NUS-ISS
Vision Systems

Module 5 - Building vision system using machine learning (1) - Detection and recognition, part 1

Dr. Tan Jen Hong
Lecturer & Consultant
Institute of System Science
National University of Singapore
issjht@nus.edu.sg

Learning objectives

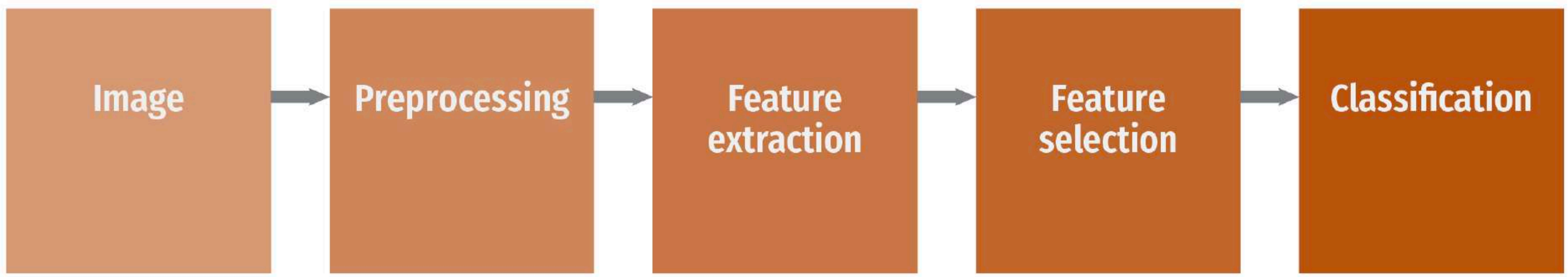
- Understand the major steps in image classification

- Understand the major challenges in image classification

- Perform image classification using opencv deep learning module

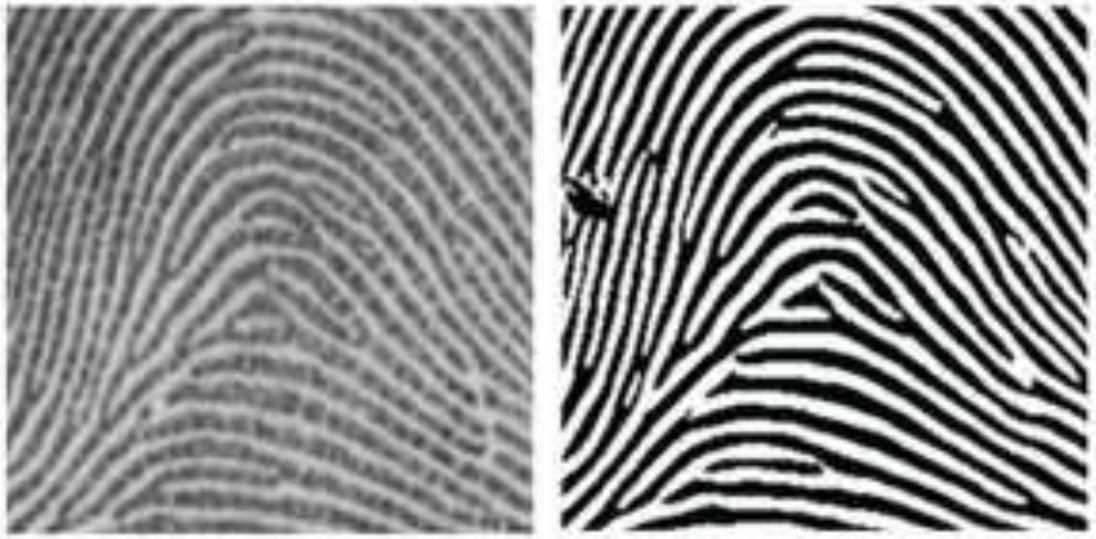
Life before deep learning …

- Some of the major steps taken to classify an image



Preprocessing

- Remove unwanted regions
- Image adjustment (rotating, resizing and etc…)
- Noise removal / image enhancement
- Contrast adjustment (histogram equalization, CLAHE, subtraction from mean and etc…)
- Perform transformation (Fourier, shearlet, wavelet, radon and etc …)



Source: https://www.researchgate.net/publication/258037532_A_Study_on_User_Authentication_Methodology_Using_Numeric_Password_and_Fingerprint_Biometric_Information/figures?lo=1

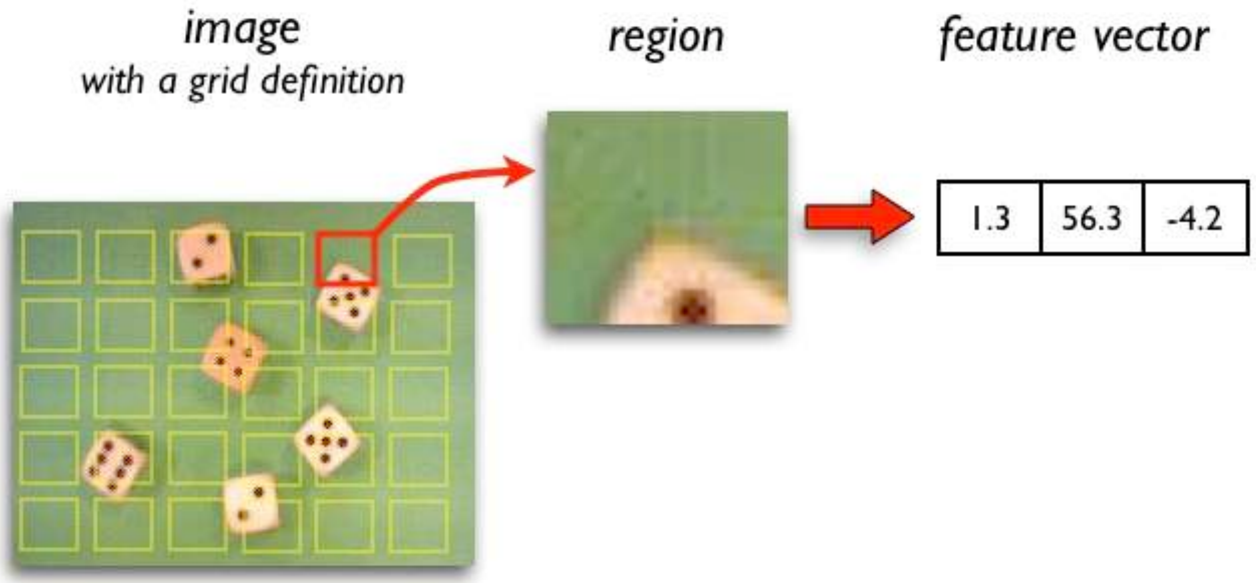
Feature extraction

- Feature: a piece of information useful for discrimination between classes

- Some are visually related; many are not

- Texture: patterns or variations among pixels

- Shapes, edges, orientations, histograms

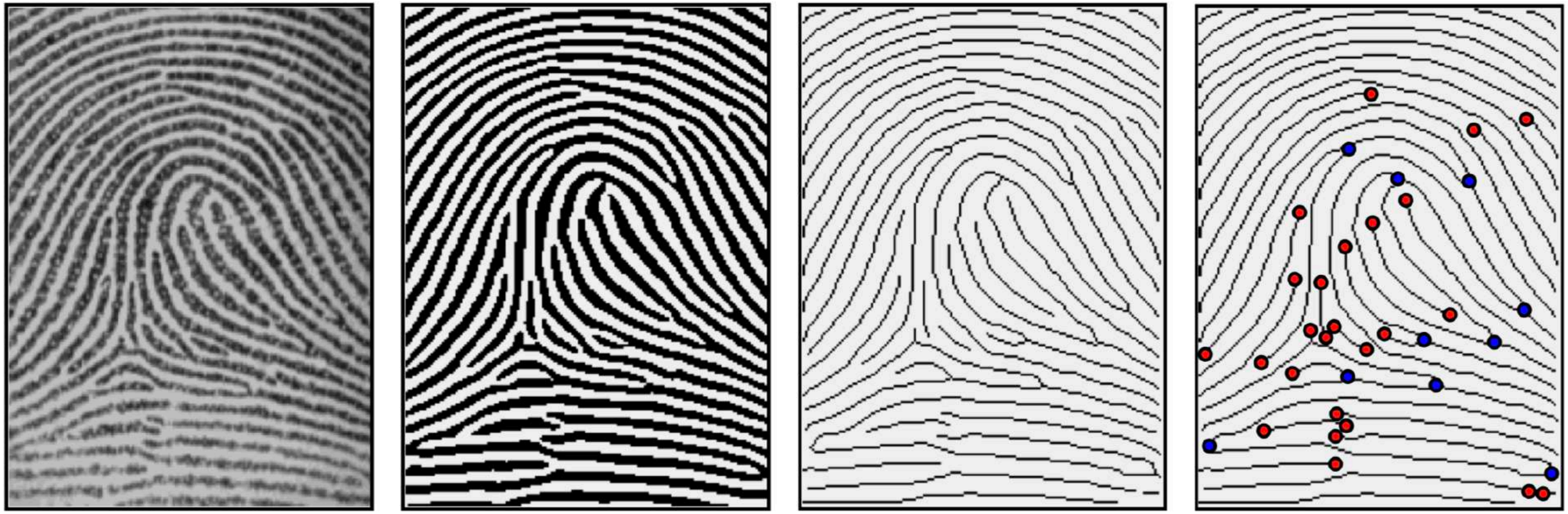


Source: http://doc.perclass.com/perClass_Toolbox/guide/feature_extraction/region.html

Features extraction is tough

The case of a finger print

- Use the position and orientation of termination (red) and bifurcation (blue) to describe a finger print



Source: https://cedar.buffalo.edu/~govind/CSE666/fall2007/FP_Tutorial.pdf

Feature selection

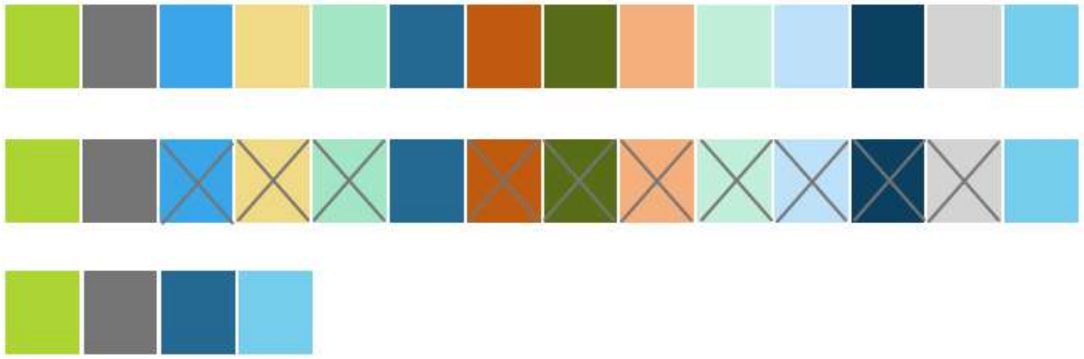
- Too many features generated; slow or not possible to train and classify

- Some features are redundant; some are irrelevant

- Improve generalization by reducing overfitting

- Some of the methods commonly used:

  Linear discriminant analysis
  Genetic algorithm
  Principal component analysis
  Partical swarm optimization
  Ants colony
  ...

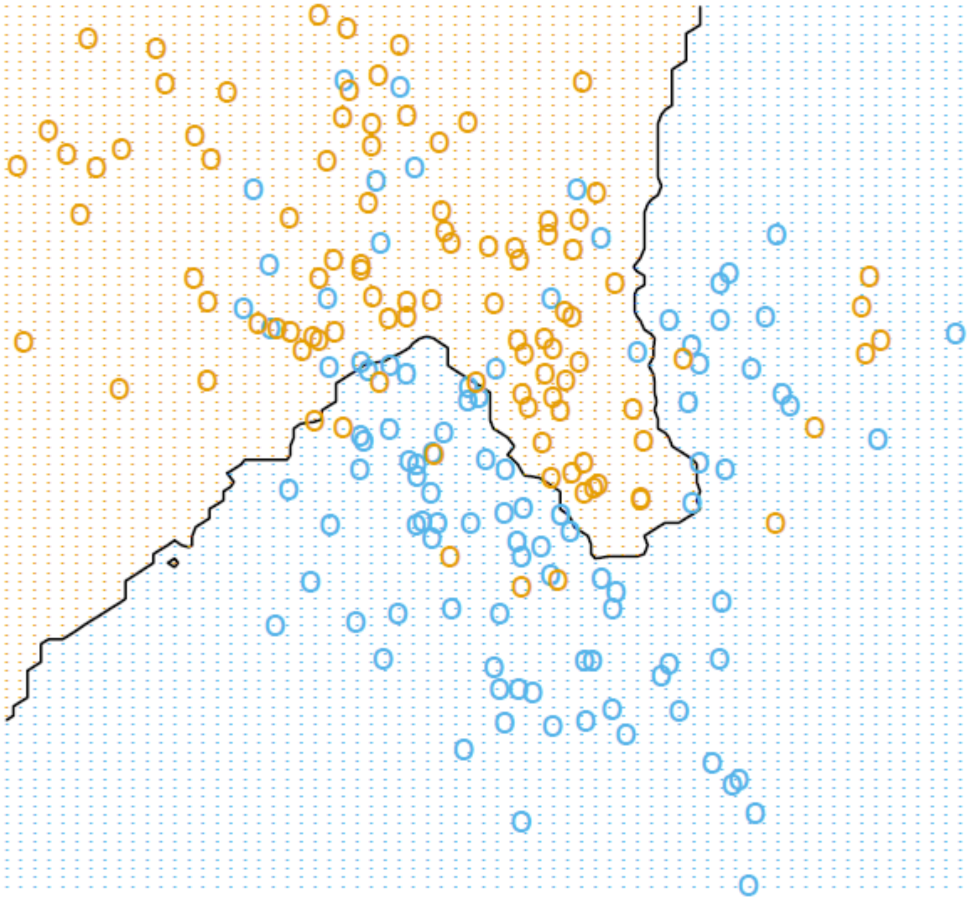


Source: https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96

Classification

- With all the features read/prepared, fed into classifier

- Get algorithms to learn about the patterns under data set, rather than specify rules to determine class



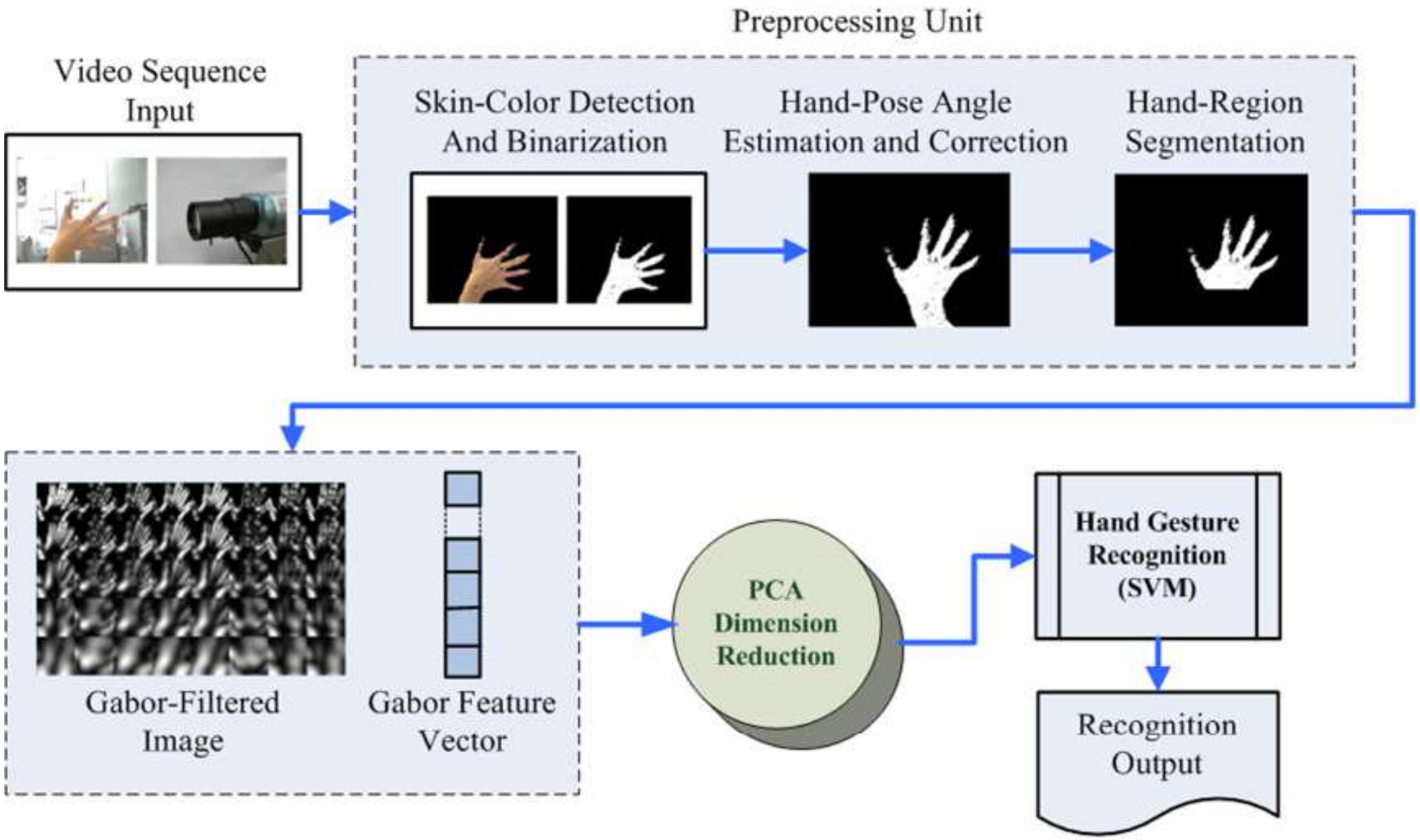
Source: https://stats.stackexchange.com/questions/21572/how-to-plot-decision-boundary-of-a-k-nearest-neighbor-classifier-from-elements-o

- Some of the classifiers commonly used:

k nearest neighbour
Neural network
Decision tree
Naive Bayes
...

Features + classifier

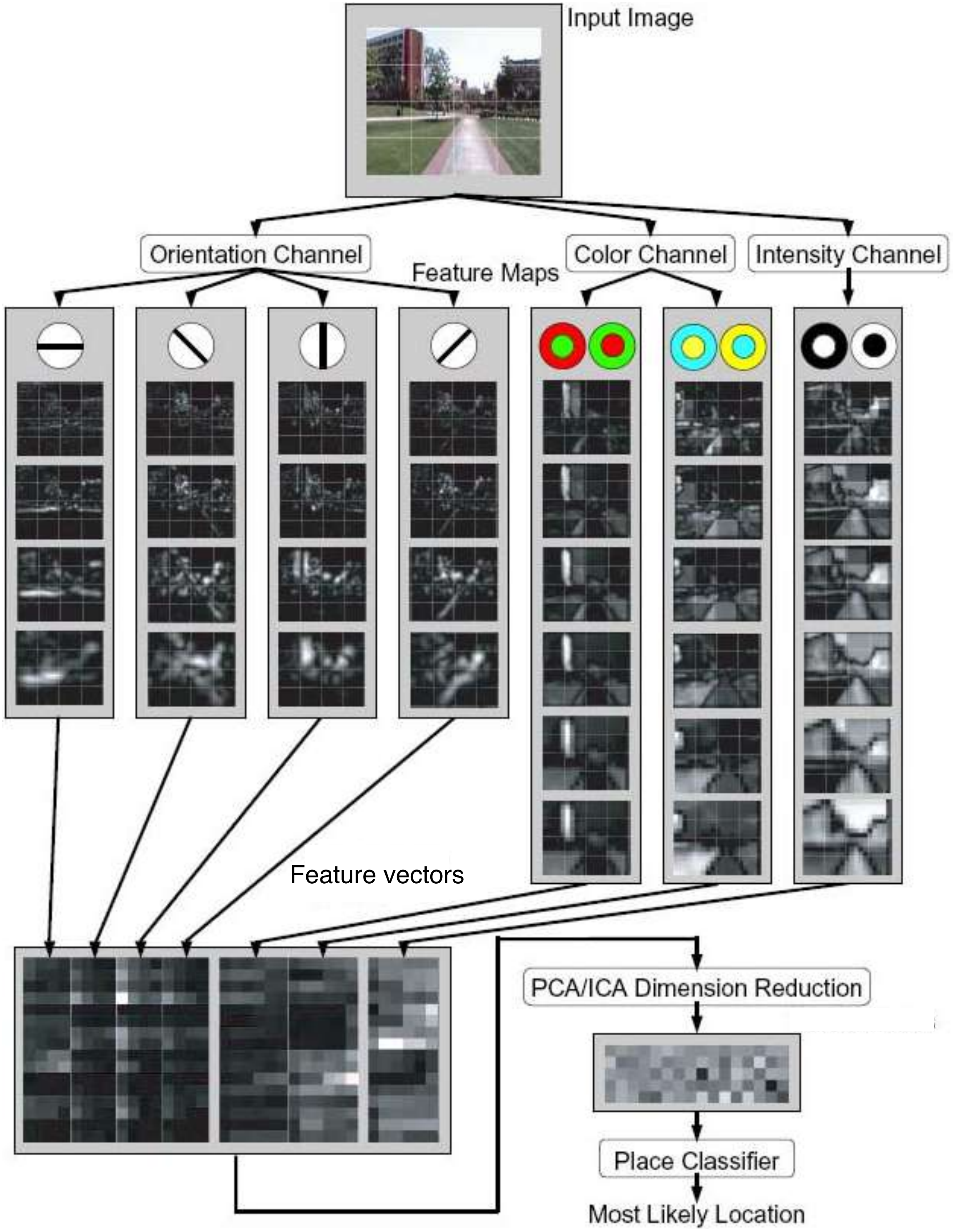
Gabor filtering



Source: https://doi.org/10.1016/j.eswa.2010.11.016

Features + classifier

Plethora of features for scene recognition



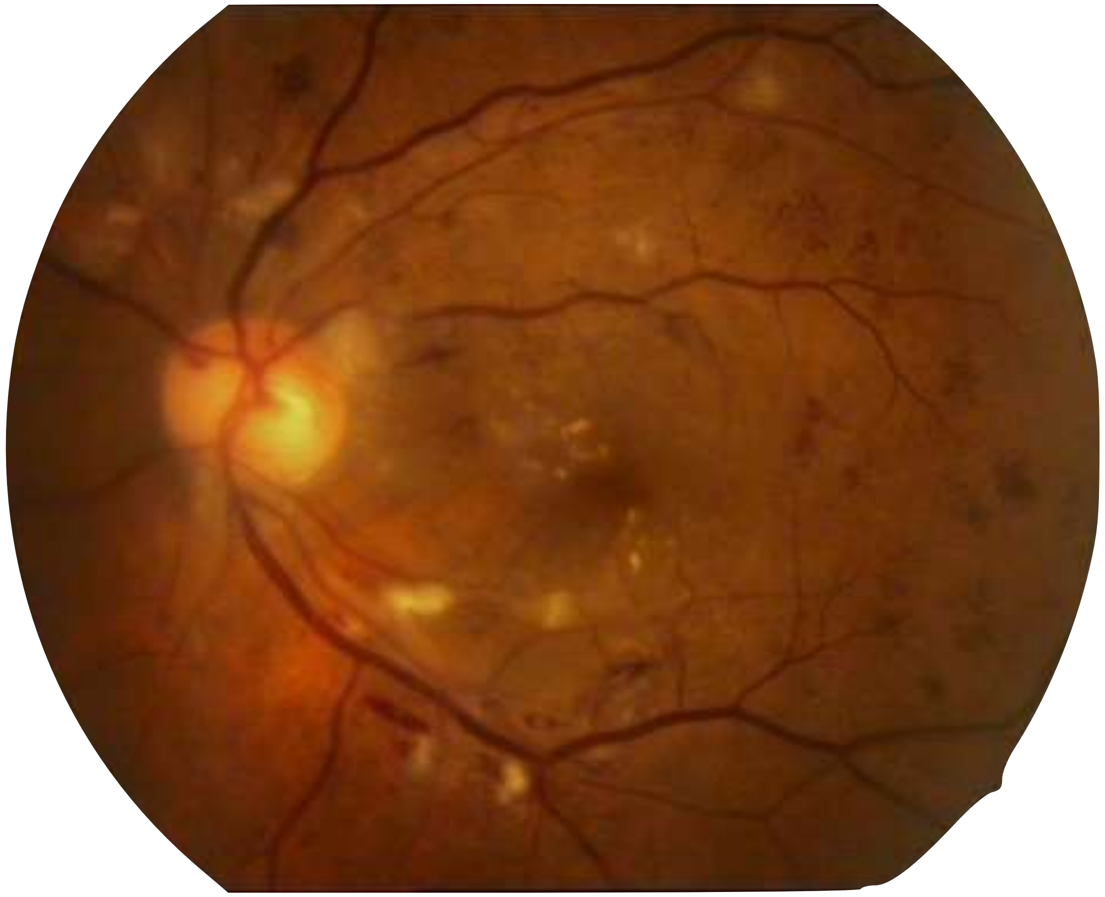
Source: http://ilab.usc.edu/siagian/Research/Gist/Gist.html

Exercise

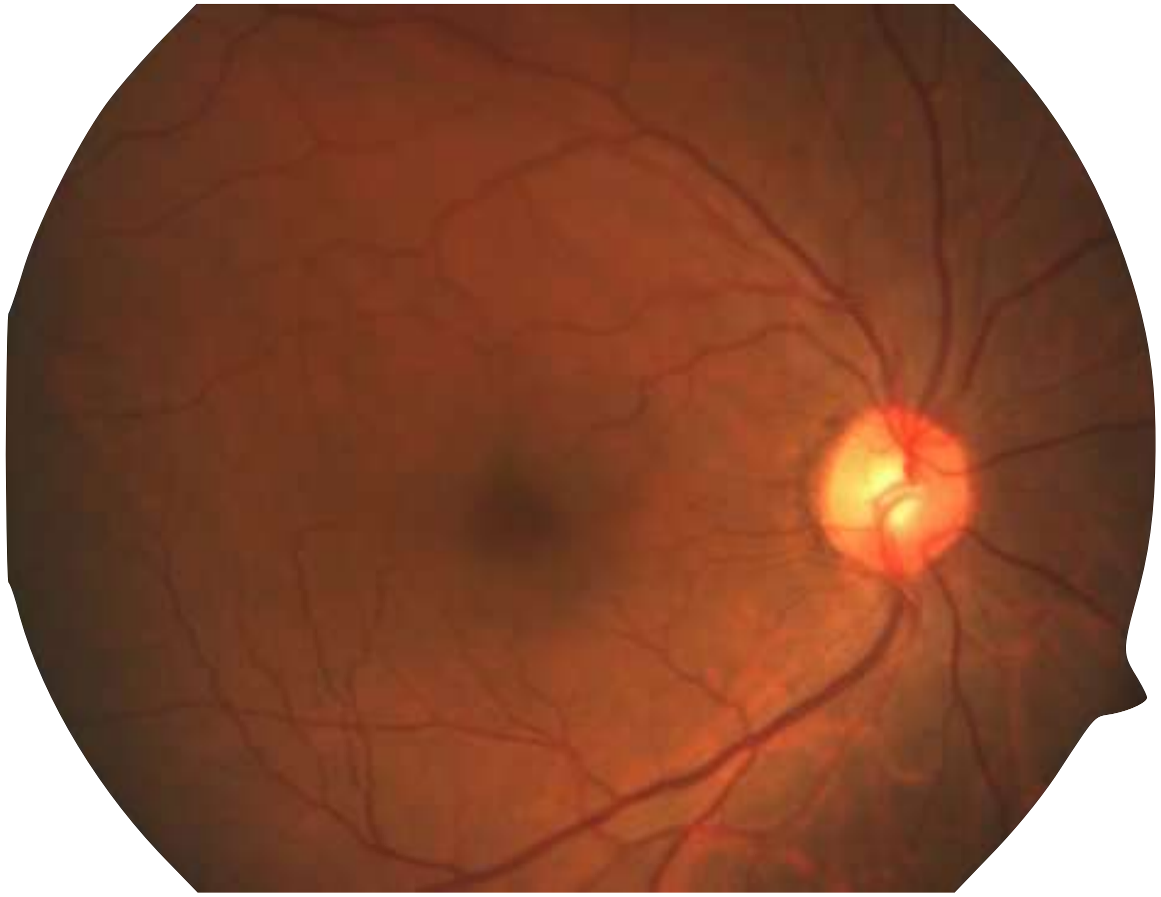
Differentiating diabetic retinopathy and normal

- Propose a solution to extract the features that can allow us to differentiate normal retinal image and image of diabetic retinopathy
- Write your solution in Word or PowerPoint
- Show preliminary results from python code if possible

Diabetic retinopathy

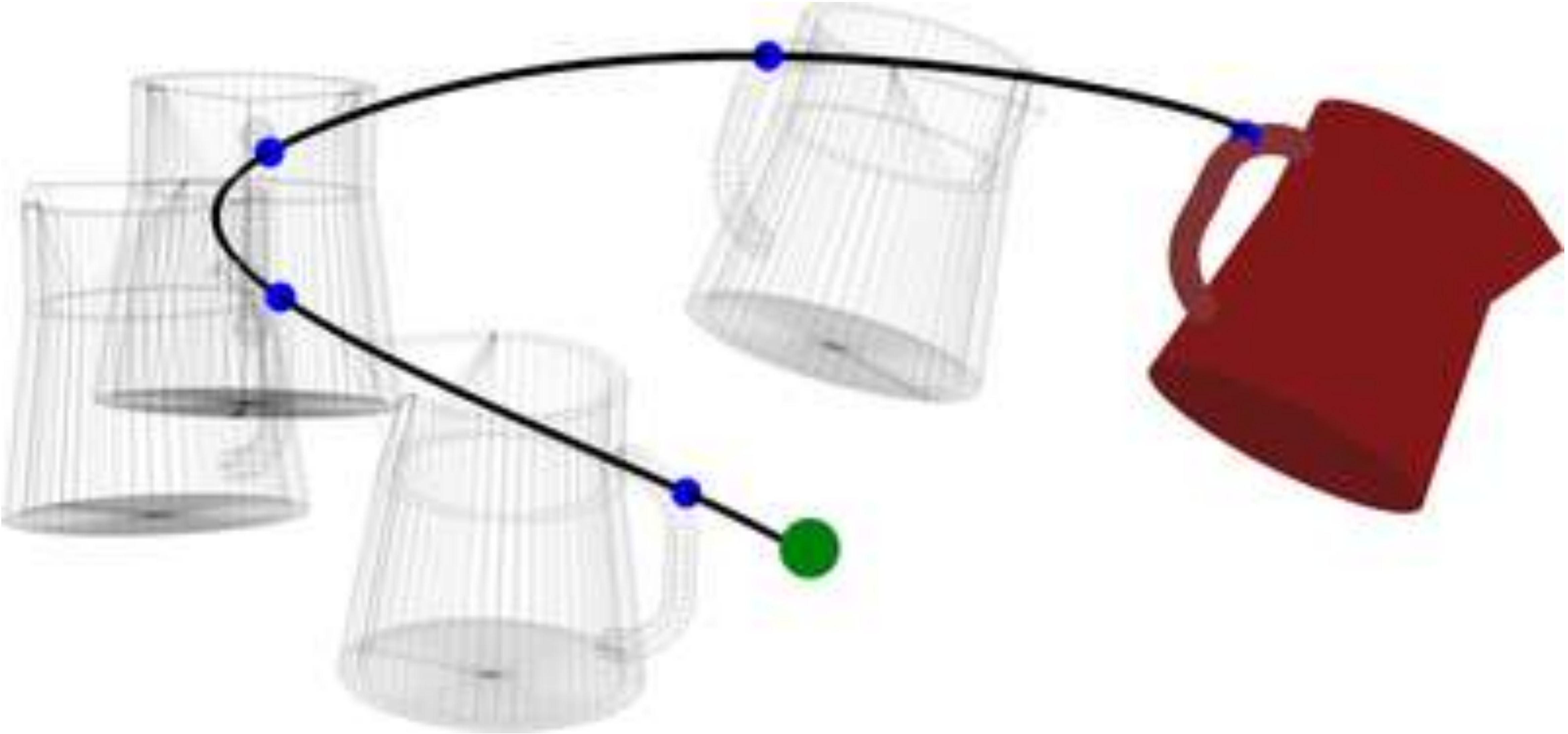


Normal



Main challenges

1. View point variation



Source: https://www.mech.kuleuven.be/en/pma/research/robotics/research/technology/invariants

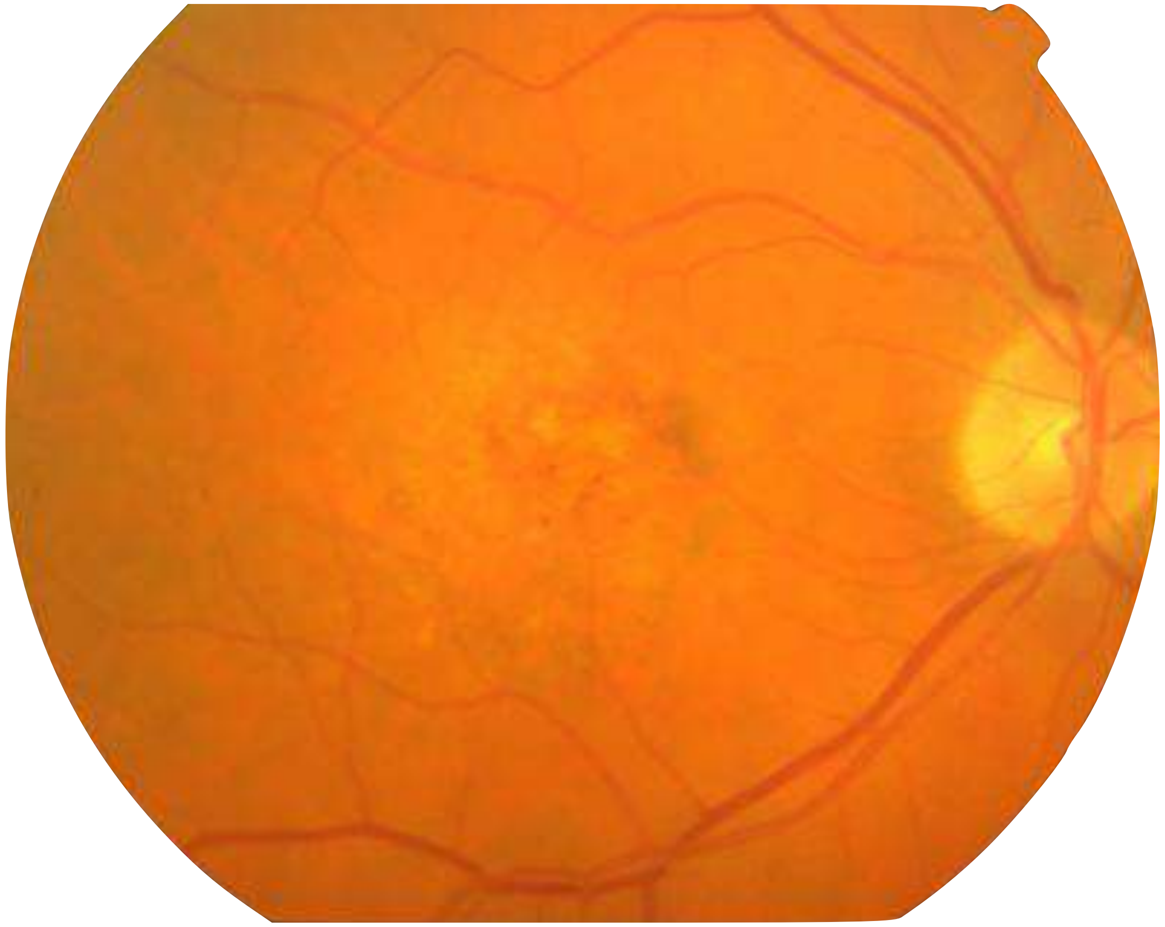
Main challenges

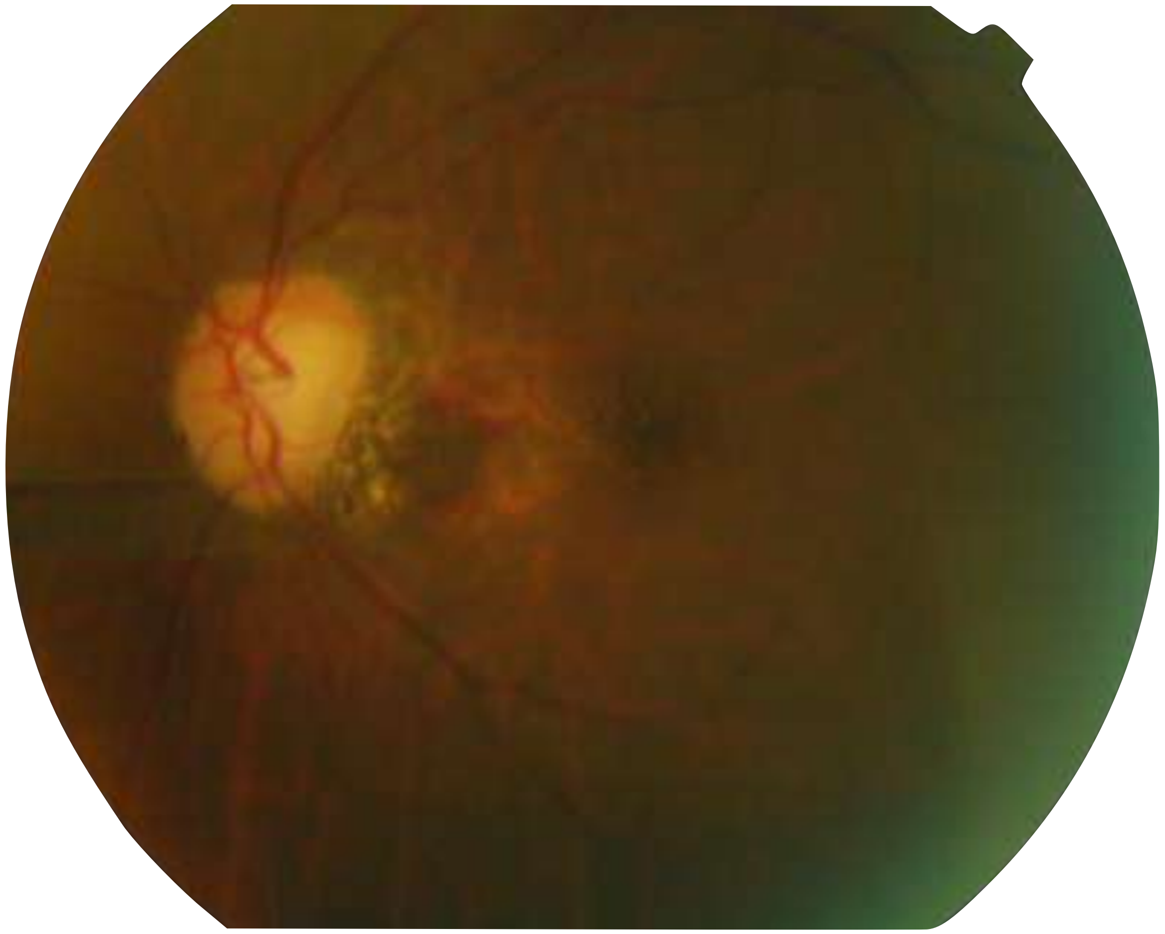
2. Illumination

Source: https://www.yankodesign.com/2017/02/15/a-variation-on-illumination/

Main challenges

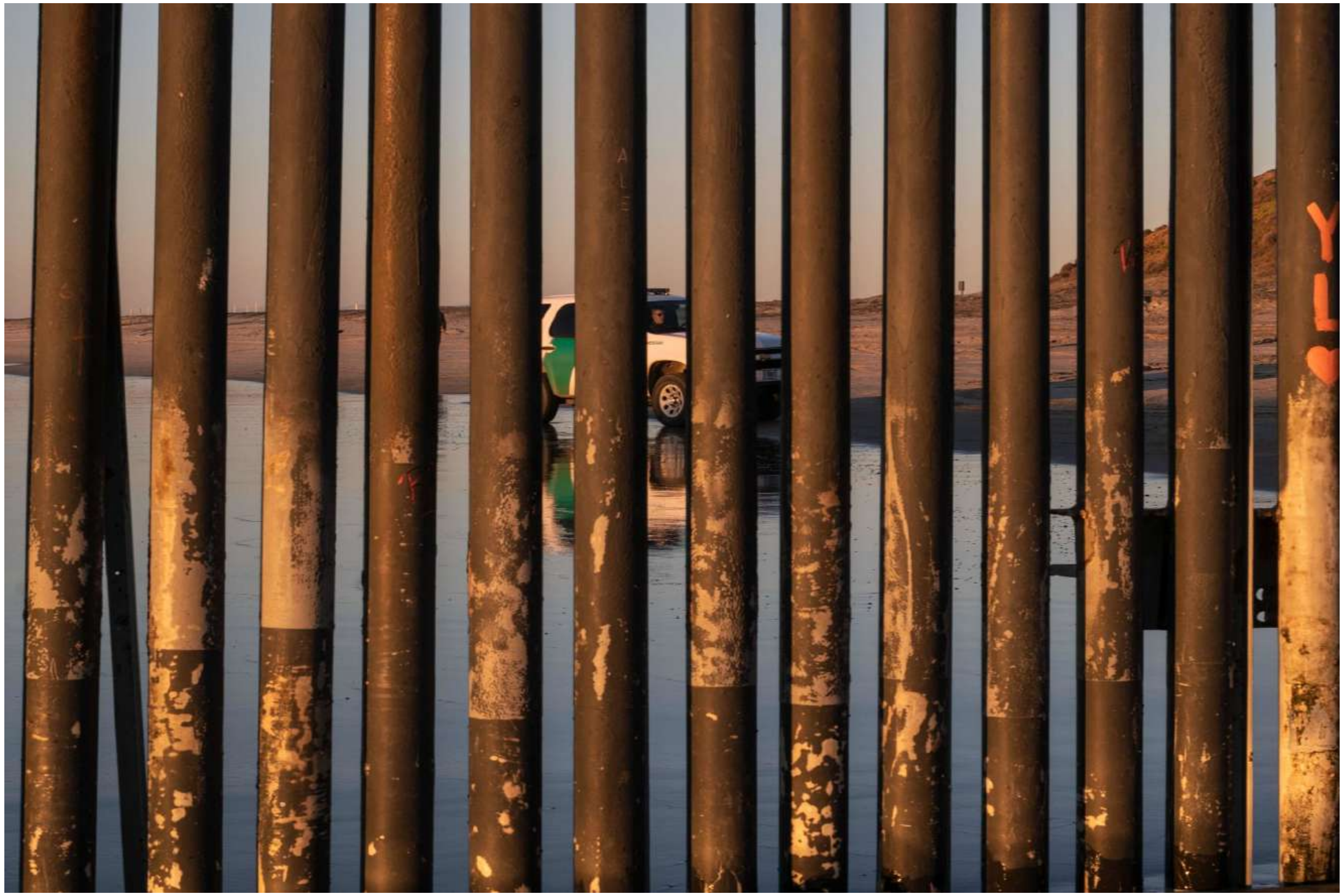
2. Illumination





Main challenges

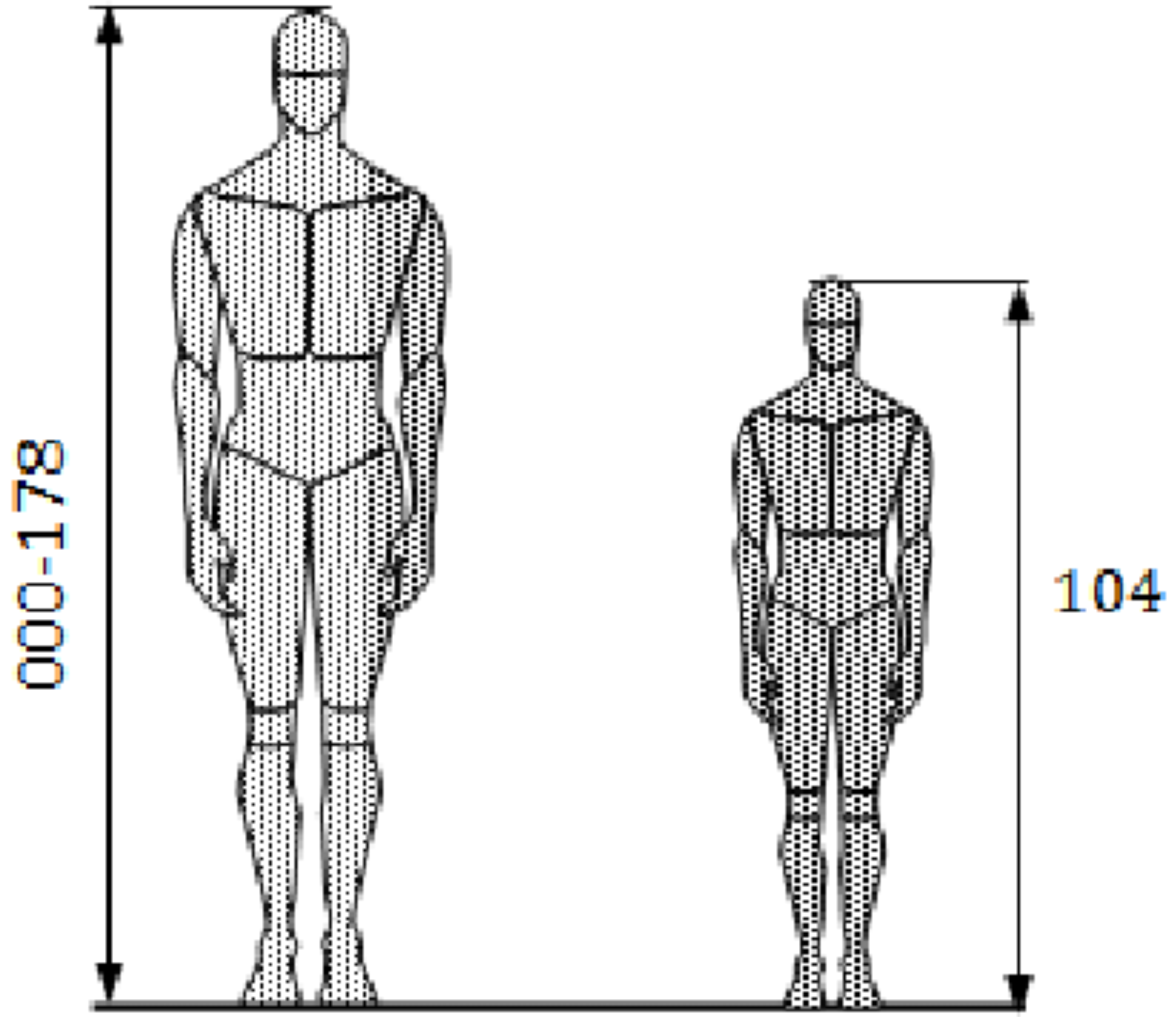
3. Occlusion



Source: Guillermo Arias/Agence France-Presse — Getty Images

Main challenges

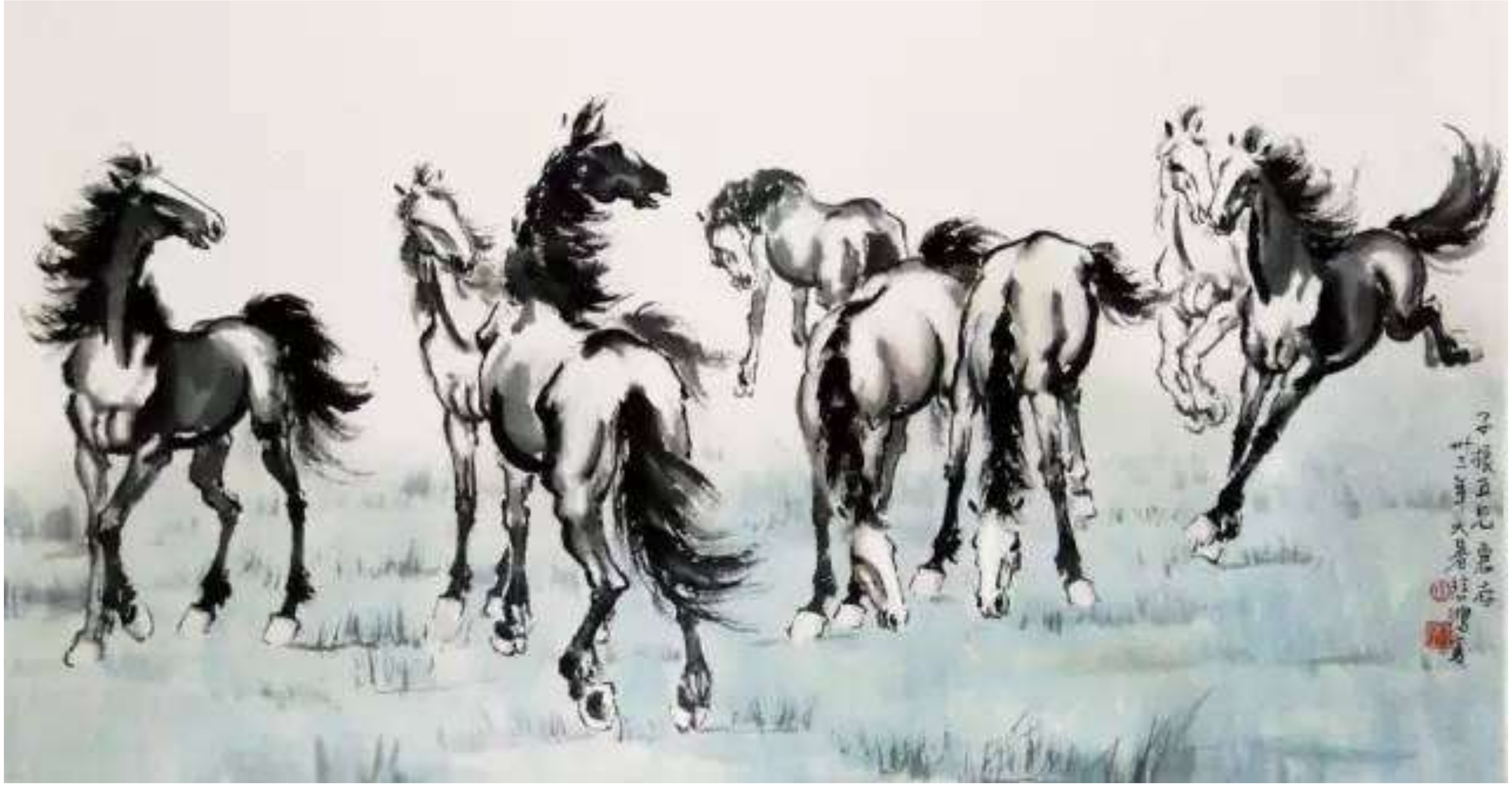
4. Scale



Source: https://www.researchgate.net/publication/327556544_ISO_Standards_and_European_Norms_for_Size_Designation_of_Clothes/figures?lo=1

Main challenges

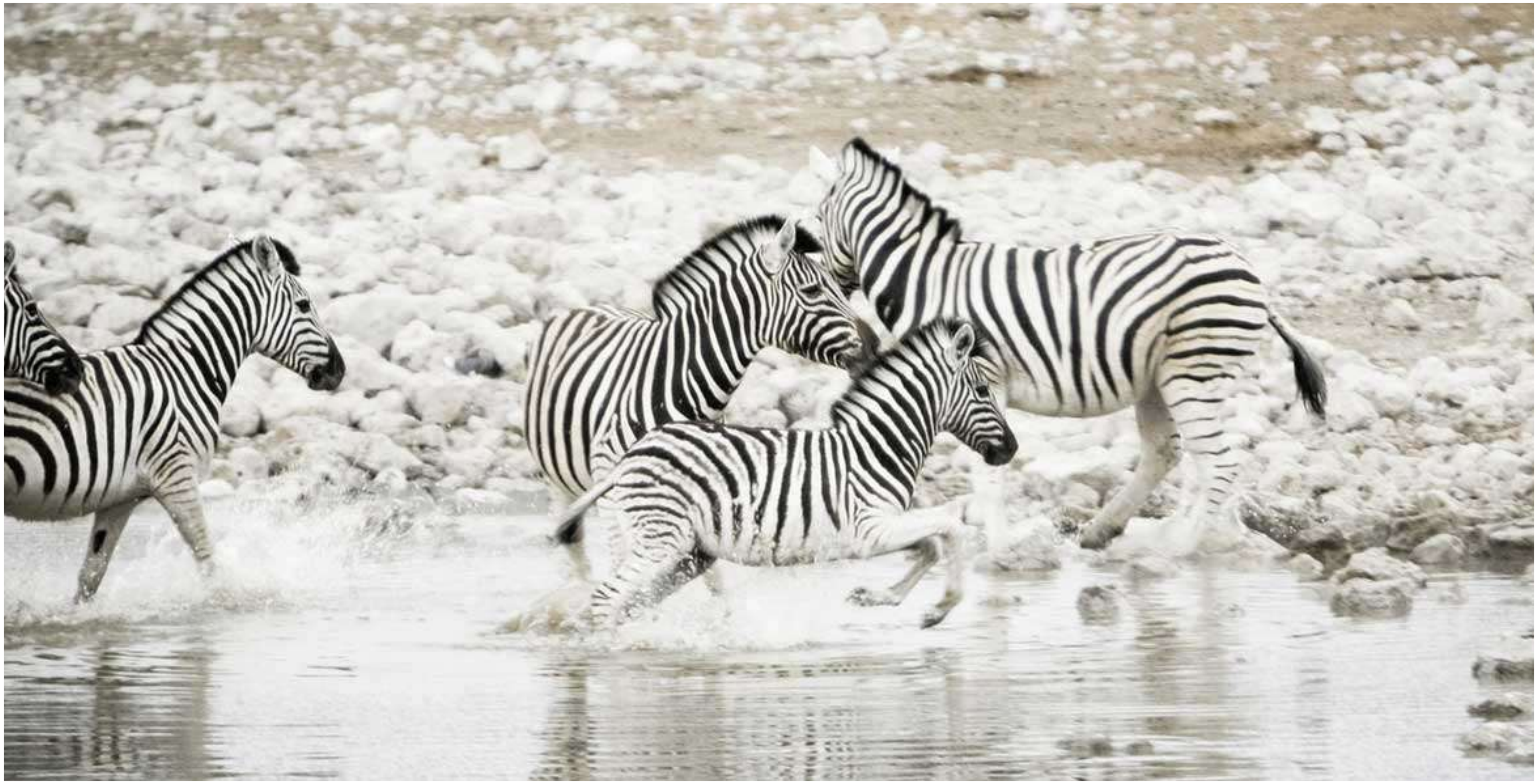
5. Deformation



Source: Xu Beihong

Main challenges

6. Background clutter



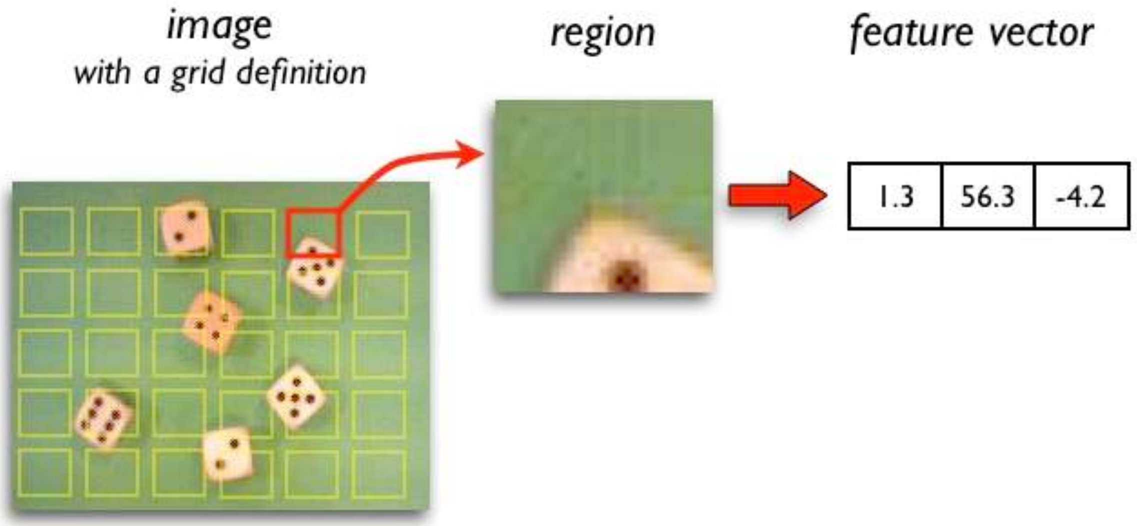
Source: http://www.photographyhut.com/background-removal/

Differences between the new and the old?

- In deep learning, feature extraction and feature selection are handled by convolutional layers

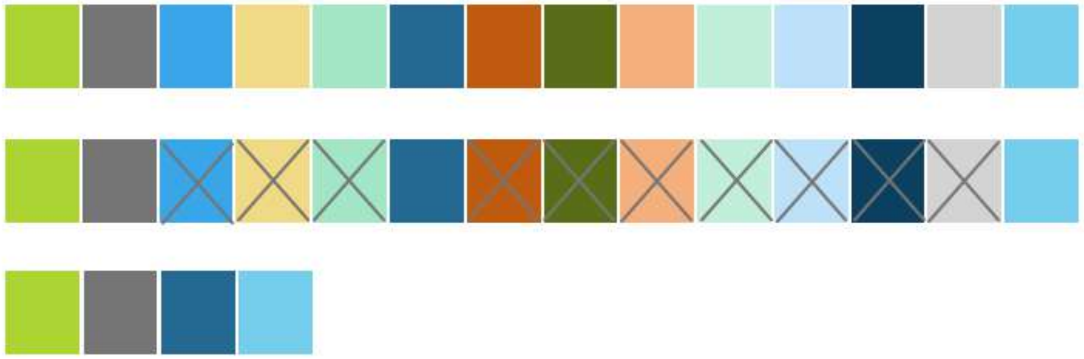
- Neural network is the classifier

Feature extraction



Source: http://doc.perclass.com/perClass_Toolbox/guide/feature_extraction/region.html

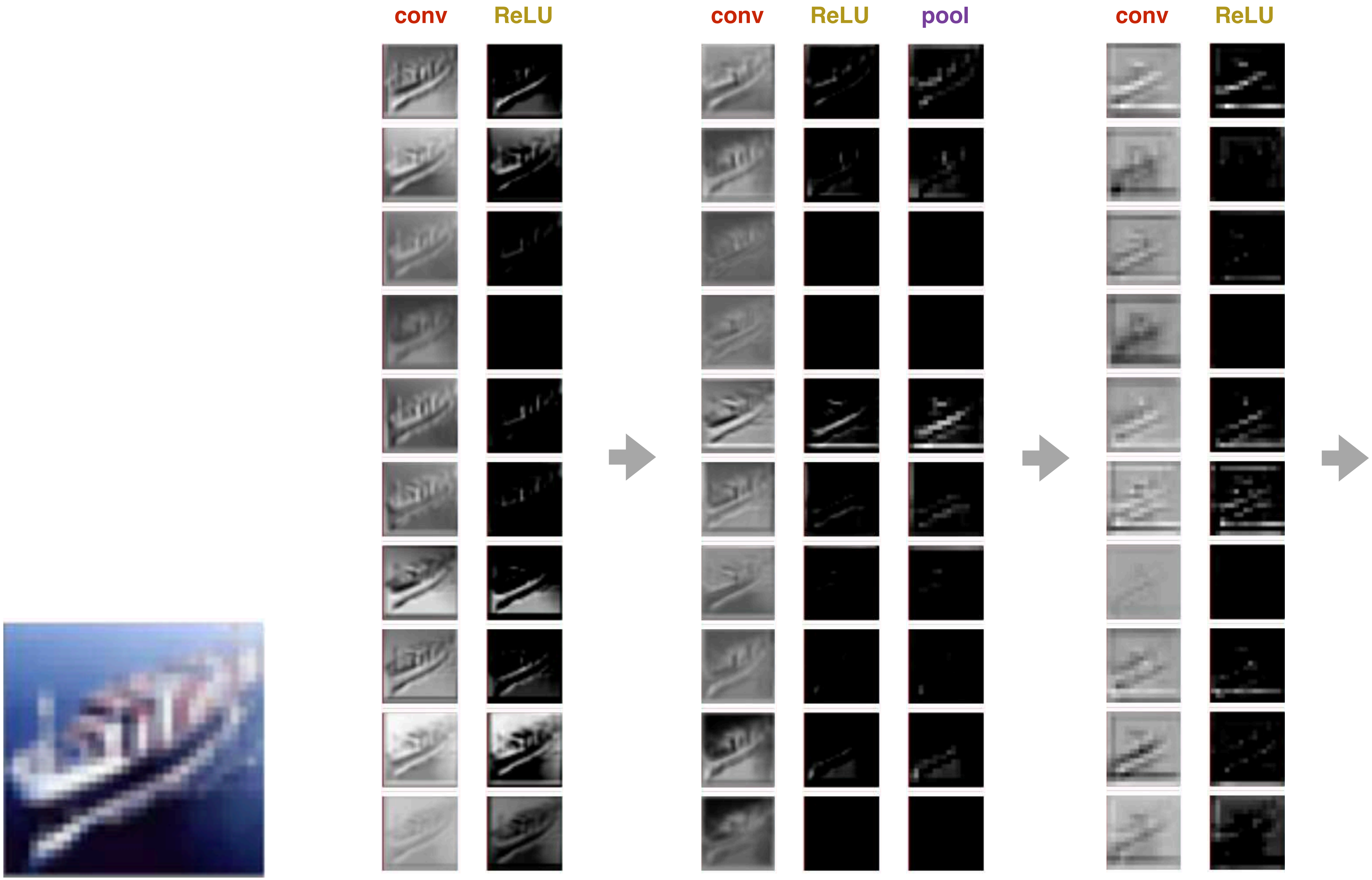
Feature selection



Source: https://medium.com/@mehulved1503/feature-selection-and-feature-extraction-in-machine-learning-an-overview-57891c595e96

Deep in action

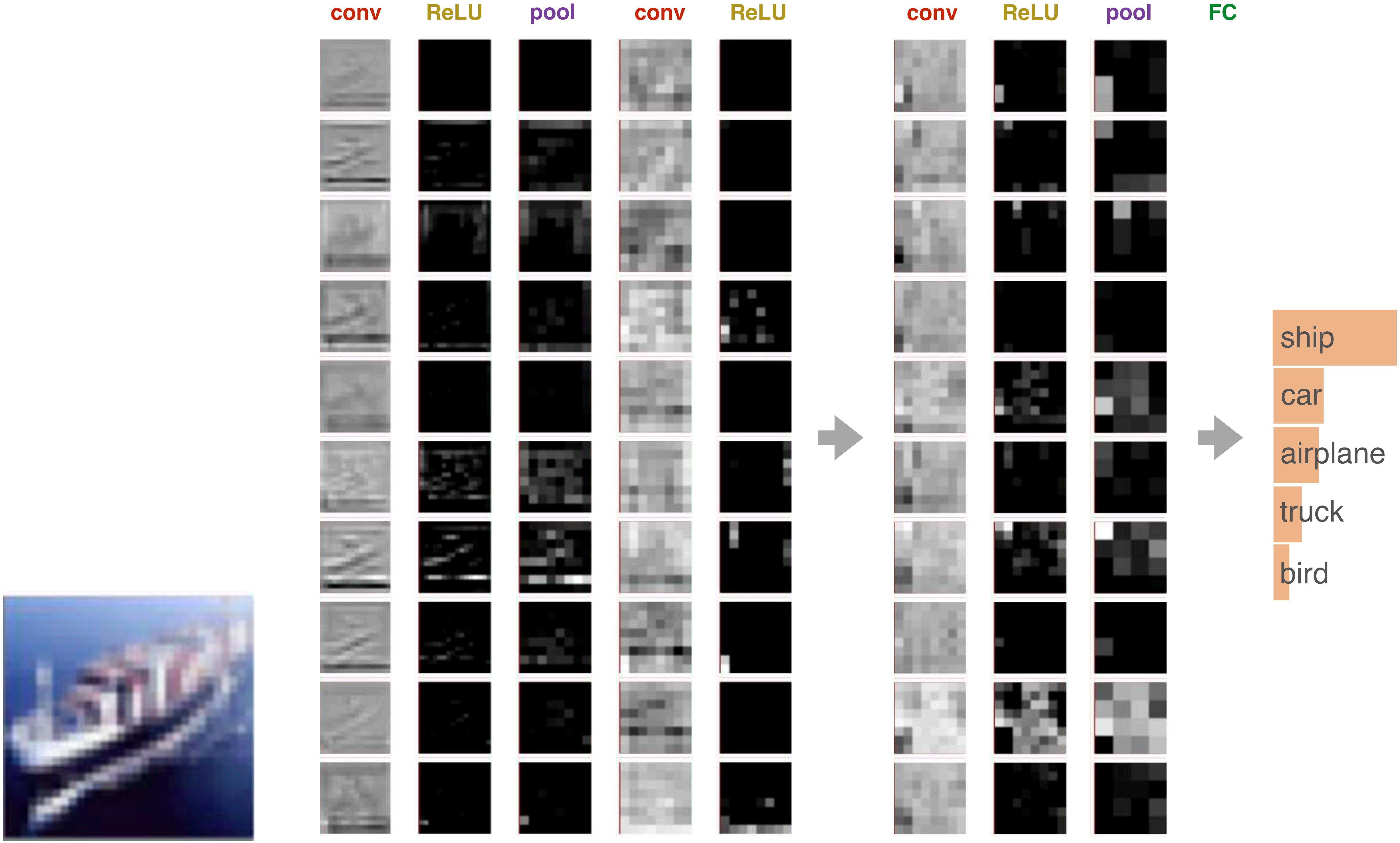
Part 1



Source: http://cs231n.stanford.edu

Deep in action

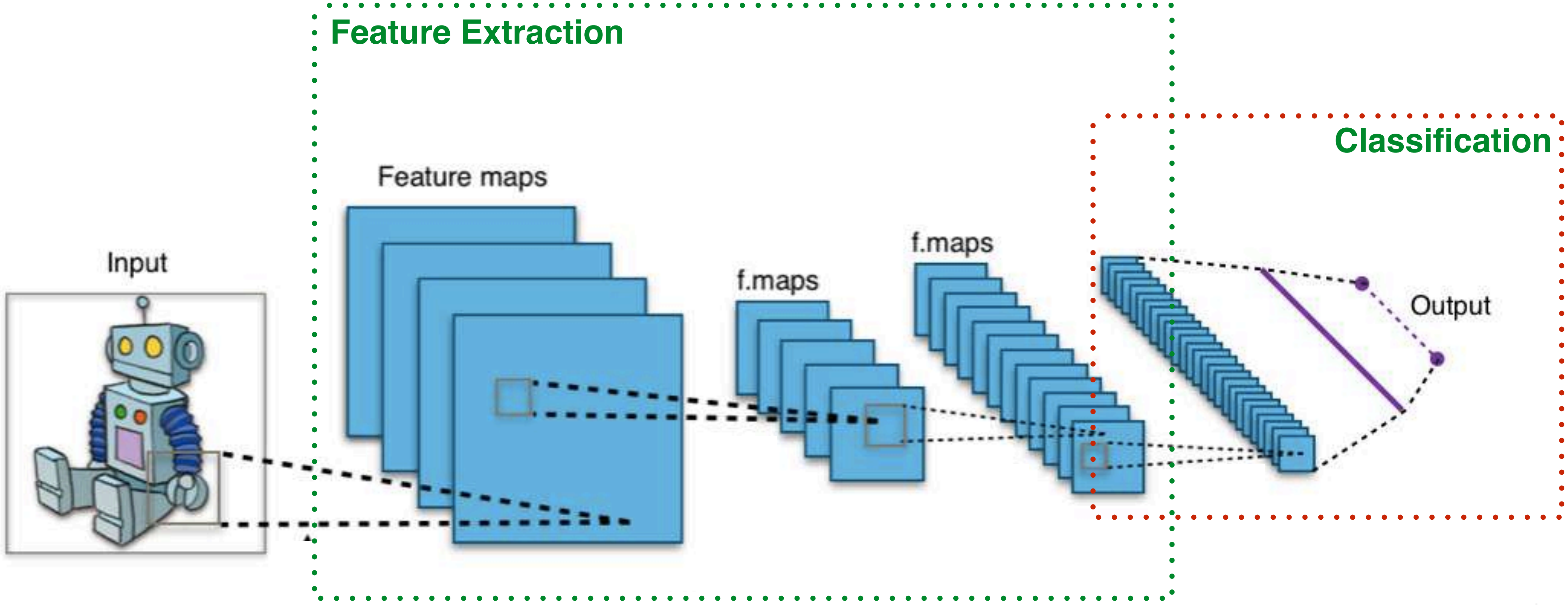
Part 2



Source: http://cs231n.stanford.edu

Learning the features

The idea behind convnet



Source: https://commons.wikimedia.org/wiki/File:Typical_cnn.png

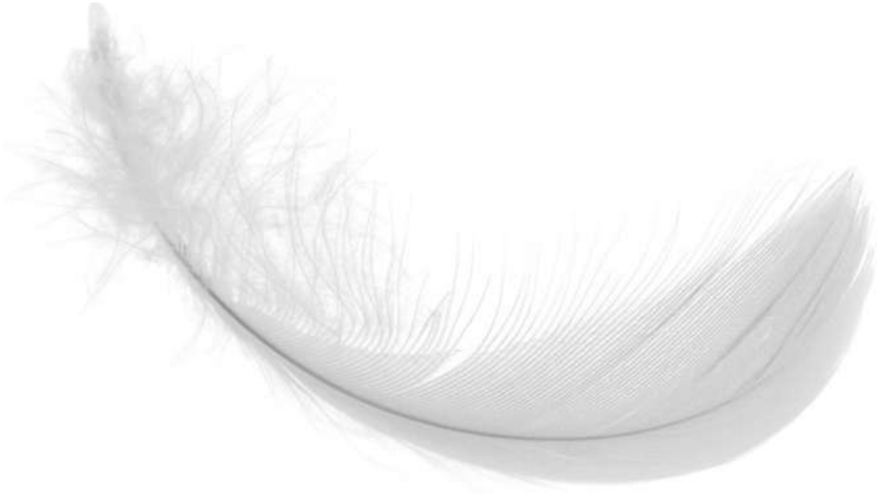
Deep learning in computer vision

- Generally train, test and run on deep learning framework, such as keras, tensorflow, pytorch, caffe2

- In production, when training is not needed, existence of the framework is redundant

- dnn module in opencv allows to run deep learning model without deep learning framework

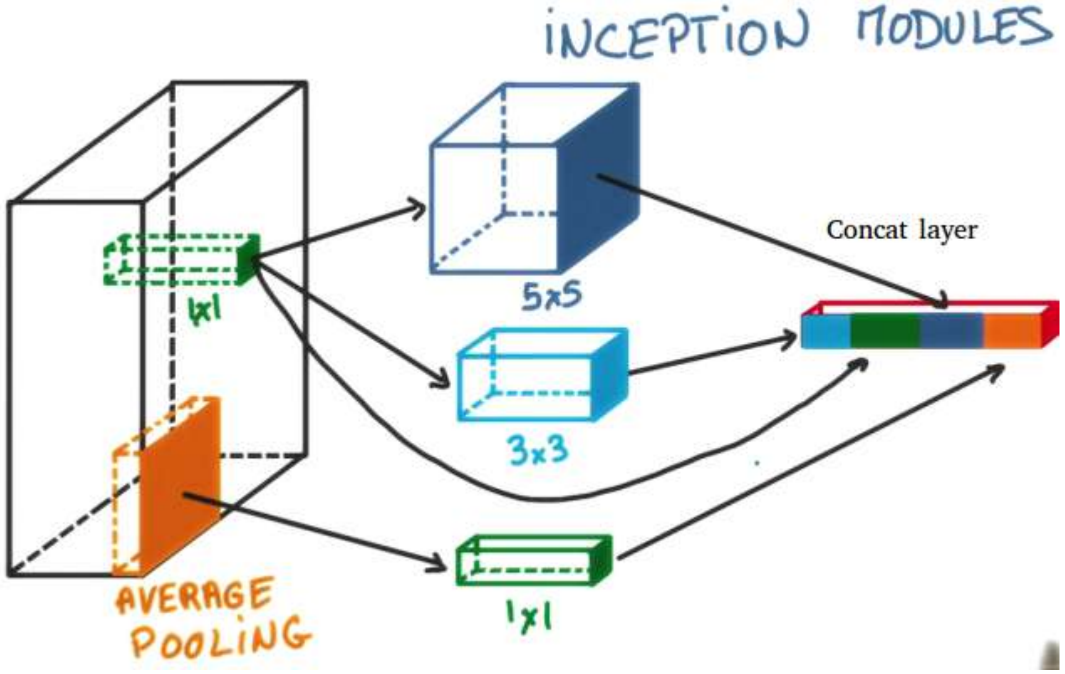
- Advantages: lightweight package, separation of code



Source: https://www.awm.gov.au/learn/memorial-boxes/1/object-list/white-feather

GoogLeNet

- Winner of 2014 ImageNet Large Scale Visual Recognition Competition
- Also called Inception v1
- https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7



Source: https://jychstar.blogspot.com/2017/04/self-driving-car-nd-a3.html

synset_words.txt

```
n01440764 tench, Tinca tinca
n01443537 goldfish, Carassius auratus
n01484850 great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias
n01491361 tiger shark, Galeocerdo cuvieri
n01494475 hammerhead, hammerhead shark
n01496331 electric ray, crampfish, numbfish, torpedo
n01498041 stingray
n01514668 cock
n01514859 hen
n01518878 ostrich, Struthio camelus
n01530575 brambling, Fringilla montifringilla
n01531178 goldfinch, Carduelis carduelis
n01532829 house finch, linnet, Carpodacus mexicanus
n01534433 junco, snowbird
n01537544 indigo bunting, indigo finch, indigo bird, Passerina cyanea
n01558993 robin, American robin, Turdus migratorius
n01560419 bulbul
n01580077 jay
n01582220 magpie
n01592084 chickadee
n01601694 water ouzel, dipper
n01608432 kite
n01614925 bald eagle, American eagle, Haliaeetus leucocephalus
n01616318 vulture
......
```

bvlc_googlenet.prototxt

```
name: "GoogleNet"
input: "data"
input_dim: 1
input_dim: 3
input_dim: 224
input_dim: 224

layer {
  name: "conv1/7x7_s2"
  type: "Convolution"
  bottom: "data"
  top: "conv1/7x7_s2"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
......
```

Implementation

- Load the label file

```
> import cv2
> import numpy as np
> import matplotlib.pyplot as plt

> lbl_file = 'synset_words.txt'
> labels   = open(lbl_file).read().strip().split("\n")
> classes  = [r[r.find(" ") + 1:].split(",")[0]
              for r in labels]
```

- Load the net model

```
> prototxt  = 'bvlc_googlenet.prototxt'
> caffemodel= 'bvlc_googlenet.caffemodel'
> net       = cv2.dnn.readNetFromCaffe(prototxt,
                                       caffemodel)
```

Implementation

- Read image and create blob

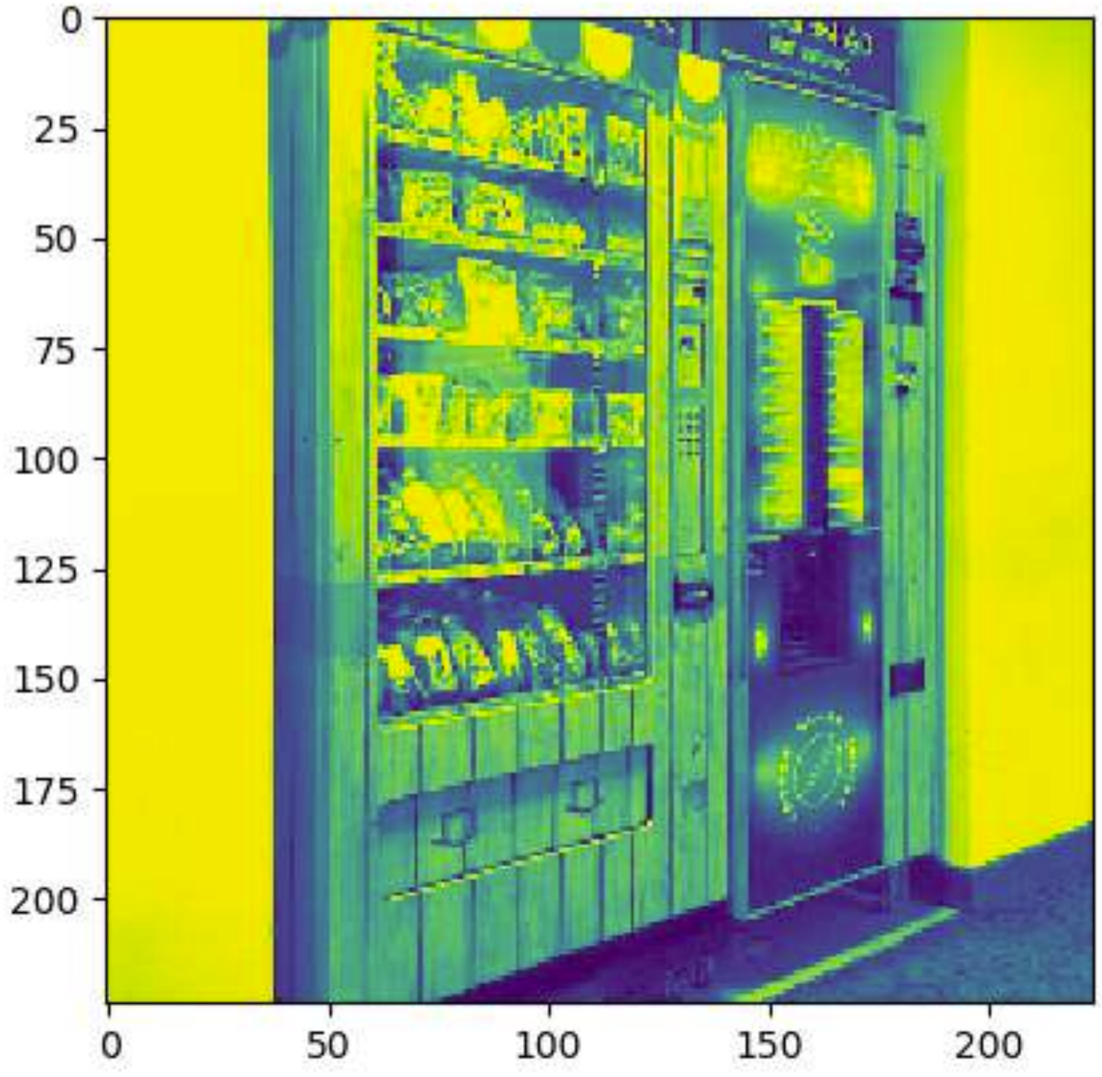
```
> img   = cv2.imread('vm.jpg')
> blob  = cv2.dnn.blobFromImage(image=img,
         scaling for image values (not image size)  scalefactor=1,
                                       output size  size=(224, 224),
           mean value to be subtracted from each channel  mean=(104, 117, 123),
                             swap R and B channel  swapRB=True,
                    Do cropping after resizing image  crop=True)
```

- Let's take a look at blob

```
> blob.shape
: (1, 3, 224, 224)

> gr  = blob[0,1,:,:]
> plt.imshow(gr)
```

Image and blob



Implementation

- Set input and run the model

```
> net.setInput(blob)
> preds  = net.forward()
```

- Check the output

```
> preds.shape
: (1, 1000)
```

- Reduce the output into 1D array (`preds[0]`) and sort

This sort gives the index, from lowest to highest

Flip the array, so the indices represent elements from highest to lowest

```
> pr_idx  = np.argsort(preds[0])
> pr_idx  = pr_idx[::-1]
```

- Prepare the caption

```
> imgtxt = classes[pr_idx[0]] + " ({:.2f}%)".format(preds[0,pr_idx[0]] * 100)
```

Output

- Put the text in the original image

```
> cv2.putText(img,
              imgtxt,
              (3, 25),
              cv2.FONT_HERSHEY_SIMPLEX,
              0.7,
              (0, 255, 0),
              2,
              cv2.LINE_AA)
```