I 202: Information Organization & Retrieval Fall 2025

Class 16: Image Classification

Today's Outline

Image Classification

Data Creation & Augmentation

Image Convolutions

Convolutional Neural Nets

Image Processing Algorithms

- Today: Image recognition brief intro, CNNs
- Later: Transformer-based image models

Slides modified from: Kraus, U Hawai'I, Manoa, Snavely CS 5670 Cornell; CS 4782 Cornell

IMAGE CLASSIFICATION

Image classification is the task of taking an input image and outputting a class or a **probability of classes** that best describes the image

input image

classification

“dog”

input image

classification

“cat”

Image Classification Application: Autonomous Driving

IMAGE ANALYSIS CHALLENGE: VARIABLE VIEWPOINT

Michelangelo 1475-1564

CHALLENGE:
VARIABLE ILLUMINATION DIFFERENCES

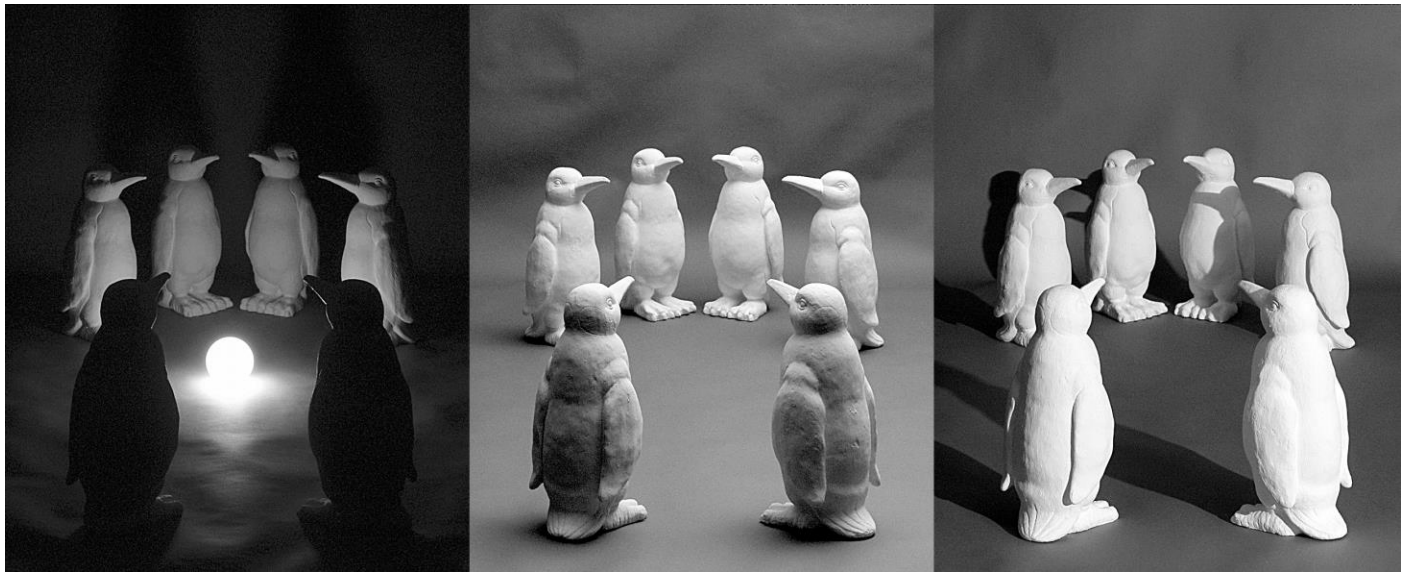
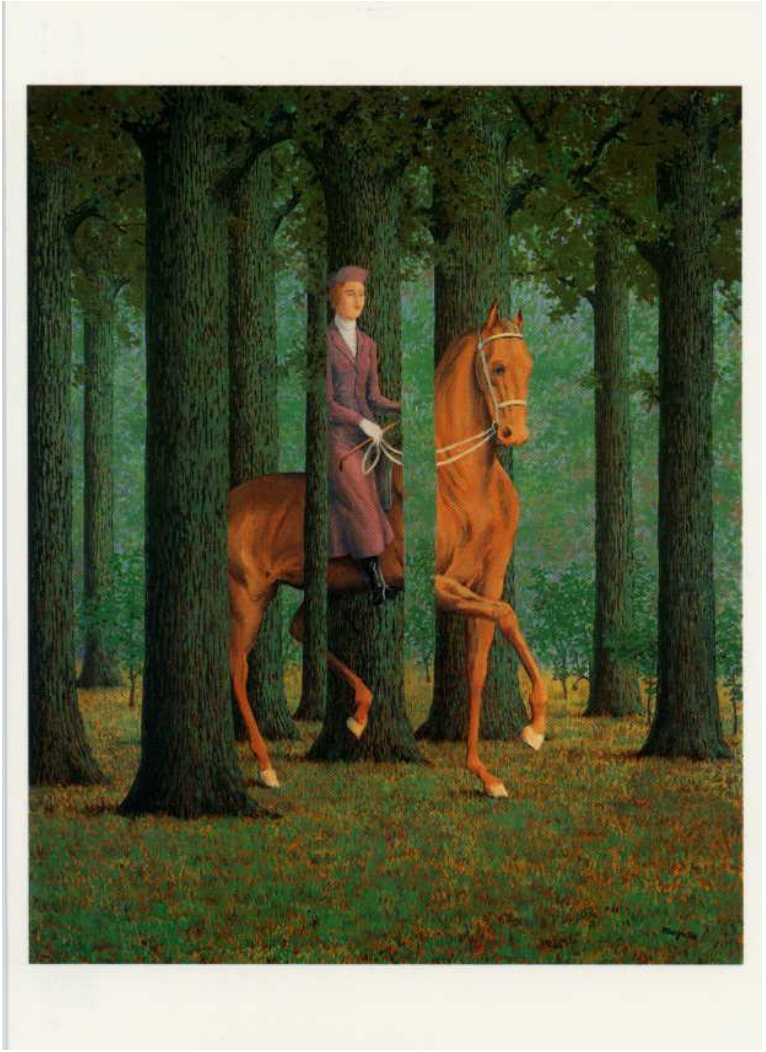


image credit: J. Koenderink

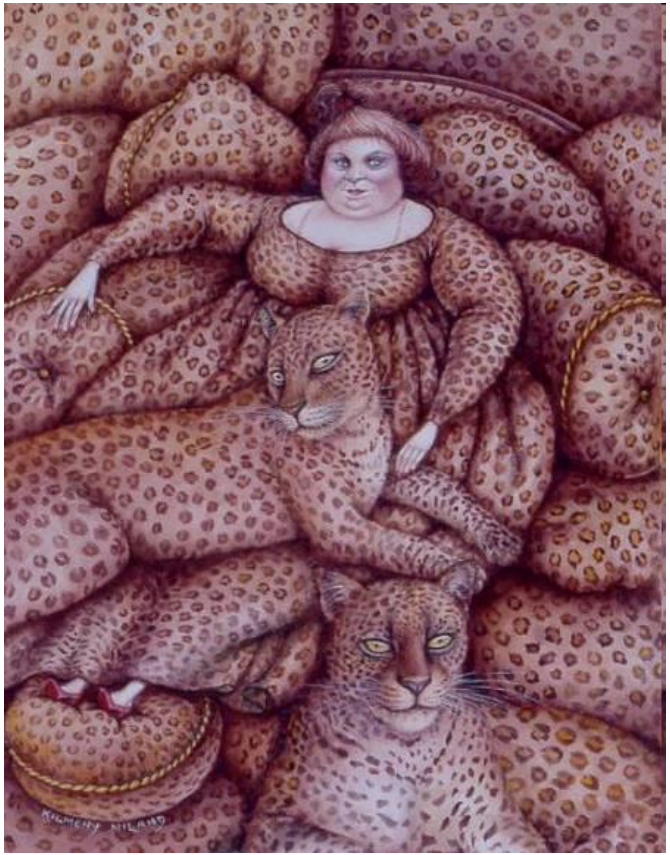
CHALLENGE: DEFORMATION

Challenge: Occlusion



Magritte, 1957

CHALLENGE: BACKGROUND CLUTTER



Kilmeny Niland. 1995

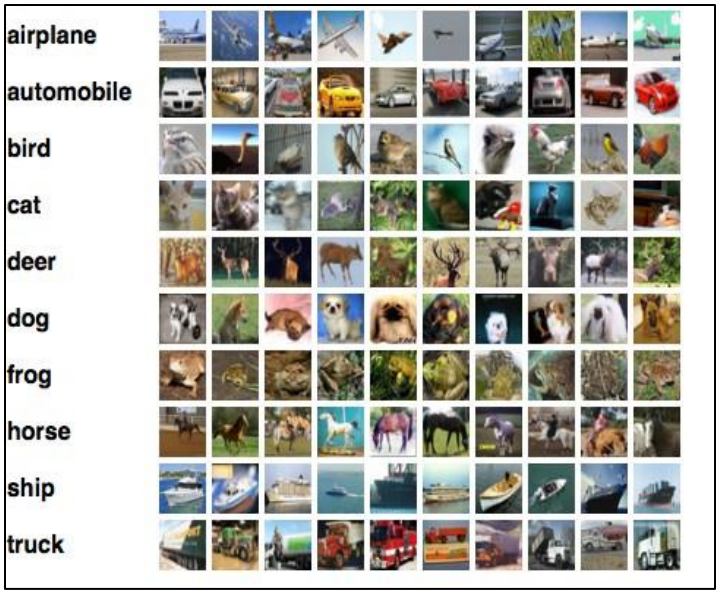
Challenge: Intra-Class Variations

Data Gathering & Augmentation

IMAGE CATEGORY DATASET

CIFAR-10 Dataset (https://www.cs.toronto.edu/~kriz/cifar.html)

•Consists of 60,000 32x32 **color** images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images.



ImageNet: A Data Gathering Breakthrough
> 14 M Images and 20,000 Categories!

IMAGENET AND WORDNET

WordNet inspired the creation of ImageNet and provides its structure

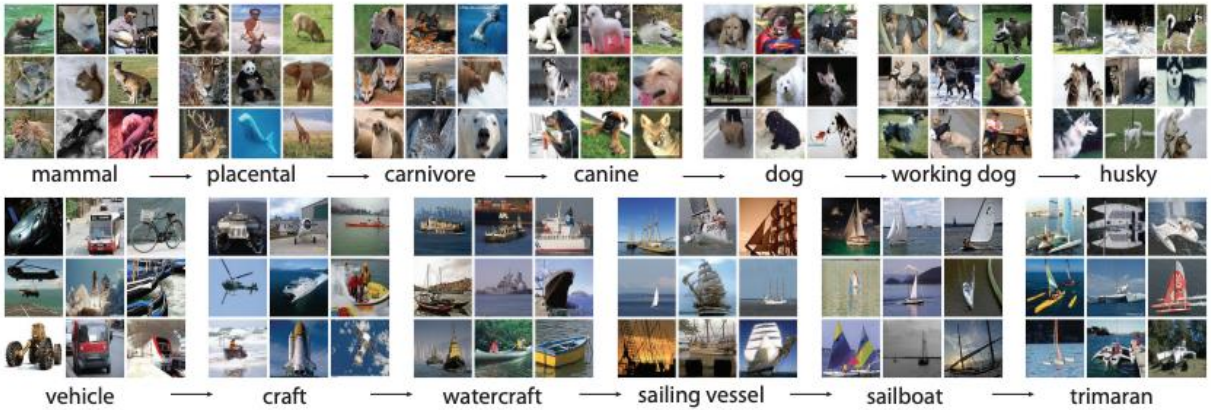
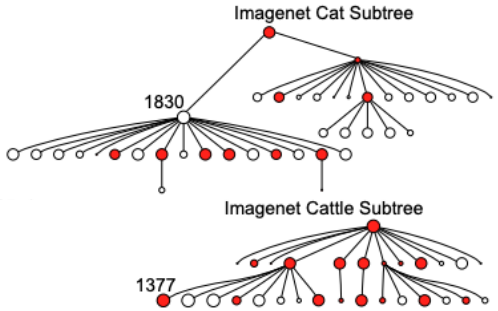


Figure 1: A snapshot of two root-to-leaf branches of ImageNet: the **top** row is from the mammal subtree; the **bottom** row is from the vehicle subtree. For each synset, 9 randomly sampled images are presented.



Deng, et al. "ImageNet: A large-scale hierarchical image database." *IEEE CVPR* 2009.

IMAGENET:
A BREAKTHROUGH IN DATA COLLECTION

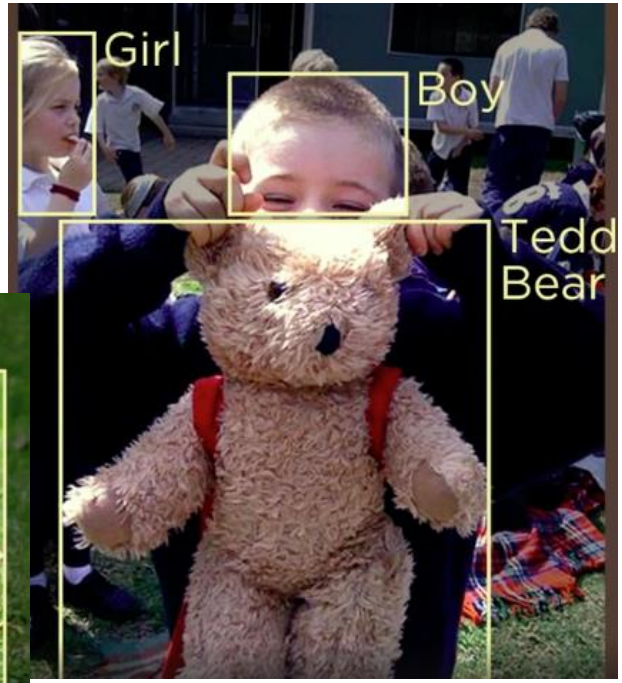
“We went to the Internet, the biggest treasure trove of pictures that humans have ever created. We downloaded nearly a billion images and used crowdsourcing technology like the Amazon Mechanical Turk platform to help us to label these images. At its peak, ImageNet was one of the biggest employers of the Amazon Mechanical Turk workers: together, almost 50,000 workers from 167 countries around the world helped us to clean, sort and label nearly a billion candidate images. That was how much effort it took to capture even a fraction of the imagery a child's mind takes in in the early developmental years”. – Fei Fei Li

https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript

Image Data Gathering Tasks

Single object, plain background

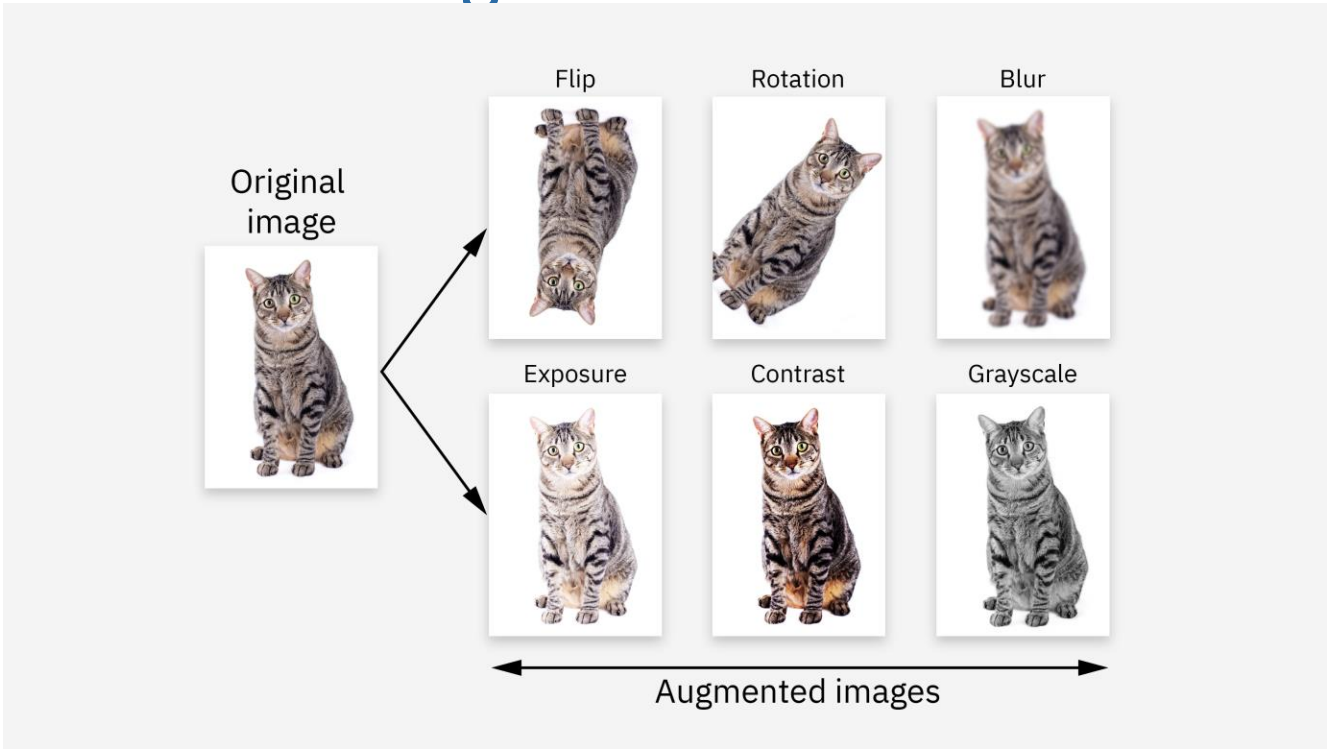
Choose a more general category when uncertain



Multiple objects

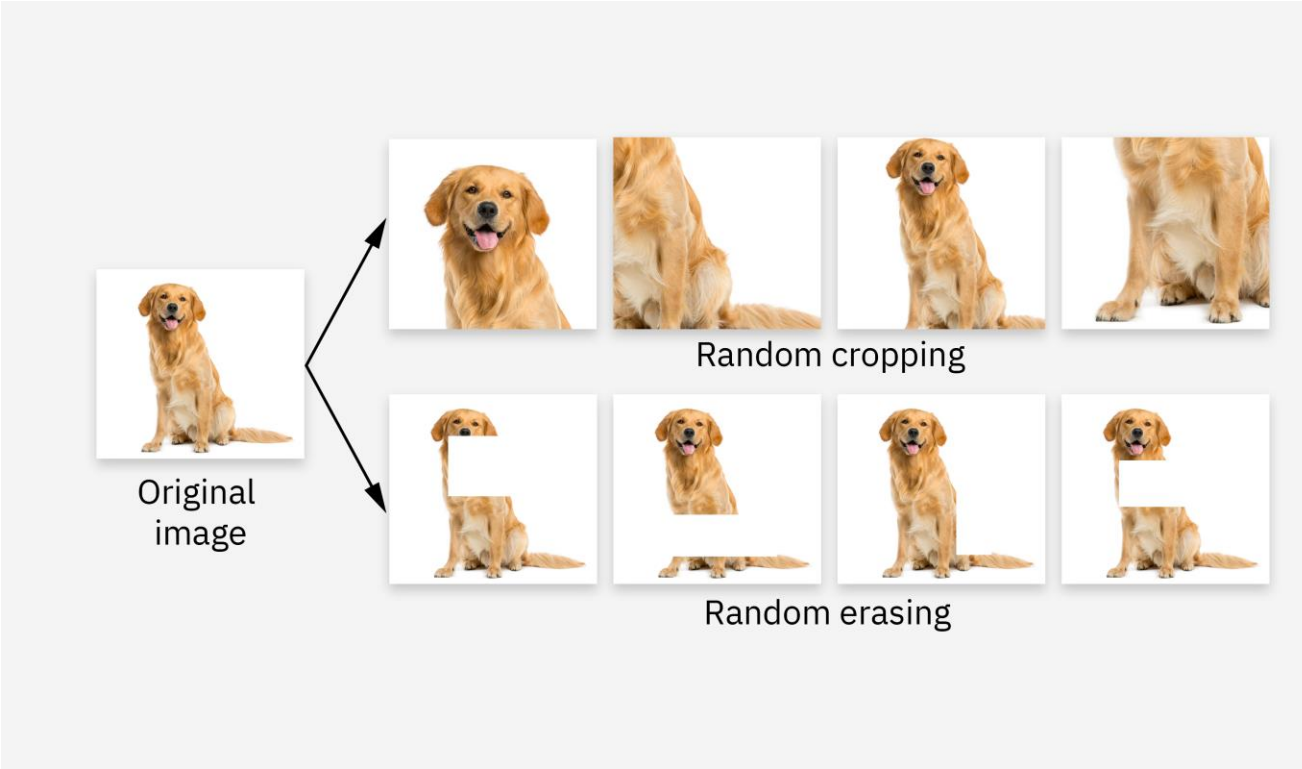
https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript

Image Data Gathering Task: Data Augmentation



https://www.ibm.com/think/topics/data-augmentation

Image Data Gathering Task: Data Augmentation



https://www.ibm.com/think/topics/data-augmentation

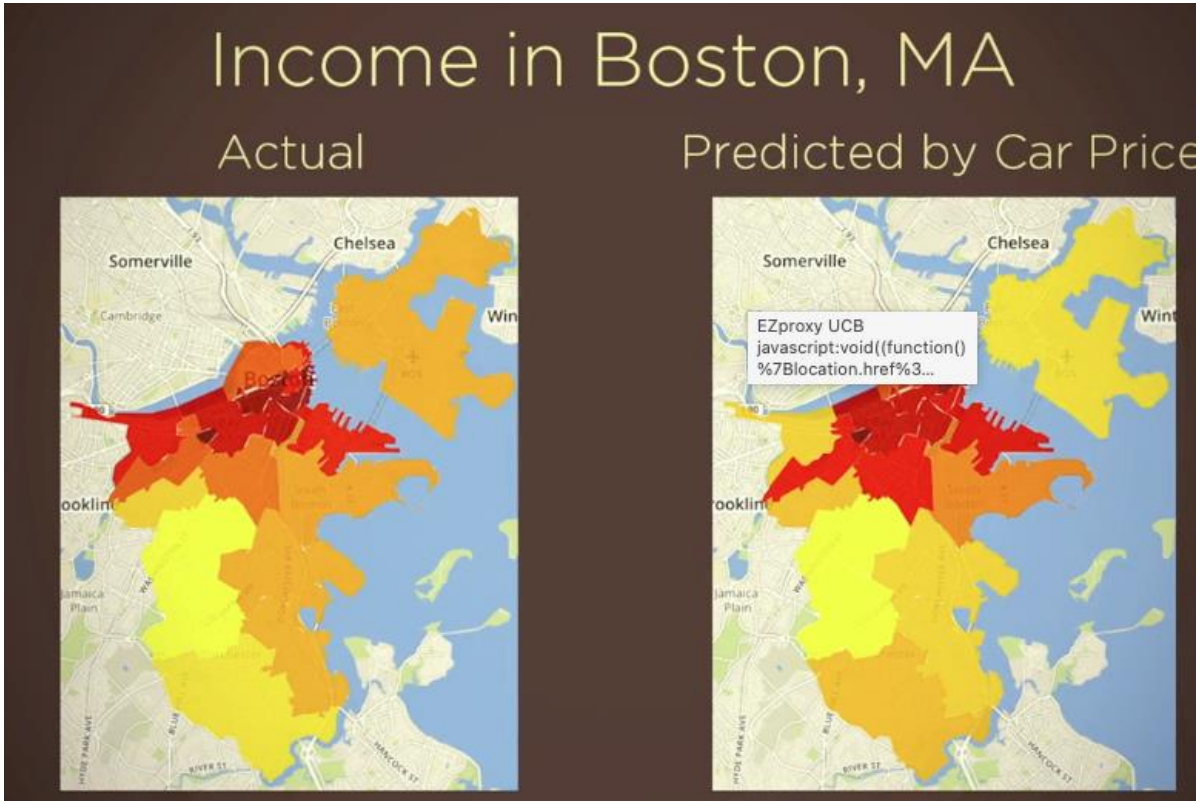
Application: Data Gathering via Image Recognition

Got millions of Google street view images, and classified the makes and models of cars

https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript

Application: Data Gathering via Image Recognition

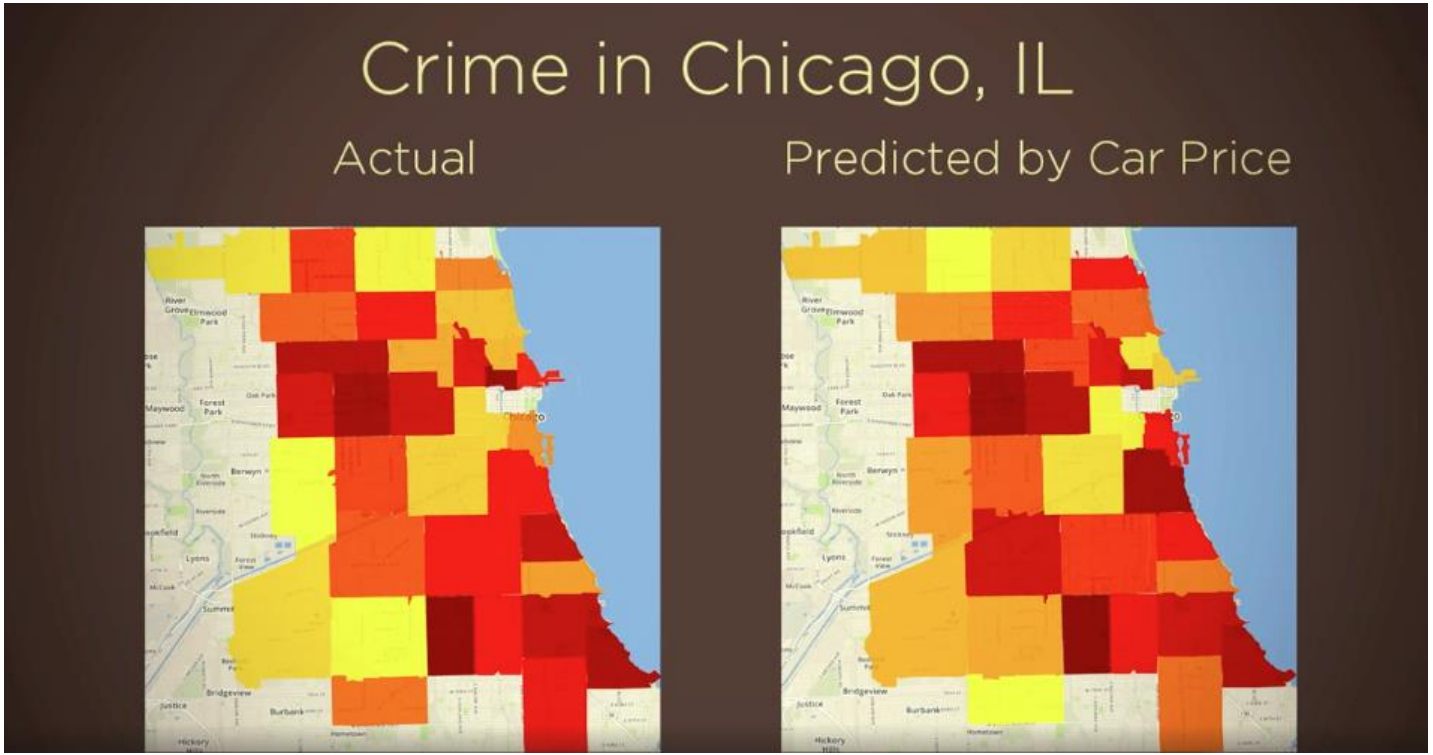
Then showed the expected correlation between car model and average income of a region



https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript

Application: Data Gathering via Image Recognition

But also found unexpected correlation between car model and crime rate



https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures/transcript

Ensuring the Category Labels are VERY Accurate

- “I ended up training on 500 validation images and then switched to the test set of 1500 images. The labeling happened at a rate of about 1 per minute, but this decreased over time. I only enjoyed the first ~200, and the rest I only did *#forscience*. ….
- The labeling time distribution was strongly bimodal: Some images are easily recognized, while some images (such as those of fine-grained breeds of dogs, birds, or monkeys) can require multiple minutes of concentrated effort. I became very good at identifying breeds of dogs.”

https://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet

Ensuring the Category Labels are VERY Accurate

Labeling Interface

I developed a labeling interface that would help us evaluate the human performance. It looked similar to, but not identical, to the screenshot below:

https://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet

Exercise: Try the Labeling Interface
Scroll Down to See the Category Choices

https://cs.stanford.edu/people/karpathy/ilsvrc/

Show answer | Show google prediction

Tibetan mastiff

GoogLeNet predictions:

Tibetan mastiff

Bernese mountain dog

Newfoundland, Newfoundland dog

Gordon setter

Appenzeller

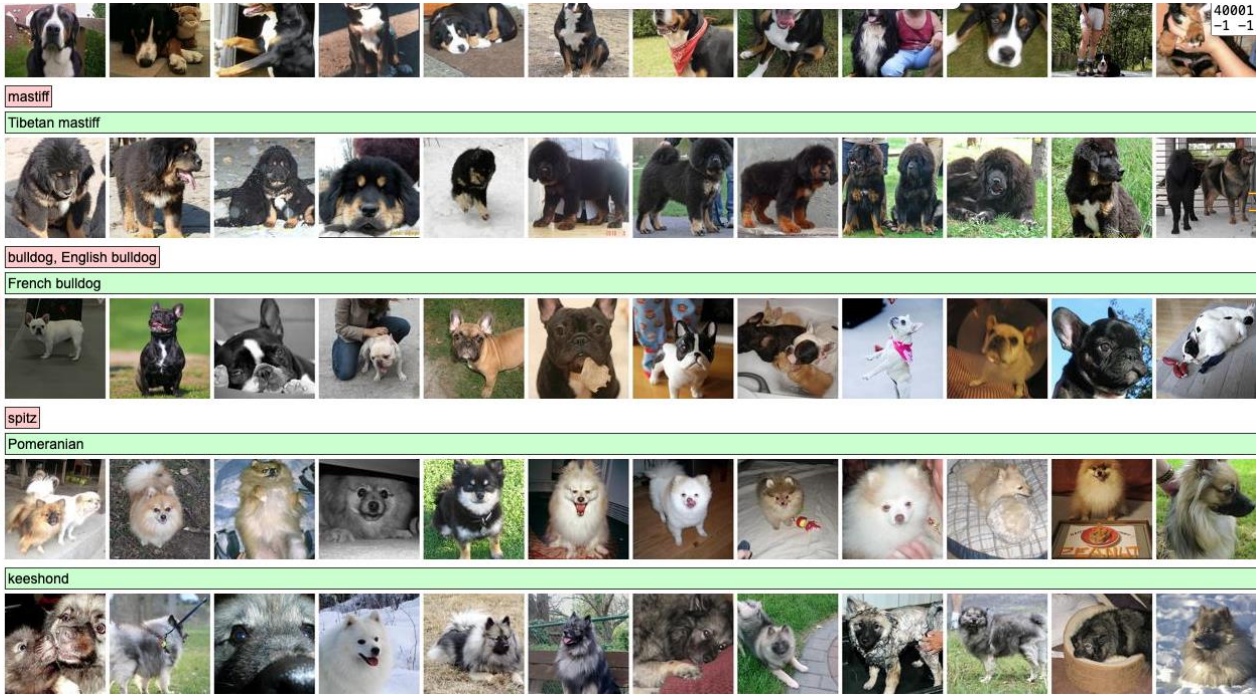
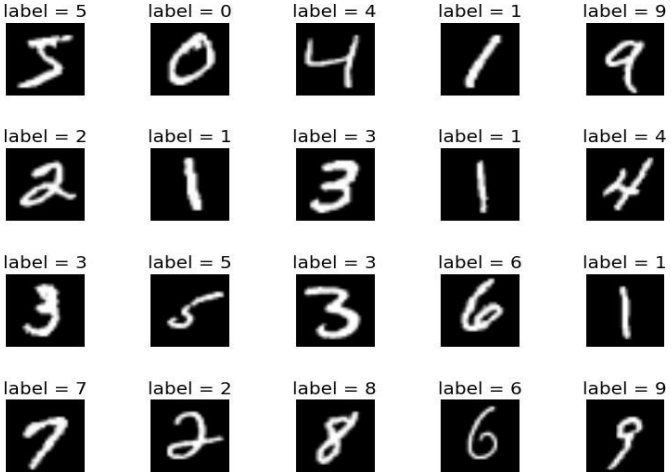


Image Representation & Convolutions

MNIST: A Classic Image Dataset

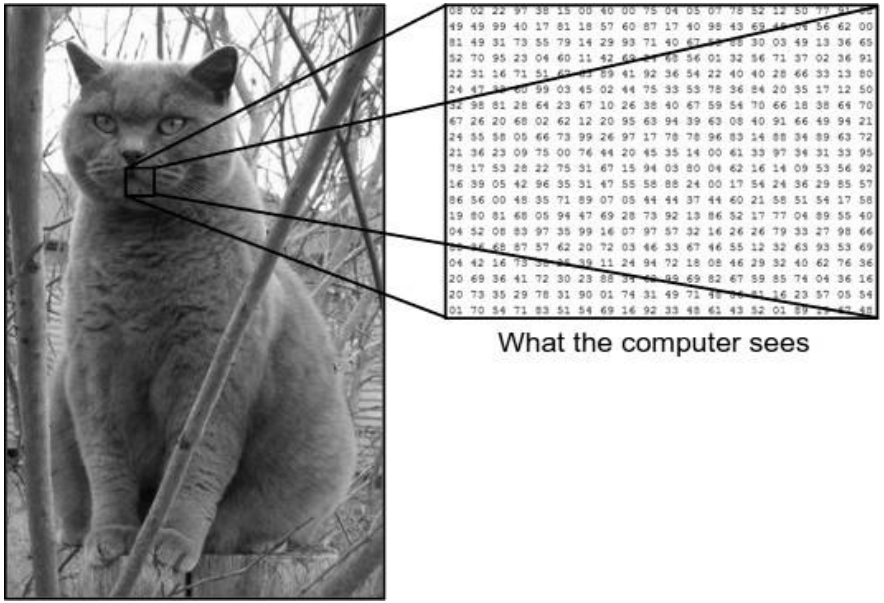
MNIST Dataset

- 60,000 training examples
- 10,000 test examples
- Rank of best Classifiers and Errors



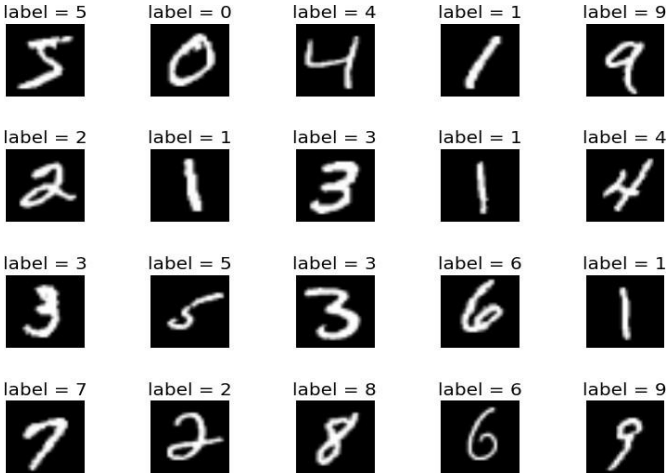
How Computers Represent Images

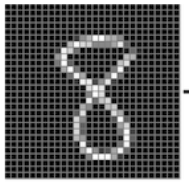
What we see vs. What computers see



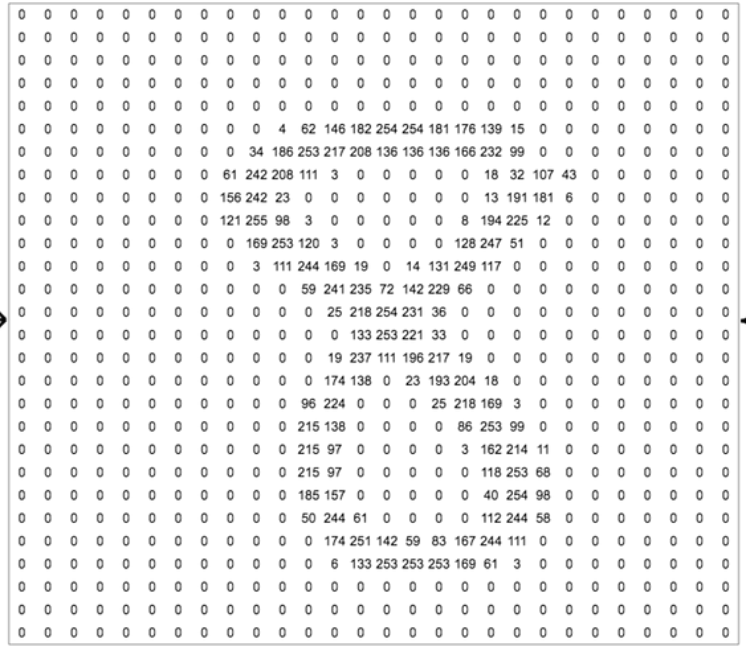
What the computer sees

MNIST Dataset

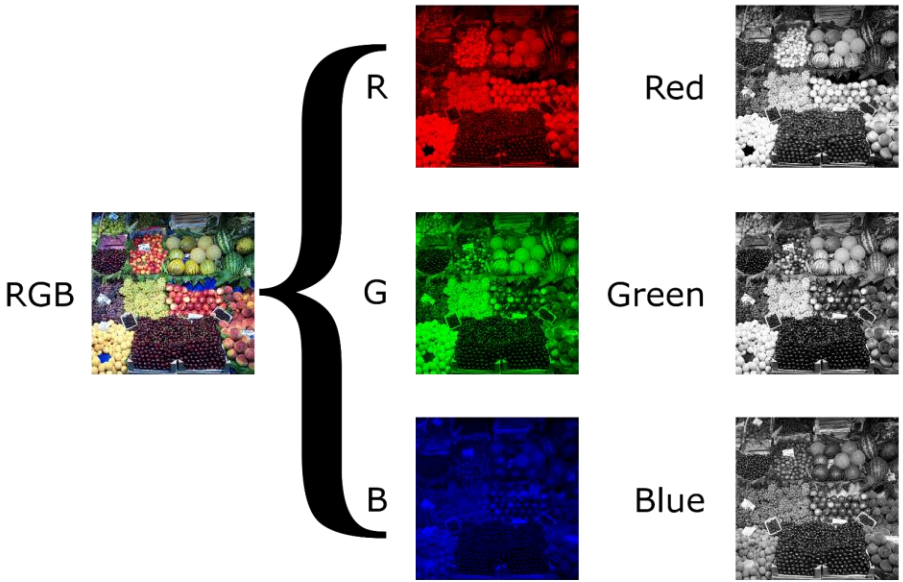




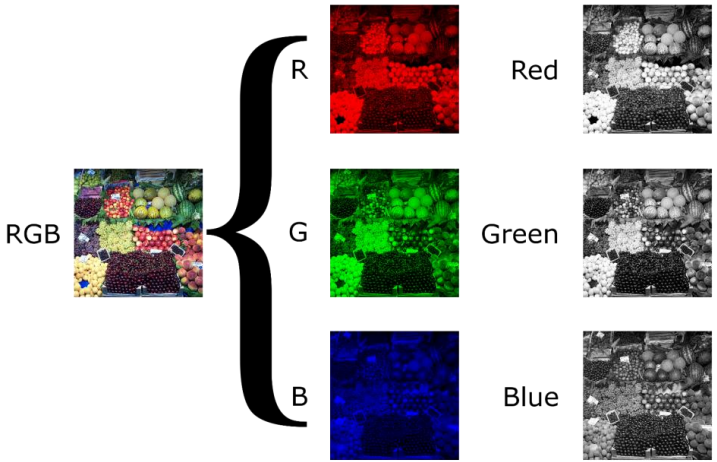
28 x 28
784 pixels

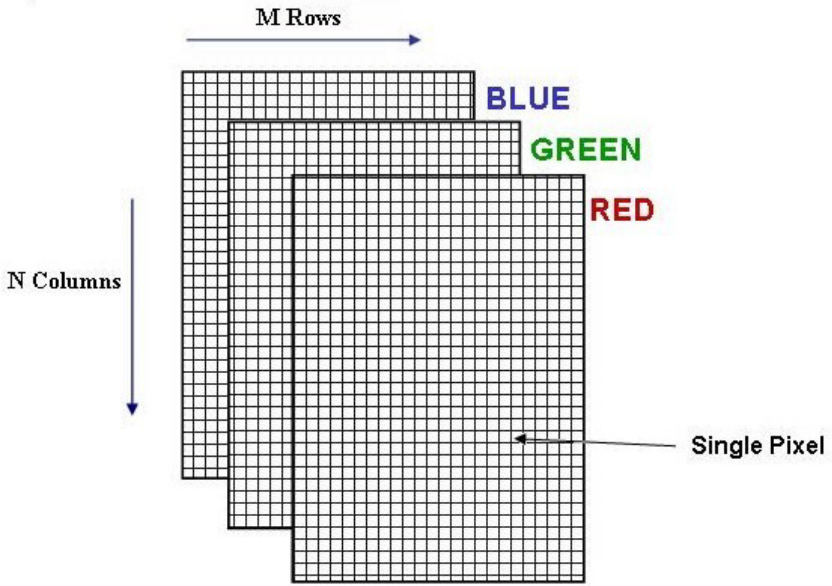


Color Images

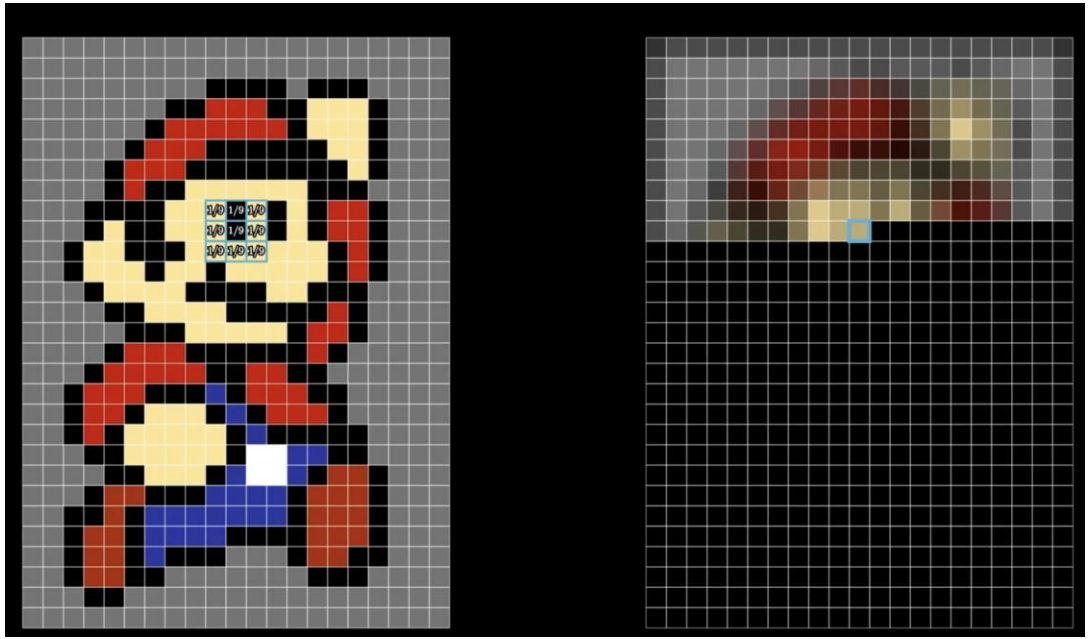


Color Images



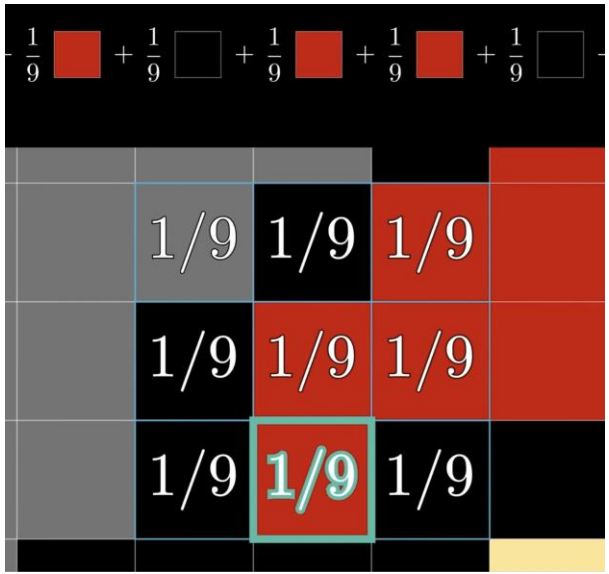


Convolutions to "average" over an Image



But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

Convolutions over an Image



Create a grid of weights
In this case, 3x3 grid
Equally weighted with 1/9

Move the grid along the image

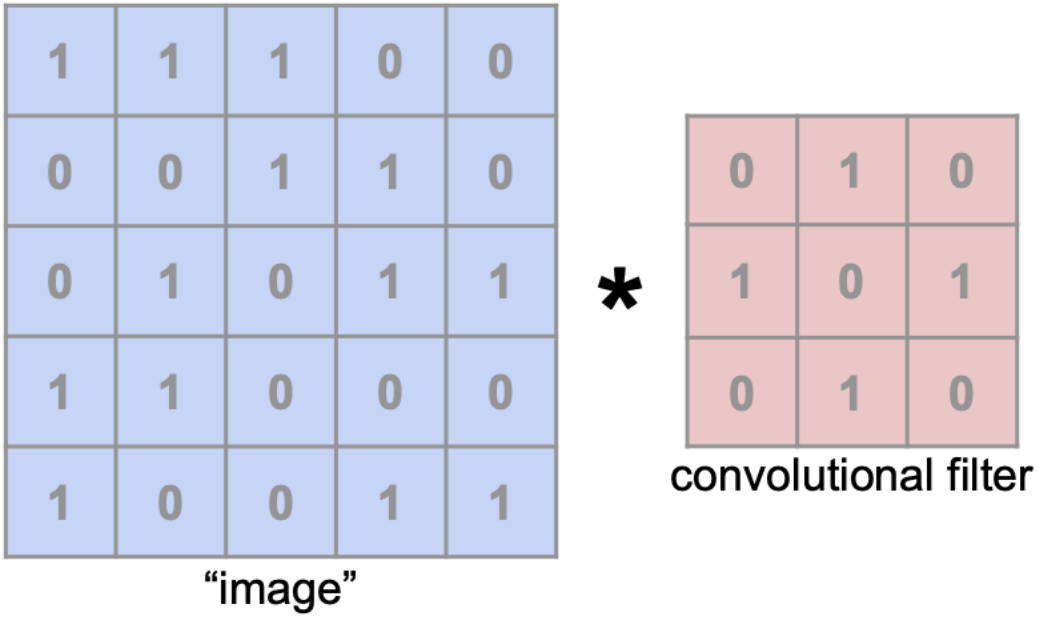
At each position in the image, multiply the grid values by the value of the color of the underlying pixel

Place that value in a new image

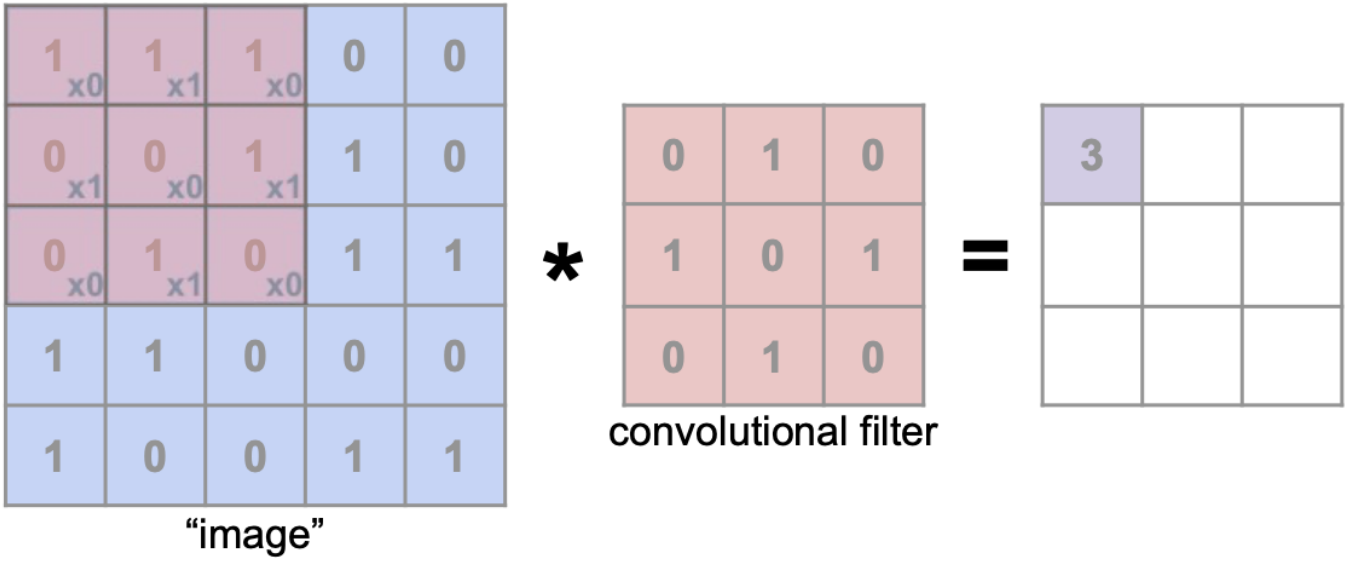
That value is the average color of the 9 pixels

But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

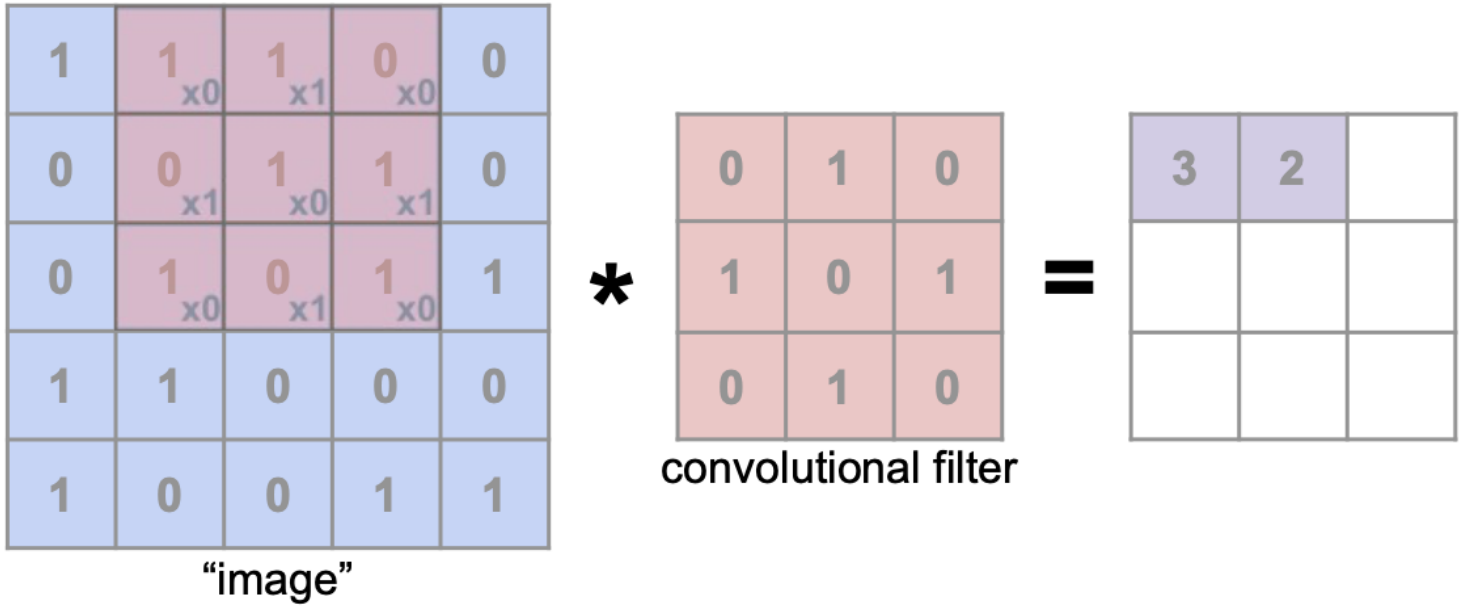
Convolutions Filters to "Average" over an Image

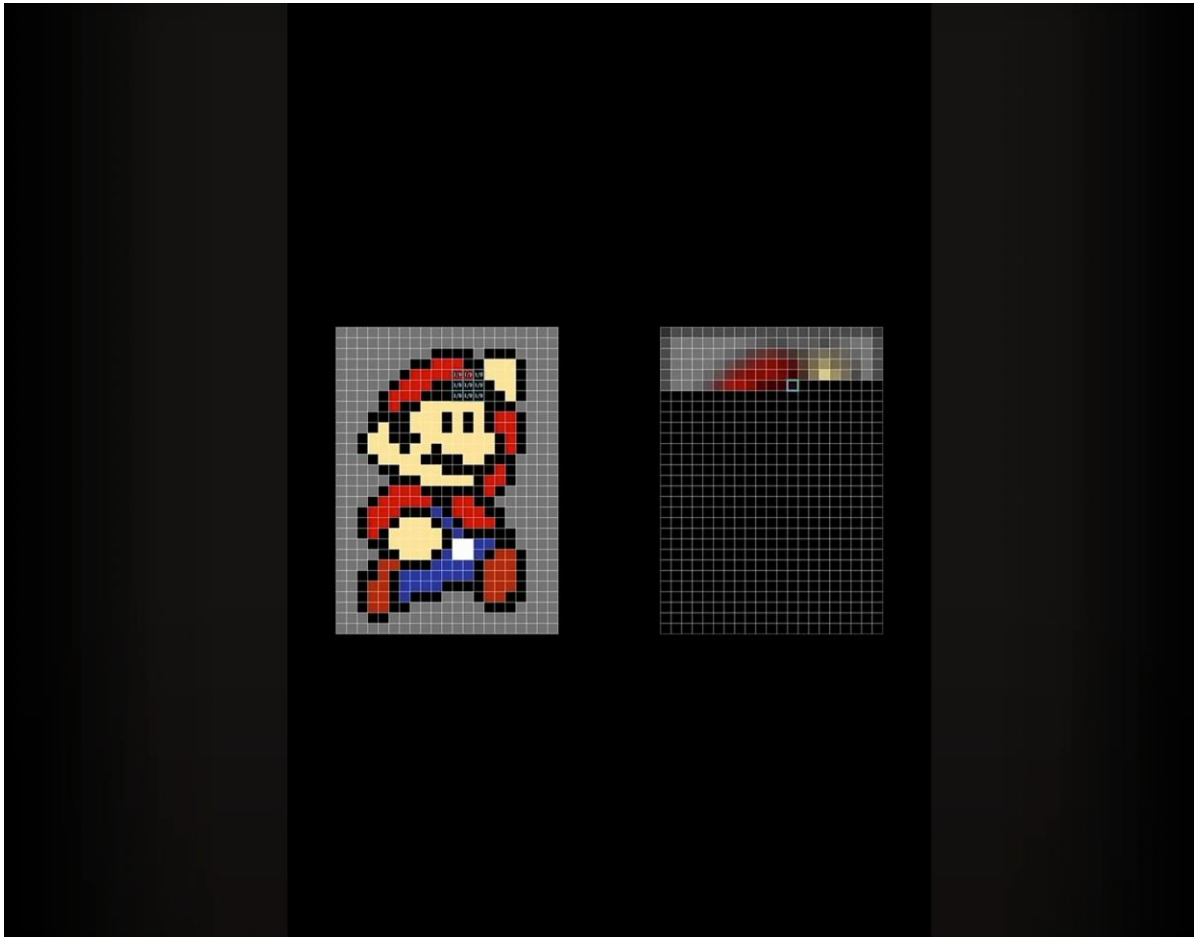


Convolutions Filters to "Average" over an Image



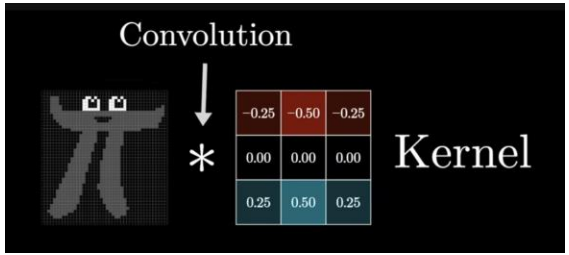
Convolutions Filters to "Average" over an Image

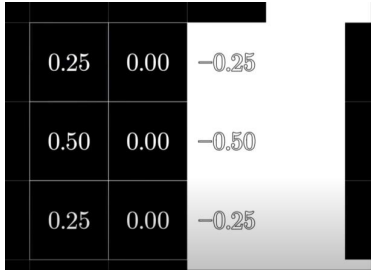


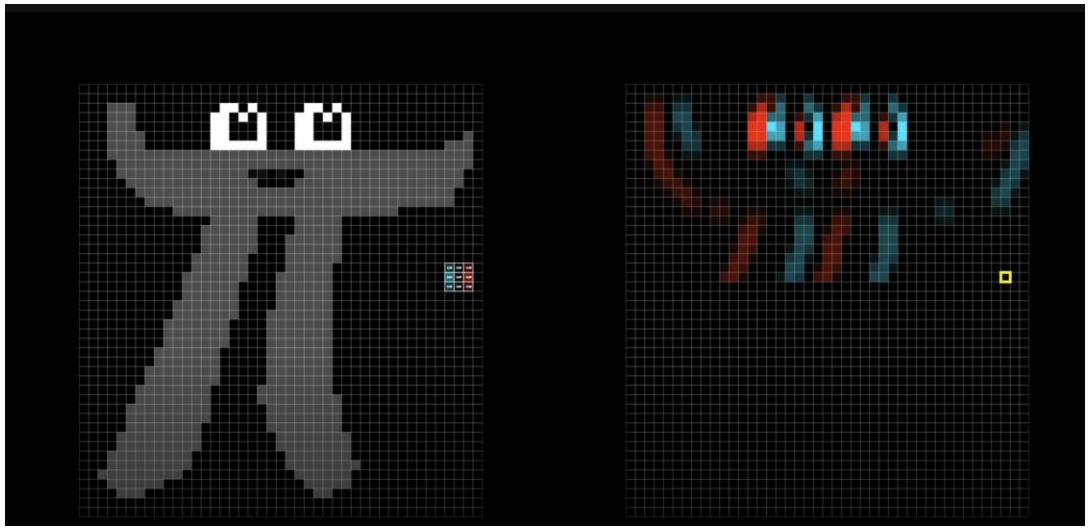


But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

Convolutions For Edge Detection

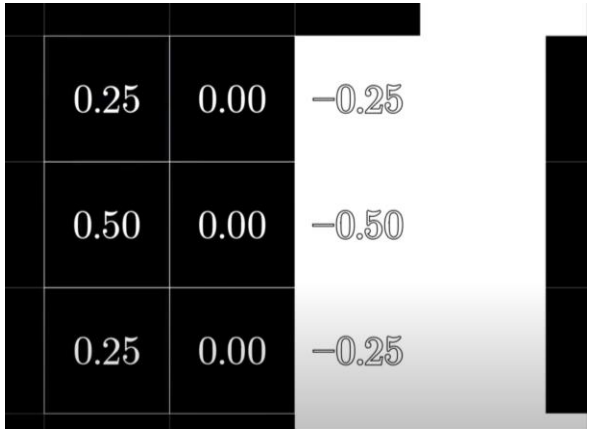






But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

Convolutions For Edge Detection

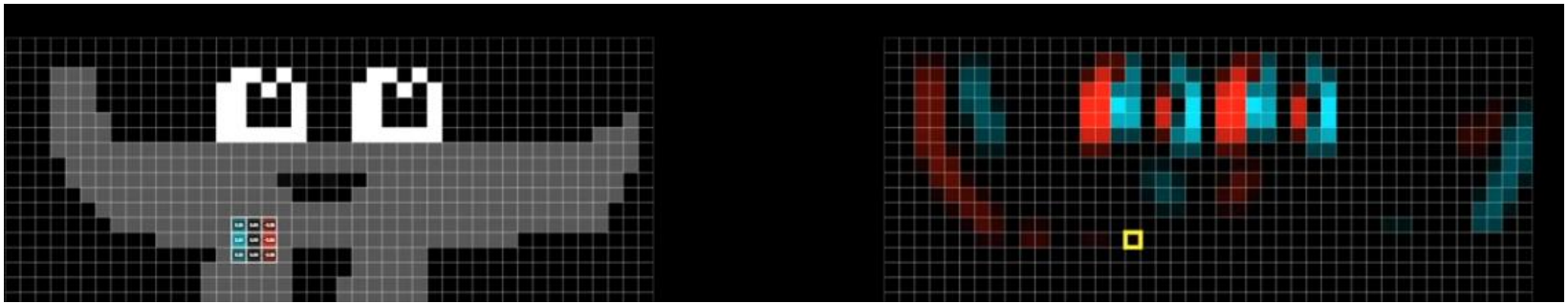


Here, black pixels are 0
So left & middle column are 0
(multiply by 0 = 0)

Here, white columns are 1
So we will get negative results in the right column (-.25, -.50, -.25)

If we color the new image:
- negative as red,
- positive as blue
- zero as black

Then we start to see red edges on the left and blue enges on the right



But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

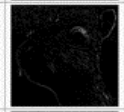
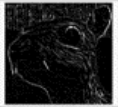
Convolutions For Edge Detection

But What is a Convolution? 3Blue1Brown https://www.youtube.com/watch?v=KuXjwB4LzSA

Varying Convolution Filters

Different filters will produce different feature maps for the same input image

Input image

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Exercise: Match the Convolutions with the Output They Produce

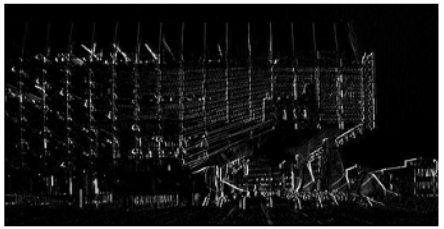
Match the following convolutional filters with the output they produce.

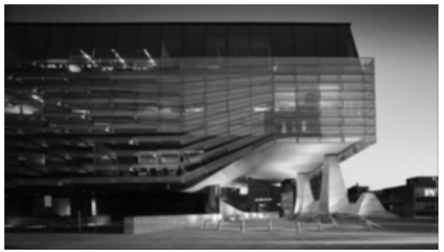
input image

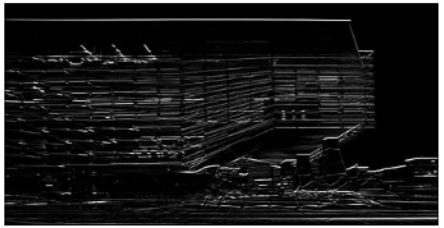
-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9





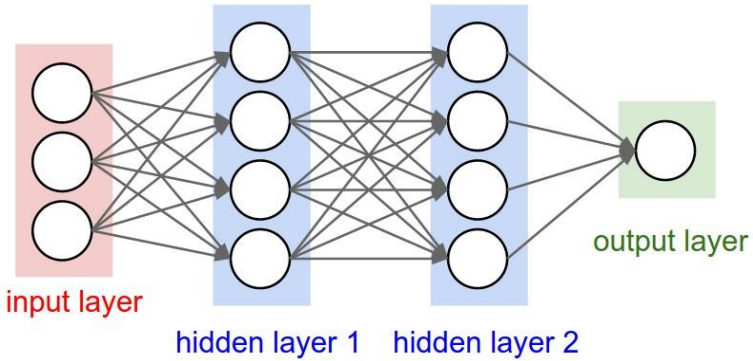


Convolutional Neural Networks (CNNs)

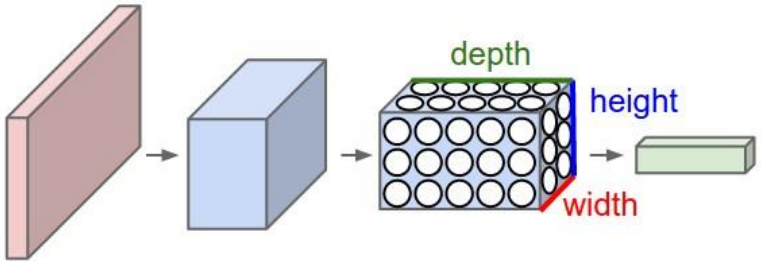
Pre-Deep Learning Image Classification

- Hand-Crafted Features
- **Texture Features**: Histogram based, Entropy, Haralick features (Co-occurrence matrix), Gray-level run length metrics, Local Binary Pattern, Fractal, etc.
- **Morphological Features**: Hu's moments, Shape features, Granulometry, Bending Energy, Roundness ratio, etc

Standard vs Convolutional Neural Nets



A standard 3-layer Neural Network.



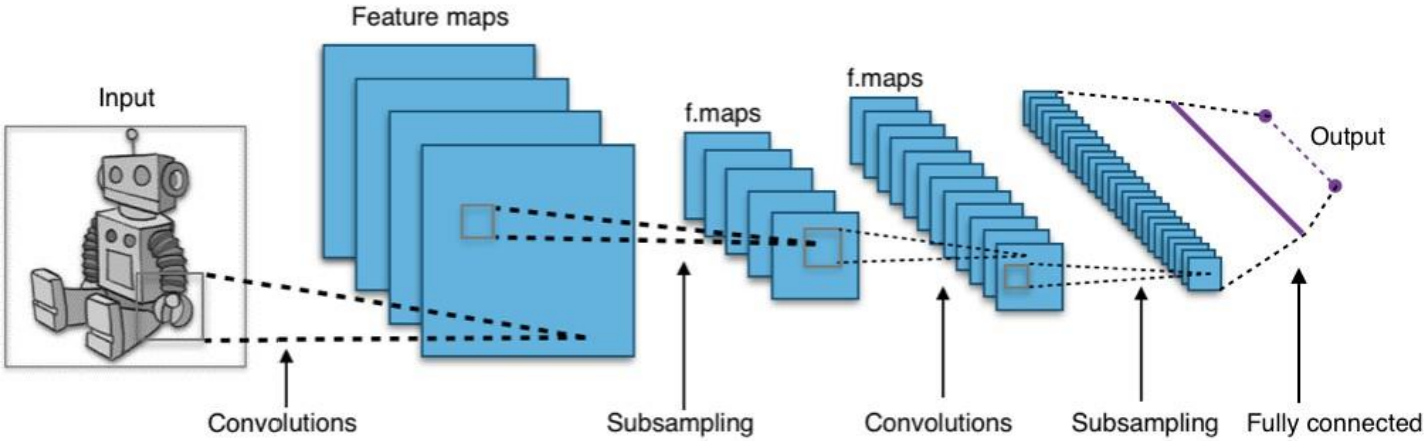
A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations.

https://cs231n.github.io/convolutional-networks/

Convolutional Neural Networks (CNNs)

Network Architecture

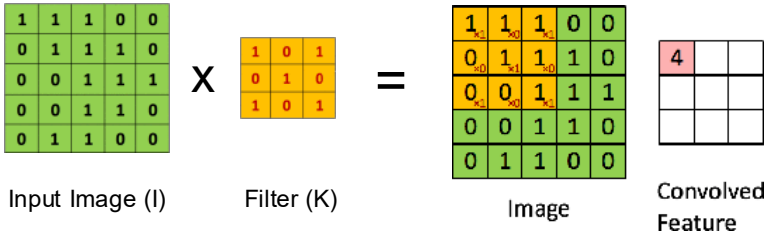
Convolutional Layer, Pooling Layer, Fully Connected Layer



CNN TERMINOLOGY

Convolution Operator

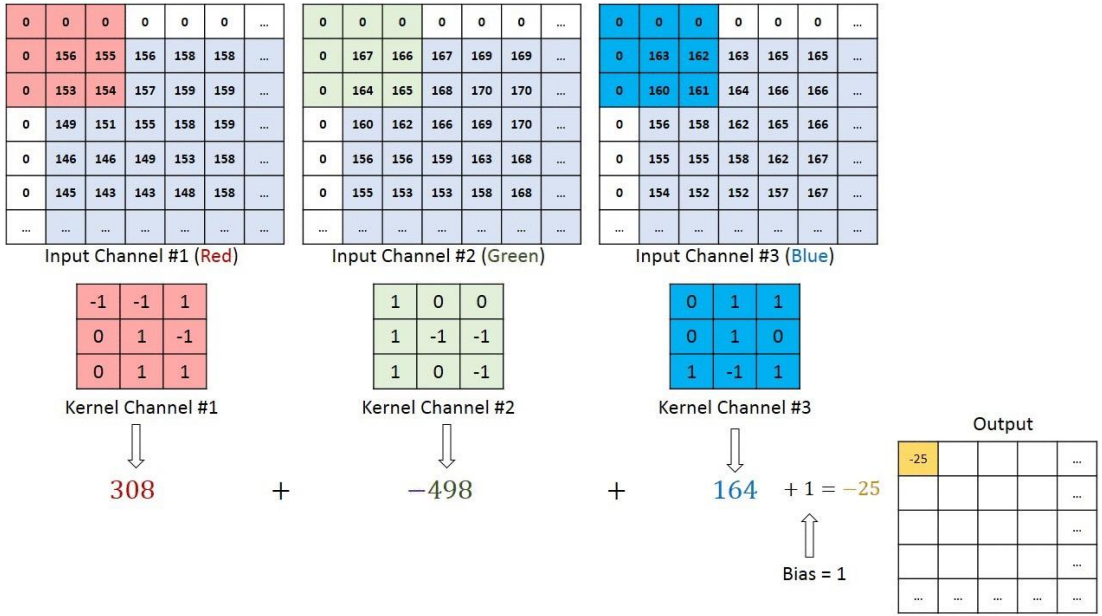
$$(I * K)_{xy} = \sum_{i=1}^{h} \sum_{j=1}^{w} K_{ij} \cdot I_{x+i-1,y-j-1}$$



•The 3×3 matrix (*K*) is called a '**filter**' or '**kernel**' or '**feature detector**' and the matrix formed by sliding the filter over the image and computing the dot product is called the '**Convolved Feature**' or '**Activation Map**' or the '**Feature Map**'.

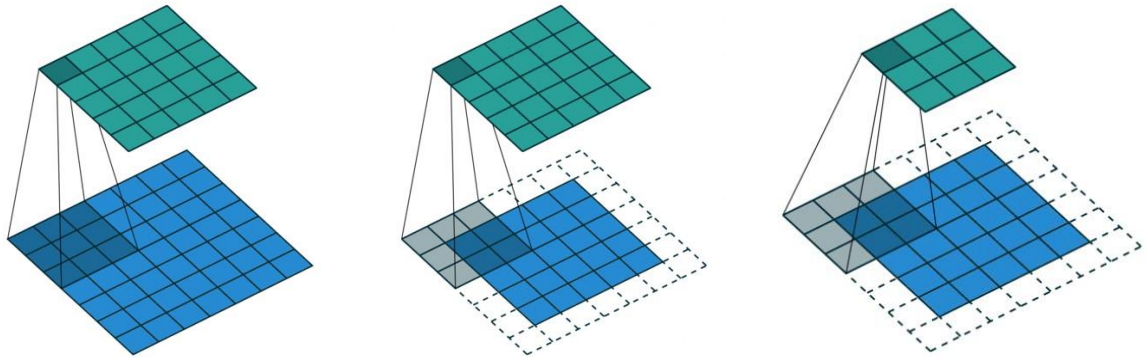
Make a convolution filter for each RGB color channel

$$\text{conv}(I, K)_{xy} = \sigma\left(b + \sum_{i=1}^{h}\sum_{j=1}^{w}\sum_{k=1}^{d} K_{ijk} \cdot I_{x+i-1,y+j-1,k}\right)$$



CONVOLUTIONAL NEURAL NETWORKS (CNNS)

- In practice, a CNN learns the values of these **filters** on its own during the **training process**
- Although we still need to specify parameters such as **number of filters**, **filter size**, **padding,** and **stride** before the training process

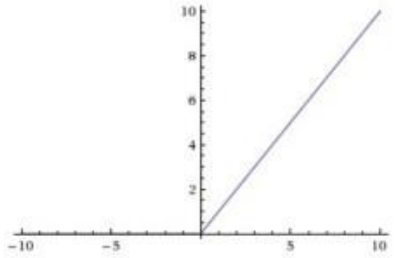


CNN Activation Layers

Activation Layer (ReLU)

•An additional operation called Rectified Linear Unit (ReLU) which can be used after every Convolution operation

Output = Max(zero, Input)

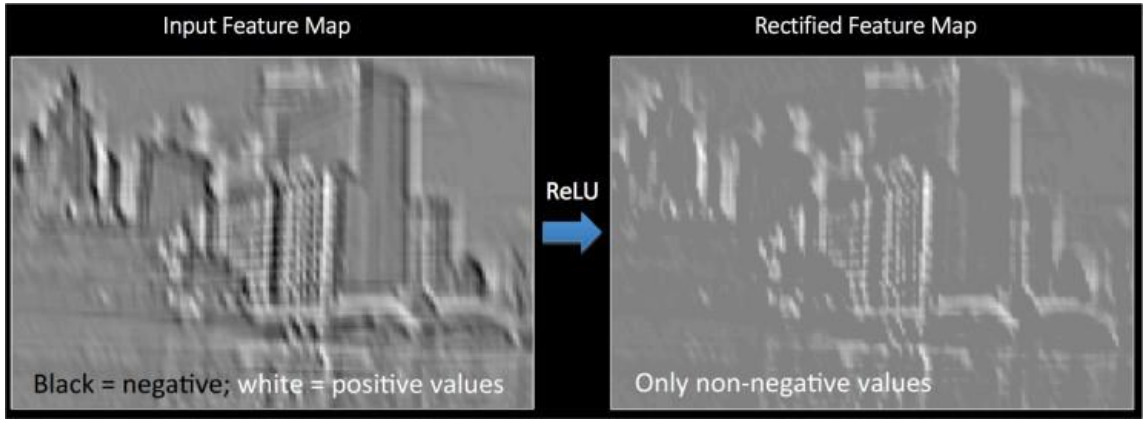


•Basically, ReLU is an element wise operation (applied per pixel) and replaces all negative pixel values in the feature map by zero

•The purpose of ReLU is to introduce non-linearity to the network

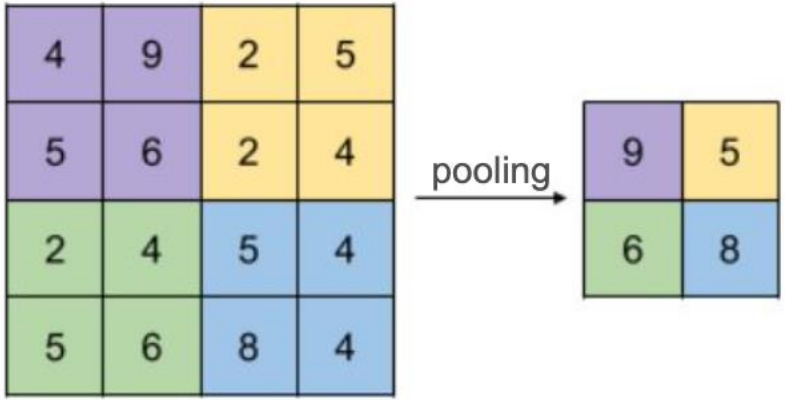
CNN Activation Layers

Activation Layer (ReLU)



•Other nonlinear functions such as ***tanh*** or ***sigmoid*** can also be used instead of ReLU, but ReLU has been found to perform better in most situations.

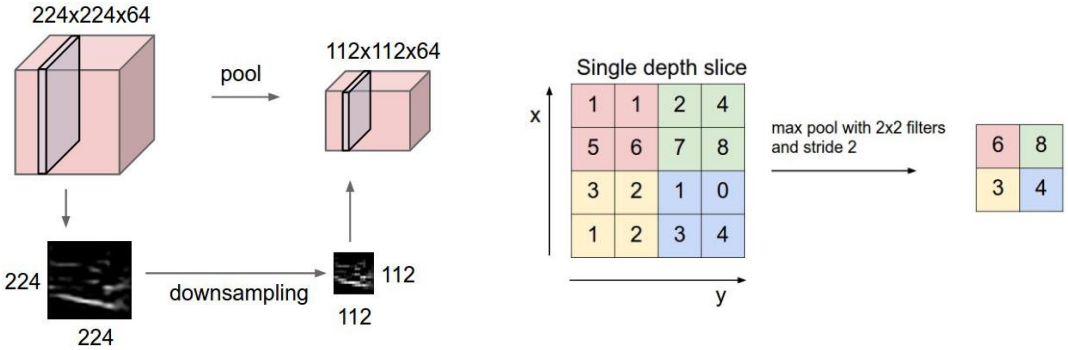
CNN Max Pooling



CNN Pooling Layer

Pooling Layer

Pooling layer **downsamples** the volume spatially, **independently** in **each depth** slice of the input

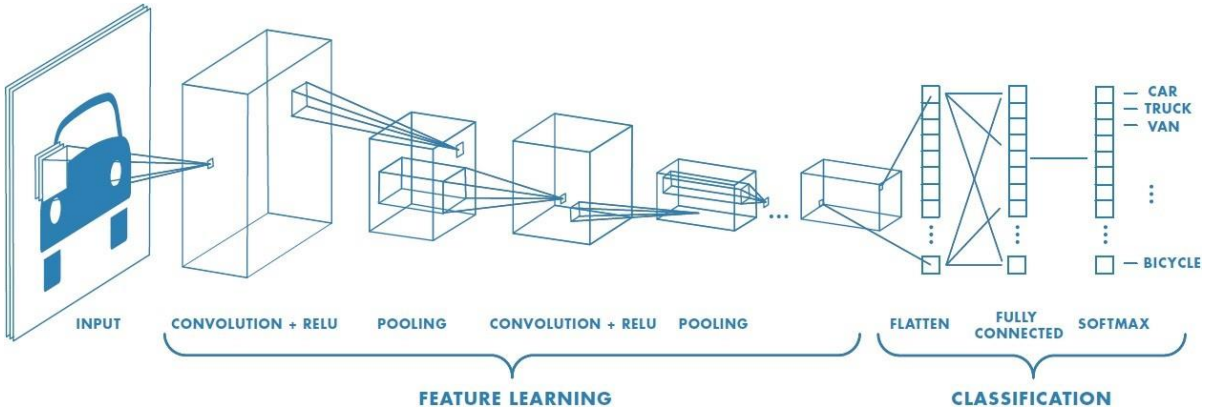


The most common downsampling operation is **max**, giving rise to **max pooling**, here shown with a stride of 2

CNN Fully Connected Layer for Classification

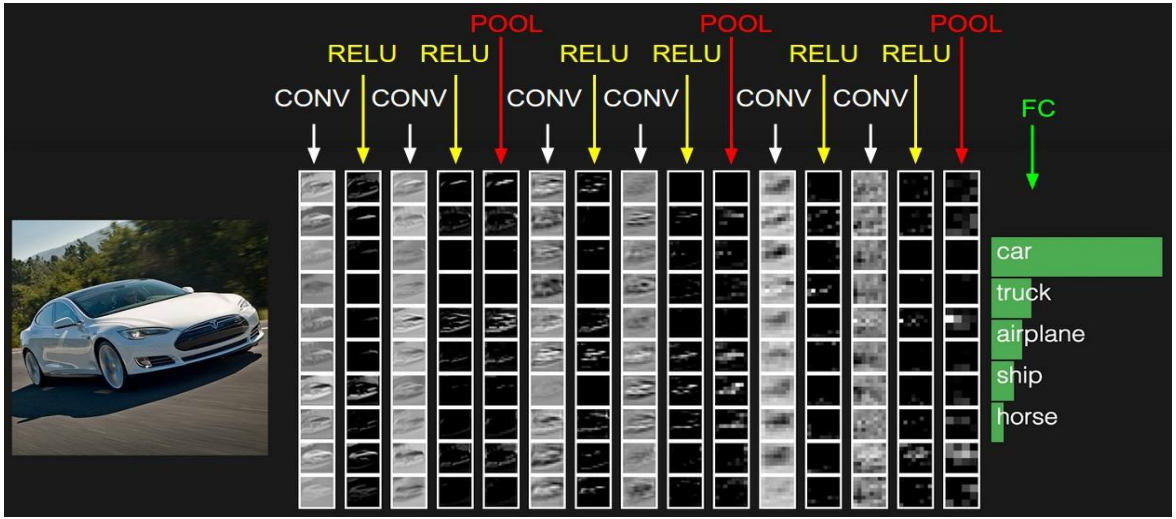
Fully Connected Layer

•Neurons in a fully connected layer have full connections to all activations in the previous layer

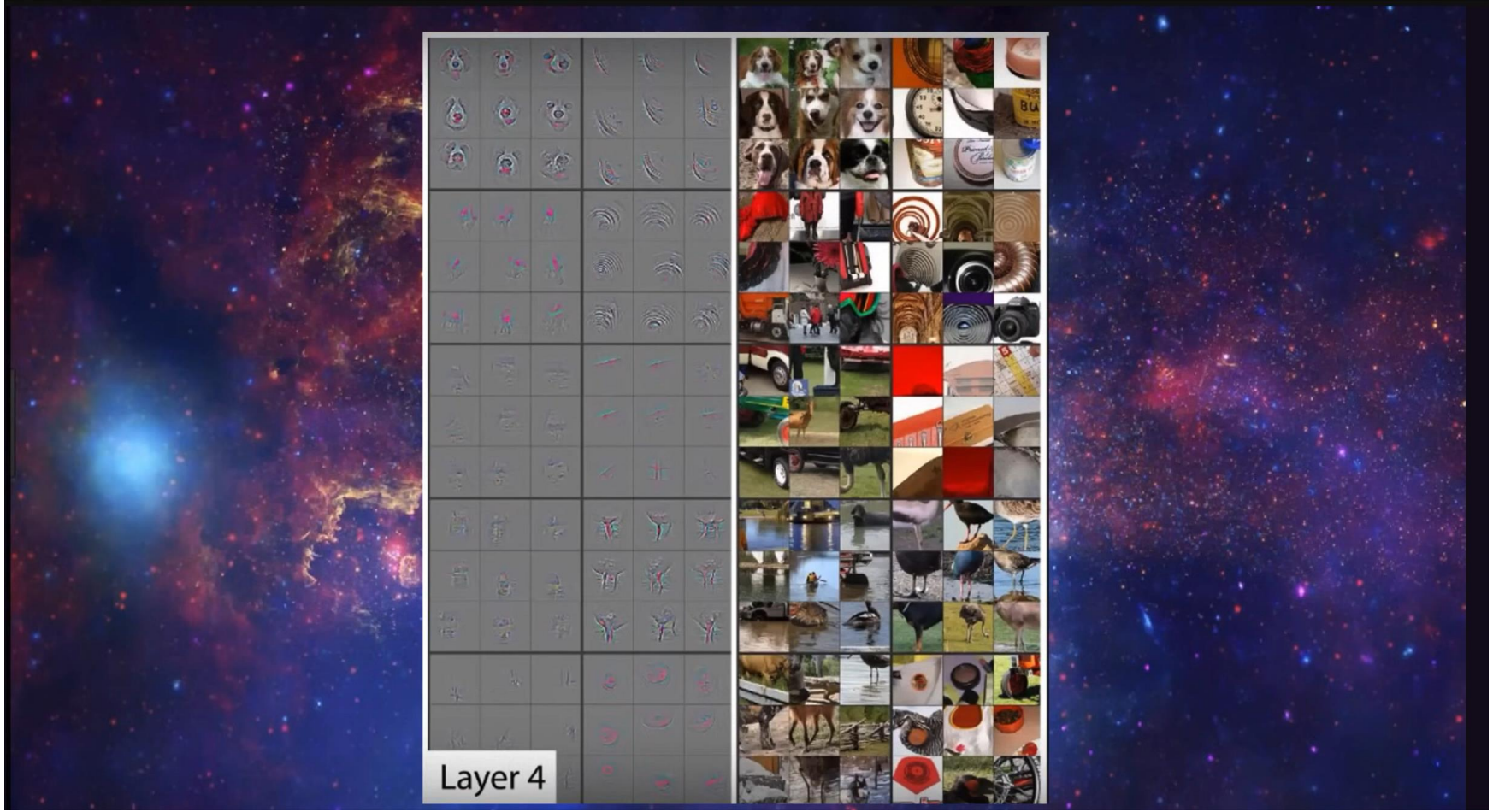


CNN Output at Each Layer

Example: Input >> [[Conv >> ReLU] * 2 >> Pool] * 3 >> FC



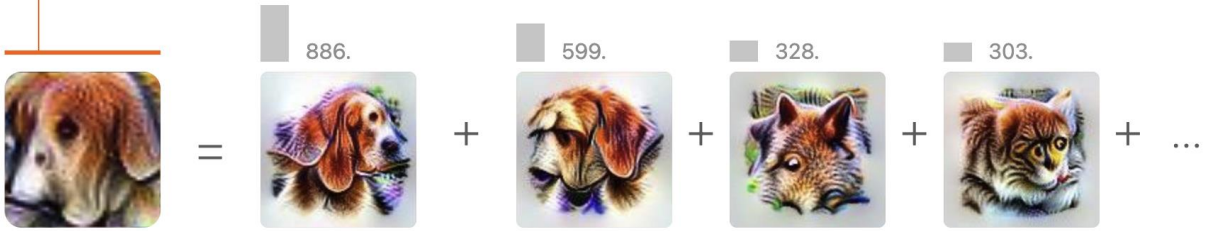
The leftmost column stores the raw image pixels and the rightmost stores the class scores (right). Since it's difficult to visualize 3D volumes, we show image slices in rows. The last layer volume holds the scores for each class.



https://www.youtube.com/watch?v=YRhxdVk_sIs

What Does the Network See?

Semantic dictionaries give us a fine-grained look at an activation: what does each single neuron detect? Building off this representation, we can also consider an activation vector as a whole. Instead of visualizing individual neurons, we can instead visualize the *combination* of neurons that fire at a given spatial location. (Concretely, we optimize the image to maximize the dot product of its activations with the original activation vector.)

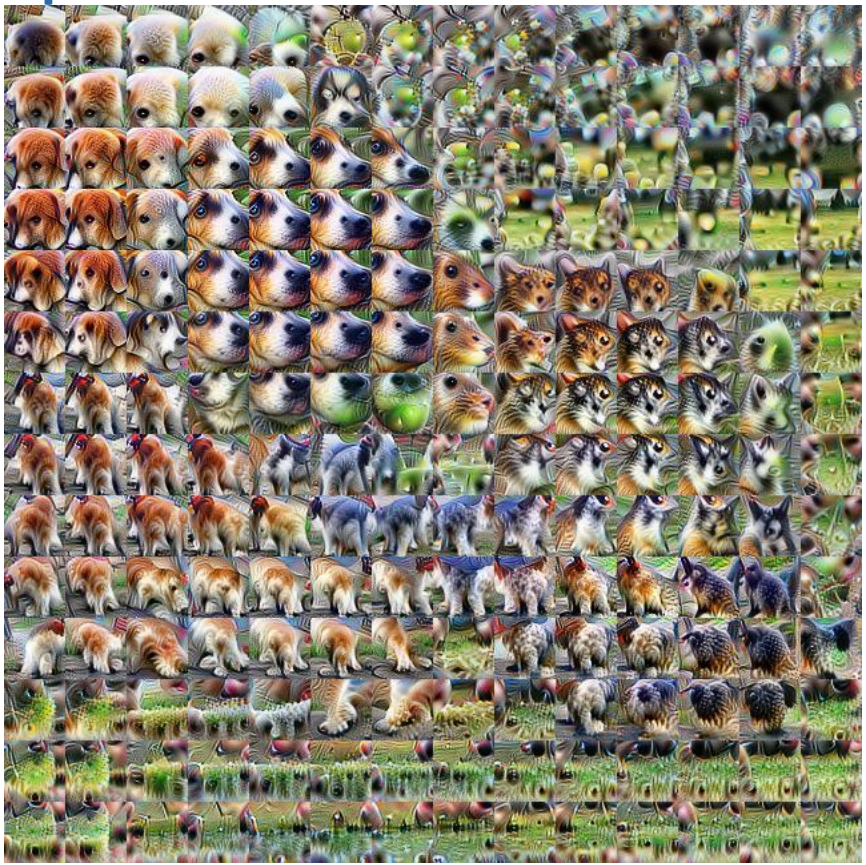


Activation Vector

Channels

https://distill.pub/2018/building-blocks/

Explore CNN Features

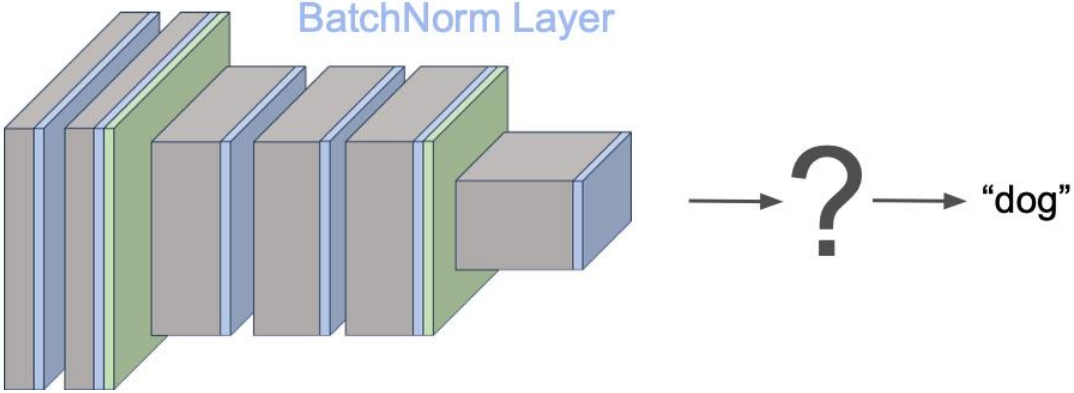


https://distill.pub/2018/building-blocks/

Advantages of CNNs for Image Classification

✅ Convolutions	Maintain spatial relation between pixels Reduce number of parameters through weight sharing
✅ Pooling	Captures key information from across different areas of the feature maps Together with convolutions allows for translational invariance
✅ **BatchNorm**	Increases speed and stability of training

input image



CNNs in Summary

- CNNs are primarily designed to process and analyze visual data, such as images and videos.
- Key components: convolution layers, pooling layers, activation functions, normalization layers
- Advantages:
  - *Translational Invariance*
  - *Parameter sharing*
  - *Feature learning*
- Can be trained with backprop
- Used for tasks such as segmentation, classification, object detection, etc.
- Achieved results around 86% on ImageNet
- Have been superseded today by Vision Transformers

To Learn More…

To learn about computer vision in detail:

Prof Hany Farid:

Info 290T

Computer Vision

3 units

Course Description	This course introduces the theoretical and practical aspects of computer vision, covering both classical and state of the art deep-learning based approaches. This course covers everything from the basics of the image formation process in digital cameras and biological systems, through a mathematical and practical treatment of basic image processing, space/frequency representations, classical computer vision techniques for making 3D measurements from images, and modern deep-learning based techniques for image classification and recognition.
Prerequisites	Linear algebra and Python (INFO 206A & B or equivalent).