

Assignment 1

Binary classification

Goal: build a CNN model for a binary image classification problem. The model has data augmentation as well. You will be able to build two models, one with two layers and the second one with three layers. Moreover, you will study the difference in performance between a CNN with two layers and one with three layers.

The project must be conducted in **alone**.

Deadline: TODO

Tool: Tensorflow / keras, sklearn, matplotlib, sklearn. Look at the instruction folder "Getting started" to install all the necessary packages and the environment.

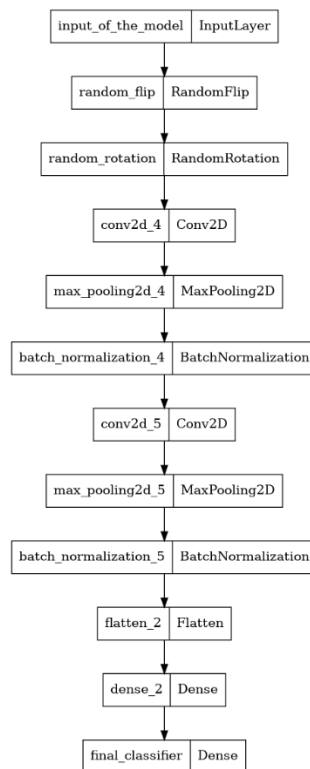
Dataset: The dataset is composed of 983 images of pizza and 983 images of all other possible food. You have three options here:

- You can download the dataset from Kaggle: [Pizza or Not Pizza? \(kaggle.com\)](https://www.kaggle.com/pizzanotpizza) an account is required.
- Download directly the dataset from the seafile: <https://seafile.utu.fi/f/43aec96c12b84020a073/?dl=1>
- As alternative and easier way, we prepared the same dataset ready to use in Moodle: "pizze_not_pizza.zip".

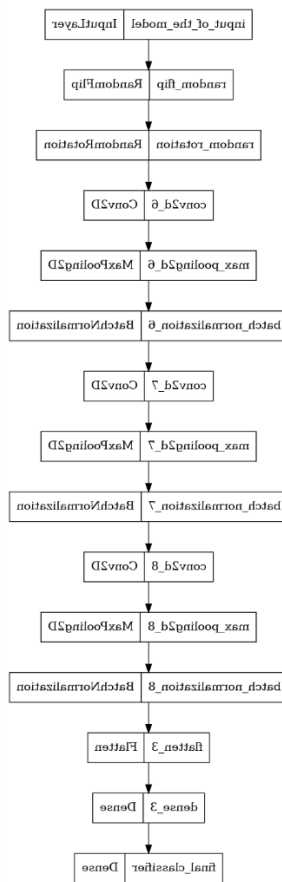
Tasks: There is a Jupyter notebook template "Assignment_1_template.ipynb" with instructions that you need to follow in order to tackle this lab. In summary you have to:

- 1- Prepare the data, use the function provided to load the data, as target size of the images use (224, 224, 3). Normalize the data and use the train test split function from sklearn library to split the data. Use test split size of 0.3.
- 2- Build a base CNN with the following characteristics:
 - a. Input layer
 - b. Data augmentation, with random flip and random rotation.
 - c. Two hidden layers each composed with the following characteristics: 16 conv2d units, max pooling 2d and batch normalization, the second one should have 24 conv2d units max pooling 2d and batch normalization.
 - d. After this, add a flatten layer and a dense layer with 8 units
 - e. Add the final classifier (a dense layer) with the correct number of output and activation
 - f. Compile the model with Adam optimizer and binary cross entropy as loss function.
 - g. Train the model with batch size 64 and epochs of 128.
 - h. Evaluate the model and report the accuracy.
 - i. Make prediction with the test set and use a threshold of 0.5 as boundaries decision between the classes.

- j. Plot some predicted images.
- k. Plot a confusion matrix or a roc curve or both 😊. Why you need these evaluations systems?



- 3- Build another base CNN, but at point c add an extra hidden layer with 32 units of conv2d. Repeat all the other steps. What happened to the accuracy of the model? Why?



Instruction for Submissions: You must convert your notebook to a PDF file and then submit the PDF file to Moodle. There are two ways for converting a notebook to PDF:

- Install PyPDF2 by running `pip install PyPDF2`
- you can manually convert the jupyter notebook to HTML (File -> Download as -> HTML (.html)), save the HTML page as a PDF

Important:

- Please make sure that the submitted notebook has been run and the cell outputs are visible.
- The description of each task should be added into the notebook as a comment in order to make your code easier for understanding.

Assignment Evaluation: The notebook is evaluated on 0-10 points scale (0 means that the work is rejected).