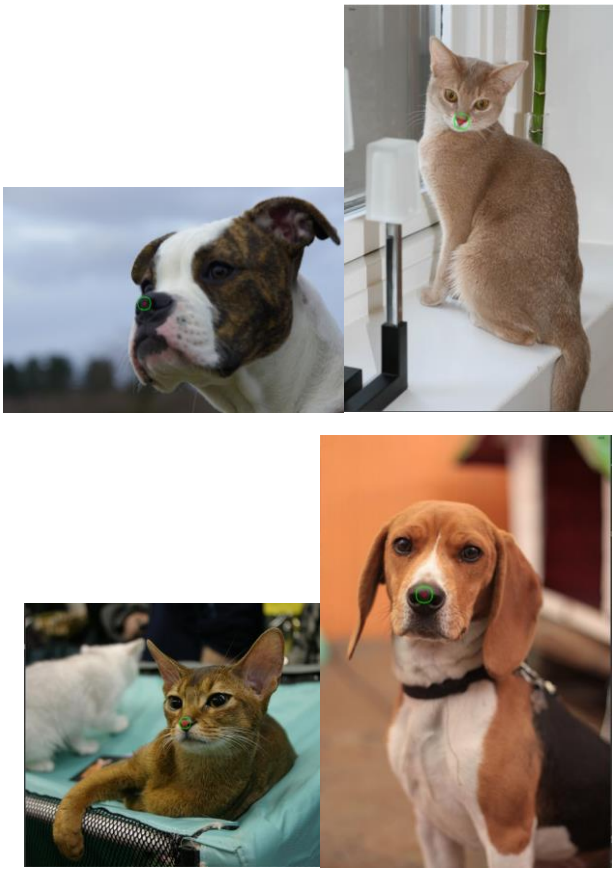

# ELEC 475

Lab 5 – Pet Nose Localization

Lab 5 – Pet Nose Localization

Michael Greenspan

# Contents

1. Introduction ..... 1
2. Task ..... 1
2.1 Step One (Prelab) ..... 1
2.2 Step Two – Model ..... 2
2.3 Step Three – Train ..... 2
2.4 Step Four – Test ..... 2
3. Deliverables ..... 2
3.1 Completed Code ..... 2
3.2 Report ..... 2
4. Submission ..... 3

# 1. Introduction

The objective of this lab is to implement a model that can localize pet noses in images. The dataset is ***oxford-iiit-pet-noses***, which is our reannotation of the ***oxford-iiit-pets*** dataset. You can reuse any existing code base in pursuit of this goal.

# 2. Task

Your task is as follows:

## 2.1 Step One (Prelab)

2.1.1 Download and install the select_pet_noses.py and show_pet_noses.py modules, from the Labs/Lab5/src directory of the course Google Drive.

2.1.2 Identify which dataset partition (0 - 166) you're responsible for, from the oxford-iiit-pet-noses.xcl spreadsheet on the course OnQ Contents->Lab Stuff directory.

2.1.3 Download your dataset partition from the course Google Drive data/oxford-iiit-pet-noses/images directory, to your local drive. (Each partition contains less than 50 images, so you won't require much storage).

2.1.4 Accurately label (i.e. run select_pet_noses.py) all images in your partition. Double check your labelling with show_pet_noses.py.

2.1.5 Once you're satisfied that your labels are accurate, copy the resulting 'labels.txt' file into the original Google Drive dataset partition directory that you were assigned, i.e. directly in your individual directory /0, /1, /2. ... /166. Do *not* put the labels.txt file in the root data/oxford-iiit-pet-noses/images directory.

### 2.2 Step Two – Model

Identify a model to regress the pet nose locations in the dataset. You can develop your own model, make use of an existing model, or modify an existing model.

### 2.3 Step Three – Train

Write a Dataloader for the dataset, and train your model, using the dataset train partition. You can optionally use the test partition for validation purposes (i.e. just to track the progress of the loss plot, and not as part of network parameter optimization).

### 2.4 Step Four – Test

Test your trained model, on the dataset test partition. Calculate the localization accuracy statistics, i.e. the minimum, mean, maximum, and standard deviation of the Euclidean distance from your estimated locations to the ground truth pet nose locations.

## 3. Deliverables

The deliverable as described below comprises the following:

- The completed code;
- A report.

### 3.1 Completed Code

In addition to the code itself, provide the following two scripts to train and test your code (e.g. from the PyCharm terminal):

**`train.txt`**

**`test.txt`**

### 3.2 Report

You report should be relatively brief (4-6 pages), and include the following information:

- Step 2: What model did you use? Why did you select this model?
  If you used a pre-existing model, then include all references (weblinks, papers, etc.) to the model, as well as a description of any modifications that you made.

If you created a new model, explain the inspiration and rationale for its design, and describe its structure in sufficient detail so that it can be reimplemented solely from the textual description.

- Step 3: Describe all hyperparameters used in training, and describe the hardware used. How long did the training take? Include the loss plot.
- Step 4: Describe the localization accuracy statistics. Did your method work well? Include some qualitative results (i.e. localized pet nose images) to support your claim. What hardware did you use to execute the inference tests, and what was the time performance (e.g. msec per image)?
- Discussion: Discuss the performance of your system. Was it as expected? What challenges did you experience, and how did you overcome them?

## 4. Submission

The submission should all of your source code, and the report described in Section 3.

All deliverables should be compressed into a single **`<zzz>.zip`** `file`, where filename **`<zzz>`** is replaced with your student number. If you are in a team of 2, then concatenate both student numbers, separated by an underscore. The zipped directory should include your code and scripts, your trained parameter files, and all output plots and images.

The report should include a title (e.g. minimally ELEC 475 Lab 5), your name and student number. If you are working in a team of two, then include the information for both partners in the report (but only make one submission in OnQ).

Your code should execute by entering the syntax provided in your two included scripts (i.e. **`train.txt`** and **`test.txt`**) on the PyCharm terminal.

The marking rubric is as follows:

| Item | mark |
|---|---|
| Step 1 | 2 |
| Step 2 | 2 |
| Step 3 | 2 |
| Step 4 | 2 |
| Report | 3 |
| Correct submission format | 1 |
| **Total:** | **12** |