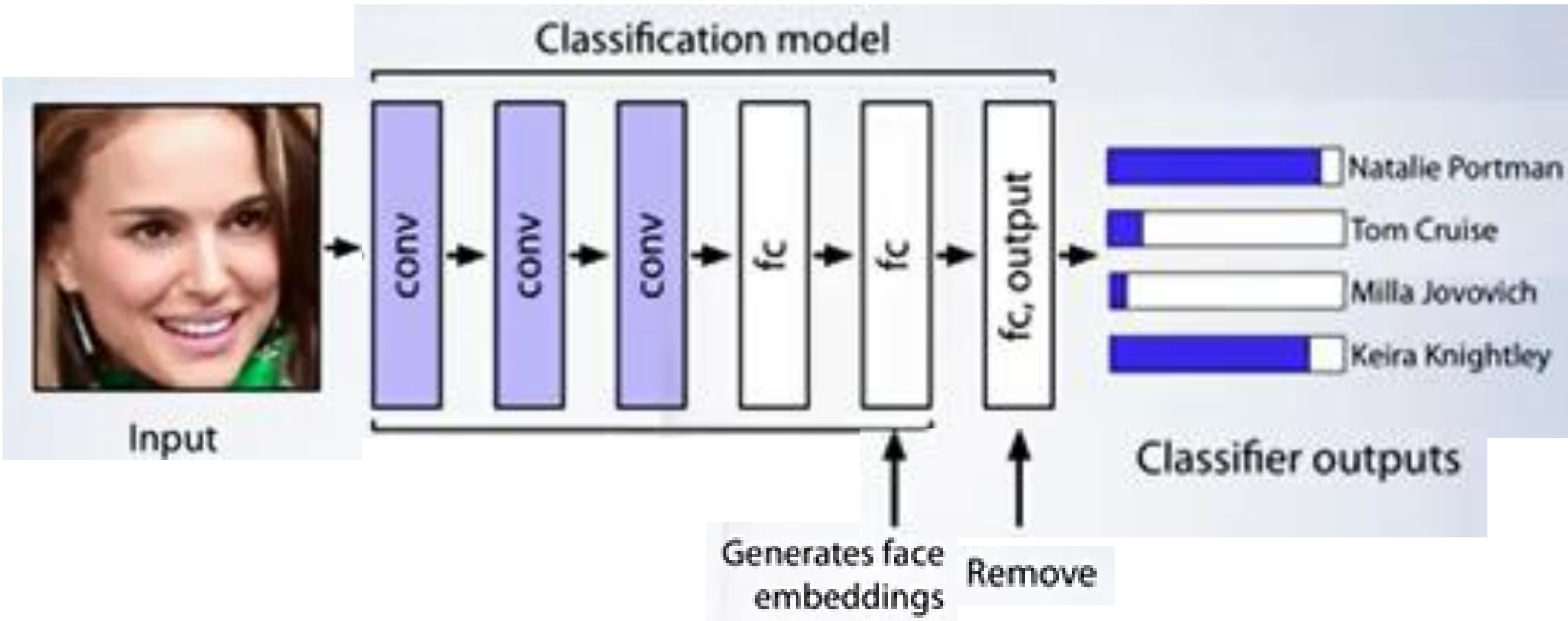
Face Verification

- Learn a "similarity " function:
  - d(img1,img2) = degree of difference between images
  - If d(img1,img2) $\leq \tau$ → same person
  - If d(img1,img2) $> \tau$ → different person

- Can be used in recognition to compare distance between query face image and all faces and select the one with small distance.

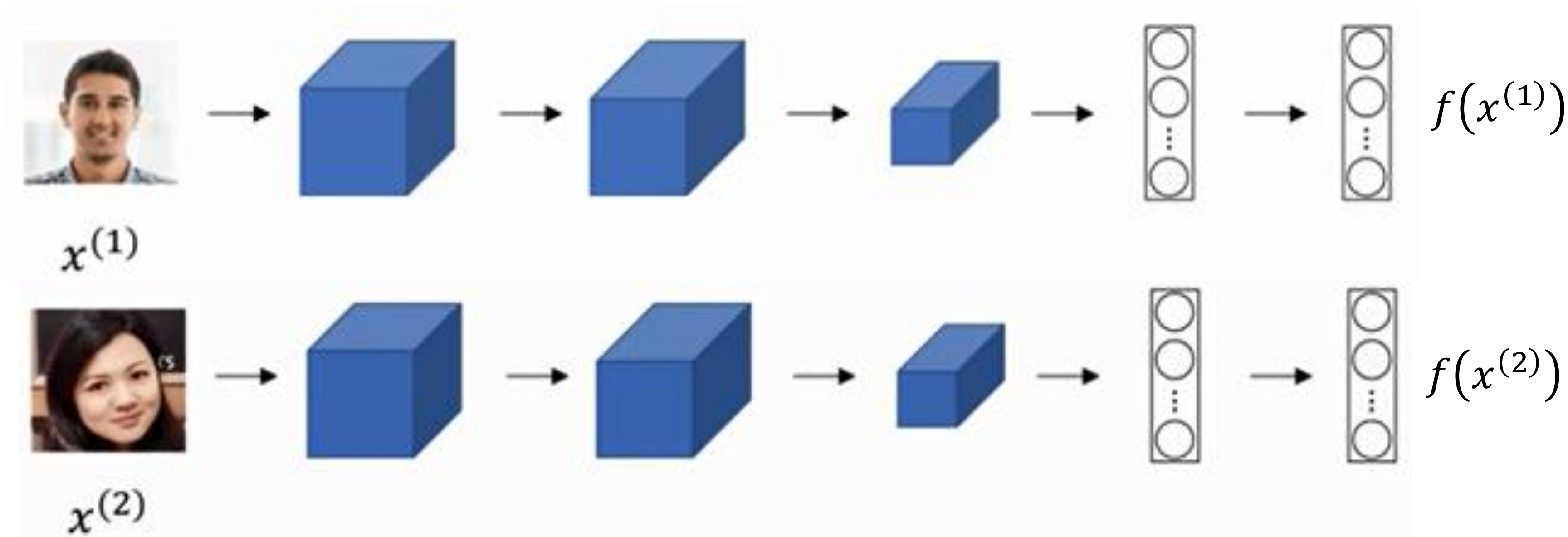
- If the person not in the database hopefully all distances are large.

Training a deep face verification model



Remove the classification layer and extract face "embeddings" and apply comparisons via Euclidean distance.

Siamese Network



$$d\left(x^{(1)}, x^{(2)}\right) = \left\|f\left(x^{(1)}\right) - f\left(x^{(2)}\right)\right\|_2^{\,2}$$

If $x^{(i)}, x^{(j)}$ are the same person, $\left\|f\left(x^{(i)}\right) - f\left(x^{(j)}\right)\right\|^2$ is small.

If $x^{(i)}, x^{(j)}$ are different persons, $\left\|f\left(x^{(i)}\right) - f\left(x^{(j)}\right)\right\|^2$ is large.

Taigman et. al., 2014. DeepFace closing the gap to human level performance

Learning Objective

$$d(A,P) = \|f(A) - f(P)\|_2^{\,2}$$
$$d(A,N) = \|f(A) - f(N)\|_2^{\,2}$$

$$\|f(A) - f(P)\|_2^{\,2} \leq \|f(A) - f(N)\|_2^{\,2}$$
$$\|f(A) - f(P)\|_2^{\,2} - \|f(A) - f(N)\|_2^{\,2} \leq 0$$

$$\|f(A) - f(P)\|_2^{\,2} + \alpha \leq \|f(A) - f(N)\|_2^{\,2}$$
$$\|f(A) - f(P)\|_2^{\,2} - \|f(A) - f(N)\|_2^{\,2} + \alpha \leq 0$$

α is a margin

Loss Function (triplet Loss)

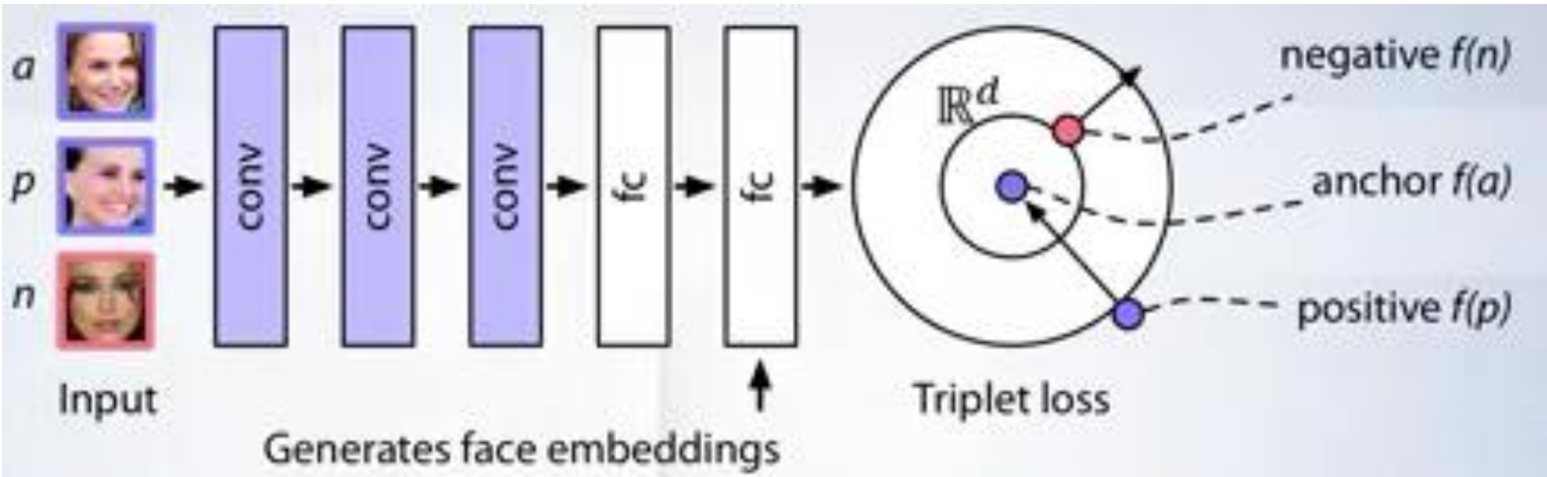
- Given images A, P, N

$$L\ (A, P, N)$$
$$= \max(\|f(A) - f(P)\|_2{}^2\ - \|f(A) - f(N)\|_2{}^2 \mathbf{+\ \alpha}\ , 0)$$

$$J = \min \textstyle\sum_{i=1}^{m} L\ (A^i, P^i, N^i)$$

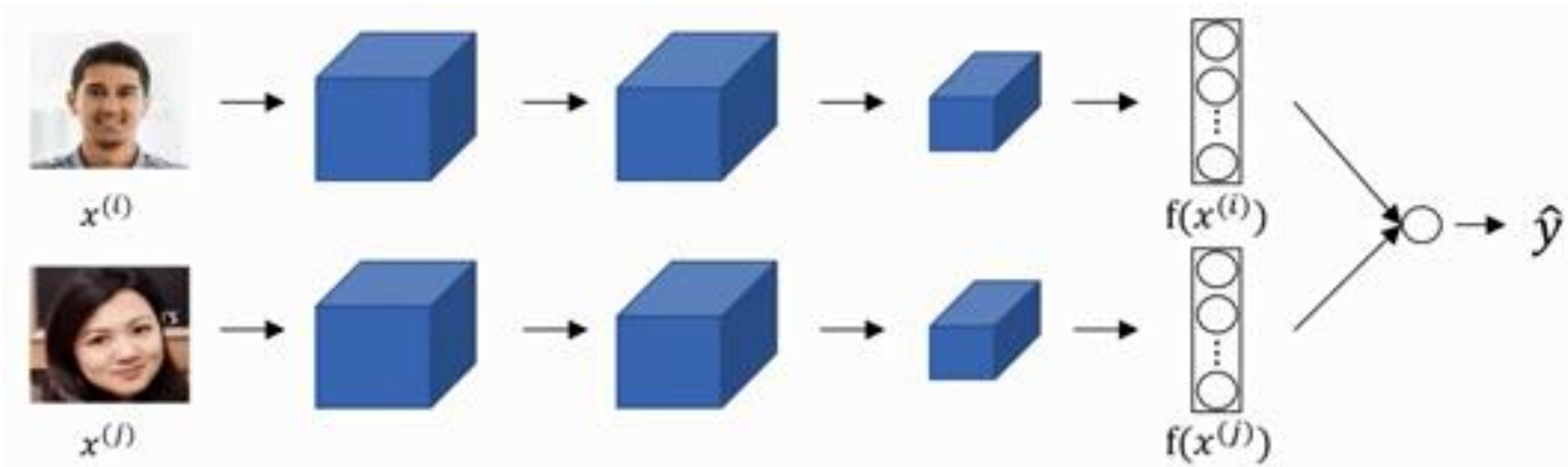
Schroff et. al., 2015. FaceNet: A unified embedding for face recognition and clustering

Learning the Triplet Loss



- Training set: 10K images for 1K persons
- So arrange the data in triplets and use this to train the model

Learning similarity pairs



- $y \in \{0, 1\}$
- 1 if similar
- 0 if not similar

Credit for

CS 4495 Computer Vision (Spring 2015)

A. Bob - College of Computing, Georgia Tech.

CSE 455 Computer Vision (Winter 2017) by Linda Shapiro - University of Washington.

CS131 "Computer Vision: Foundations and Applications" by University of Stanford (Fall 2019)