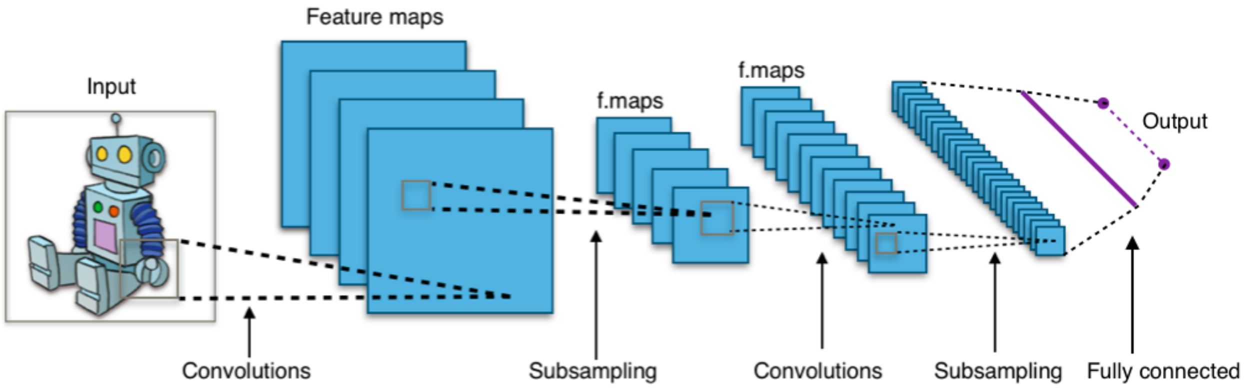
Practice 4 – Deep Learning

Pattern Recognition

In this practice you will use Deep Learning for image classification. In concrete, you are going to work with Convolutional Neural Networks (CNN), a kind of deep learning algorithm that takes advantage of the structure of the images to drastically reduce the number of needed parameters.



To understand well CNNs, their structure, and how do they work, you can visit the webpage http://cs231n.github.io/convolutional-networks/.

Work to do

This lab is divided in two parts. In the first part you will work with an already trained CNN called Alexnet and use it to classify the images from the CKBD dataset. In the second part you will have to build your own CNN and train it with your data. You have to deliver your code and a report containing all what it is asked below, reflecting that all of the concepts are well understood.

Part1: Feature extraction with Alexnet

To do this part, we will follow the matlab example DeepLearningImageClassificationExample.m, that you can find explined in https://es.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html.

Using this example as a guide, do the following:

- Load the pre-trained Alexnet Network and take a look at its structure using the command *net.Layers*. Ennumerate the different kind of layers and explain them (you might need to search on the Internet).

- Plot the weights of the first convolutional layer as it is done in the example and explain intuitively which kind of features do you think that the network is extracting.

-As you can see in the parts *Extract Training Features Using CNN* and *Train A Multiclass SVM Classifier Using CNN Features*, CNNs can be used to extract features from your images and then use this features to train a classifier such as a SVM. Do this with the CKBD dataset and report the

accuracy obtained. Remember that you have to divide the dataset in a test set and a training set, use the first one to train your SVM, and the second one to validate it (compute the accuracy).
Help: The Alexnet needs the data in a specific format (shape [227,227] and RGB). So, you have to modify your dataset using the following two commands:

imagesData = cat(3,imagesData,imagesData,imagesData);
imagesData=imresize(imagesData,[227,227]);

Part 2: CNN classifier

In this part, you will build your own CNN to classify the data from the CKBD dataset. To do this, use as a guide the Matlab example shown in https://es.mathworks.com/help/nnet/examples/create-simple-deep-learning-network-for-classification.html and:

- Try different configurations of layers and analyse the results, **try to get an accuracy as high as possible!** You can choose how many convolutional layers do you want to use and with which parameters and activation functions, which other layers will you apply... There is not a correct way to do that, but you should understand well what each layer is doing in order to have some intuition about what could be useful.

- Explain in detail your final CNN, their layers, and how and why have you arrived to that configuration.

- Report the accuracy of your model and compare it with the one obtained in part 1. This accuracy will be

-Plot the different filters of the first convolutional layer as you have done in part 1 and comment the results.