



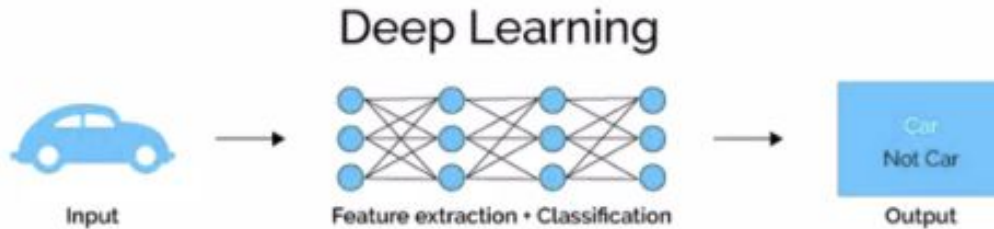
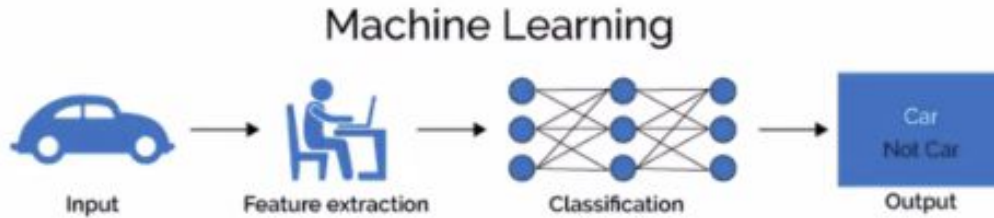
MSc in Digital Humanities
Image Processing/Analysis

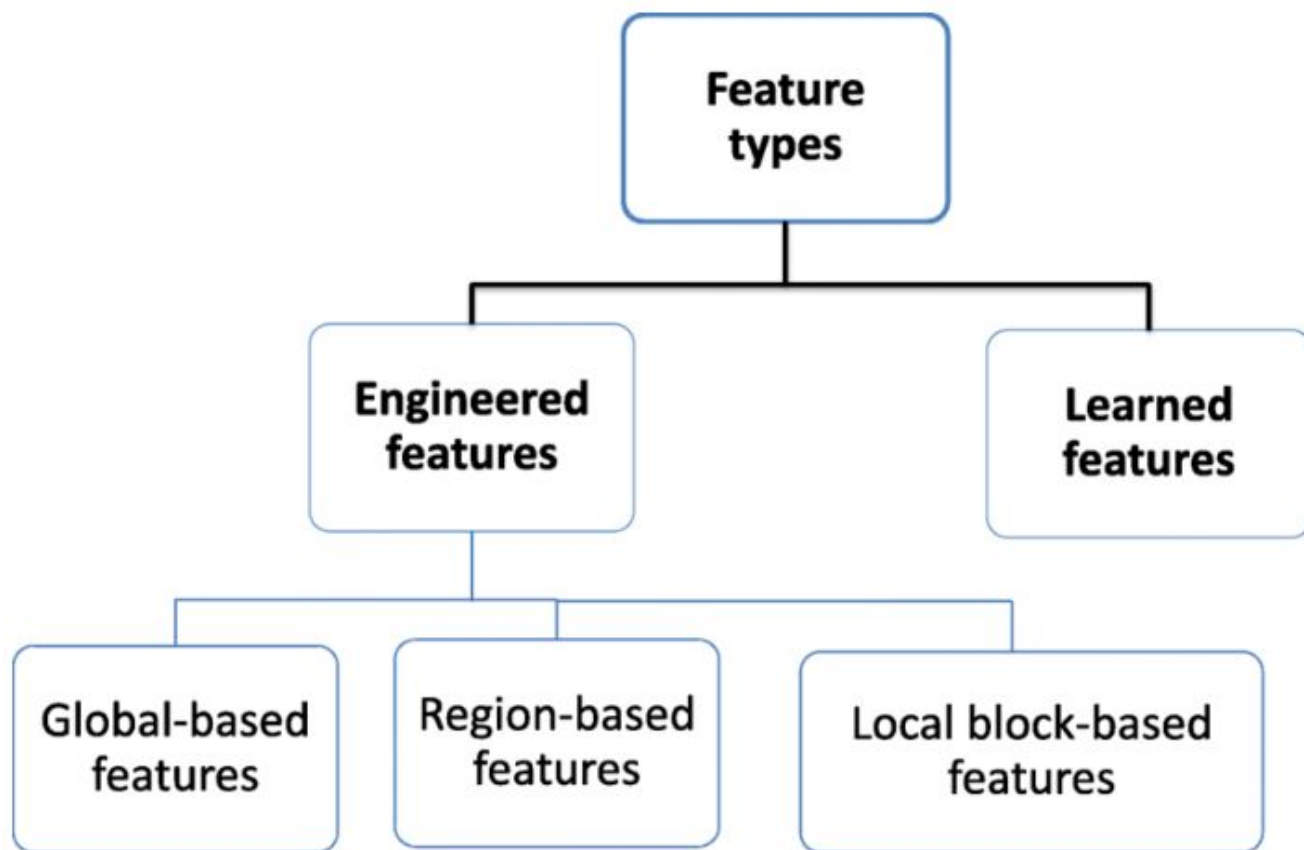
Processing, Classification, Segmentation II

Paraskevi Tzouveli, Christos Tsinos, George Paylidis

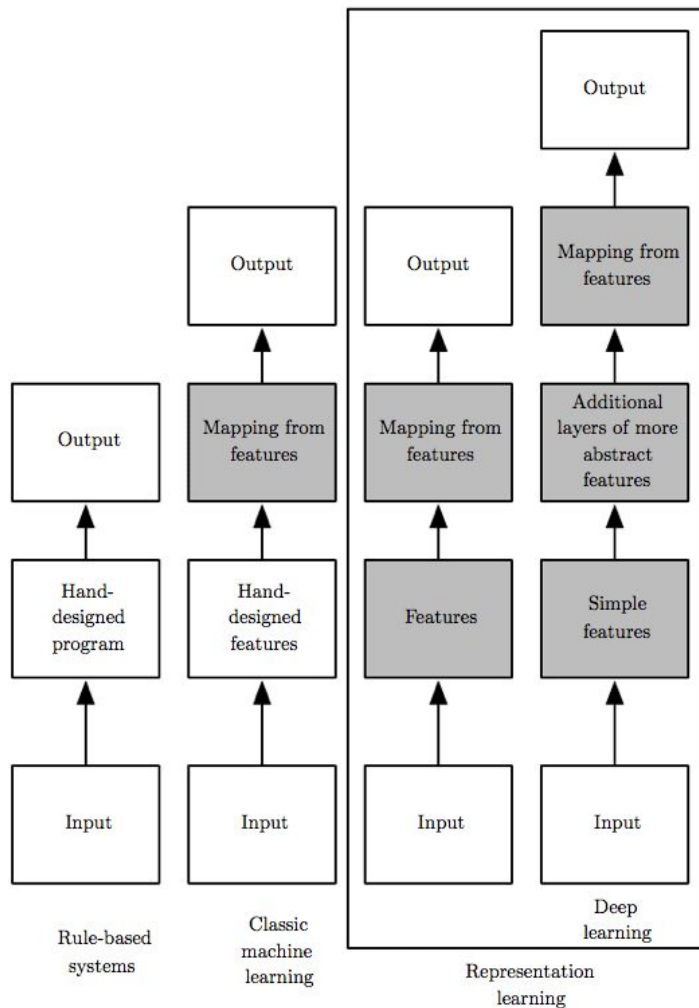
e-mail: parasktz@gmail.com

Machine Learning vs Deep Learning





Features Extraction Techniques over the time



Traditional Computer Vision and Deep Learning

```
>>> img = cv2.imread('fly.png',0)

# Create SURF object. You can specify params here or later.
# Here I set Hessian Threshold to 400
>>> surf = cv2.SURF(400)

# Find keypoints and descriptors directly
>>> kp, des = surf.detectAndCompute(img,None)

>>> len(kp)
699
```

Traditional CV

- Explicit feature extraction (e.g. SIFT, HOG)
- From images we create feature vectors

Learning Feature Hierarchy with DL

- Deep architectures can be more efficient in feature representation;
- Natural derivation/abstraction from low level structures to high level structures;
- Share the lower-level representations for multiple tasks (such as detection, recognition, segmentation).

Feature representation



3rd layer
"Objects"



2nd layer
"Object parts"



1st layer
"Edges"



Pixels

Deep Learning (CNN)

- Implicit feature extraction
- The output layers of each convolutional layer are the features of the next Convolutional layer

What is a CNN

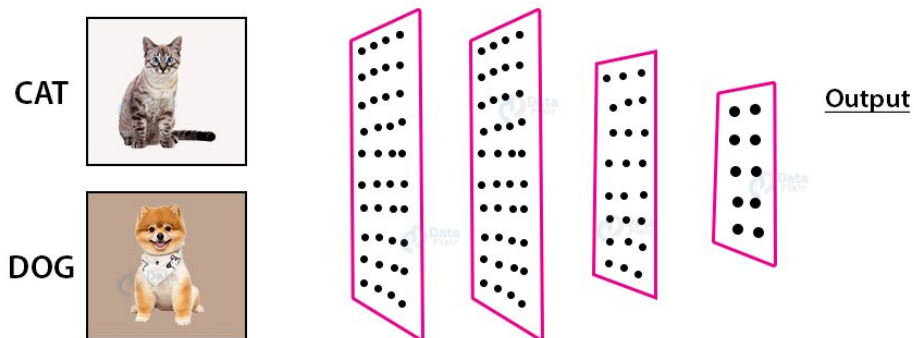
Imagine your CNN is a factory assembly line designed to look at an image (like a picture of a cat) and tell you exactly what it is.

The main job of the CNN is to start with **pixels** and end with a **label** ("Cat").

The Input: Raw Material

- **Action:** You feed the image into the network.
- The image is just a massive grid of numbers (pixels).

E.g. For a color image, it's three grids: one for Red, one for Green, and one for Blue.



What is a CNN

Step 1: The Feature Extractor (The Convolutional Layer)

This is the most unique and important part of the CNN.

- **Action:** Small digital "magnifying glasses" called **filters** (or kernels) slide across the entire image, inspecting tiny pieces one by one.
- The network doesn't look at the whole cat. It sends in 10, 20, or 30 tiny inspectors, each trained to look for one specific, simple thing:
 - One inspector looks only for **vertical lines**.
 - Another looks only for **diagonal edges**.
 - A third looks only for **rough texture**.
- **Output:** The result is a set of **Feature Maps**, which are new images that highlight **where** in the original picture that specific feature was found.
 - *Example:* The "vertical line" map will be bright wherever a vertical line (like a cat's whisker or a chair leg) was found.

What is a CNN

Step 1: Feature Extractor This stage aims to automatically learn hierarchical representations of the input data.

Convolutional Layer - Core Concept: Parameter Sharing and Sparsity of Connections.

- **Mechanism:**

1. A small, learnable matrix called a **filter** (or kernel) is slid across the input volume (the image).
2. At each location, a dot product is computed between the filter and the input pixels covered by the filter, this operation is a discrete convolution

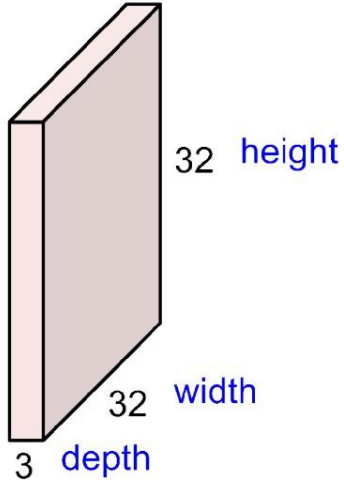
- **Purpose:**

Feature Detection: Each filter learns to detect a specific feature (e.g., edges, textures, corners).

- **Hyperparameters: Filter Size** (e.g., 3×3), **Stride** (the step size of the filter), and **Padding** (adding zeros to maintain spatial dimensions).
- **Activation:** The result is typically passed through a non-linear activation function, most commonly **ReLU** (Rectified Linear Unit), defined as : $f(x) = \max(0, x)$.

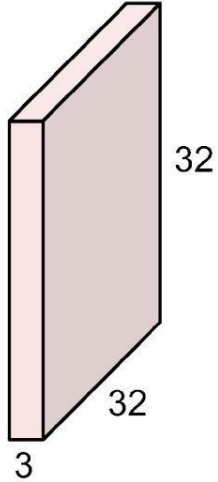
Convolution Layer

32x32x3 image -> preserve spatial structure

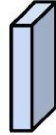


Convolution Layer

32x32x3 image

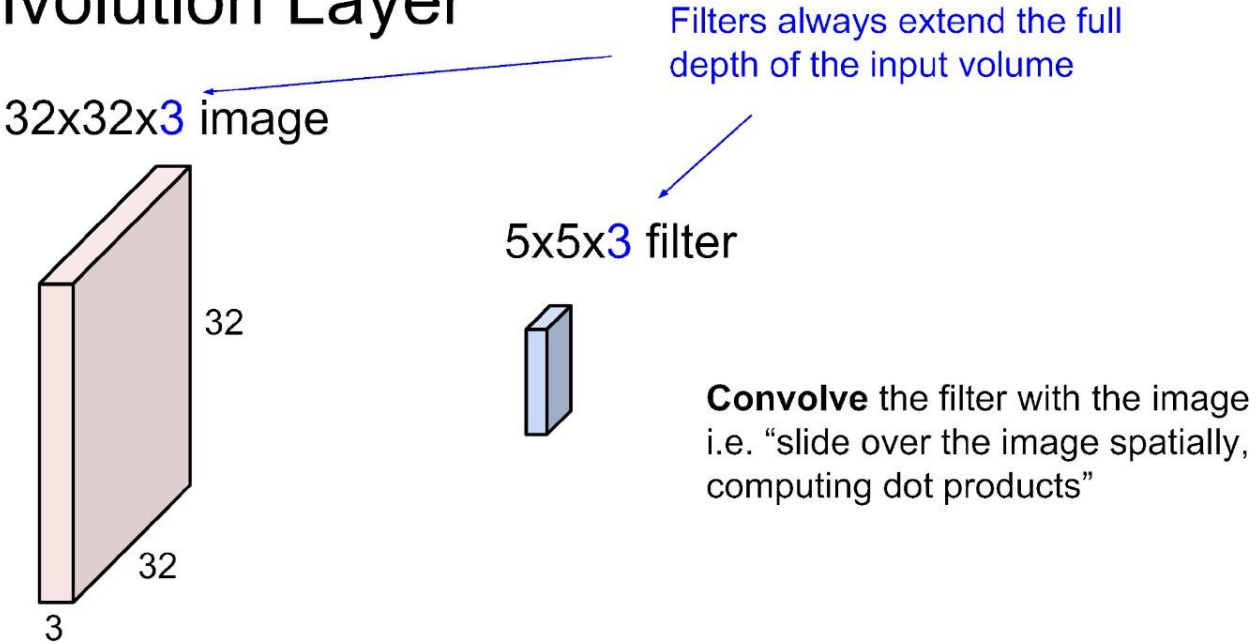


5x5x3 filter

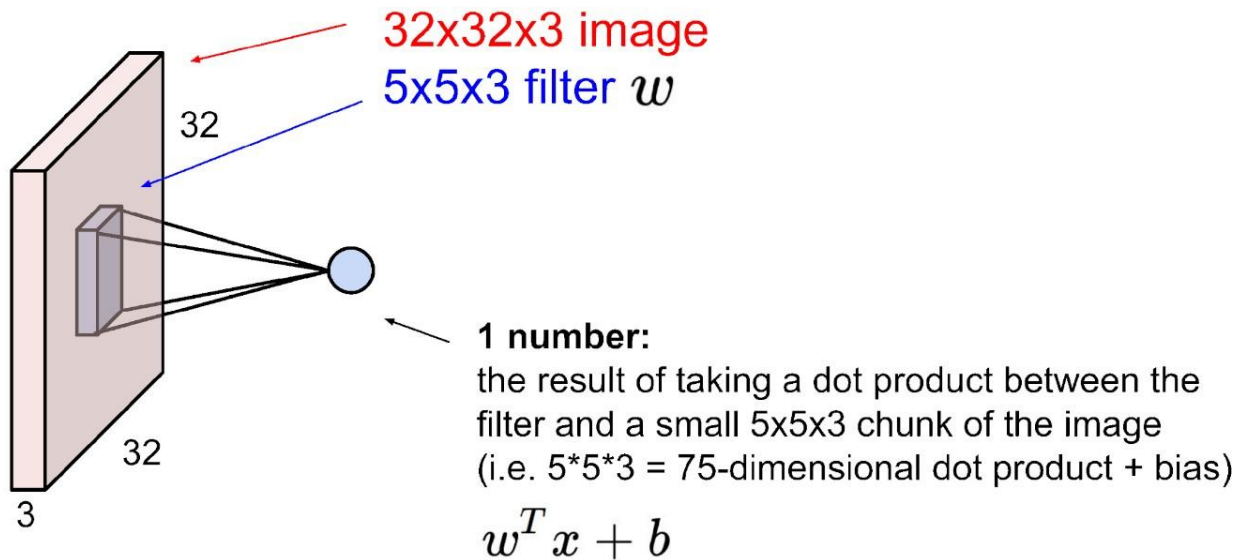


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

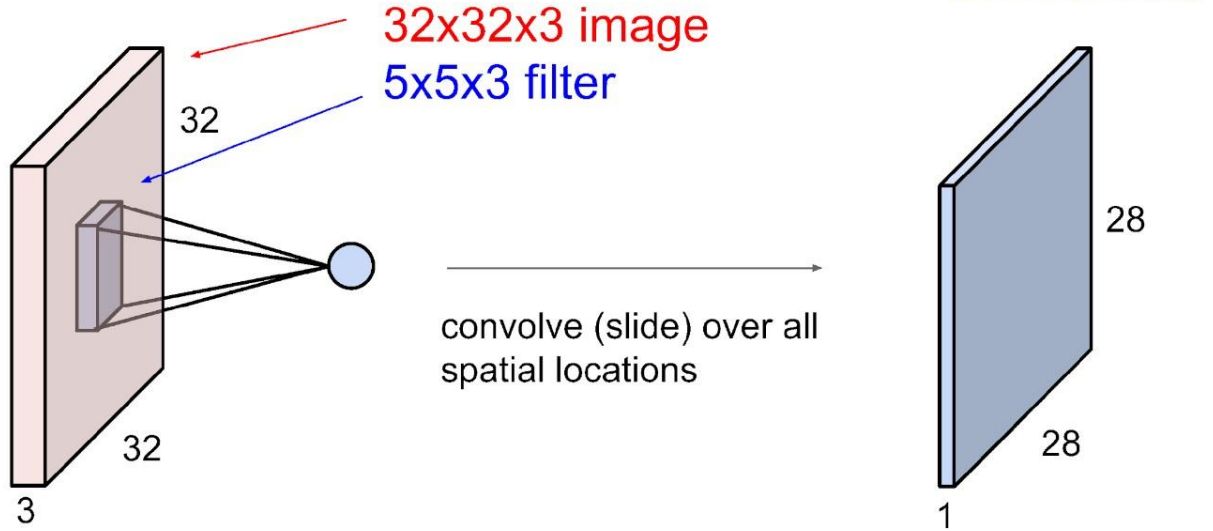
Convolution Layer



Convolution Layer

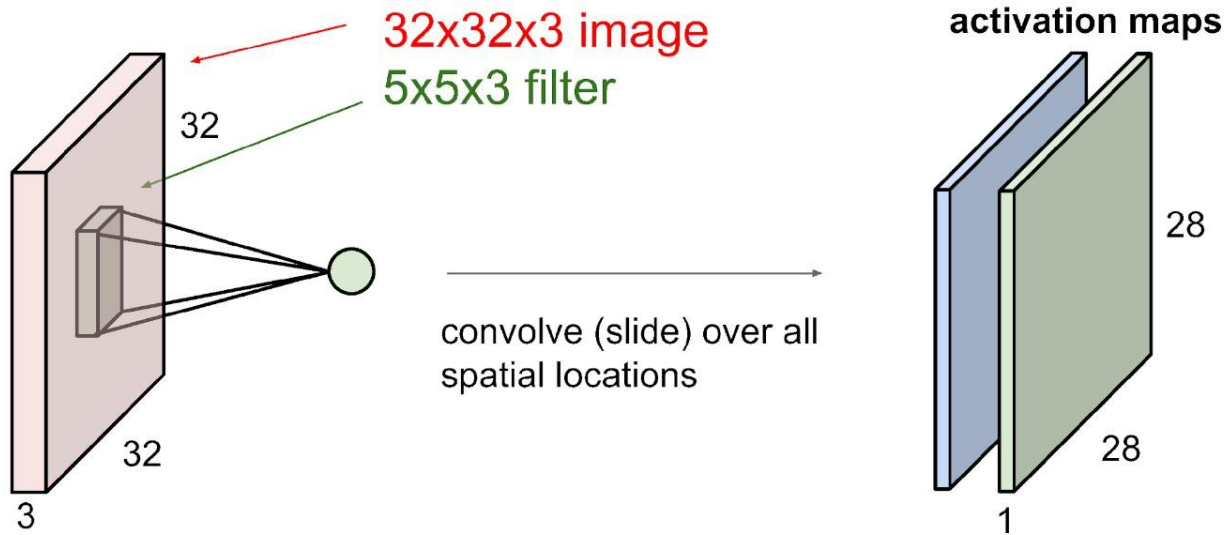


Convolution Layer

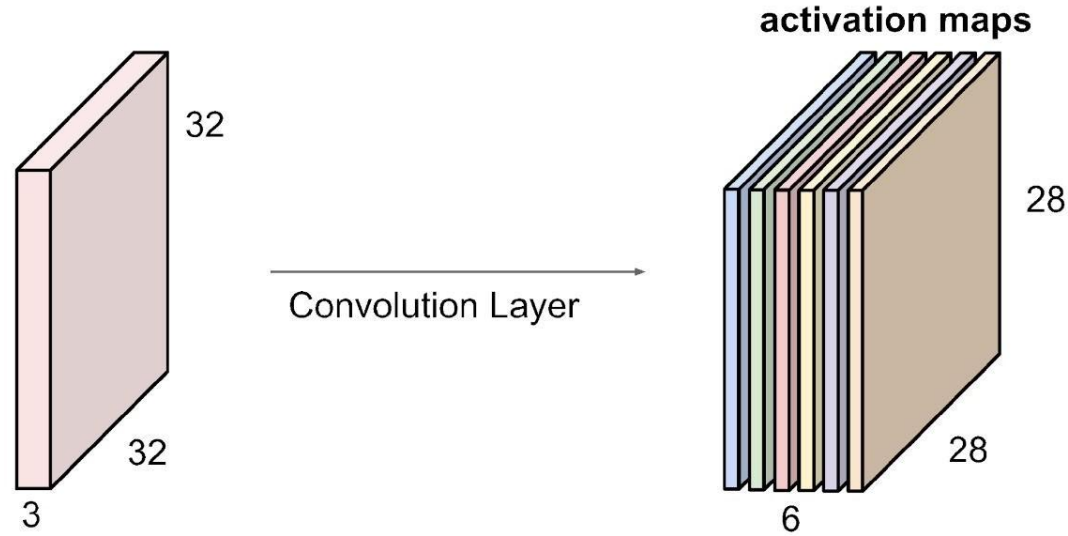


Convolution Layer

consider a second, green filter

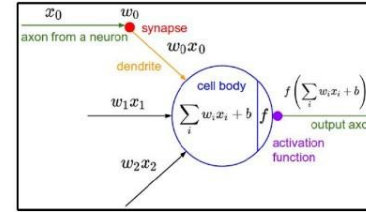
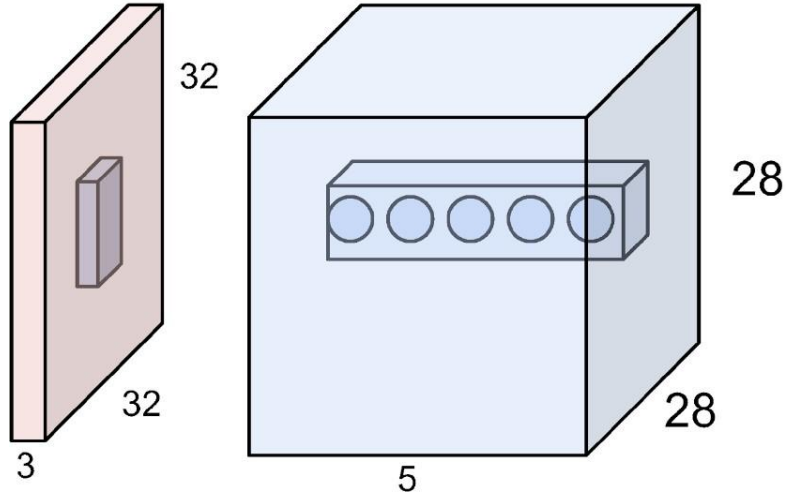


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



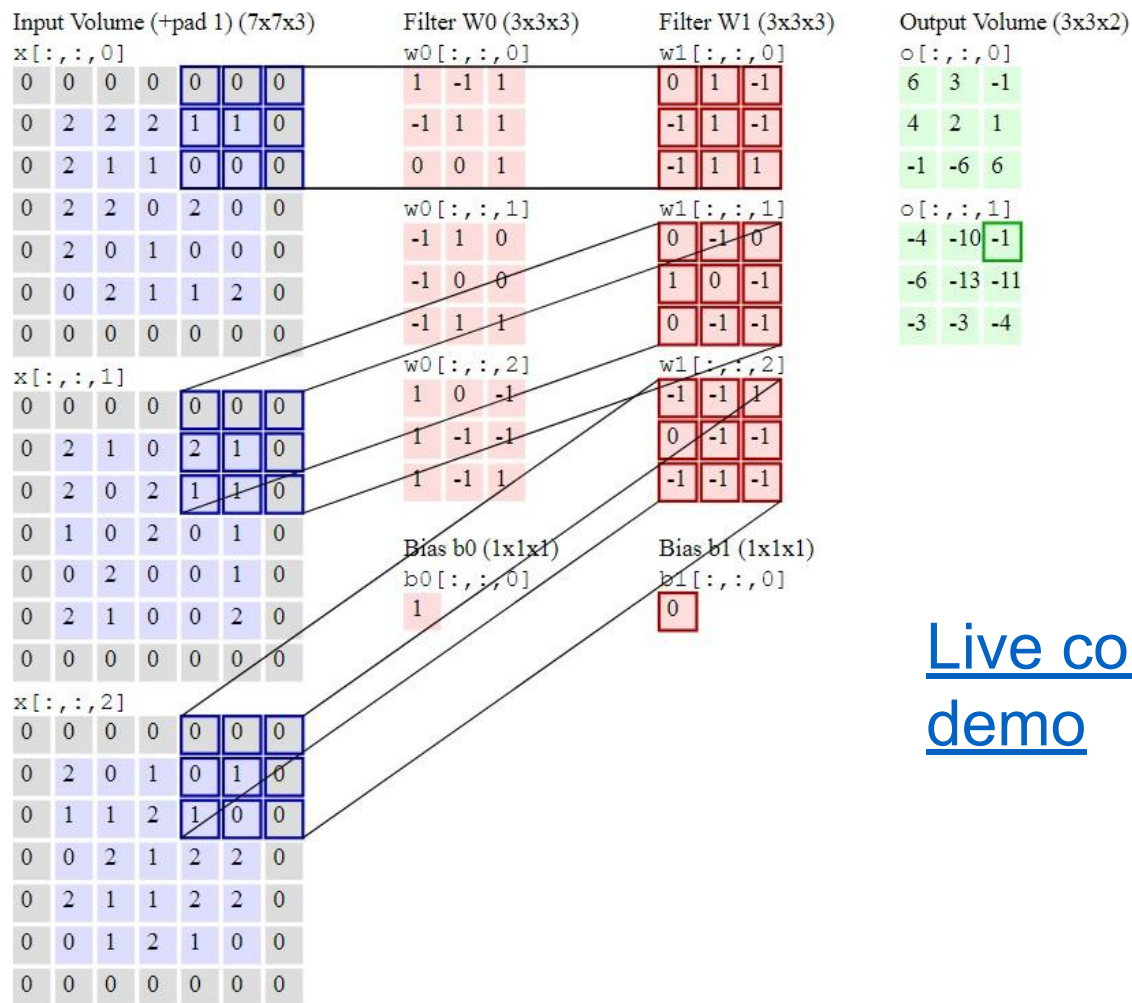
We stack these up to get a “new image” of size 28x28x6!

The brain/neuron view of CONV Layer



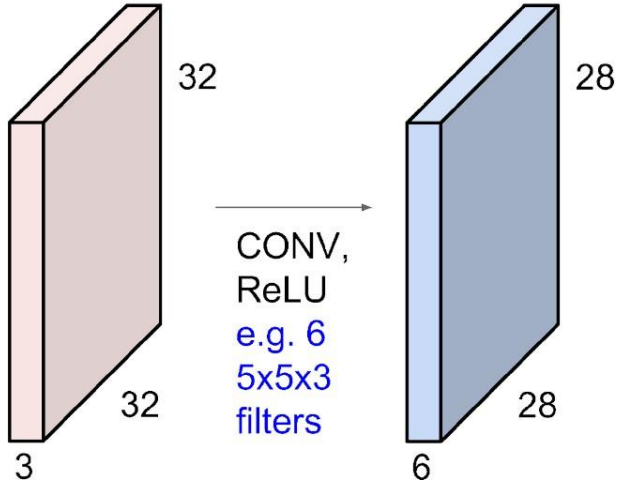
E.g. with 5 filters,
CONV layer consists of
neurons arranged in a 3D grid
(28x28x5)

There will be 5 different
neurons all looking at the same
region in the input volume

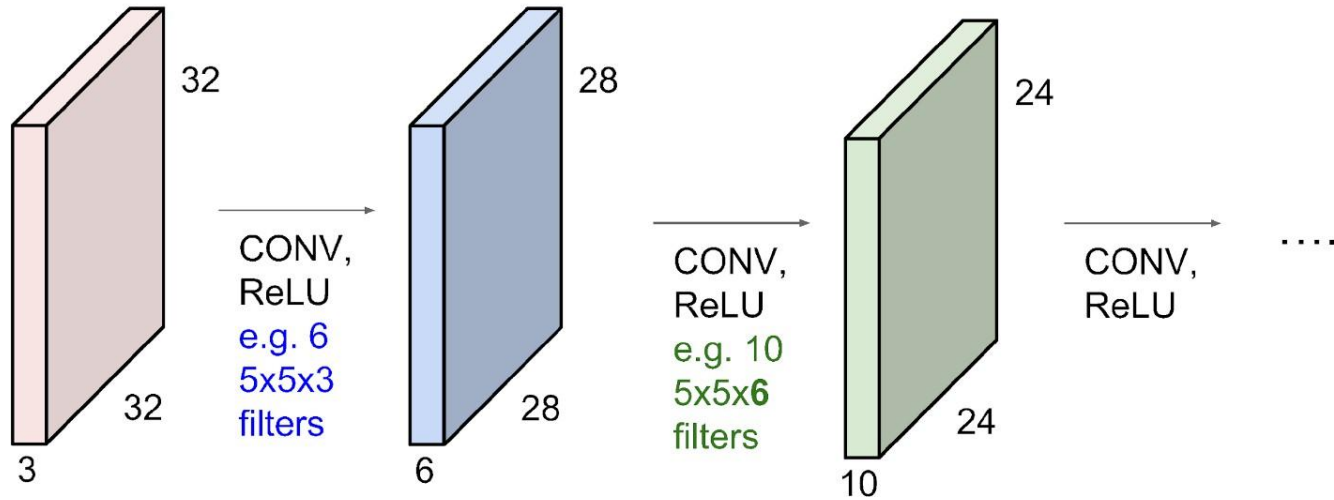


Live convolution
demo

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



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



What is a CNN

Step 2: The Sizer and Compressor (The Pooling Layer)

The feature maps created in Step 2 are still too large.

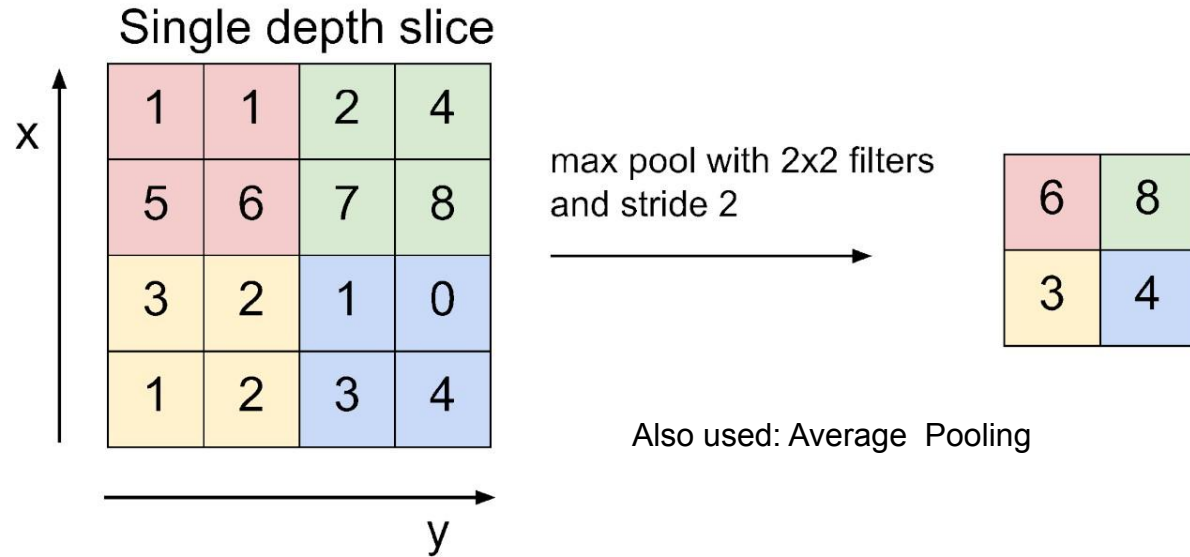
→ We need to shrink them down while keeping only the most important information.

Action: The network runs a "Max Pooling" filter over the feature maps. It slides a small window (e.g., 2x2) over the map and only keeps the **biggest number** from that window.

- This is like drawing a rough sketch. We don't need to know *every single pixel* where an edge was found, we just need to know the **most important location** where that feature showed up in a general area.

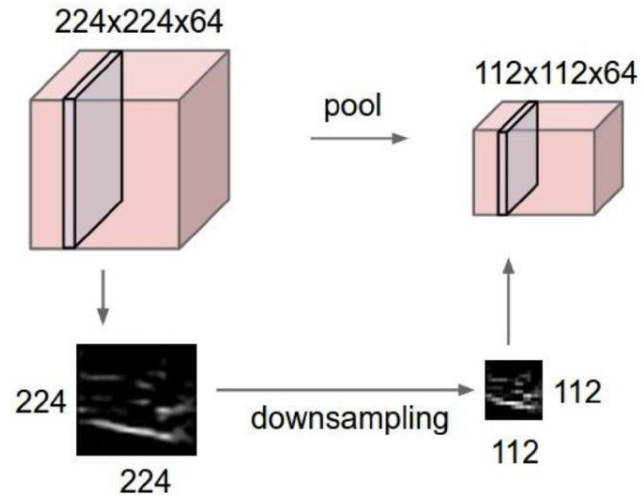
Benefit: This dramatically reduces the size of the data and speeds up the process, making the network less sensitive to where exactly the cat is positioned in the photo.

MAX POOLING



Pooling layer

- makes the representations smaller and more manageable
- operates over each activation map independently:



What is a CNN

Step 3: The Assembly Line Flattening (The Flatten Layer)

Before we can make a final decision, we have to convert the processed image data into a single, long list of numbers.

- **Action:** All the compressed feature maps (which are still 2D grids) are unrolled and stacked end-to-end to create one massive 1D vector (a long list).
- You are taking all the little sketches and notes that the inspectors and compressors made, and compiling them into **one final, organized report**.

→ This is the network's understanding of the image: a list of the *strength* of every feature it found.

What is a CNN

Step 4: The Final Decision-Maker (The Fully Connected Layers)

The fully connected layers are the "brain" or the traditional part of the neural network.

- **Action:** The long list of features from the Flatten layer is passed through several layers of standard neurons.
- This is the executive decision. The network asks:
 - "If I see a lot of **whiskers**, an **ear shape**, and some **fur texture**, does that usually mean 'Cat' or 'Dog'?"
 - The connections between these final neurons (the weights) are trained to weigh these features and combine them to reach a conclusion.

What is a CNN

The Output: The Prediction

- **Action:** The final layer calculates the probability for every possible outcome.
- The network gives its final confidence score:
 - **Cat:** 95%
 - **Dog:** 3%
 - **Bird:** 1%
 - **Truck:** 1%
- The highest score is the network's final answer: **Cat!**

What is a CNN

The Learning Process: Training a CNN

The entire training process relies on two passes through the network:

Forward Pass

- Input data flows from left to right, computing the output prediction
- The **Loss** (or Cost) is calculated by comparing the prediction $\hat{y}$ to the true label y .

Backward Pass (Backpropagation)

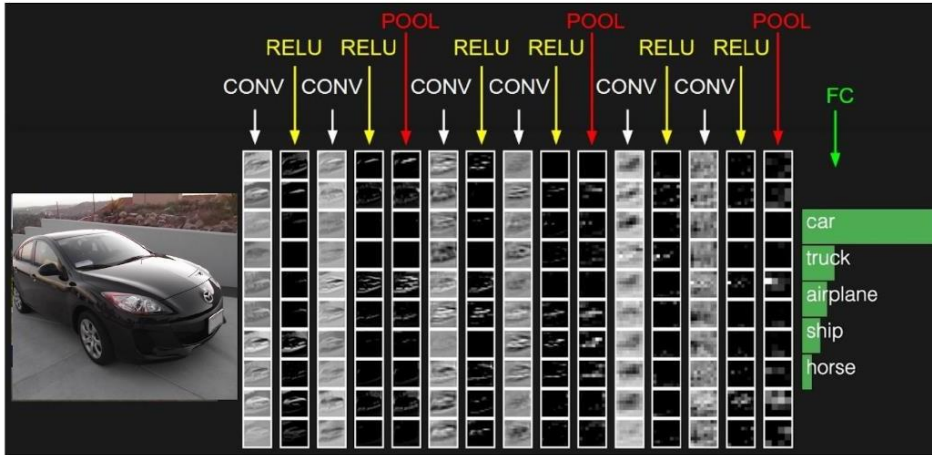
- The gradients of the Loss function with respect to every weight in the network are calculated using the **Chain Rule**.
- Crucially, the CNN layers (Conv and Pooling) use the same backpropagation method as standard NNs, but applied through their specific operational geometry.

Optimization

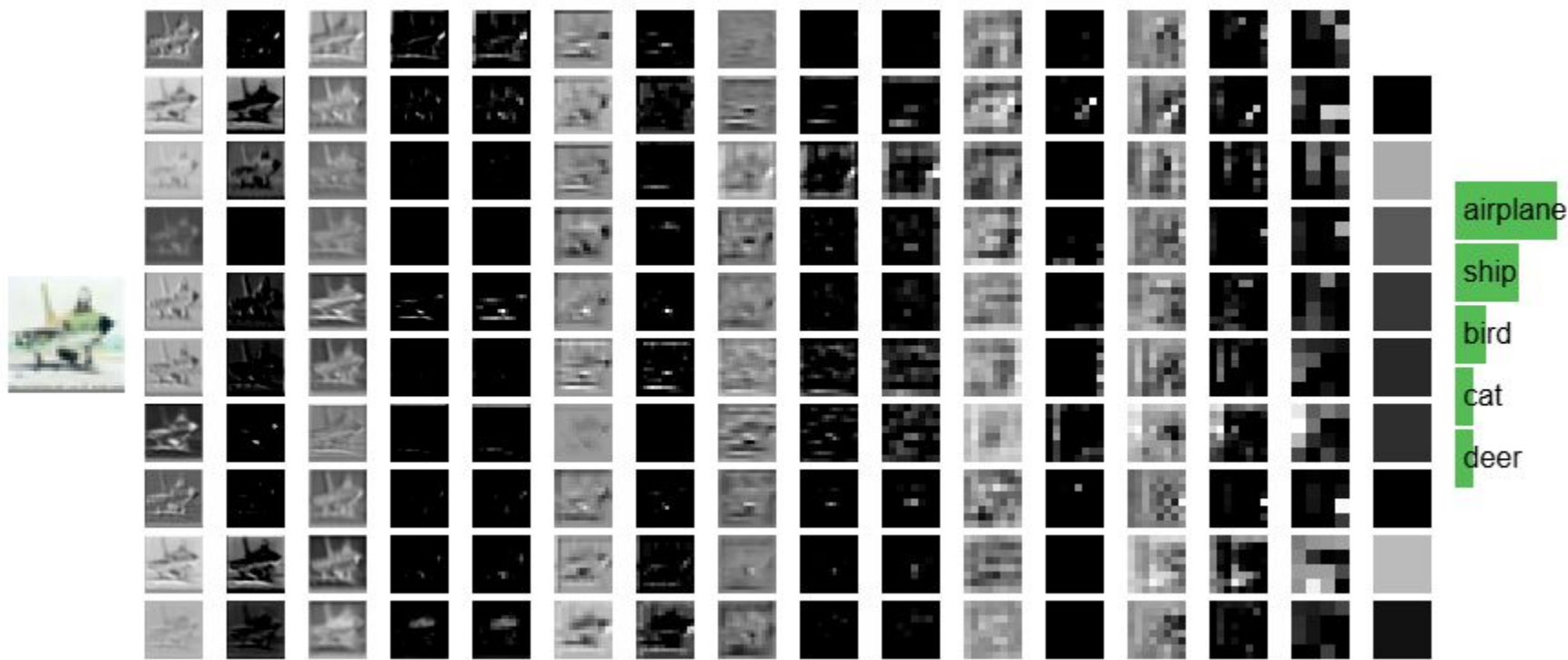
- The **Optimizer** (e.g., Adam, SGD) uses the calculated gradients to update the weights, moving the network toward a minimum of the loss function.

Fully Connected Layer (FC layer)

- Contains neurons that connect to the entire input volume, as in ordinary Neural Networks



CS231n

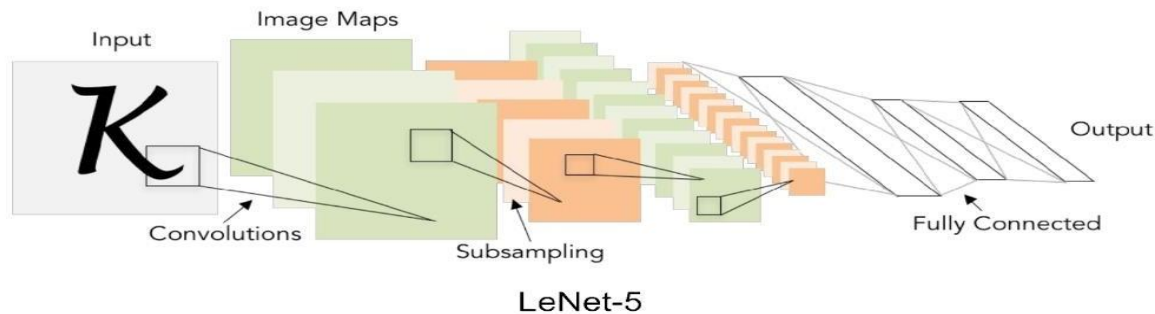
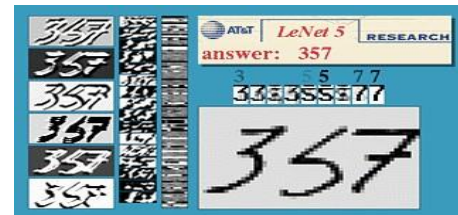


*This network is running live in your browser

A bit of history

Gradient-based learning applied to document recognition

[LeCun, Bottou, Bengio, Haffner 1998]



A bit of history

ImageNet Classification with Deep Convolutional Neural Networks

[Krizhevsky, Sutskever, Hinton, 2012]

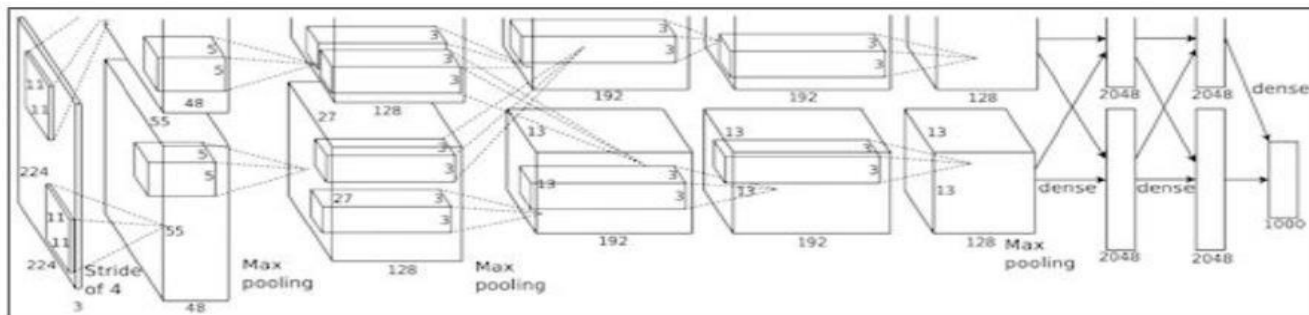


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

“AlexNet”

[CNN Visualization - MNIST](#)

CNN Architectures (ImageNet)

IMAGENET

14,197,122 images, 21841 synsets indexed

[Explore](#) [Download](#) [Challenges](#) [Publications](#) [CoolStuff](#) [About](#)

Not logged in. [Login](#) | [Signup](#)

ImageNet is an image database organized according to the **WordNet** hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

[Click here](#) to learn more about ImageNet, [Click here](#) to join the ImageNet mailing list.



<http://www.image-net.org/>

A very large labeled images dataset
Currently: 14.197.122 images in
21.841 categories

CNN Architectures (ImageNet)

IMAGENET Large Scale Visual Recognition Challenge (ILSVRC)

Competition

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

For details about each challenge please refer to the corresponding page.

- [ILSVRC 2017](#)
- [ILSVRC 2016](#)
- [ILSVRC 2015](#)
- [ILSVRC 2014](#)
- [ILSVRC 2013](#)
- [ILSVRC 2012](#)
- [ILSVRC 2011](#)
- [ILSVRC 2010](#)

- **LeNet** 1998 (the grandfather)
- **AlexNet 2012**. The first work that popularized Convolutional Networks in Computer Vision was the AlexNet, developed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton. The AlexNet was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up (top 5 error of 16% compared to runner-up with 26% error). The Network had a very similar architecture to LeNet, but was deeper, bigger, and featured Convolutional Layers stacked on top of each other (previously it was common to only have a single CONV layer always immediately followed by a POOL layer).
- **ZF Net**. ILSVRC 2013 winner
- **GoogLeNet**. ILSVRC 2014 winner
- **VGGNet**. The runner-up in ILSVRC 2014.
- **ResNet**. winner of ILSVRC 2015.

CNN Architectures

(LeNet $\rightarrow$ ImageNet)

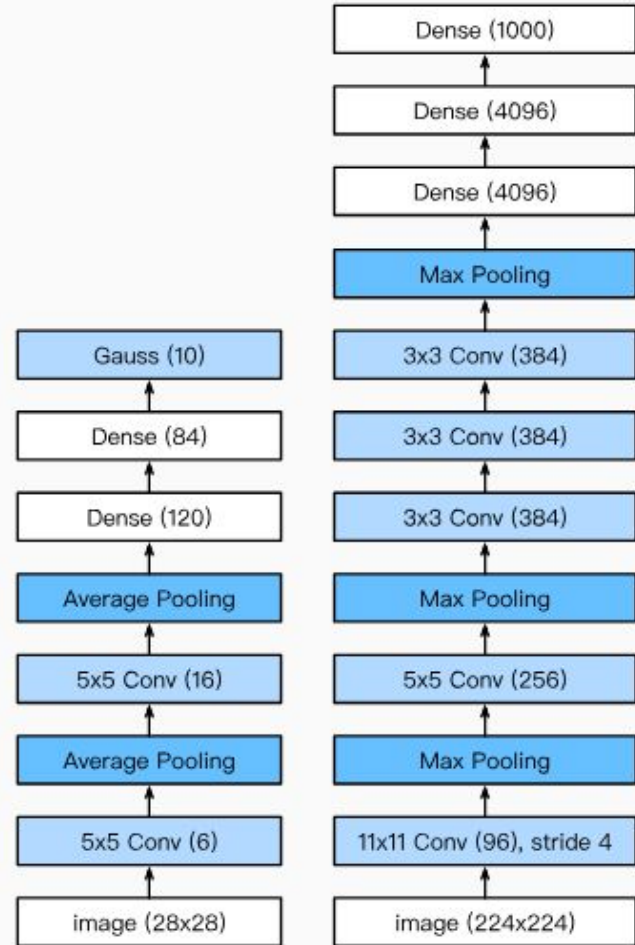
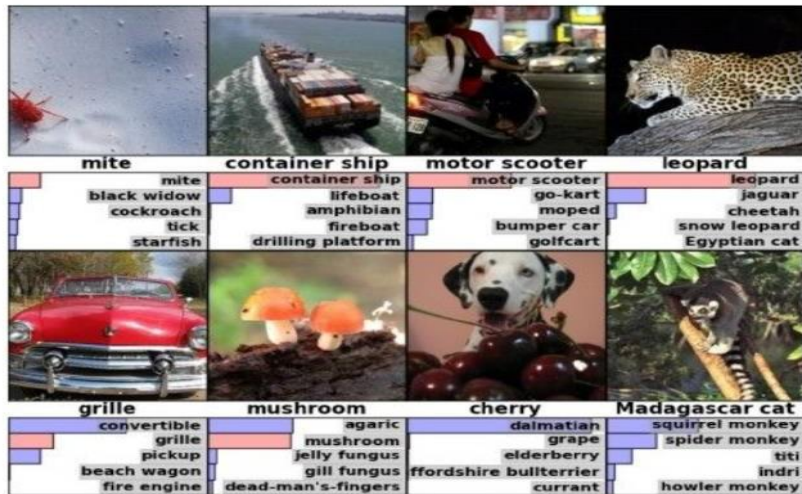


Fig. 6.10 LeNet (left) and AlexNet (right)

Fast-Forward to today: ConvNets are everywhere

Classification



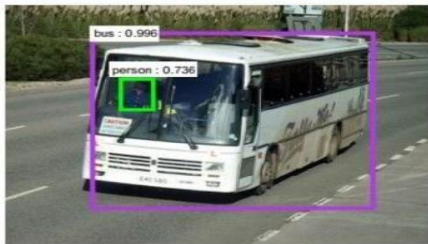
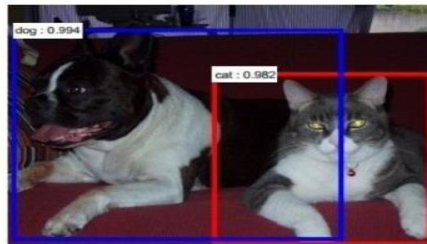
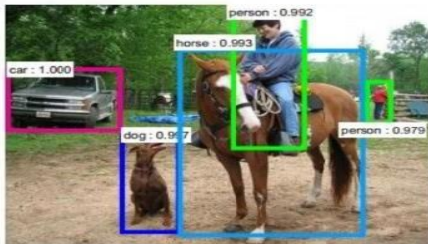
Retrieval



Figures copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Fast-Forward to today: ConvNets are everywhere

Detection



Figures copyright Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, 2015. Reproduced with permission.

[Faster R-CNN: Ren, He, Girshick, Sun 2015]

Segmentation



Figures copyright Clement Farabet, 2012.
Reproduced with permission.

[Farabet et al., 2012]

Transfer Learning

“You need a lot of a data if you want to
train/use CNNs”

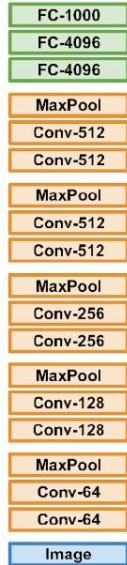
Transfer Learning

“You need a lot of data if you want to train/use CNNs”

BUSTED

Transfer Learning with CNNs

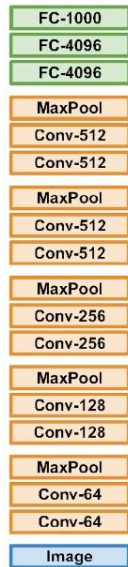
1. Train on Imagenet



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

Transfer Learning with CNNs

1. Train on Imagenet



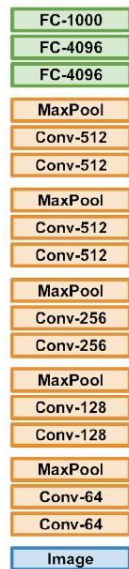
2. Small Dataset (C classes)



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

Transfer Learning with CNNs

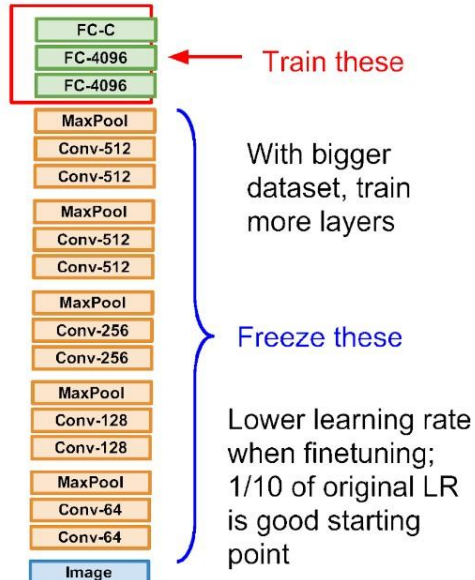
1. Train on Imagenet



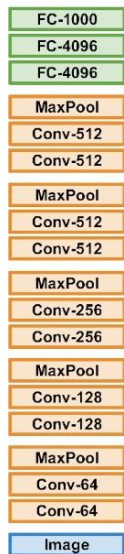
2. Small Dataset (C classes)



3. Bigger dataset



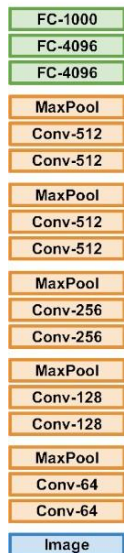
Donahue et al, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", ICML 2014
Razavian et al, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition", CVPR Workshops 2014



More specific

More generic

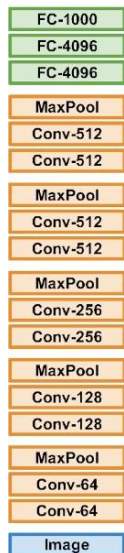
	very similar dataset	very different dataset
very little data	?	?
quite a lot of data	?	?



More specific

More generic

	very similar dataset	very different dataset
very little data	Use Linear Classifier on top layer	?
quite a lot of data	Finetune a few layers	?



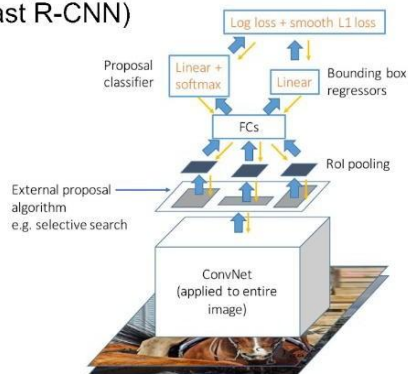
More specific

More generic

	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

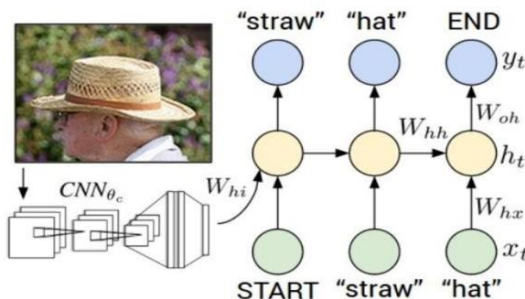
Transfer learning with CNNs is pervasive... (it's the norm, not an exception)

Object Detection (Fast R-CNN)



Girshick, "Fast R-CNN", ICCV 2015
Figure copyright Ross Girshick, 2015. Reproduced with permission.

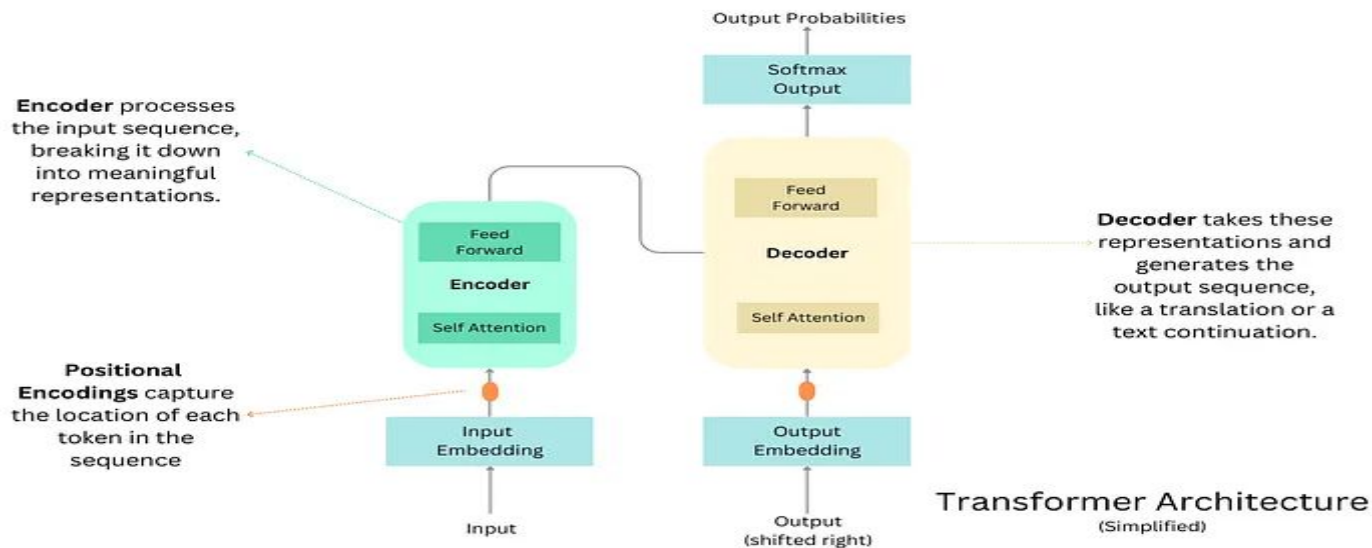
Image Captioning: CNN + RNN



Karpathy and Fei-Fei, "Deep Visual-Semantic Alignments for
Generating Image Descriptions", CVPR 2015
Figure copyright IEEE, 2015. Reproduced for educational purposes.

Transformer

A deep learning architecture that fundamentally changed artificial intelligence by using a mechanism called self-attention to process data in parallel, overcoming the limitations of previous sequential models.



This innovation has been key to the development of modern large language models (LLMs) like GPT and Llama and modern Large Visual Language models (LVLMs)

Bibliography

1. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
2. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786), 504-507.
3. Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097- 1105).

Image Segmentation



Types of Image Segmentation



SEMANTIC IMAGE
SEGMENTATION



INSTANCE
SEGMENTATION



PANOPTIC
SEGMENTATION

Method of Image Segmentation



Time for practice

- [Image Basic Analysis Code.ipynb](#)
- [Image Classification using Hugging Face Transformers](#)
- [Object Detection using Hugging Face Transformers](#)
- [Object Segmentation using Hugging Face Transformers](#)