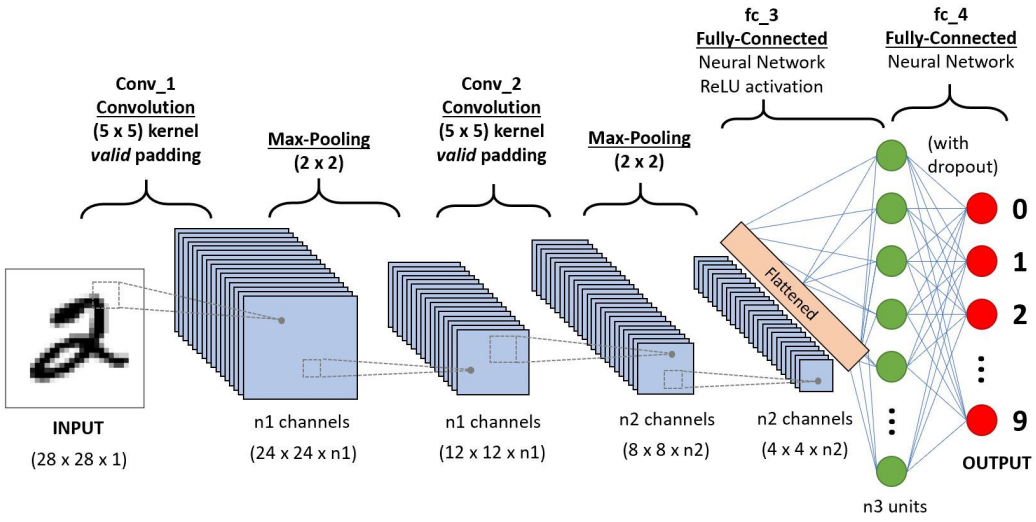
CSCI 303

Introduction to Data Science

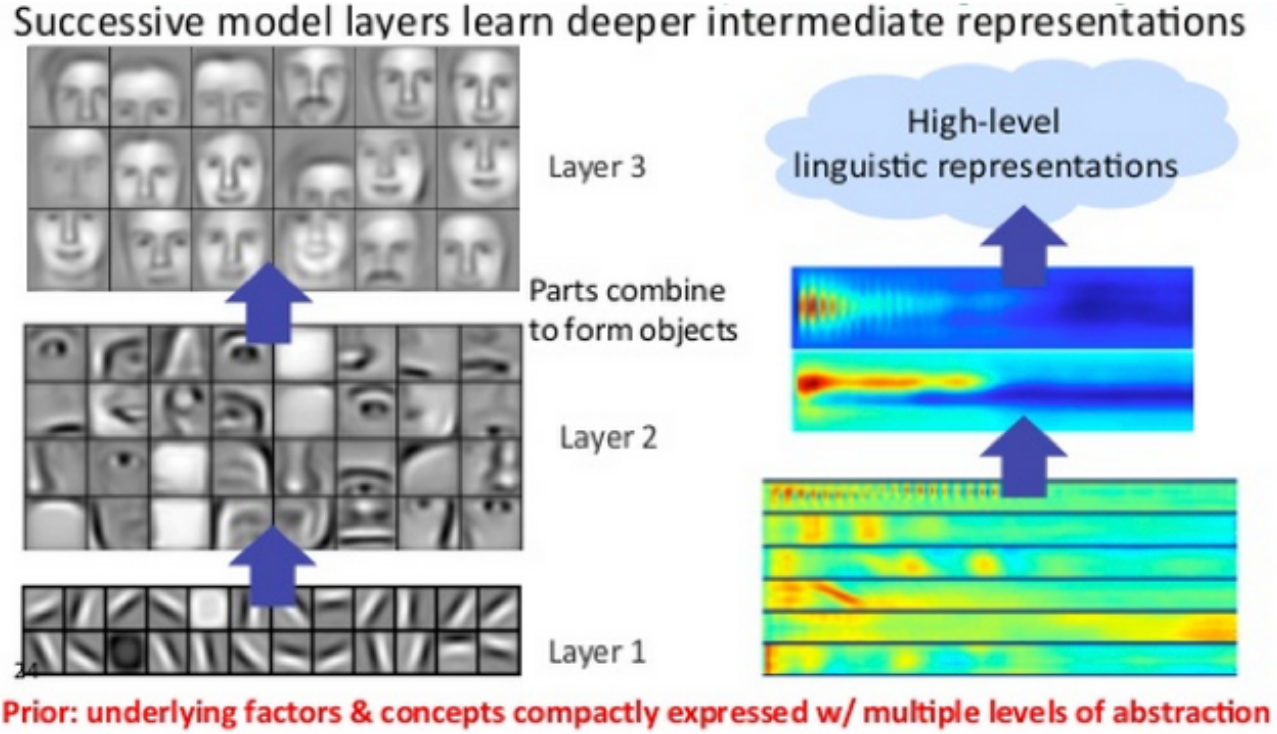
Convolutional Neural Networks



Convolutional Neural Networks (CNN)

- CNNs are NNs with a specialized architecture that makes them suited for **computer vision tasks**.
- We want algorithms that can detect an object, regardless of where it in the image ("**translational invariance**")
- CNNs are **hierarchical**
  - **Early layers** help detect **simple image features** (lines, arcs, corners, blobs)
  - **Higher layers** help detect more **complex image features** (ears, eyes, wheels, chrome)
- Translation invariance and hierarchy concepts are inspired by neuroscience (to some degree)

Assumption: Object have hierarchical representations

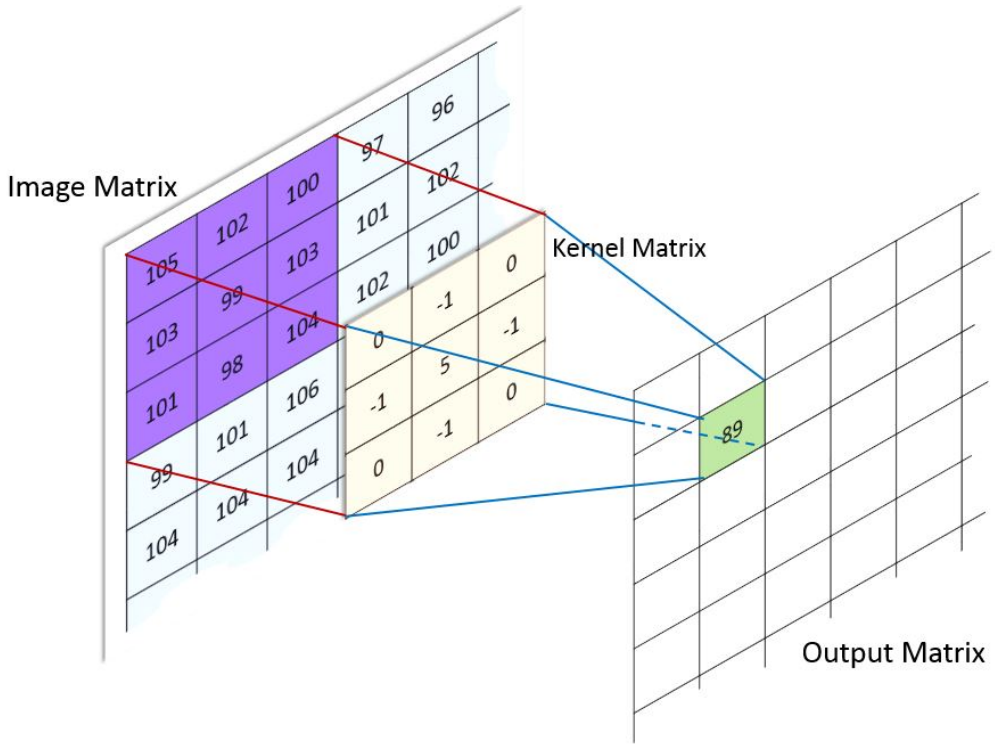


Convolutional kernels

Most "layers" of a CNN are "convolutional layers"

Convolution: Filtering of an image with a small "kernel" that looks like a useful features (line, arc, etc.)

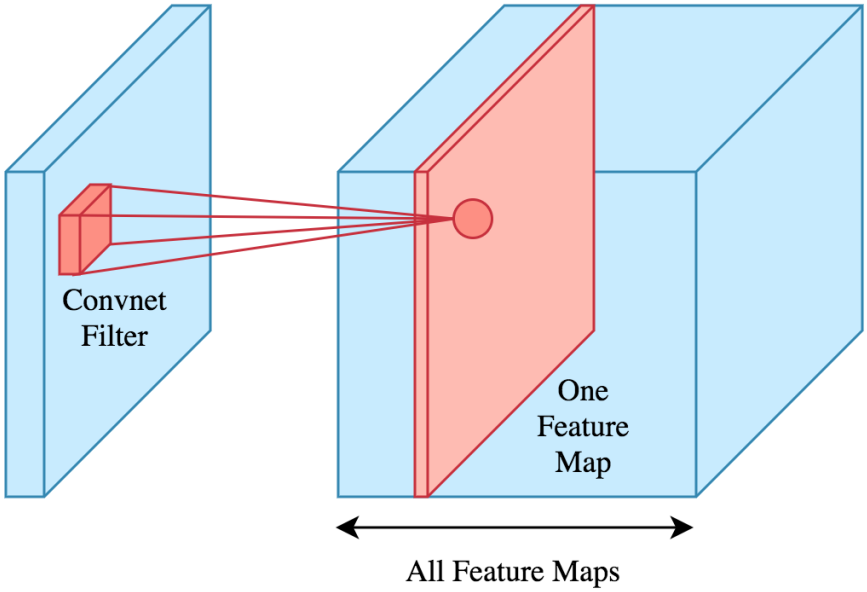
- Each neuron within a layer has an **kernel of weights** that are **arranged in a 2D matrix**.
- There are identical neurons (all have the same kernel), also arranged in a 2D pattern.
  - In practice, we don't really have an array of neurons in a 2D pattern.
  - Instead, **the kernel is shifted across the input image (or input from the previous layer) and applied to the inputs that it overlaps with. This is the so-called convolution operation.**



Many different kernels (filters) are used/applied in a given layer, e.g.,

- a kernel for an up-down line
- a kernel for a left-right line
- a kernel for a point/blob
- etc.

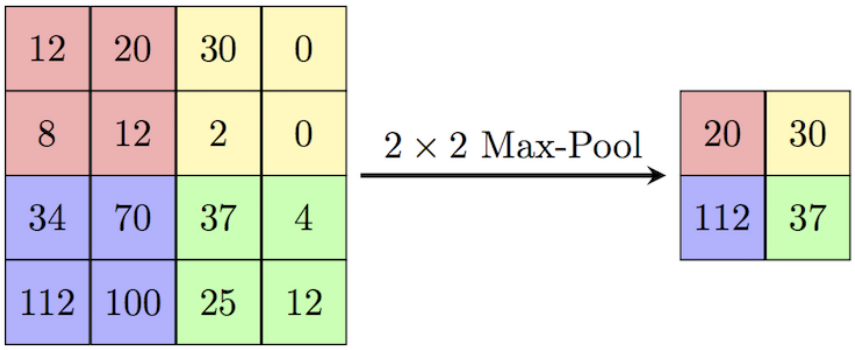
So, a layer that takes in an MxN 3-channel (R,G,B) image may have C different kernels, and thus output an MxNxC array.



(Max) Pooling

- A hierarchy model implies that:
  - More complex **features found at higher layers** of the model will often be **spatially larger**.

- Because features at layer L+1 are made up of features at layer L, not of pixel-level features, the **resolution at layer L+1 can be lower than that at layer L**.
- To lower the resolution, we compute the maximum value over small areas and **replace the area with that max value** (e.g., replace a 2x2 patch with a 1x1 value).



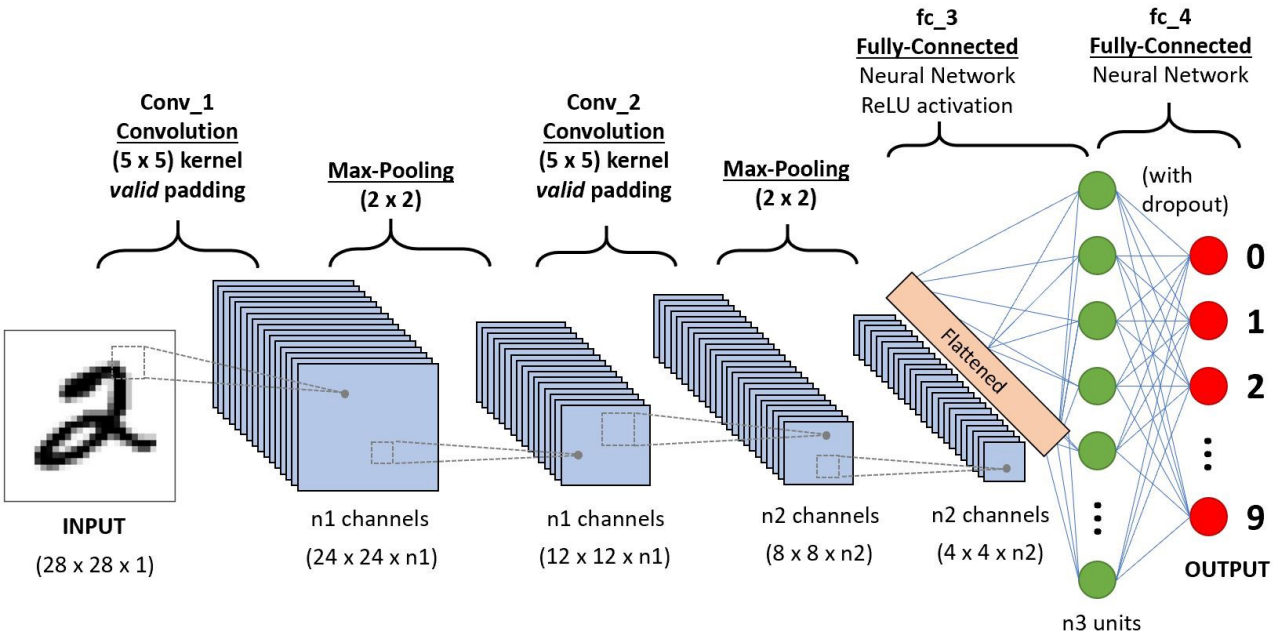
A complete CNN model

Basic CNN models commonly have a sequence of layers like this, although the number of layers may varying widely, depending on the comlexity of the task/images.

- Convolutional layer with ReLU activation
- Max-pooling layer
-
- Convolutional layer with ReLU activation
- Max-pooling layer
-
- ...
- Convolutional layer with ReLU activation
- Max-pooling layer
-
- Flatten the MxNxC array into a MNCx1 array
-
- Dense layer with ReLU
-
- Dense layer without ReLU

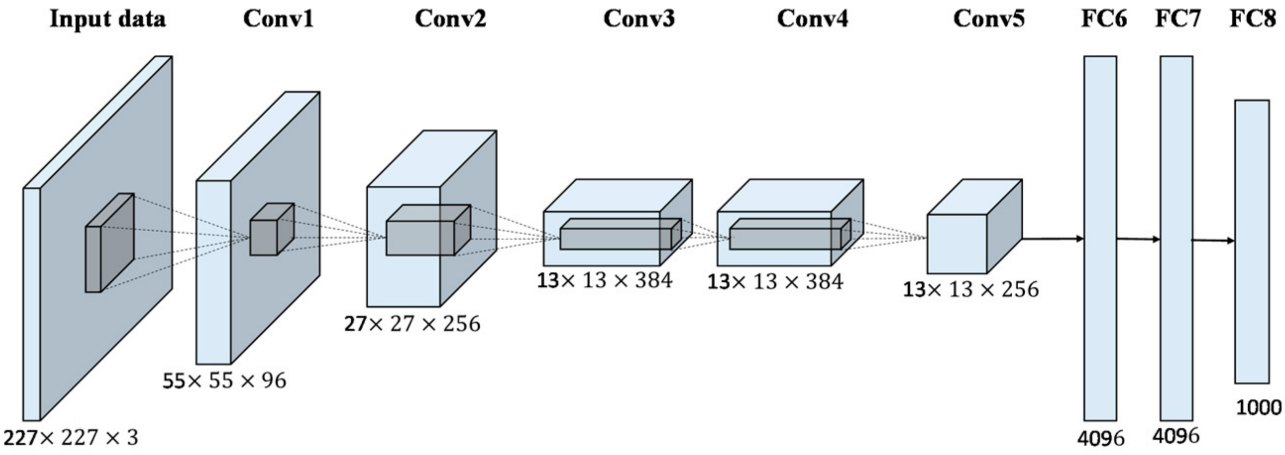
Example figure 1

- 3D shape of kernels is not highly apparent in this figure



Example figure 2

- 3D shape of kernels is clearing in this figure, but
- Max-pooling is not highly apparent in this figure (but can be inferred)

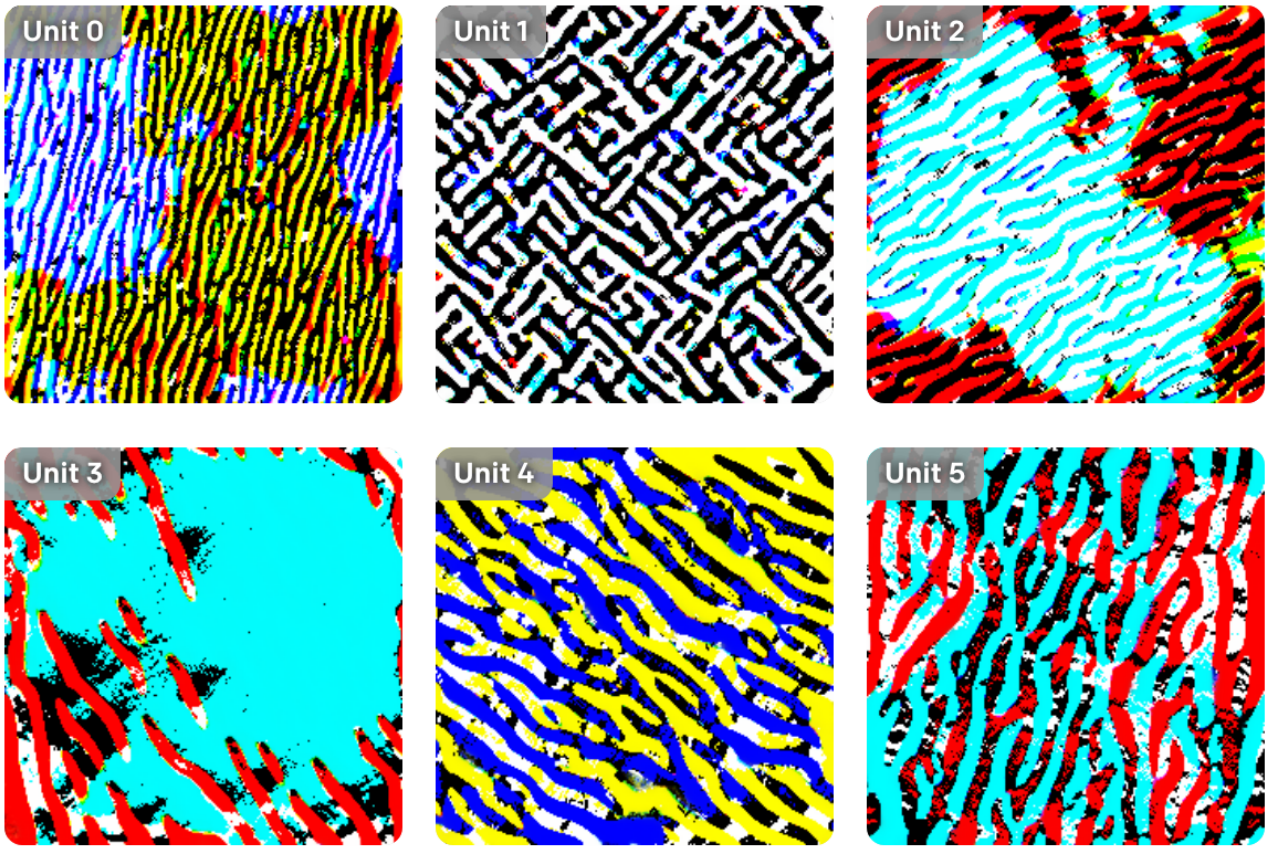


Fun with OpenAI Microscope

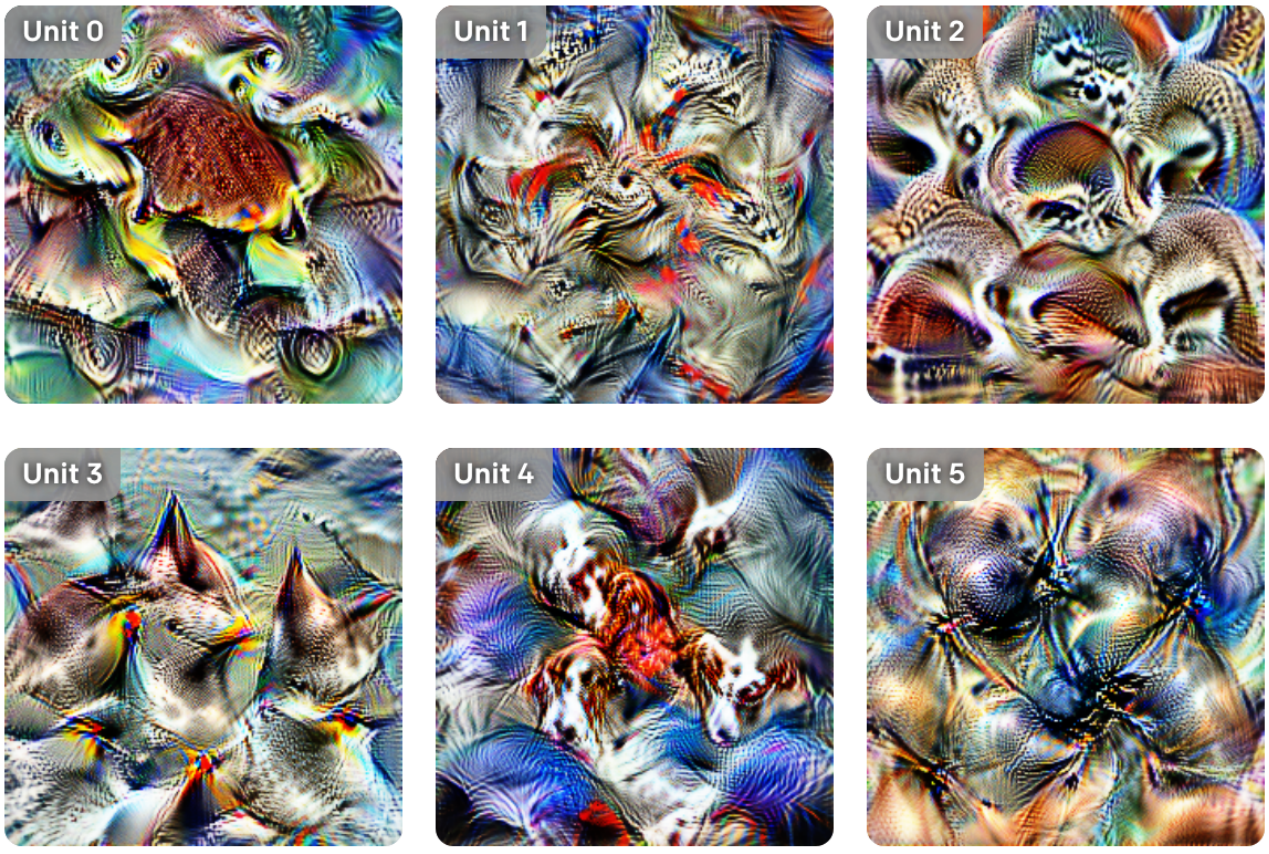
What features are some of the neurons in a CNN sensitive to?

https://openai.com/blog/microscope/ (https://openai.com/blog/microscope/)

Low level features in the AlexNet model:

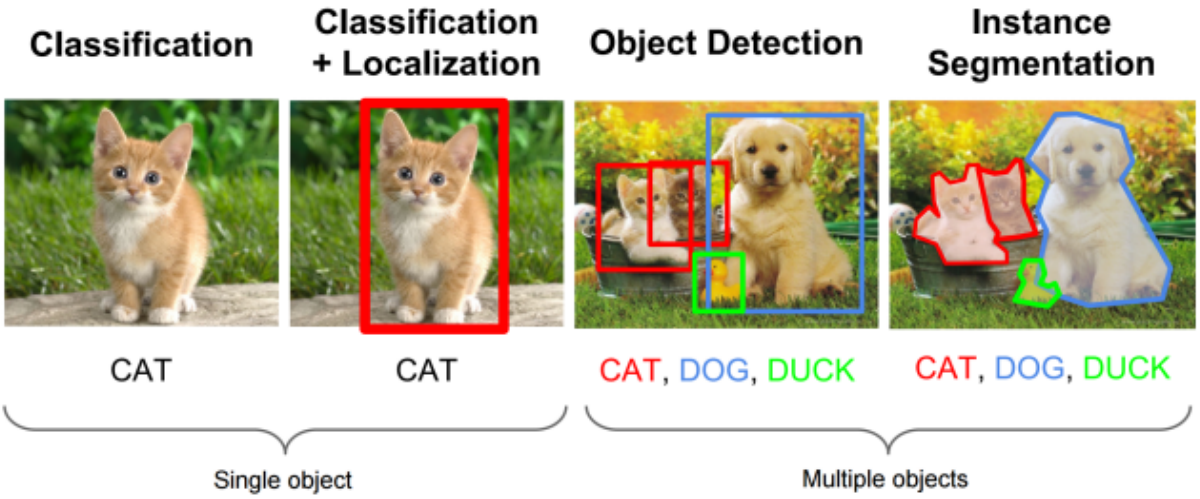


High level features in the AlexNet model:



Other CNN vision applications

Object localization and segmentation

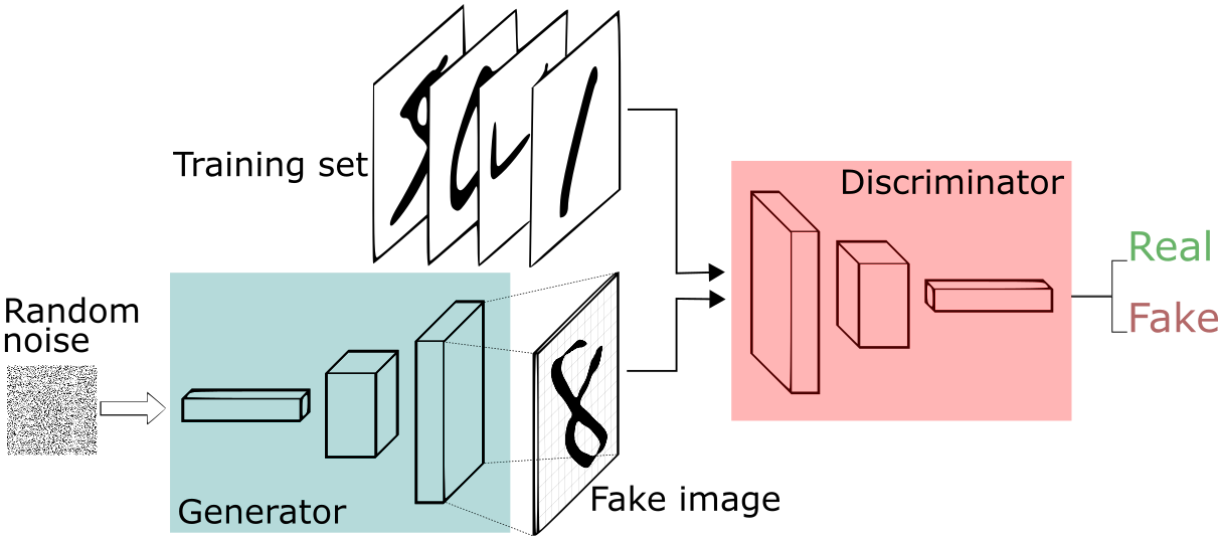


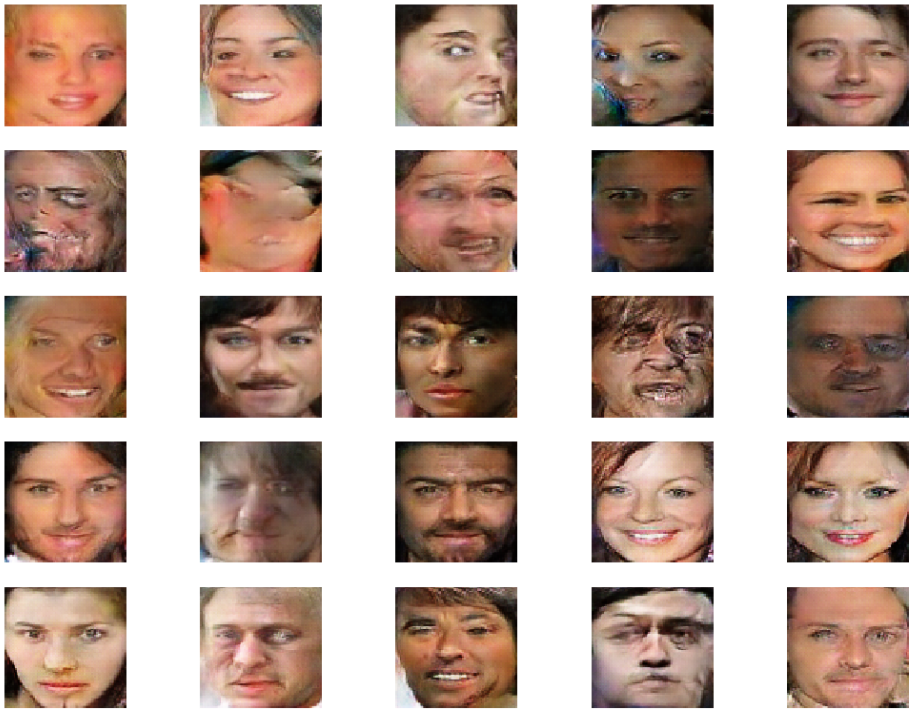
Art: Neural style transfer

Image generation: Generative Adversarial Networks (GAN)

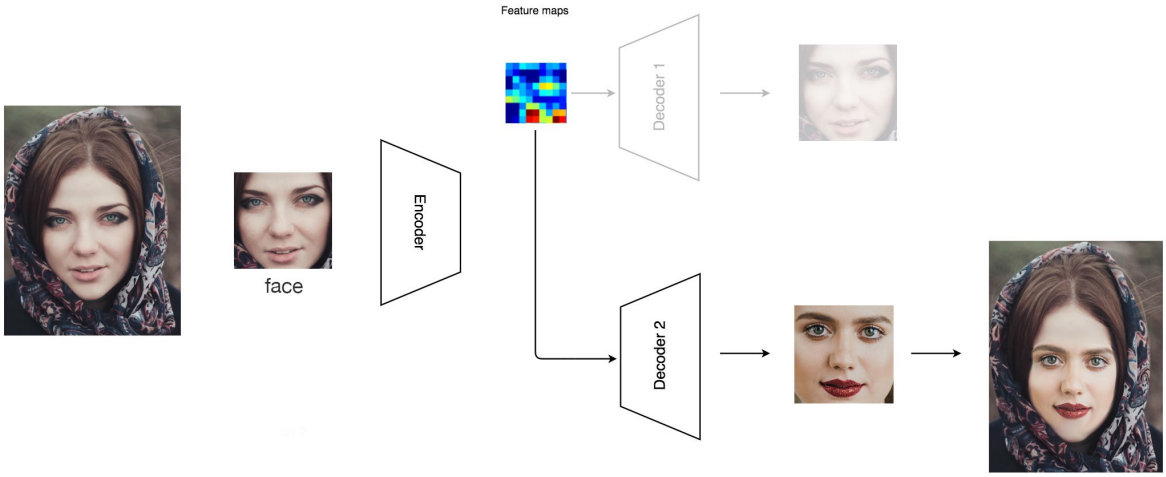
Two networks trained simultaneously

- **Generator** network tries to create realistic-looking image from noise inputs
- **Discriminator** network tries to distinguish real images for synthetic ones from the generator





"Deep Fakes"



Building CNNs

Tensorflow has great tools for building and training CNNs (and so does PyTorch).
We'll be exploring Tensorflow CNN construction and training in a separate assignment.