Image classification

Raúl Alonso Calvo

Required steps

- Image Preprocessing
- Detection of ROI or objects
- Feature definition and extraction
- Training and Classification

Training and Classification

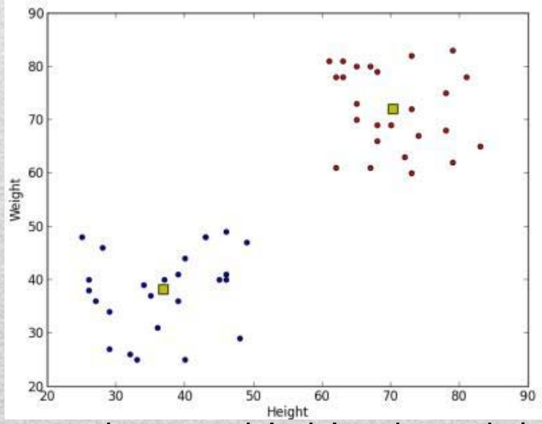
- You will find that classification on images could be performed using
  - Statistical methods such as:
    - Support Vector Machine
    - Decision Tree
    - Clustering
  - Machine learning methods
    - Artificial Neural Networks

Clustering

- Probably you have learned about these algorithms in other courses
- The important point is adapting the data
  - Image -> to numbers
- We will perform a Color Quantization
  - reducing number of colors in an image

Clustering

- OpenCV provides K-means algorithm
- Data in RGB -> each data value (pixel) will be composed of three features:
- Usual example of clustering algorithm with two features:



https://docs.opencv.org/master/d1/d5c/tutorial_py_kmeans_opencv.html

Practice 1

- Clustering example proposed in moodle:
  - https://moodle.upm.es/titulaciones/oficiales/mod/page/view.php?id=1326191
- Can we use a different color-space
- Could we adapt the code for using in other color-space ?

Neural Networks

- Trending topic for classification and prediction nowadays
- Special data need adaptations of classical NN
- Convolutional Neural Networks
  - Adds pre-processing layers for feature extraction
  - Final features are the input to a NN

CNN basics

- Image segmentation aims to obtain regions of interest (ROIs) from an input image

- Regions obtained in segmentation aid to represent the image as a sum of pixel sets

- These sets represent objects or parts of objects

- Each region is a set of pixels
  - we could say that image segmentation order pixels into larger components

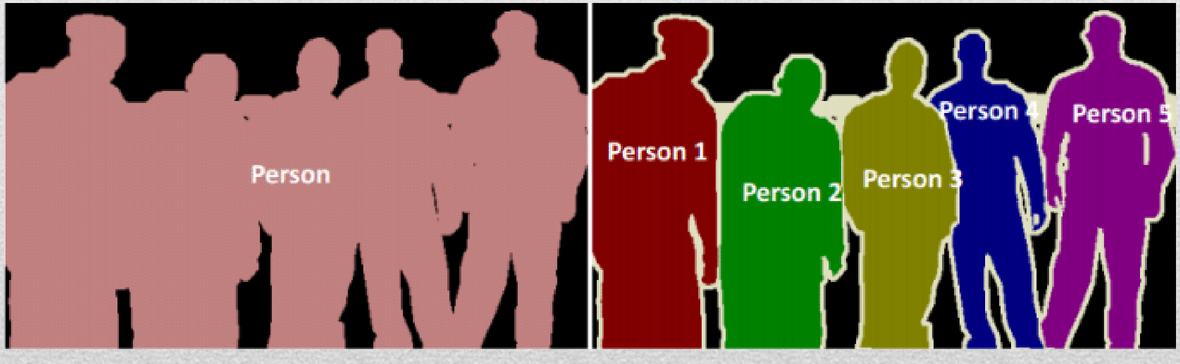
CNN basics

- Current trends in image analysis are:

  - Classification – categorizing the image into a class such as “people”, “animals”
  - Object detection – detecting objects within an image and drawing a rectangle around them
  - Segmentation – identifying parts of the image and understanding what object they belong to

CNN basics

- Inside the image segmentation research are, we can even distinguish:
  - Semantic segmentation which classifies pixels of an image into meaningful classes, and
  - Instance segmentation which identifies the class of each object in the image.



Varatharasan, Vinorth & Shin, Hyo-Sang & Tsourdos, Antonios & Colosimo, Nick. (2020). Improving Learning Effectiveness For Object Detection and Classification in Cluttered Backgrounds.

CNN basics

- Is a deep learning algorithm whose structure imitates the visual cortex of the human eye to distinguish different characteristics from input image

- The goal of identifying objects appearing in the image

- These CNN are based in work performed by Hubel and Wiesel in the 1950s, studying cat and monkey visual cortexes and neurons

CNN basics

- CNNs can receive a source image (or a sequence)

- And assign biases and weights to the different types of objects appearing in the image

  To discriminate between such object categories

- Therefore, a CNN receives as input one tensor
  - with (number of images) x (image width) x (image height) x (image depth)

CNN basics

- The architecture of a CNN is composed of several hidden hierarchical layers
  - the first layers are able to identify simple shapes such as lines
  - the following layers specialize in recognizing objects in increasing order of complexity

- As any other neural network based-method, the specialization is achieved by training the CNN
  - often using hundreds or thousands of object instances for each class (type of object)
  - regardless the object’s colour or perspective

CNN basics

- Notes
  - In order to enhance CNN performance usually original dataset should contain a lot of examples
    - Data augmentation
  - Usually CNN cannot process the input image as a whole
    - CNN implementations define tensor input
    - Sliding window techniques

CNN basics

- If we plan to work with images of 28×28 pixels we would need a NN with 784 input neurons for each band
  - For three 3 channels (red, green, blue) will need 28x28x3 = 2352

- Also we would usually need to normalize values
  - "value/255" for obtaining values in range 0 to 1

CNN basics

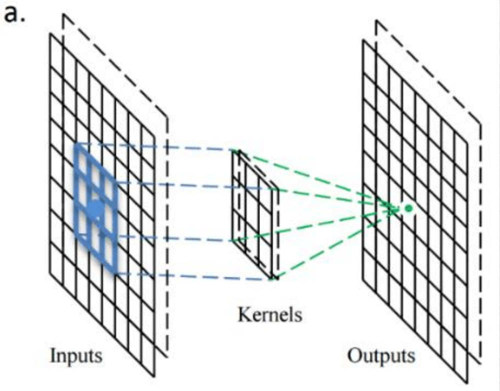
- The distinctive characteristic from CNN is the convolution

- A kernel is applied to input image
  - A matrix operator
  - Involves surrounding pixels
  - Generates an output image

CNN basics

- The distinctive characteristic from CNN is the convolution

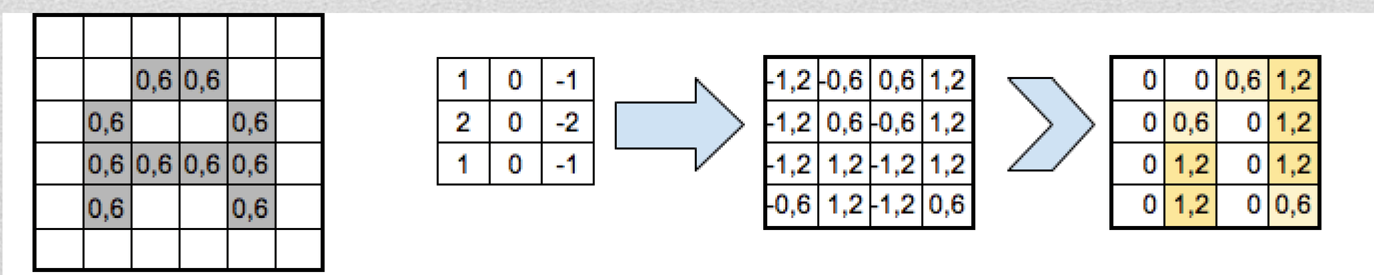
- A kernel is applied to input image
  - A matrix operator
  - Involves surrounding pixels
  - Generates an output image



A stride one 3x3 convolutional kernel acting on a 8x8 input image, outputting an 8x8 filter/channel.
Source: https://www.researchgate.net/figure/a-Illustration-of-the-operation-principle-of-the-convolution-kernel-convolutional-layer_fig2_309487032

CNN basics

- Each kernel is defined to extract a characteristc
- When multiple convolutional kernels are applied within a convolutional layer
  - many channels/feature maps are created
  - one from each convolutional kernel
  - An activation function (usually ReLU) is used for each feature map

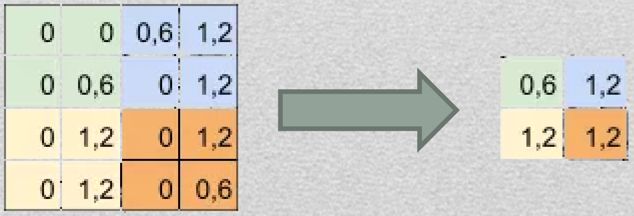


CNN basics

- Then we can 'reduce' the feature maps using a Pooling Layer

- It is used to generalize: is a destructive which aims to reduce overfitting

- A function for summarizing each feature map
  - Average pooling
  - Min pooling
  - Max pooling:



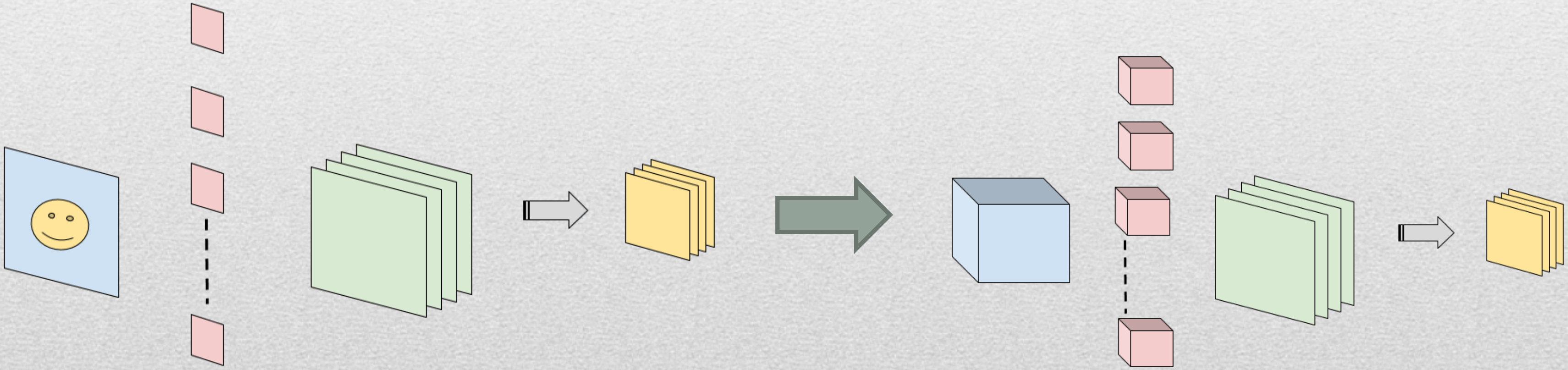
CNN basics

- Architecture:
- A set of:
  - Two or more of convolutional layers
  - A pooling layer

- Finally a fully connected layer is used at the output
  - May be stacked one, two or more deep

CNN basics

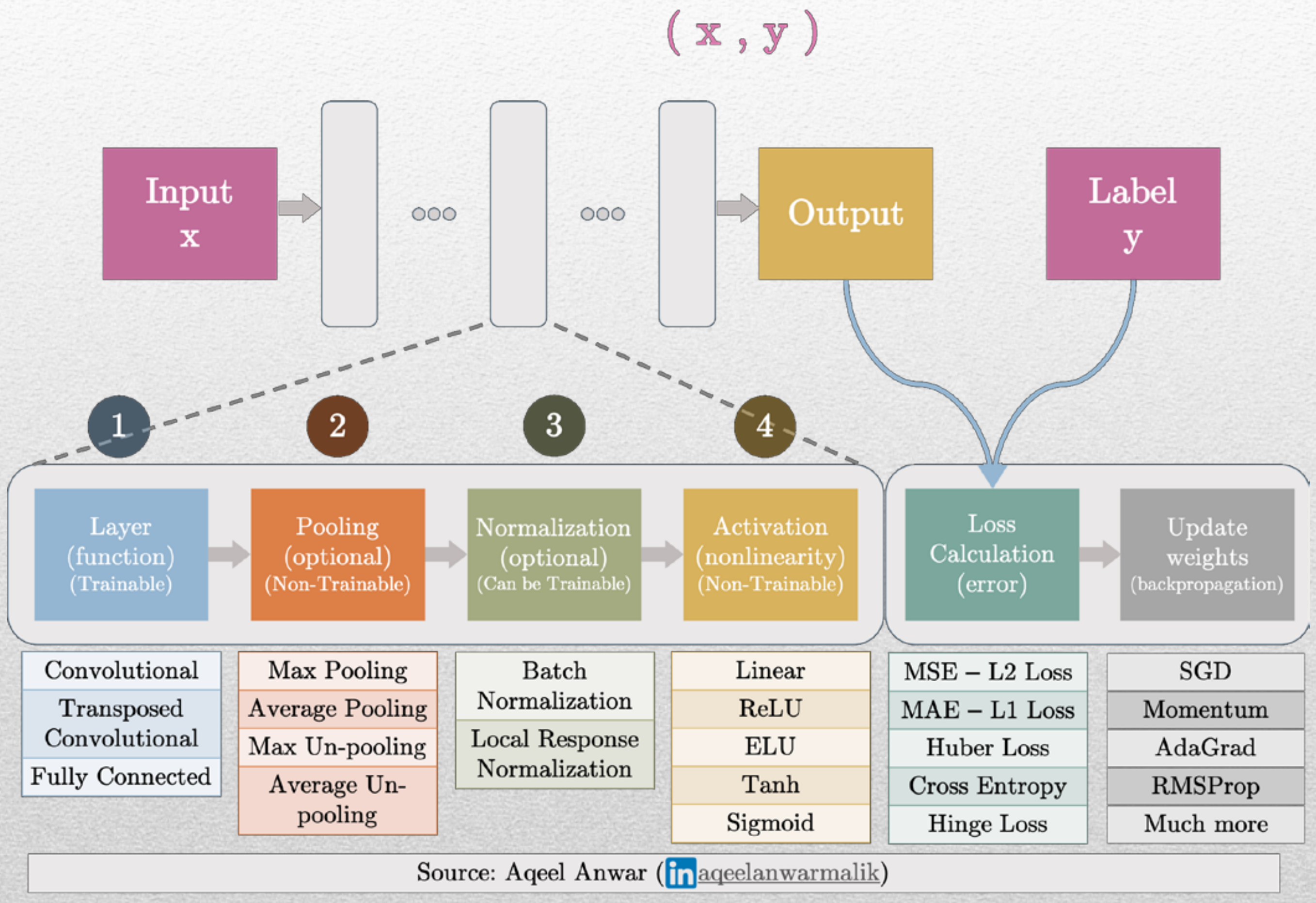
- Convolutional Layer:
  - Input from the layer
  - Apply kernel to obtain feature map
  - Apply pooling to reduce dimensions



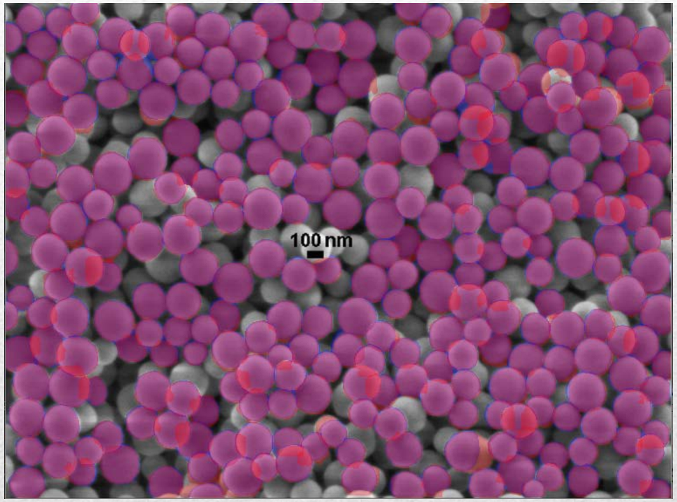
CNN basics

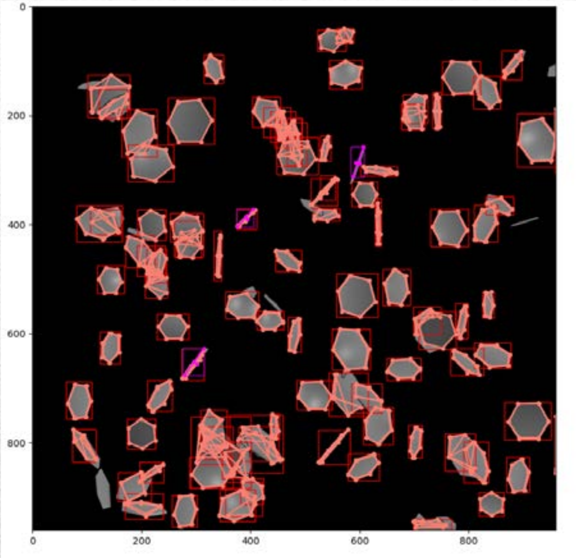
- Fully connected Layer
  - we can flatten out the square feature maps into a traditional flat fully connected layer
  - I.e.: For 10 features, downsized to 14x1
    - we can define the fully connected layer with 200 hidden neurons each with input connections
    - 1960 + 1 weights per neuron
    - that is a total of 392,200 connections and weights to learn in this layer
  - We can use a sigmoid or softmax transfer function to output probabilities of class values directly.

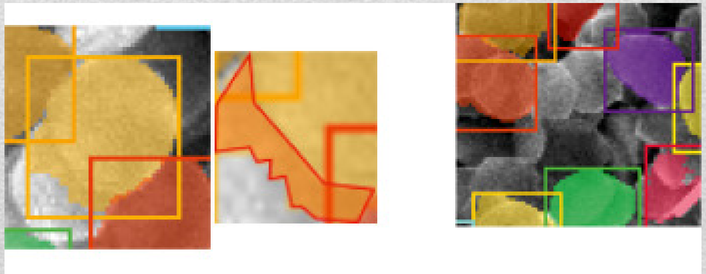
CNN


(x , y)
Input
x
Output
Label
y
1
2
3
4
Layer
(function)
(Trainable)
Pooling
(optional)
(Non-Trainable)
Normalization
(optional)
(Can be Trainable)
Activation
(nonlinearity)
(Non-Trainable)
Loss
Calculation
(error)
Update
weights
(backpropagation)
Convolutional
Transposed Convolutional
Fully Connected
Max Pooling
Average Pooling
Max Un-pooling
Average Un-pooling
Batch Normalization
Local Response Normalization
Linear
ReLU
ELU
Tanh
Sigmoid
MSE – L2 Loss
MAE – L1 Loss
Huber Loss
Cross Entropy
Hinge Loss
SGD
Momentum
AdaGrad
RMSProp
Much more
Source: Aqeel Anwar (in aqeelanwarmalik)

CNN







(a) (b)

Practice 2

- Semantic segmentation exercise to upload to moodle

- Test Yolo3, guide available in moodle

- Explore Yolov5 !
    - https://github.com/ultralytics/yolov5
    - https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

Other methods

- https://docs.opencv.org/master/d1/d73/tutorial_introduction_to_svm.html

- https://www.geeksforgeeks.org/license-plate-recognition-with-opencv-and-tesseract-ocr/

Other links

- https://towardsdatascience.com/visual-interpretability-for-convolutional-neural-networks-2453856210ce
- https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet/
- https://towardsdatascience.com/visual-interpretability-for-convolutional-neural-networks-2453856210ceinc.com/es/es/blog/machine-learning-support-vector-machines