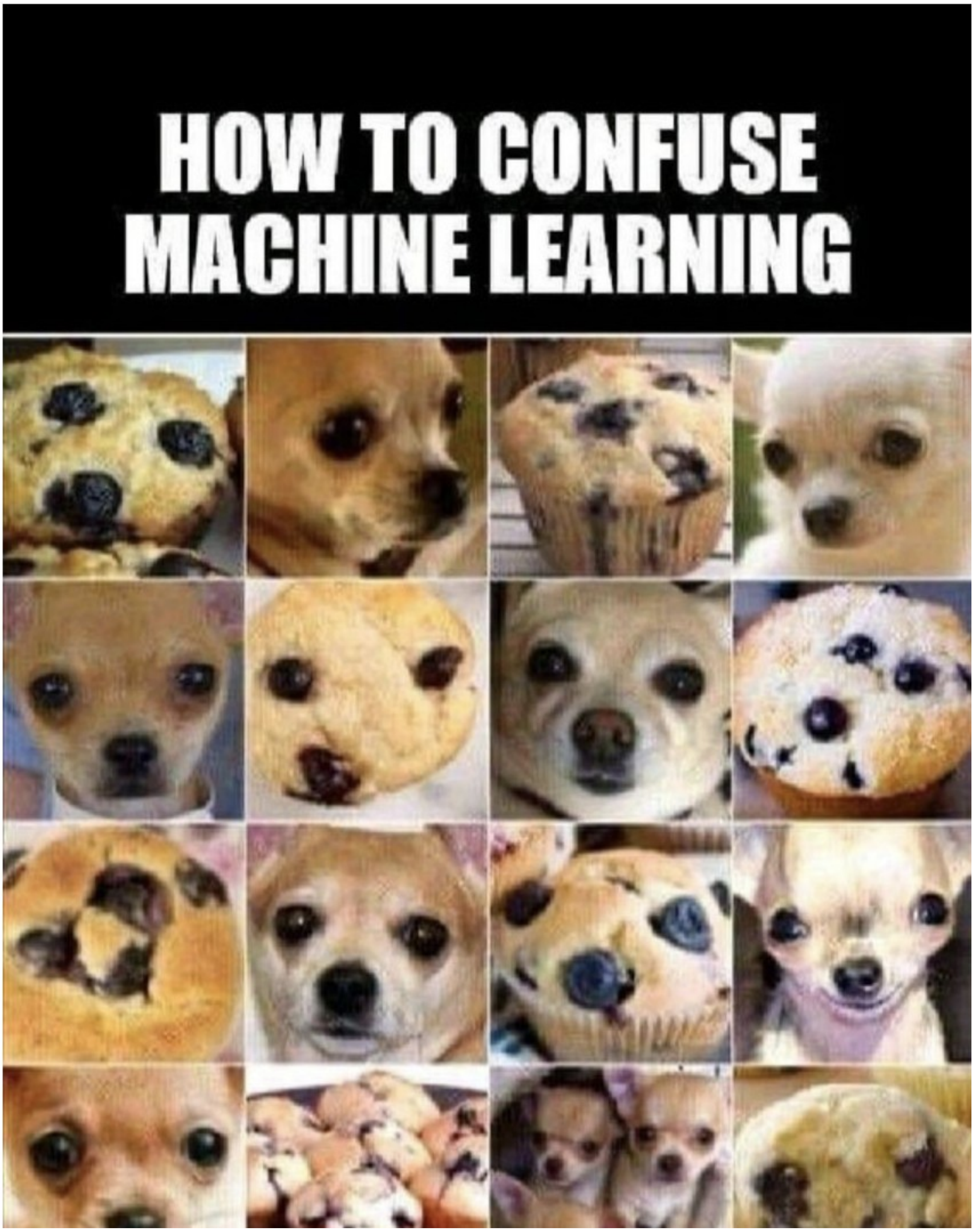
Assignment 1

Binary classification

This assignment aims to solve a practical problem in Deep Learning using 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: 03.02.2025

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

Dataset: The dataset is composed by two folders. The train folder contains 2559 chihuahua and 2174 muffins. The test set contains 640 chihuahua and 544 muffins.

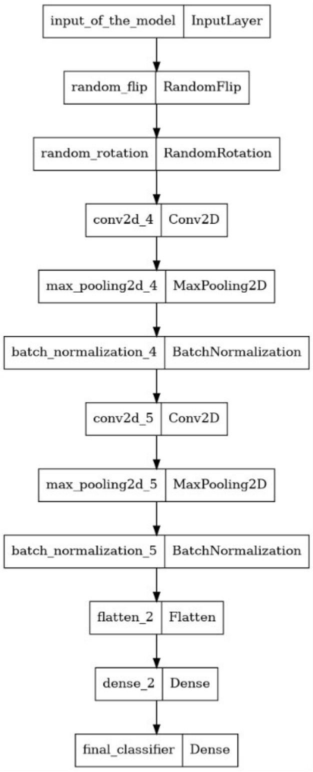
- You can download the dataset from Kaggle how to confuse a ML model (kagge.com) an account is required.
- Download directly the dataset from the seafile: https://seafile.utu.fi/f/43d6fc9295464e68916b/?dl=1
- As alternative and easier way, we prepared the same dataset ready to use in Moodle: "cookies_vs_chihuahua.zip".

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 30.
   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 and roc curve. Why you need these evaluations systems?


input_of_the_model | InputLayer
random_flip | RandomFlip
random_rotation | RandomRotation
conv2d_4 | Conv2D
max_pooling2d_4 | MaxPooling2D
batch_normalization_4 | BatchNormalization
conv2d_5 | Conv2D
max_pooling2d_5 | MaxPooling2D
batch_normalization_5 | BatchNormalization
flatten_2 | Flatten
dense_2 | Dense
final_classifier | Dense

3- Now at this point you will see that in the we have hardcoded the prediction threshold at 0.5. However, we do not know if this 0.5 is a good threshold. Try to find a way to calculate the best possible threshold. Hints:
   a. You can get the list of all possible threshold from the roc_curve functions.
   b. You can extract the best possible threshold index using this formula:
      i. Np.argmax(tpr - fpr)
   c. Make proper calculations and re-plot the confusion matrix.
   d. What happen?

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 describtion 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).