Image feature extraction
(shallow and deep)

Andrew Kondrich

August 2, 2019

Class schedule

Curriculum Plan

Image Representation

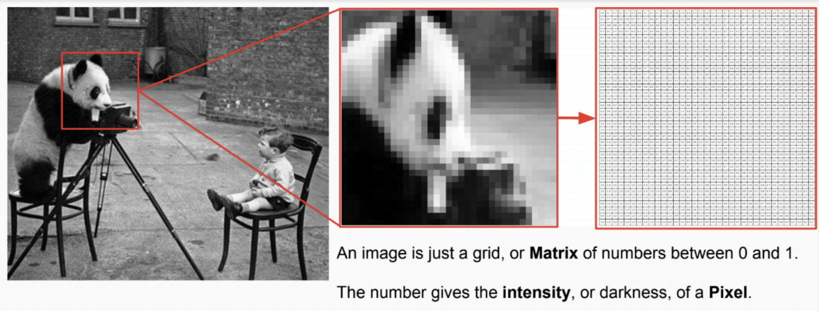


Figure: Black & White image of panda is a matrix

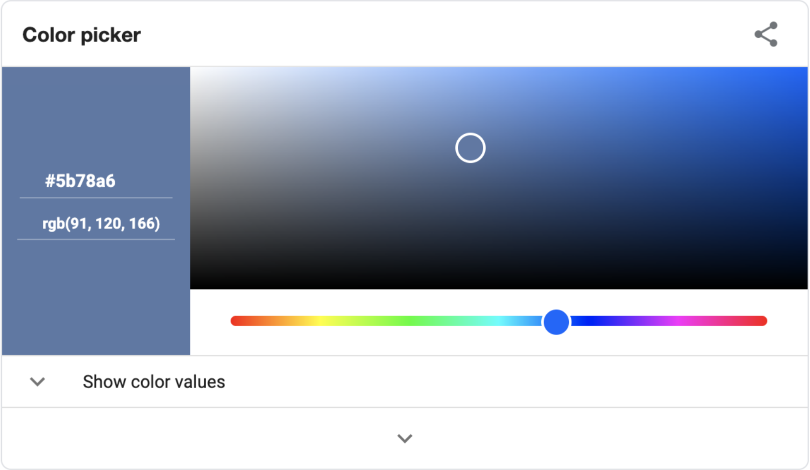


Figure: Color of single pixel is represented by combination of 3 RGB values

Representing Color

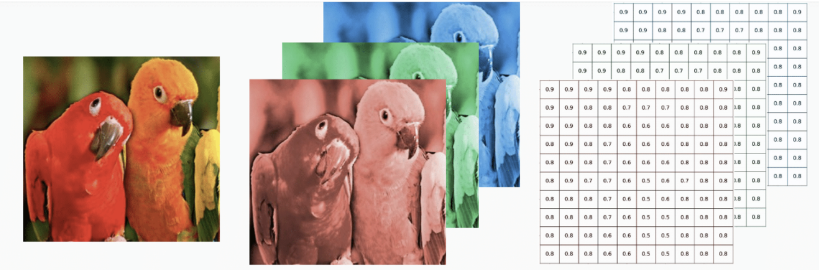


Figure: Stacking RGB layers forms final picture

Representing Color

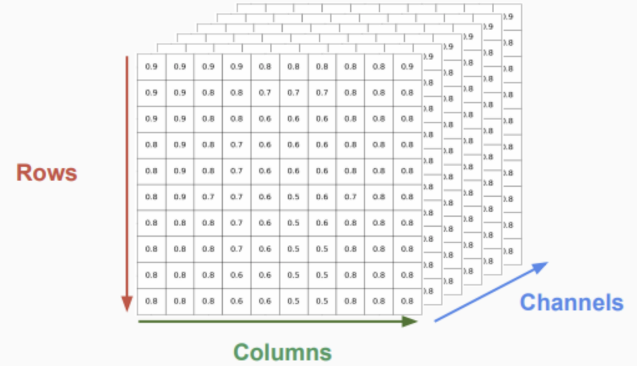


Figure: Numerically, images are 3-dimensional matrices with rows, columns, and 3 **channels**. We also call these **volumes**.

Curriculum Plan

Challenges of Computer Vision

- Because images are very high-dimensional data, it is difficult for models to capture all possible variety and edge cases included in your data.
- Example: When using logistic regression, what happens when you shift an entire image left by one pixel?

Challenges of Computer Vision

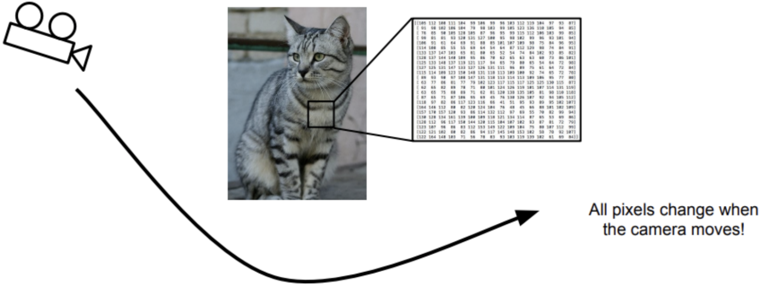


Figure: Viewpoint Variation

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

Figure: Background Clutter

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Illumination

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Deformation

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Occlusion

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

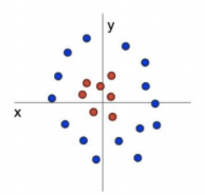
Curriculum Plan

Neural Networks

Visual Viewpoint

Linear classifiers learn one template per class

Geometric Viewpoint



Linear classifiers can only draw linear decision boundaries

Figure: Motivation: Linear models are fairly weak

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Neural Networks

- We can model more complex functions by "stacking" more weights.
- Instead of

$$\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x})$$

We can now also do this:

$$\hat{y} = \sigma(\mathbf{w}_1 \cdot \sigma(\mathbf{w}_2 \cdot \mathbf{x}))$$

$$\mathbf{w}_1 \in \mathbf{R}^M, \mathbf{w}_2 \in \mathbf{R}^{M \times D}, \mathbf{x} \in \mathbf{R}^D$$

Each row of the matrix $\mathbf{w}_2$ is called a "neuron"

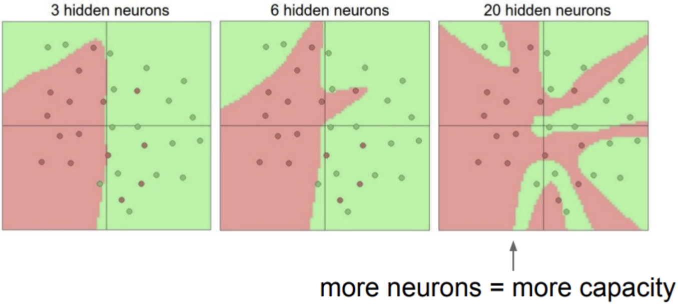


Figure: The more neurons you have, the more **expressive** your model is.

[0] http://cs231n.github.io/neural-networks-1/

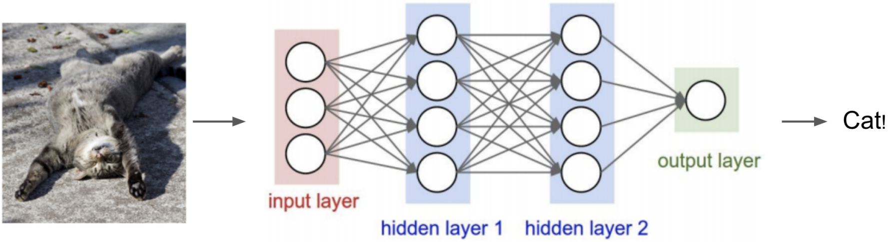


Figure: By stacking more and more layers and adding neurons, we can get models capable of representing all cats!

Curriculum Plan

Invariant Local Features

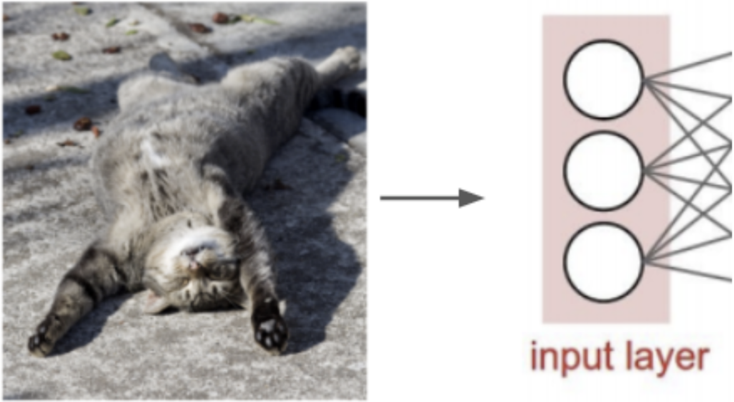


Figure: What is the best way to pass an image into a neural network?

Invariant Local Features

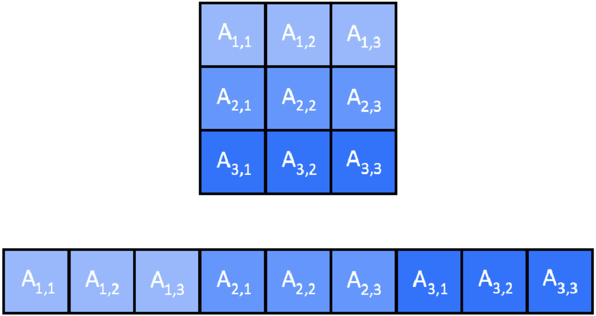


Figure: What could be wrong with just flattening images?

Invariant Local Features

What could be wrong with just flattening images?

- Images are huge: How many features in a 128x128x3 image?

Invariant Local Features

What could be wrong with just flattening images?

- Images are huge: How many features in a 128x128x3 image?
- For every change in viewpoint, you would change 49,152 features.

Invariant Local Features

What could be wrong with just flattening images?

- Images are huge: How many features in a 128x128x3 image?
- For every change in viewpoint, you would change 49,152 features.
- You don't take advantage of **locality**

Invariant Local Features

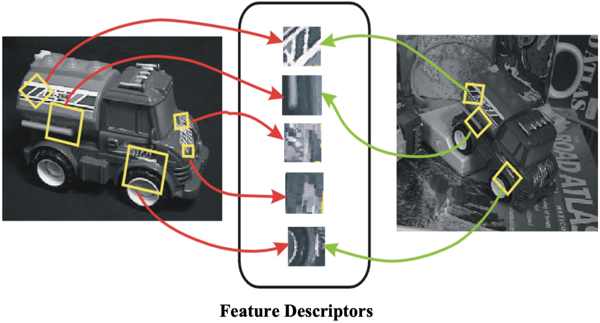


Figure: Patterns in images should remain the same even if picture changes

[Source: N. Snavely]

Invariant Local Features

What are advantages of local features?

- Locality: features are local, so robust to occlusion and clutter
- Quantity: hundreds or thousands in a single image
- Distinctiveness: can differentiate a large database of objects

Invariant Local Features

Suppose we only consider a small window of pixels

- Look for image regions that are unusual: lead to unambiguous matches in other images
- How to define "unusual"?

Invariant Local Features

Suppose we only consider a small window of pixels

- What defines whether a feature is a good or bad candidate?
- (This intuition is used in the famous Harris Corner Detector)

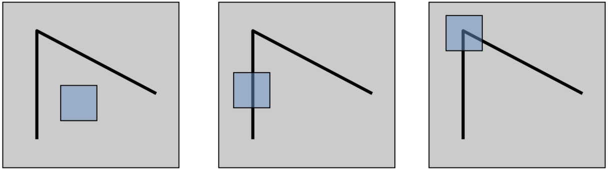


Figure: Which window would change the most if you shift it?

[Source: S. Seitz, D. Frolova, D. Simakov]

Curriculum Plan

Classical Feature Extraction

- **Problem:** Our images are too high-dimensional and flattening doesn't take advantage of innate local properties of images.
- **Solution:** Can our images be reduced to a set of features (also named a feature vector).
- In "classical" feature extraction, we rely on our human understanding of what a "good" local features is to decide what matters in a picture and what doesn't.

Classical Feature Extraction

Figure: Standard Image of a campus

Classical Feature Extraction

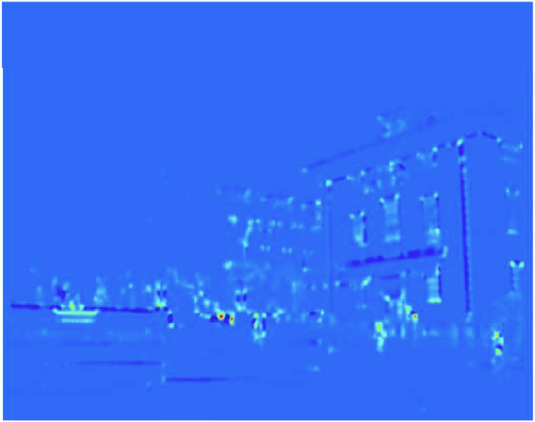


Figure: Search for parts of the image that have the most "cornerness"

Classical Feature Extraction

Figure: Based on our heatmap, extract points of interest

Classical Feature Extraction

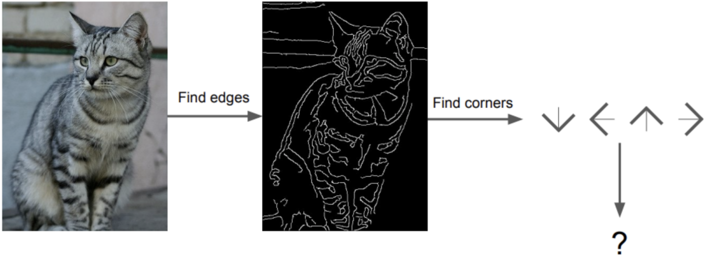


Figure: Extracting edges and corners can greatly reduce our input dimensionality while improving the model's understanding of local features

Curriculum Plan

Deep CNN

- Histograms and edges are called “shallow” features because we build them by hand
- We know we can learn neural network and linear model weights by **Gradient Descent**
- Can we also learn our features?

Deep CNN

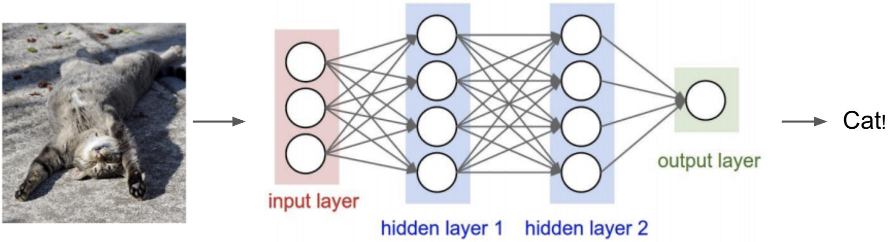


Figure: Our neural network predicting cats must take either flattened cat or extracted features as input.

Deep CNN

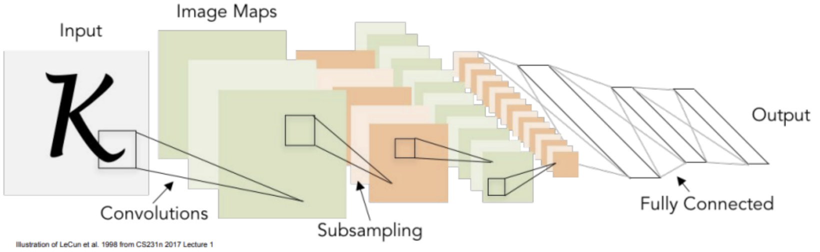


Figure: In modern Computer Vision, we rely on **Convolutional Neural Networks** to learn meaningful features by gradient descent.

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf

Deep CNN

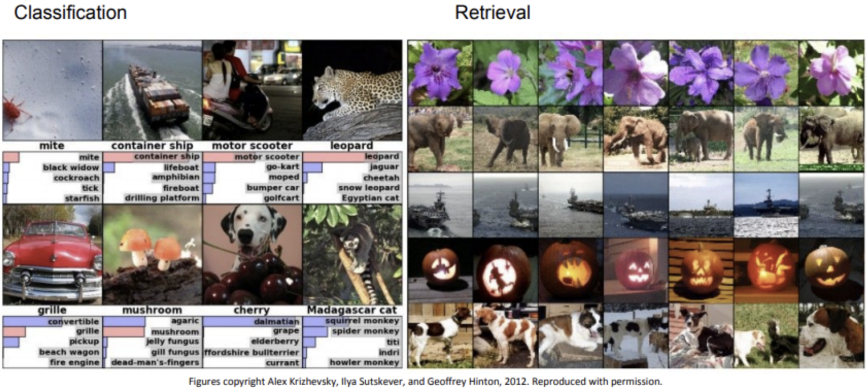


Figure: Deep Convolutional Neural Networks are everywhere

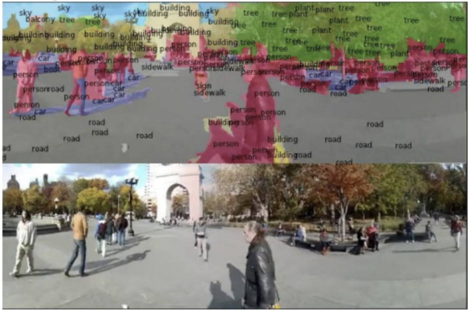
[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf

Detection

Figures copyright Shaoqing Ren, Kaiming He, Ross Girschick, 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]

Figure: Deep Convolutional Neural Networks are everywhere

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf

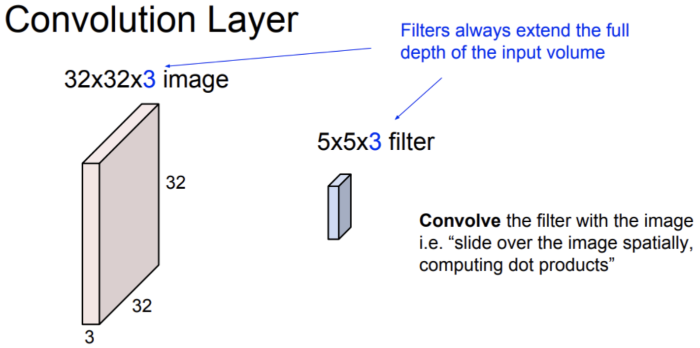


Figure: Convolutional Layers

[0] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture05.pdf

Deep CNN

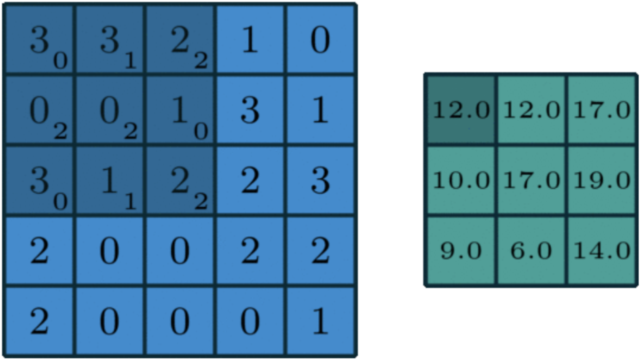


Figure: Convolutional Layers

Deep CNN

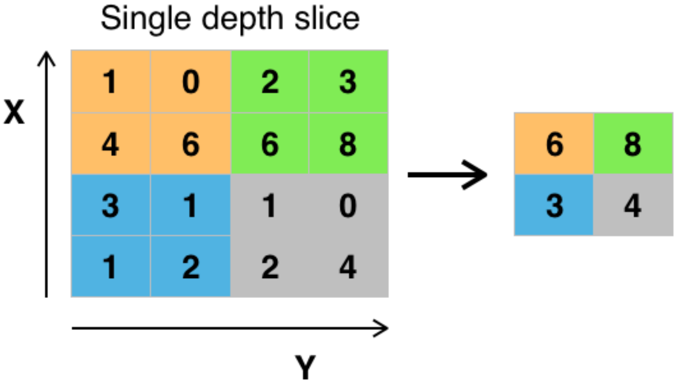


Figure: Max Pooling Layers

Deep CNN

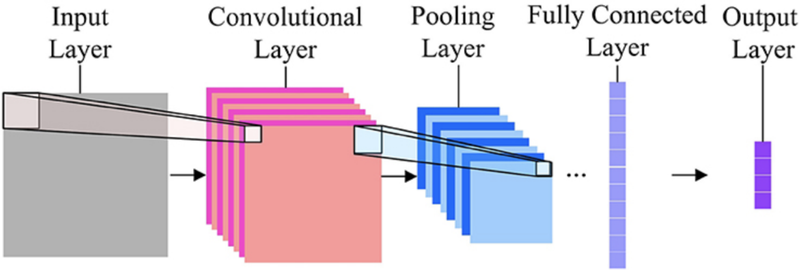


Figure: Full CNN Structure

Image feature extraction
(shallow and deep)

Andrew Kondrich

August 2, 2019