# Homework 05

Binary Classification Metrics

### Overview

In this homework, we are trying to measure the performance of a binary classification model.

### Description

We want to use transfer learning on a pre-trained **ResNet50** CNN model trained on the ImageNet dataset [1] (this dataset spans 1,000 object classes and contains 1,281,167 training images) to perform classification for recognizing images of horses and camels using the dataset provided in [2]. More specifically, fine-tuning refers to the practice of "unfreezing" a few of the top layers of a frozen model base and jointly training both the newly added classifier layers and the last layers of the base model. This allows us to "fine-tune" the higher-order feature representations in the base model to make them more relevant for the specific task that we retrain our model to perform. Note: the link of the notebook about training a CNN from scratch using Keras can be found in the Reading sheet (R6).

### Tasks

- Part A (Submission deadline by the end of the class):
    - Import a ResNet50 model (10 points)
    - Make sure the model is initialized with the ImageNet Weights (10 points)
- Part B (Submission by coming Monday 11:59 PM):
    - Import the Horses vs Camels Dataset to fine tune your model (10 points)
        - Make sure you use a training, validation and test set
        - For the validation set isolate a subset of 20 images each of the classes of the original training partition
    - Make sure that the classifier layer of the model is adapted to support binary classification (use a dense layer with a single output value and a sigmoid activation function) (10 points)
    - Train your model with binary cross entropy loss function. Freeze all layers except of the classifier layer(s) (10 points)
    - Plot a training learning curve (loss-epochs) and discuss if your model is overfitted, under-fitted, or well-trained. (10 points)
    - Plot confusion matrix. (5 points)
    - Calculate and report Precision. (5 points)
    - Calculate and report Accuracy. (5 points)
    - Calculate and report Sensitivity. (5 points).
    - Calculate and report Specificity. (5 points)
    - Plot ROC of your trained model. (5 points)

- Re-initialize the model by loading the default image net weights and repeat **all** of the above steps from **Part A and Part B without freezing any model layer** and perform training and evaluation **for the full model** (10 points)
- Write a small paragraph to comment on the difference observed in the results produced by the two models (10 points)
- Provide answers to the fullowing questions (10 points)
  - What is a confusion matrix ?
  - What is accuracy and how is it measured?
  - What is Precision and how is it measured?
  - What is Sensitivity and how is it measured?
  - What is Specificity and how is it measured?
  - What is a ROC curve and how is it computed?
  - When is it best to use a softmax versus a sigmoid activation function in the last layer of a neural network?

## Submission Guidelines

1. Upload the notebook file and a pdf copy of in in Teams.

## References

1. "ImageNet Dataset", https://www.image-net.org/download.php
2. "Horses vs Camels Dataset", https://www.kaggle.com/akrsnv/horses-and-camels
3. "Transfer learning and fine-tuning with Tensorflow", https://www.tensorflow.org/tutorials/images/transfer_learning
4. "Transfer learning and fine-tuning with Keras", https://keras.io/guides/transfer_learning/