Hands-on Session

An Interactive Introduction to Pattern Recognition

Recognizing human actions

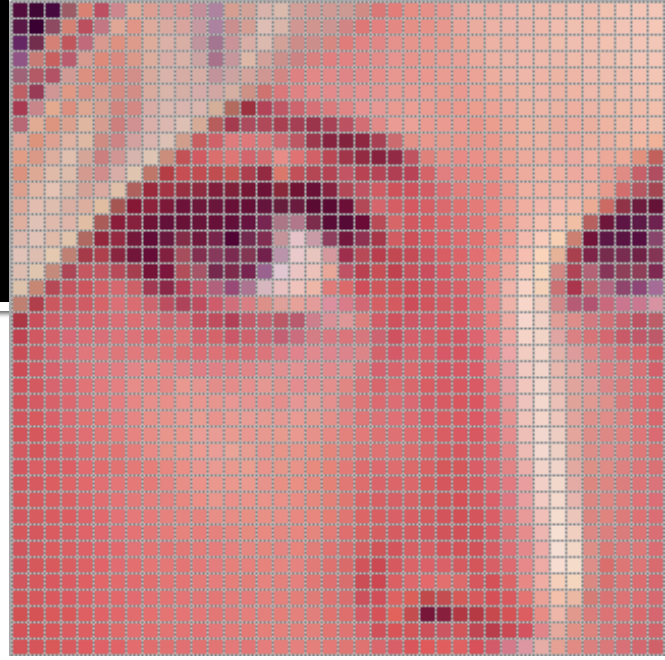
- In this hands-on session, you will explore different features and classifiers to recognize human actions:
  - Due to space constraints, perform actions while seated, eg. waving hands, boxing, nodding your head.

(a) bend

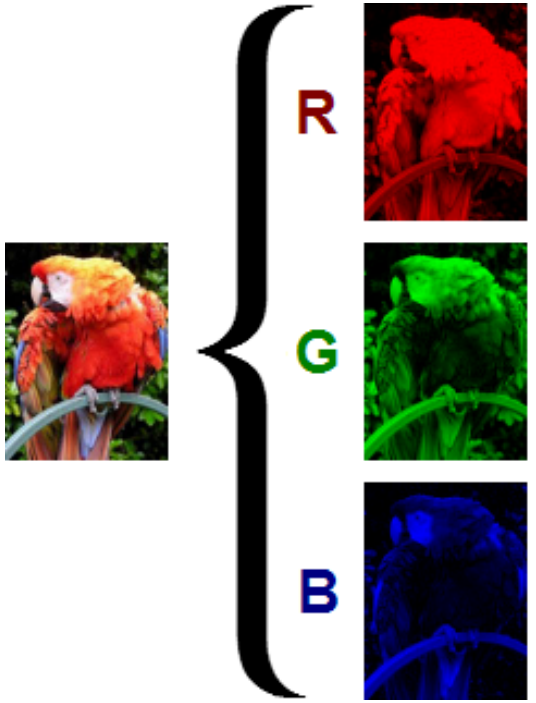
(b) jack

(c) jump

Image processing basics

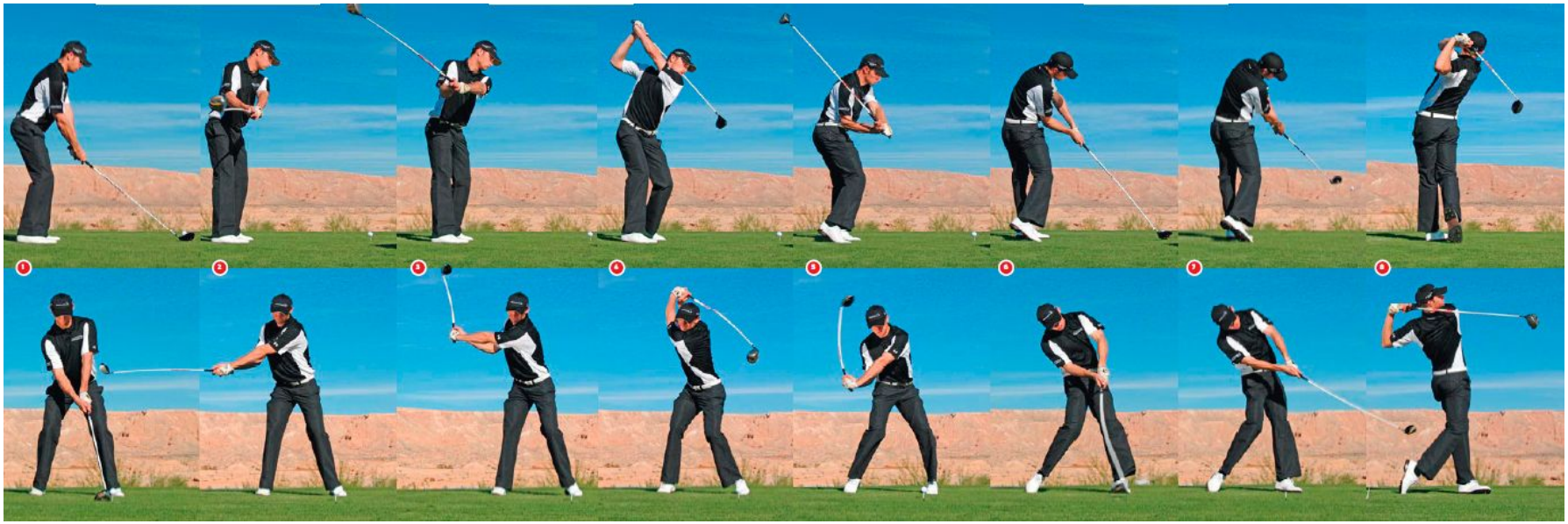


- A digital image is made up of dots, called pixels (picture elements).
- Each pixel has 3 channels: red, green, blue, that blend together to produce a color.



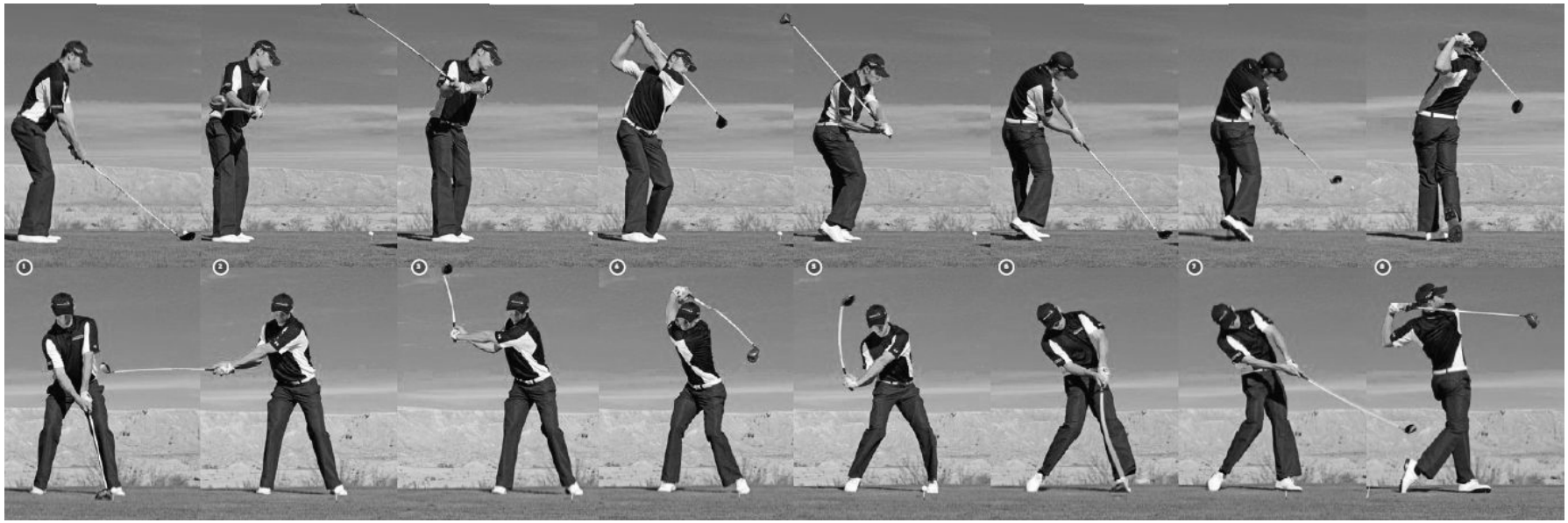
Video

- A video is a sequence of still images (frames).
- When played back at 15 frames per second (fps) or faster, we perceive motion.

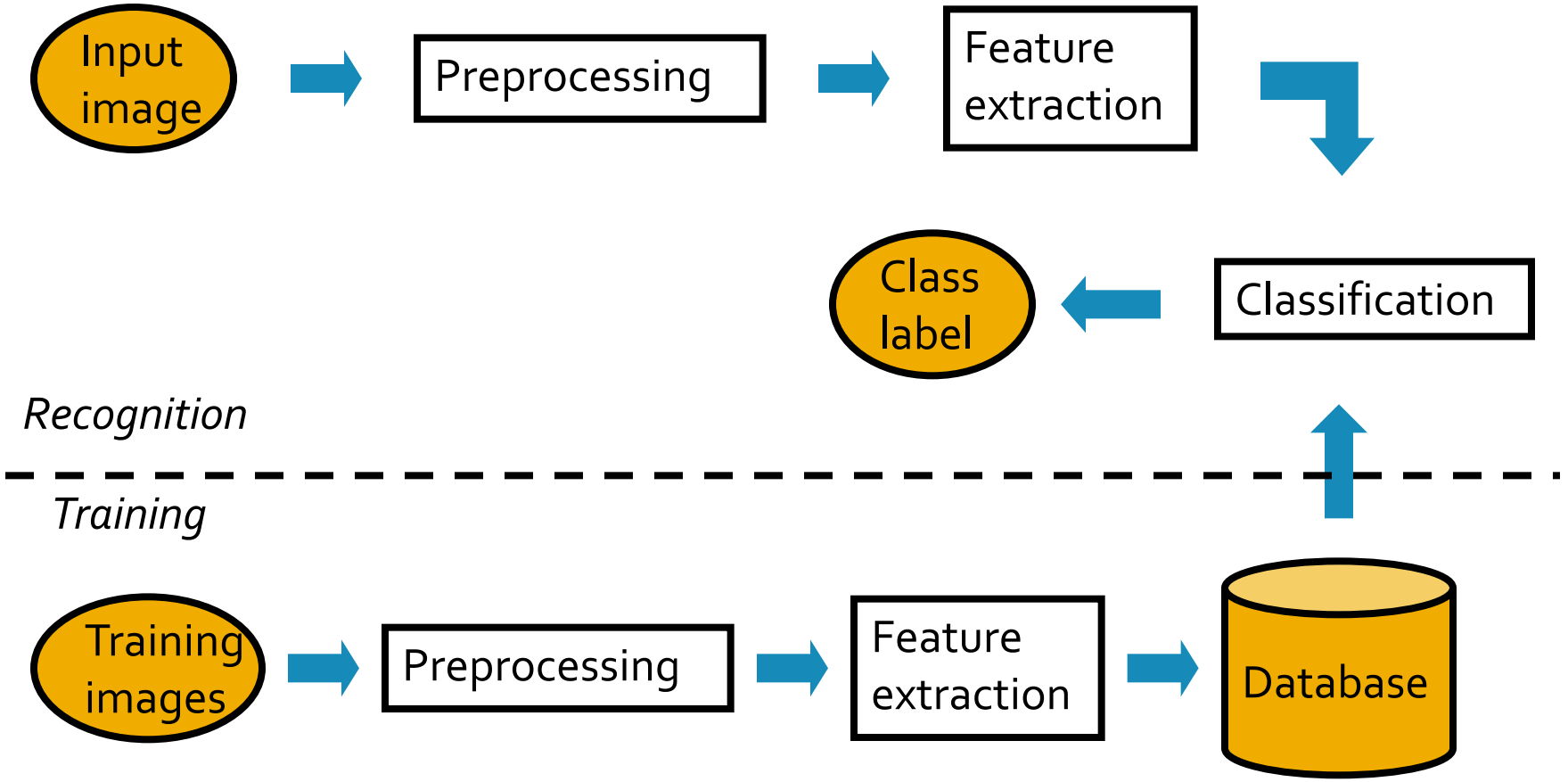


Color not needed

- For many image processing tasks, color is not useful. Grayscale is sufficient.
  - Color only triples the amount of memory needed.

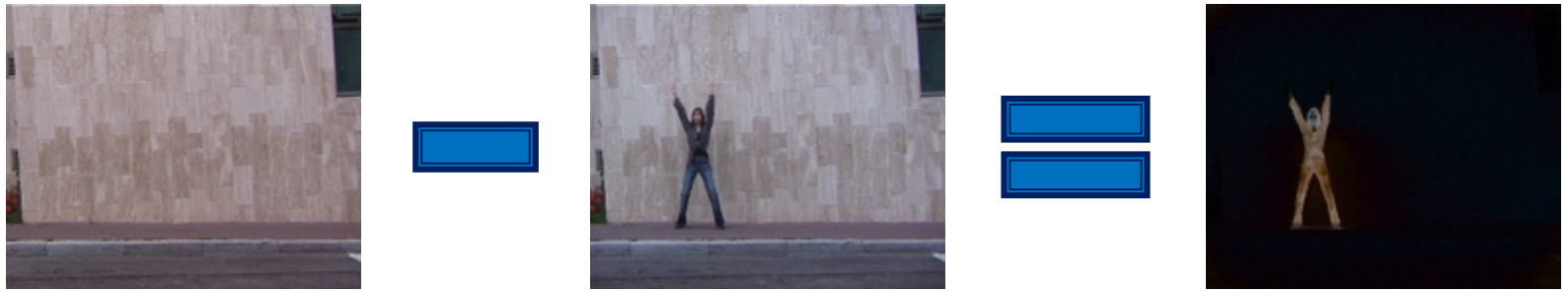


Typical pattern recognition pipeline



Pre-processing

- Detecting foreground vs. background
  - Assume camera is fixed



- Make sure you record the background before performing your action.
- Features are only computed from the foreground object, not the background.

Feature 1: aspect ratio

- From the foreground image we can determine the bounding box.
- Let its width and height be *w* and *h*, respectively.
- Define aspect ratio: $\alpha = w/h$

- We compute the aspect ratios of 10 frames and put these values into a feature vector:
- Question: how will the aspect ratio change when you move your arm?

$$x = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{10} \end{bmatrix}$$

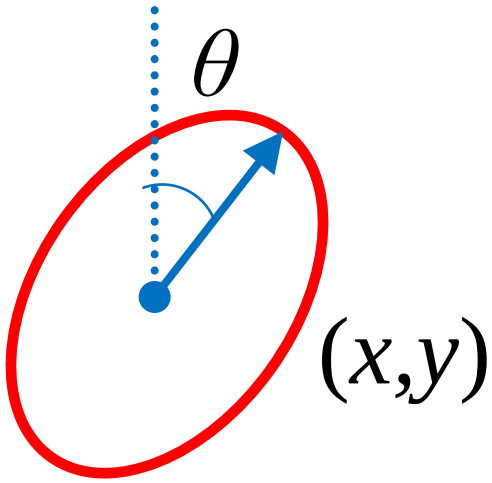
Feature 2: trajectory

- From each foreground image, we can fit an ellipse on it.
- We then compute the coords of the centre, and the angle the ellipse makes with the vertical.

- Our feature vector is then these 3 values over 10 frames:

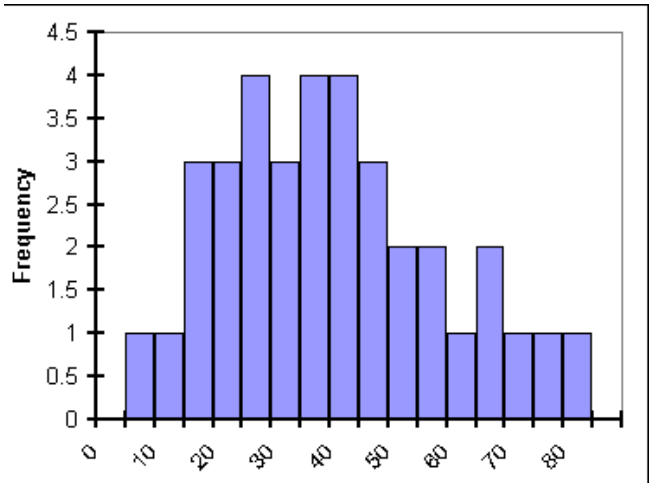
$$x = [\ x_1, y_1, \theta_1, x_2, y_2, \theta_2, \ldots, \theta_{10}\ ]$$

- This feature captures how the rough shape is changing.



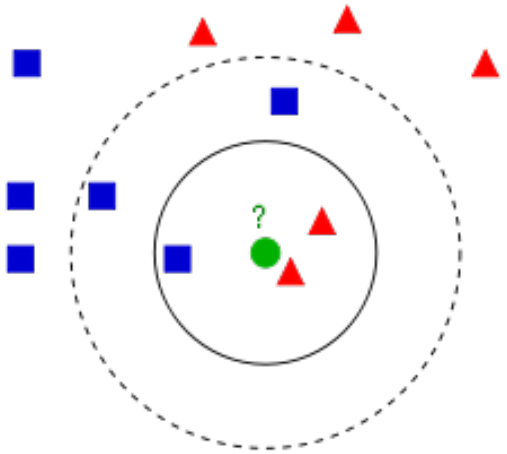
Feature 3: Histogram of oriented gradients (HOG)

- Foreground frames: $f_1, f_2, \ldots, f_{10}$
- Calculate average frame: $m = (f_1 + f_2 + \cdots + f_{10})/10$
- At each pixel in *m*, compute derivatives: $\frac{\partial m}{\partial x}, \frac{\partial m}{\partial y}$
- Compute the histogram of these derivatives:
- This feature captures the speed and direction of change in different parts of the image.



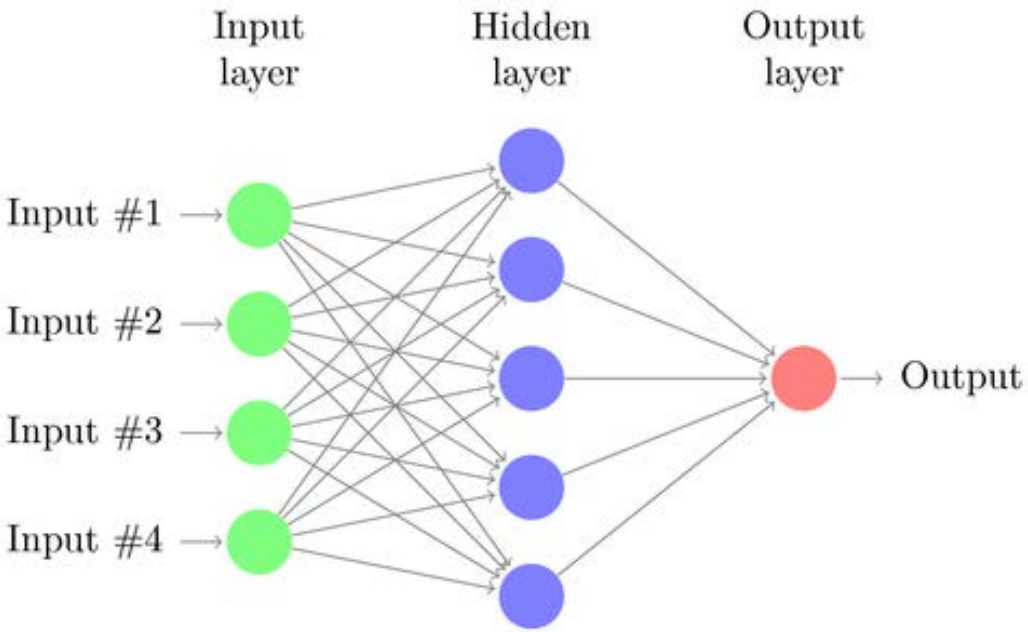
Classifier: nearest neighbor

- Blue squares and red triangles are training data.
- To classify a new point (green),
  - Determine the 3 nearest neighbors of the green point.
  - Green is assigned the label of the majority vote (in this case, red)
- Your program uses 5-nearest neighbor



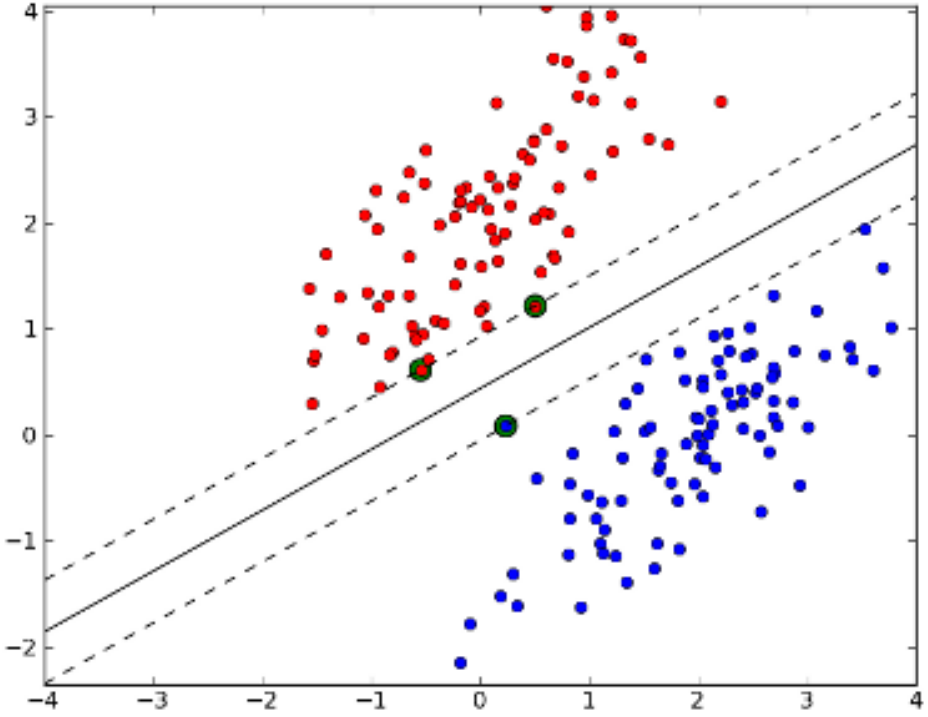
Classifier: Neural network

- Imitates the neurons in a brain.
- Inputs are the pixel values
- Output is the label of the class



Classifier: Support vector machine (SVM)

- Determines a linear boundary with the largest margin of separation between the classes



Your job

- Use the program to recognize different actions that you perform.
- Try different features and classifiers.
- You will need to record the (static) background, as well as the videos for testing and training.
- Answer the following questions. Then come up to present your answers.

Questions

- Which feature-classifier combination is most accurate at recognizing your actions?
  - Explain how you would determine this combination.
- Think of two actions between which the aspect ratio feature cannot distinguish.
  - Perform these actions, and verify that they cannot be distinguished. (Use a fixed classifier)
  - If, instead, you use HOG, can these actions be distinguished?
- Will your system work in outdoor lighting?
  - What must be changed, or what extra steps are needed, to make it work outdoors?