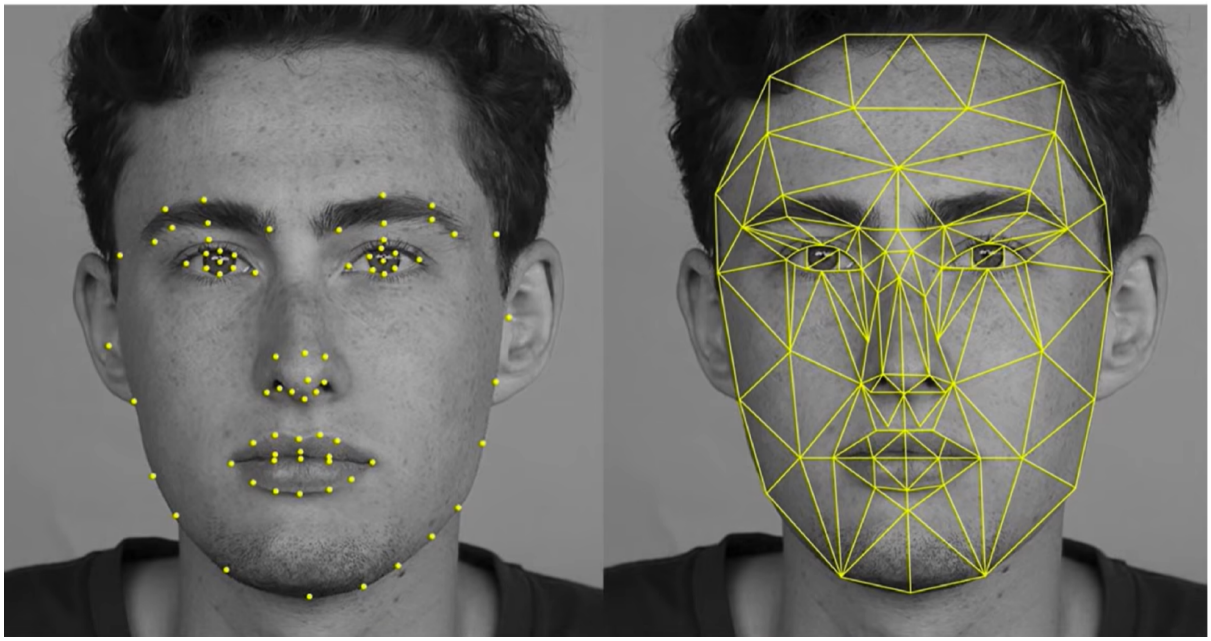
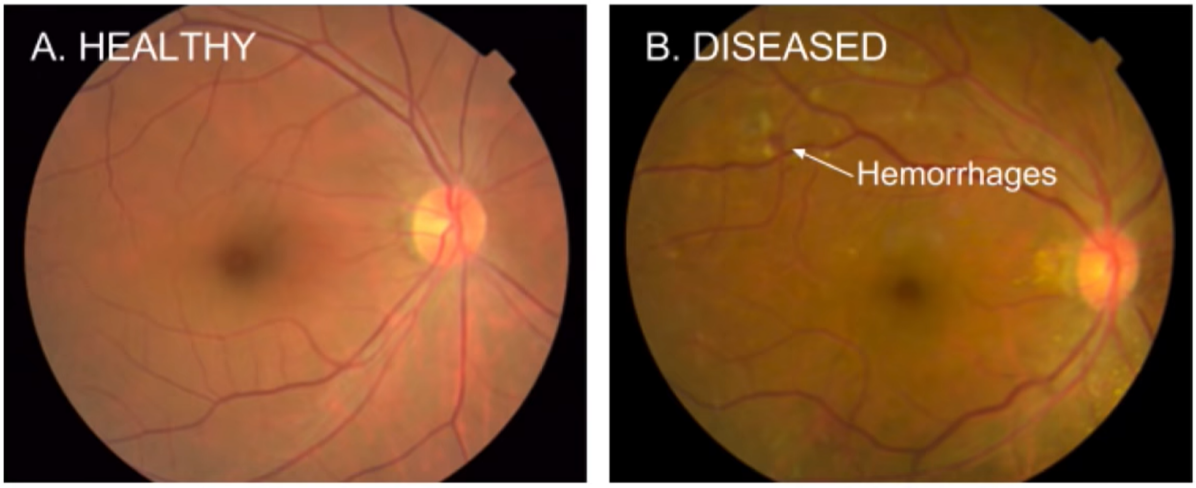
[3]Convolutional Neural Networks

Impacts

Facial Recognition



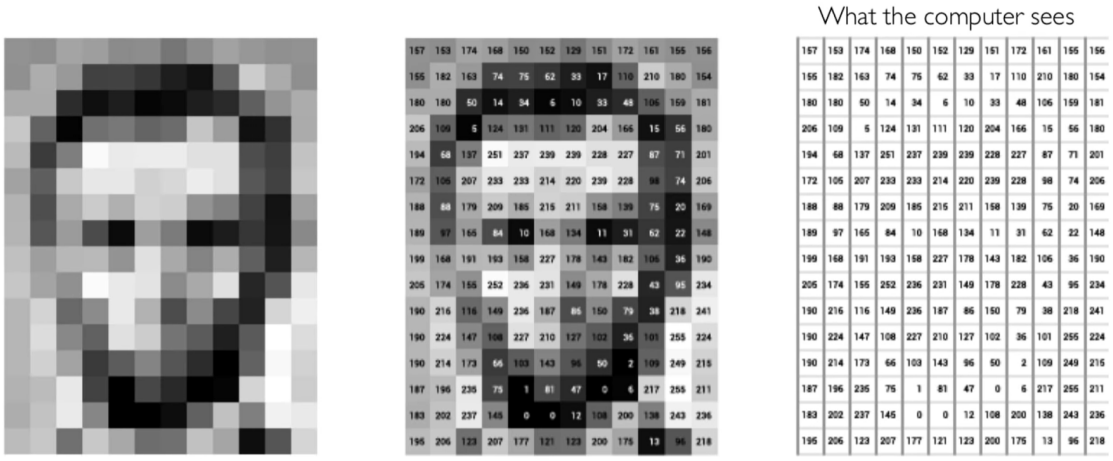
Medical, Biology, Healthcare



Self Driving Cars

What Computer "see"

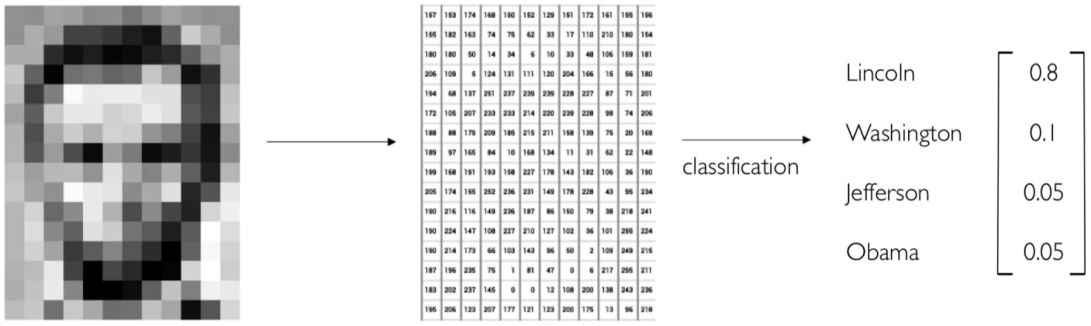
- The computer sees the image as a two dimensional matrix with gray scale
- If the image is color (RGB) then the colors stack on to each other



An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

Task for Computer Vision

- Regression and classification are the common tasks for computer vision
  - Regression: The output variable takes continuous value
  - Classification: output variable takes class label and can produce a probability of belonging to each class



High Level Feature Detection

- To tell the difference between each image, there are specific features of each image and that's a way to tell the difference between one image and another

Let's identify key features in each image category

Nose,
Eyes,
Mouth

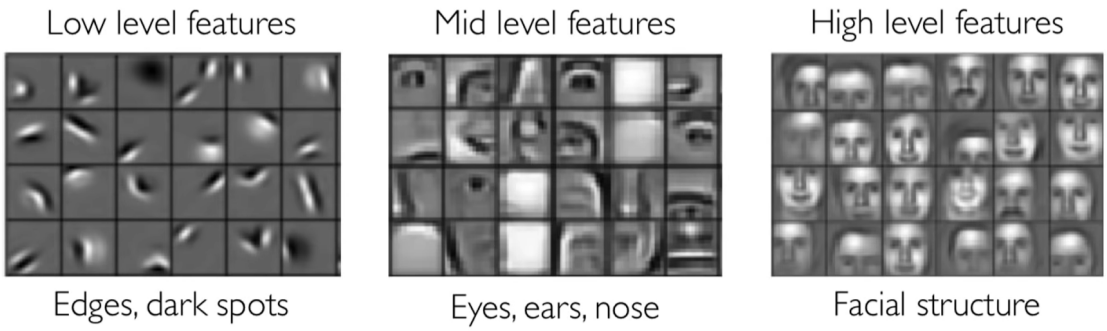
Wheels,
License Plate,
Headlights

Door,
Windows,
Steps

Manual Feature Extraction

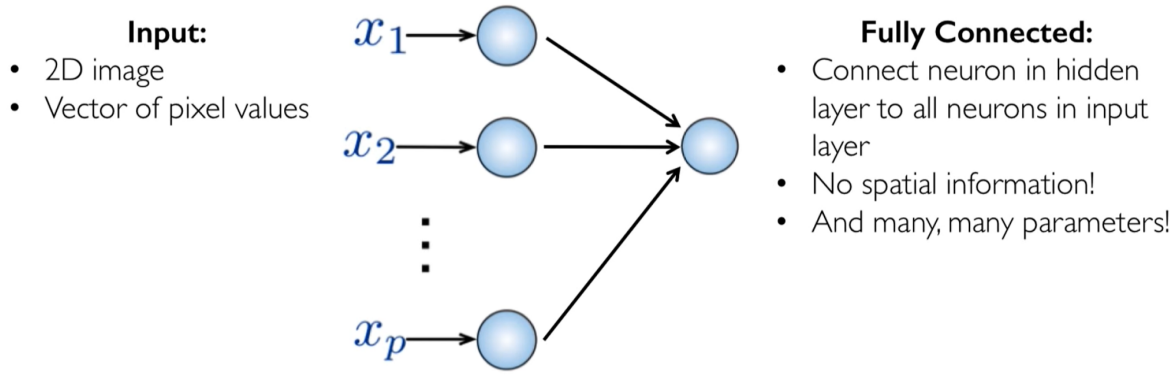
- The detection hierarchy is the bottleneck
- There could be lots of variation between images

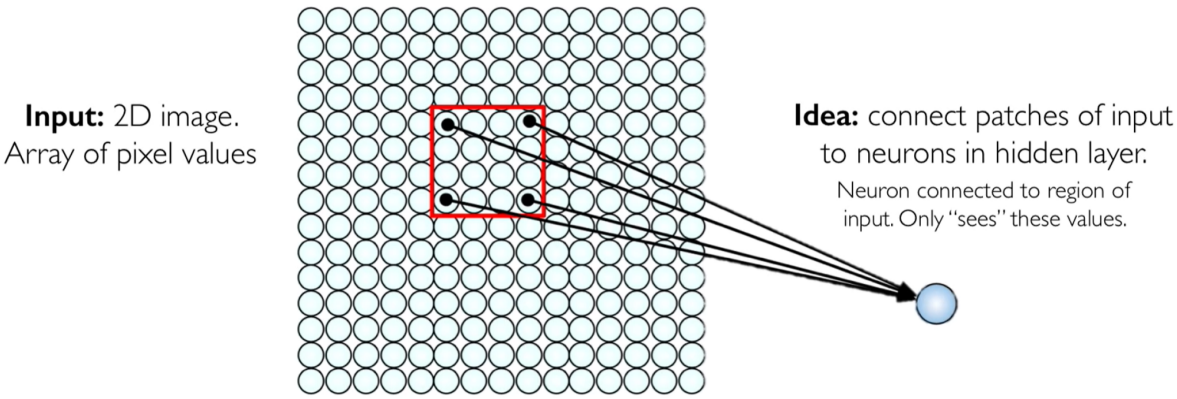
- We can learn a hierarchy of feature directly from the data instead of hand engineering



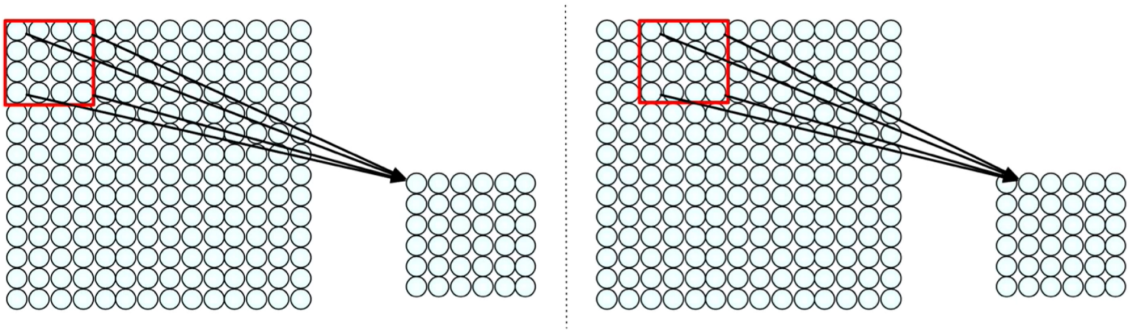
Learning Visual Features

- Using 2d image to compress that down into a 1d vector and putting it into the fully connected (Dense) layer





- Connect patches to input layers to a single neuron in subsequent layer
- Use a slide window to define the connection as shown below

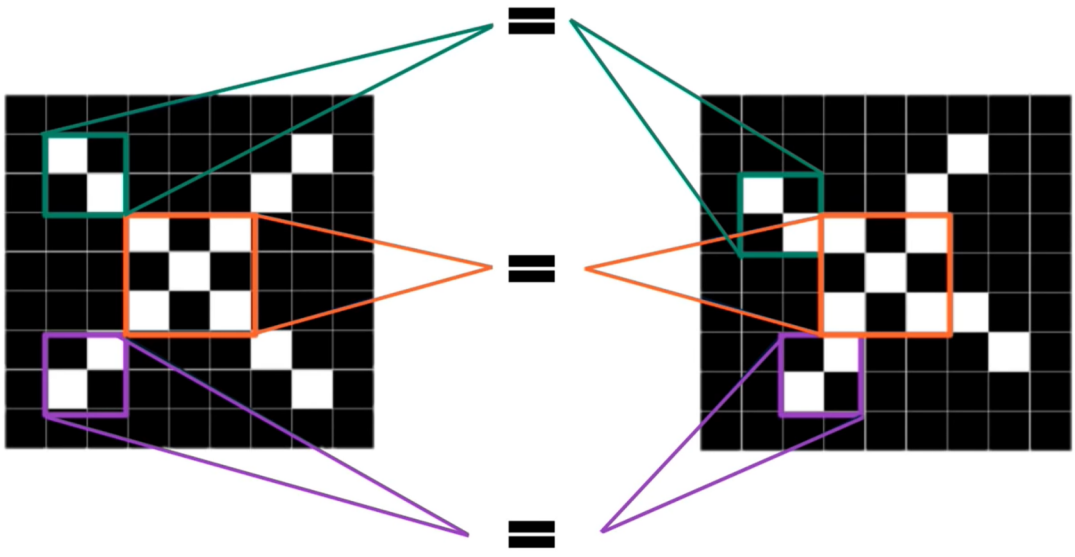


Feature Extraction with Convolution

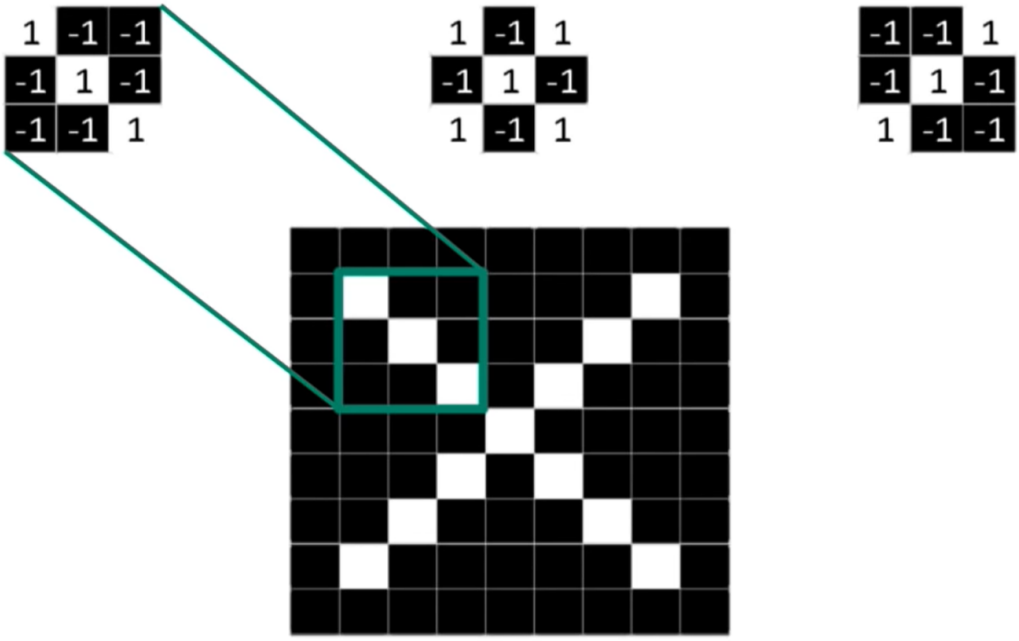
- Apply a set of weight, a filter to extract the local features
- Use multiple filters to extract different features
- spatially share parameters of each filter

Feature Extraction and Convolution

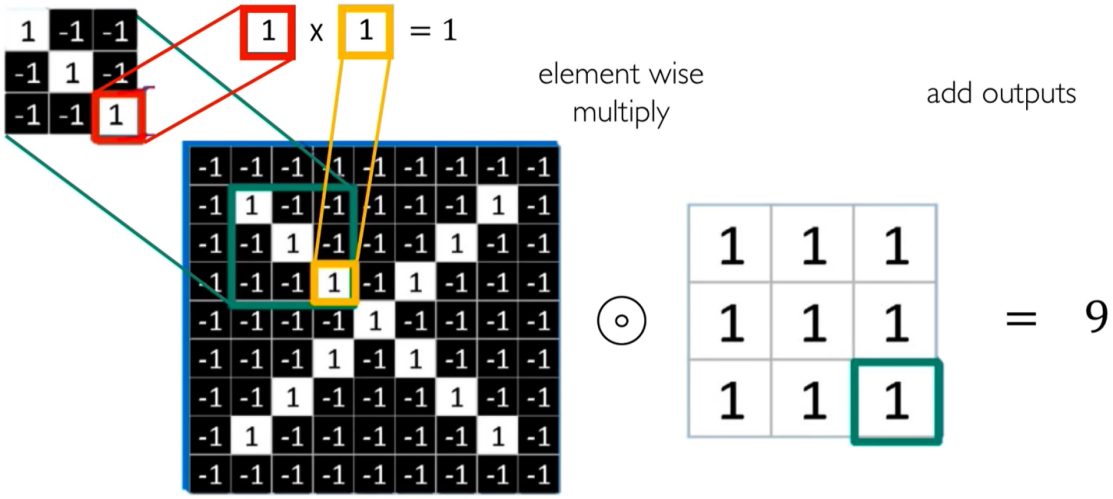
- Example below compares the weather the two x are the same



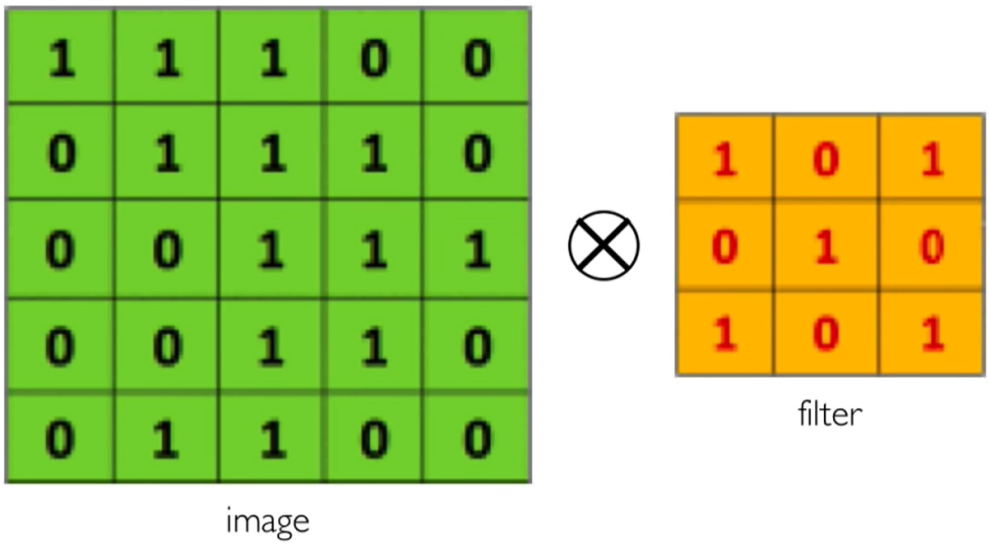
- The model could see that there are a few main features that are the same therefore they are both x
- Using filters, we want to capture features that is important to the image



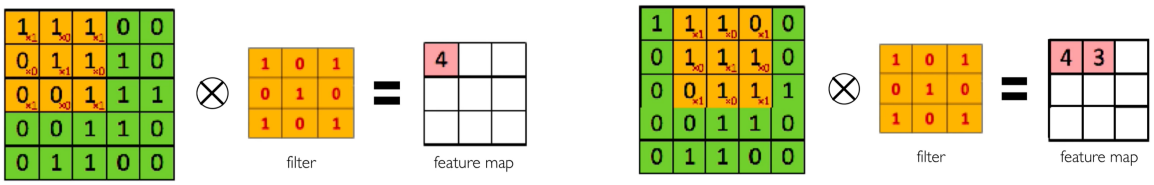
- The image below shows what we do with the operation of the filters

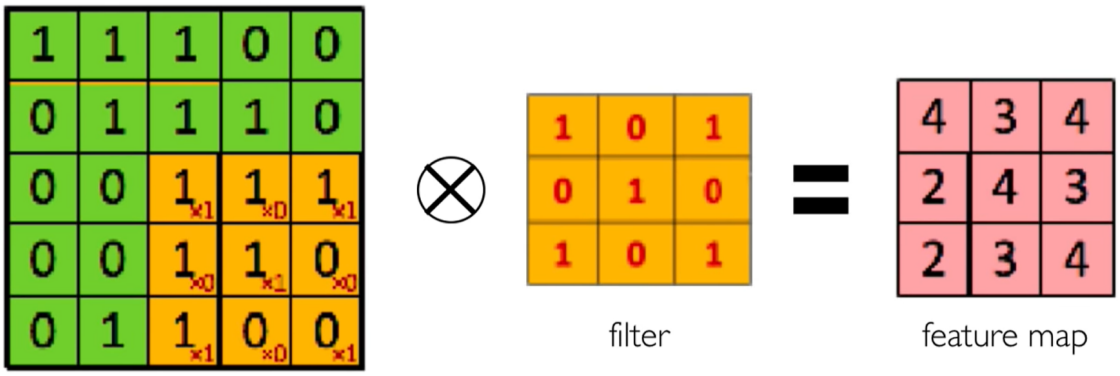


- If we want to perform a convolution for a 5*5 grid with a 3*3 filter, the image shown below



- We want to slide the filter through the image and multiply the output
- Here are the steps of how it's done:



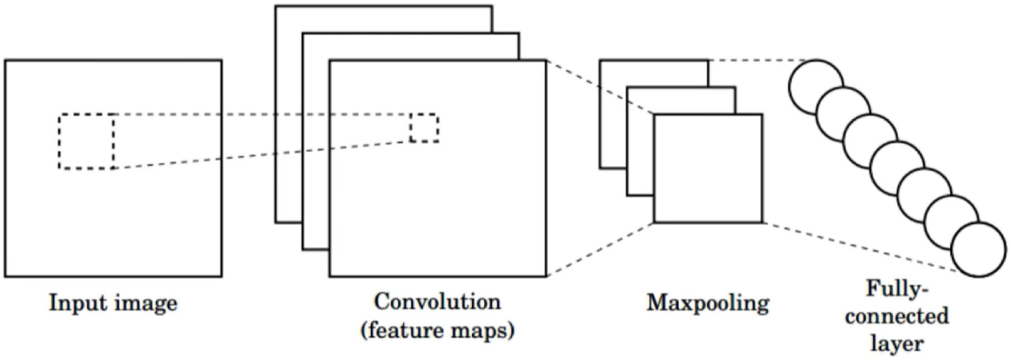


- Keep repeating this process until the feature map has been filled completely

Producing Feature Map

- By changing the filter of the image it could change what the convolution is looking for

Convolution Neural Networks (CNNs)



1. Convolution: Apply Filter to generate feature maps

`tf.keras.layers.Conv2D`

1. Non-linearity: Often ReLU

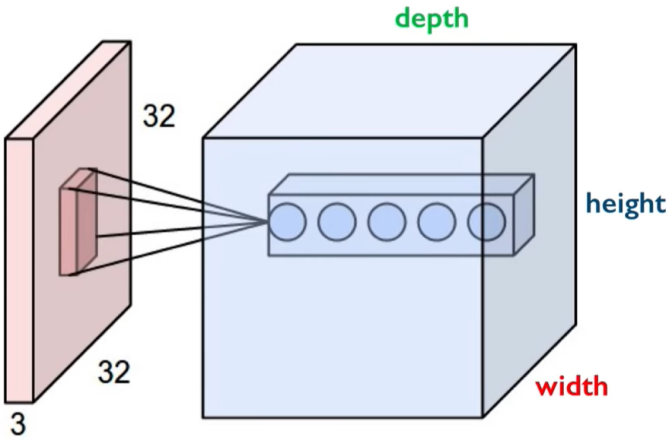
`tf.keras.layers.MaxPool2D`

1. Pooling: Down-Sampling operation of each feature map

`tf.keras.activations.*`

CNNs special arrangement of output volume

- The convolution output could be in multiple dimensions



Layer Dimensions:

$h \; x \; w \; x \; d$

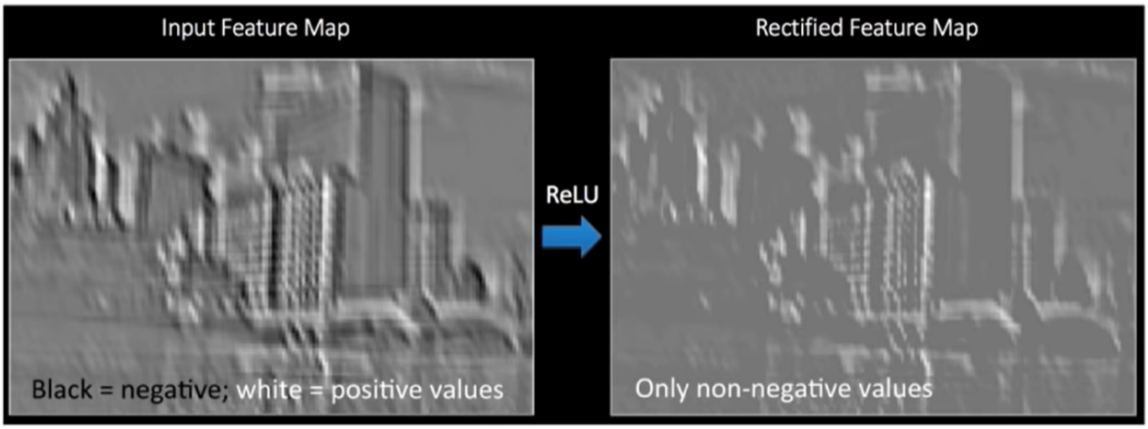
where h and w are spatial dimensions
d (depth) = number of filters

Stride:
Filter step size

Receptive Field:
Locations in input image that
a node is path connected to

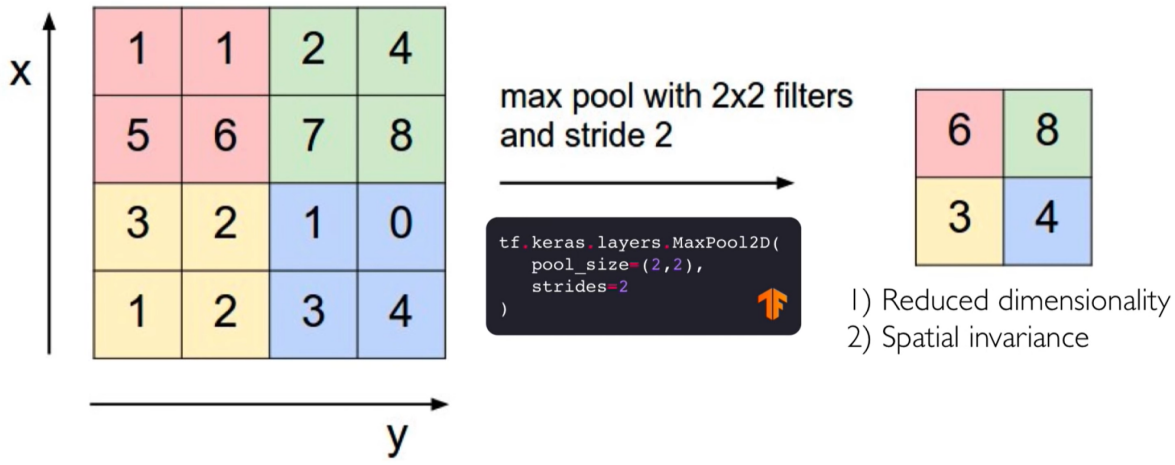
Non-linearity

- Non-linearity is applied after every convolution operation
- ReLU replaces all negatives to zero



Polling

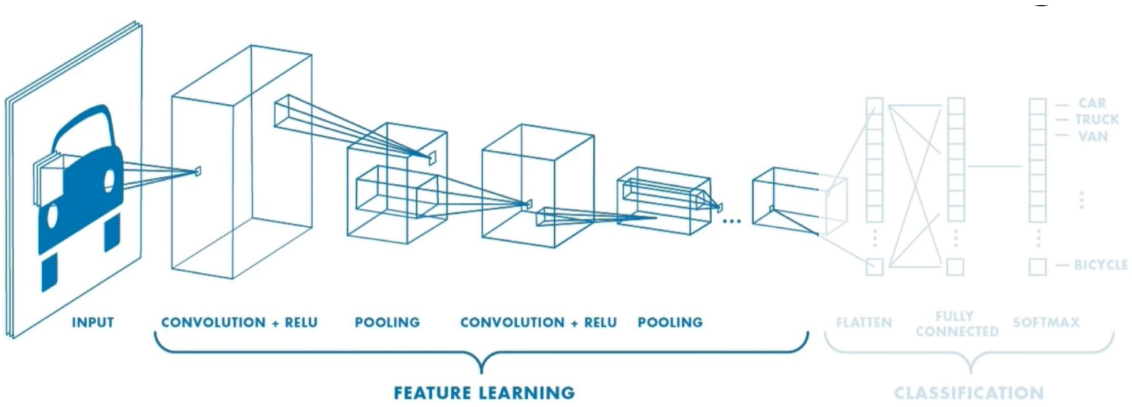
- Polling selects a greatest or lowest value in the specific area and replace the entire area with that single number



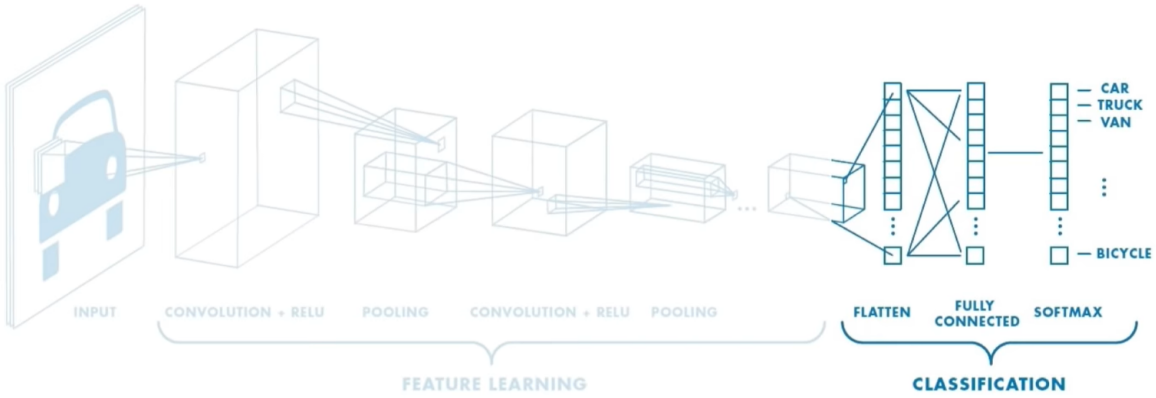
- This down-samples the image
- It allows us to shrink the image and still maintain the spatial structure

Steps for CNN networks

- The first part is Feature learning
  - Learns the image through convolution
  - introduces non-linearity
  - Reduce size and dimension through pooling

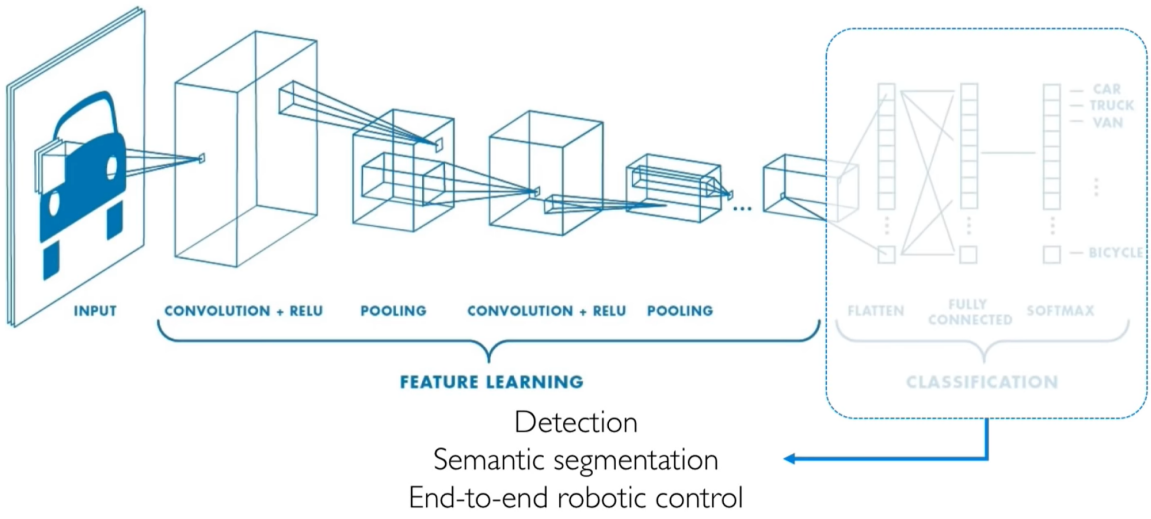


- The second part is classification
  - CONV and POOL output high level features of input
  - fully connected layers use this to classify the input image
  - express the probability of it belonging to a particular class
- The activation is softmax

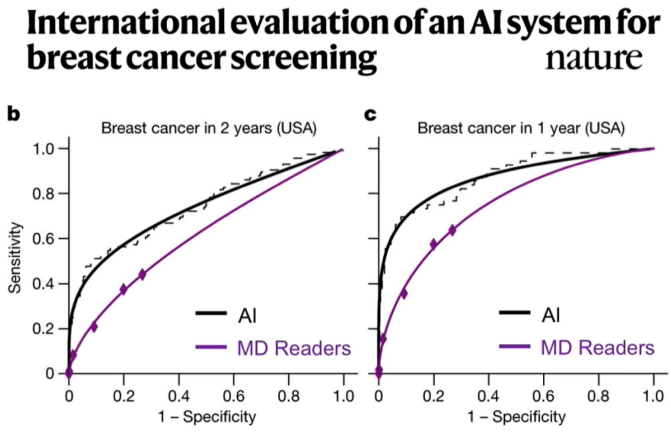


Applications

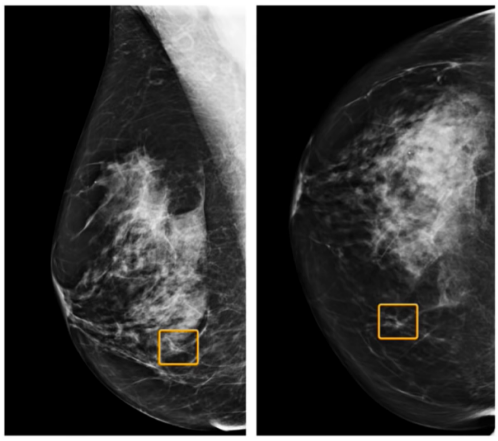
- CNNs not only can be used for classification, it could also be used for
  - Detection
  - Sematic segmentation
  - End-to-end robotic control



Detection: Breast Cancer Screening



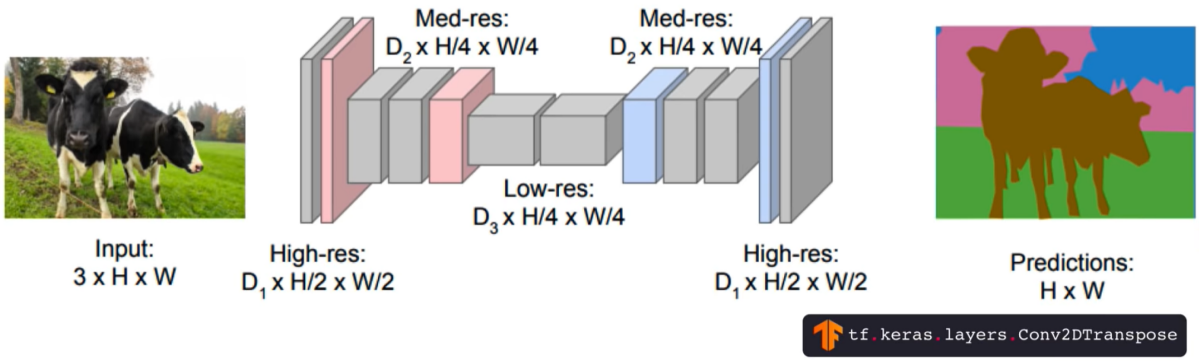
CNN-based system outperformed expert radiologists at detecting breast cancer from mammograms



Breast cancer case missed by radiologist but detected by AI

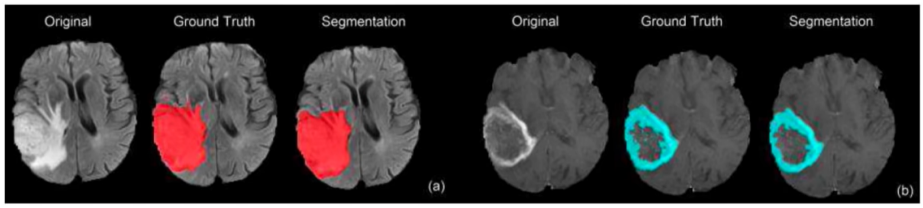
Semantic Segmentation: Fully Convolutional Network

FCN: Fully Convolutional Network.
Network designed with all convolutional layers,
with **downsampling** and **upsampling** operations

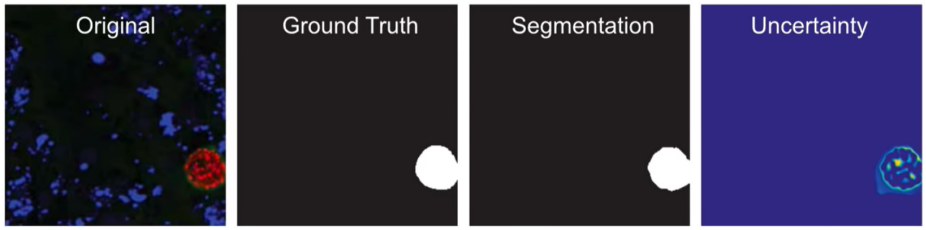


Semantic Segmentation: Biomedical Image Analysis

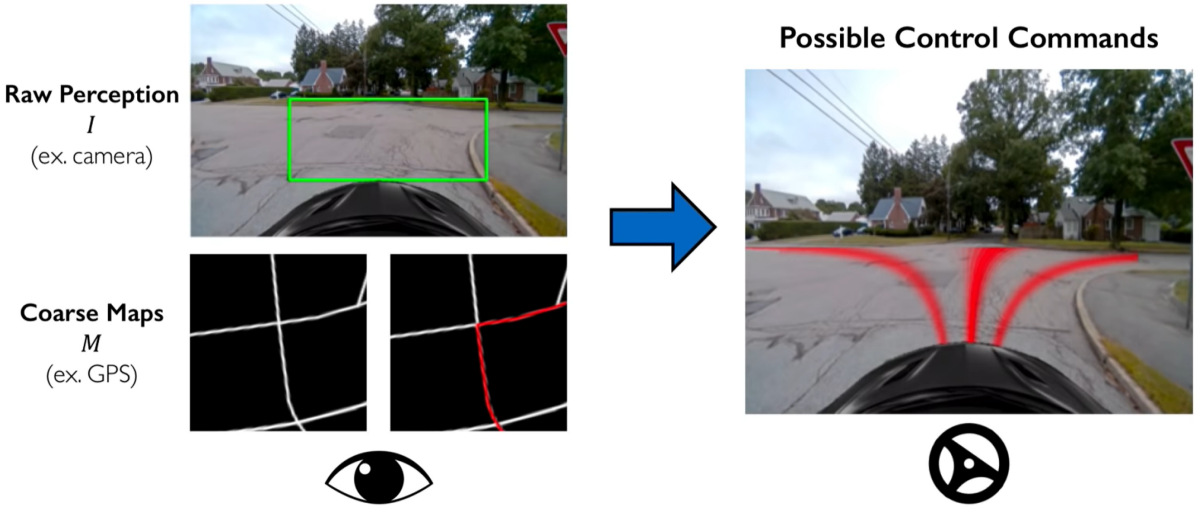
Brain Tumors
Dong+ *MIUA* 2017.



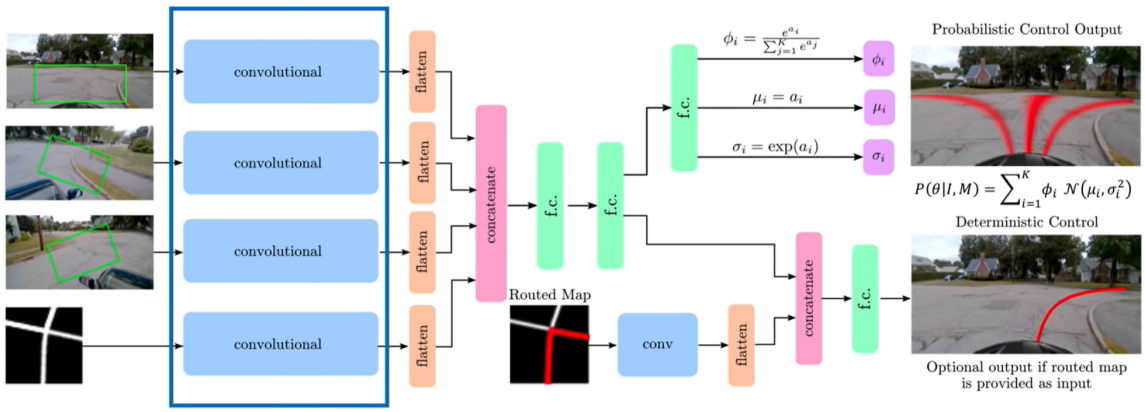
Malaria Infection
Soleimany+ *arXiv* 2019.



Self-Driving Car: Navigation from Visual Perception



Entire model is trained end-to-end **without any human labelling or annotations**



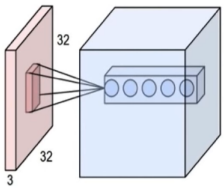
Summary

Foundations

- Why computer vision?
- Representing images
- Convolutions for feature extraction

CNNs

- CNN architecture
- Application to classification
- ImageNet



Applications

- Segmentation, image captioning, control
- Security, medicine, robotics