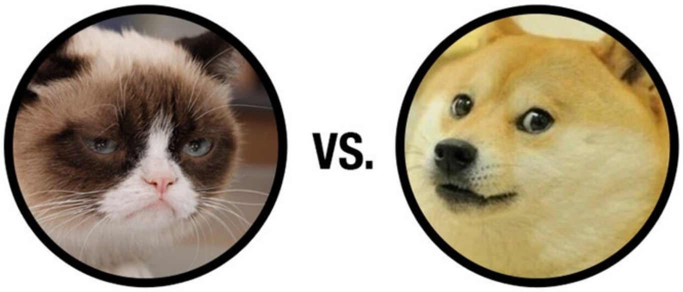
CAP 4401: Digital Image Processing

Assignment 5: Image classification

In this project you will design and implement a deep learning solution for image classification based on images of cats and dogs.



Goal: to build a deep learning solution in MATLAB that is capable of predicting whether an input color image containing an animal represents a *cat* or a *dog*.

Learning objectives:

- Learn how to implement a complete, fully functional, deep learning solution for a contemporary computer vision challenge using MATLAB.
- Get acquainted with representative contemporary datasets, challenges and problems in computer vision and machine learning.
- Learn how to use deep neural networks under the paradigm of *transfer learning*.
- Demonstrate the ability to perform model selection, fine-tuning, and performance evaluation of different solutions to the same problem.

Starter package

- **MATLAB starter code**: ***cats_dogs_starter.m*** (and associated data) (available on Canvas)
  - **Note**: This code is functional, (hopefully) correct, but old (ca. 2016) (and incomplete). Part of your job will be to refactor and expand the code and improve its documentation.

Instructions:

- **This is a group activity. Groups of 1-3 students are allowed.**
- Document all your findings, steps, conclusions, lessons learned, insights, etc. in your **report** *(think of it as a "lab notebook")*

Procedure:

1. Download the **MATLAB starter code** (and associated data) and add the files to your working folder, adjusting the MATLAB path if necessary.
2. Convert the .m file to a .mlx file and use the resulting Live Script as the starting point for the remaining steps.
3. Run the starter code and ensure that it works as intended.

4. Make sure you understand what each step / part is doing and write down questions you might have as well as ideas for improving/expanding the code and/or rewriting parts of it to reflect latest developments in MATLAB and associated best programming practices.

At this point, there are many things you might want to do. Try to tackle them one at a time, test your code, ensure that it works, and then move on to the next idea. Make sure you document the process extensively.

To help you out, these are some **things I expect you to do** (in no particular order):

- Train the network for longer time (i.e., **increase the number of epochs**)
- Train the classifier with a **larger number of images** (*)
- Try different (%) values for **partitioning the dataset into training and validation**
- Choose different **hyperparameters** (optimizer, learning rate, etc.) for network training
- Start from a **different pretrained model** (note that this might have implications elsewhere in the code, e.g., need to resize the images to a different target size)[1]
- Plot the **confusion matrix** for each variation of your model
- Try **different image data augmentation options** (and values/ranges)
- Add code to **inspect** (a subset of) the **incorrect classifications**

And these are some **things you are allowed (but not required) to do** as well:

- Change the problem to another (comparable and meaningful) 2-class image classification problem (e.g., classify a picture of food into "hot dog" or "not hot dog"[2])
- Compute and plot the ROC curve for each variation of your model
- Compute additional metrics (e.g., sensitivity, specificity) for your model
- Build and train your own neural network from scratch (instead of using transfer learning)
- Test the best solution using the test dataset from Kaggle (*)
- … (and much more… the sky is the limit)

(*) I suggest you consider downloading the **train.zip** data file from https://www.kaggle.com/c/dogs-vs-cats/data. You can disregard **test1.zip** (for now) and the **sampleSubmission** CSV file (forever).

[1] See https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

[2] With a nod to the TV series "Silicon Valley" ☺

What to submit

- Your **final Live Script** containing all relevant code, figures, plots, text, links, etc.
- (OPTIONAL) A separate **report** (PDF), if you prefer to keep the "story" of what you did, how, and what you learned from doing so in a separate file.
- (OPTIONAL) A README file, if appropriate.
- (OPTIONAL) Specific (few) test images, if appropriate.

Hint: You might want to summarize some of your steps (and their impact on results) using tables, such as the one below:

Model	Validation accuracy	Test accuracy
1. Transfer learning using pretrained CNN (AlexNet)		
2. Model 1 + data augmentation		
3. Transfer learning using pretrained CNN (VGG16)		

Deliverables
You should submit **a single zip file** via Canvas. One file per group is enough (make sure my graders and I can tell who is in the group).