CAP 4630 – Intro to AI – Dr. Oge Marques

Final Project: Cats vs. Dogs

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

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

You can choose whatever preprocessing image processing techniques, machine learning algorithms, and MATLAB toolboxes you wish, but keep in mind that the most important points are:

- To demonstrate the use of a sound **methodology** for every step of your work (e.g., feature selection, dataset selection, cross-validation techniques, performance measures, etc.).
- To **document** each step (and associated design decisions, criteria, thresholds, references, algorithms, strengths and weaknesses) of your work in a clear and readable manner.

The expected deliverables are:

- Self-contained MATLAB code to run your solution (and details on how to install and run it)
- A descriptive report containing the most important steps, building blocks, design decisions, examples of incorrect prediction results, and a ***summary table*** that should contain overall accuracy rates for, **at least, two different machine learning algorithms (and their variants, if applicable).**

Goals:

- Learn how to implement a complete, fully functional, machine 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 feature selection, model selection, fine-tuning, and comparison, and performance evaluation of different solutions to the same problem

Starter package

- **Dataset**: cats and dogs dataset from Kaggle (https://www.kaggle.com/c/dogs-vs-cats/data)
- **Reference links**:
    - https://www.kaggle.com/c/dogs-vs-cats (more details about the Kaggle challenge associated with the dataset)
    - (OPTIONAL) Dogs vs. Cats Redux: Kernels Edition (a playground offered by Kaggle, if you want to learn more about Kernels, Jupyter Notebook, and TensorFlow - https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition
    - https://www.mathworks.com/help/nnet/examples/transfer-learning-using-convolutional-neural-networks.html (transfer learning using MATLAB)

- **MATLAB starter code**: ***final_project_starter_cap4630.m*** (and associated data) (available on Canvas)

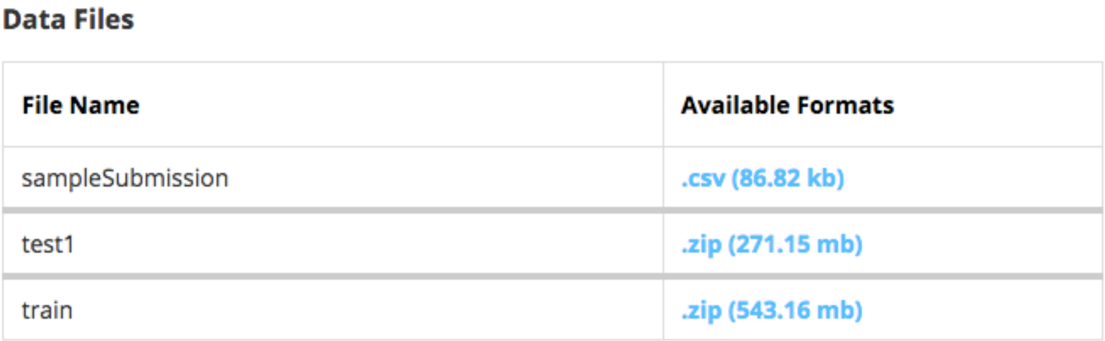
Instructions:

- This is a group activity
- Open collaboration among groups is **not** allowed

Procedure:

1. Download MATLAB starter code (and associated data) and add to your working folder, adjusting the MATLAB path if necessary.

2. Download the **train.zip** data file from the dataset link indicated above. You can disregard test1.zip (for now) and the sampleSubmission CSV file (forever).



Data Files

File Name	Available Formats
sampleSubmission	.csv (86.82 kb)
test1	.zip (271.15 mb)
train	.zip (543.16 mb)

3. Run "Part 1" of the starter code and ensure that it works as intended.
   Inspect the contents of variable **convnet**.

4. Run "Part 2" of the starter code and ensure that it works as intended.
   What type of preprocessing is performed by the auxiliary function *readAndPreprocessImage*?

5. Run "Part 3" of the starter code and ensure that it works as intended.
   What can you say about the montage with network weights for the second convolutional layer?

6. Run "Part 4" of the starter code and ensure that it works as intended.
   What other MATLAB function could have been used to build the SVM classifier in this case (binary classification problem)?

7. Run "Part 5" of the starter code and ensure that it works as intended.
   Did your classifier recognize 'Doge' as a dog? If not, can you tell why?

8. The starter code used a very small subset of the Kaggle dataset (20 images of dogs and 20 images of cats) out of the 25,000 images (2 x 12,500) available. This has to be changed.

In this step, you should partition your dataset into:

Training:	TR %
Cross-validation:	CV %
Testing:	TS %

Recommended values for TR, CV, and TS are 60, 20, and 20, respectively. If you use different values (or "play" with different choices), make sure to document that.

9. (OPTIONAL) Perform additional image preprocessing (e.g., sharpening, brightness/contrast/color adjustments, etc.) on all images from the dataset.

10. **Select** a (family of) classifier(s) to use **first**. Recommended techniques include (but are not limited to):

a. Logistic Regression
b. SVM (linear, quadratic, cubic, RBF Gaussian kernel, etc.)
c. K nearest neighbors (kNN)
d. Naïve Bayes
e. Decision Tree
f. Discriminant Analysis
g. Neural networks
h. Deep neural networks

11. Implement the entire workflow to test your solution and produce meaningful figures of merit and plots (see Warm-up Exercise).

12. (OPTIONAL) Tweak and fine-tune meaningful parameters associated with your first solution. Then freeze it and add meaningful results to the "summary table".

13. **Select** a different (family of) classifier(s) to use **second**.

14. Implement the entire workflow to test your solution and produce meaningful figures of merit and plots (see Warm-up Exercise).

15. (OPTIONAL) Tweak and fine-tune meaningful parameters associated with your second solution. Then freeze it and add meaningful results to the "summary table".

16. (OPTIONAL) Perform, test, and document any relevant changes or improvements (additional classifiers, different metrics, different thresholds, different partitioning of the dataset, etc.)

17. Produce a meaningful technical report. In addition to the summary table include, if applicable, other tables and plots, as well as representative samples of best and worst results.

Deliverables

You should submit **a single zip file** via Canvas.

It should contain:

- All necessary m-files to properly reproduce your results.
- All relevant third-party MATLAB scripts and functions that were ***not*** provided in the starter package (with an acknowledgment of their source), if any.
- A ***detailed*** report (PDF) containing your comments, remarks, examples, answers and comments associated with the Warm-up Exercise, relevant tables and plots, etc.
- A README file describing what is in the package, and (optionally) containing instructions on how to "install" (e.g., directory structure, dependencies, etc.) and run your code.

It should ***not*** contain:

- Any images or files that were already distributed as part of the starter package.
- The actual dataset.
- Older versions of your code or any leftovers that are not relevant for (and might even hurt) grading your work.

Grading rubric

Code: quality	10%
Code: correctness	30%
Code: inline documentation	5%
README file	5%
Report	50%