MLab Fall 2023 Project: Bringing Down BARCO

1 Introduction

In the MLab Fall Project, you will sharpen your deep learning skills by engaging with a high-impact real-world problem. In doing so, you will learn about infrastructure, data processing, implementation, model evaluation, and other critical tools of a machine learning practitioner. By the end of the project, you will have gained valuable skills and expertise in the field of machine learning.

2 Problem statement

> Birds Aren't Real, Co. (BARCO) have deployed a squadron of bird-shaped drones, equipped with machinery to perform high-accuracy fecal deposition on human heads.
>
> In other words, they poop on us.
>
> As part of the Machine Learning Anti-poop Bureau (MLAB), you will be trying to identify the drones by type. There are 20 types of BARCO drones. Because each type of drone requires different measures to take care of them while minimizing public disturbance, it is imperative that you identify the drones correctly.

This year's Fall Project is on **bird classification**. Visually identifying bird species is an important task in monitoring and protecting biodiversity, rescuing endangered animals, and generally evaluating the quality of the environment. Because birds are generally easier to monitor on a large scale than other indicator species, being able to accurately keep track of bird species is extremely important to ecological conservation.

Traditionally, identifying bird species by eye often requires high levels of expertise, especially for visually similar species. The size of the ornithology and professional birdwatching communities, as well as the inaccuracy of more casual birdwatchers, thus puts strict limits on our ability to keep track of bird populations on the large scale frequently.

We would like to use **computer vision** to automatically learn features from images of birds to predict the species of the bird in the image.

For this project, you will tackle a miniature version of this problem by attempting to learn and predict the species of birds across 20 different classes:

Baltimore oriole	Northern cardinal	Myna	Barn owl	Ostrich
Purple finch	Peacock	Albatross	Ruby-throated hummingbird	Hawfinch
Robin	Spoonbill	Sora	Scarlet ibis	Oyster catcher
Magpie goose	House finch	Green jay	Golden eagle	Frigate

Specifically, the problem statement is:

> Build a model that, given a *photo*,
> outputs the *predicted bird species* for the bird in the image.

You will train and test your model on a provided set of 3,113 images from the 20 classes. We will then evaluate your model's ability to generalize the bird species on unreleased test sets drawn from the same 20 species. This is analogous to using images labeled for species by bird experts in order to make predictions when we do not have expert labels.

3 Onboarding Contest Structure

Please form teams of up to **4** people. Don't worry if you don't have a team in mind – we're happy to help match you up into teams. Once you have formed a team or decided you want us to match you up, please fill out the form in **#mlab-general**, which you should do by this **Wednesday, October 26**.

3.1 Prizes

Stay tuned :D

4 Files

All files necessary to complete this project have been posted in the **#mlab-general** channel for your convenience.

The files should be in this structure:

```
project
├── bird_data.csv
├── bird_data.hdf5
├── bird_data
│   ├── 0-BALTIMORE ORIOLE
│   │   ├── 0_1.jpg
│   │   └── ...
│   └── ...
├── MLab_Onboarding_Project_Fall_2023.ipynb
├── small_data.csv
├── small_data
│   ├── 0-GREAT XENOPS
│   │   ├── 0_1.jpg
│   │   ├── 0_2.jpg
│   │   └── ...
│   ├── 1-KIWI
│   │   ├── 1_1.jpg
│   │   └── ...
│   └── 2-OSPREY
│       └── ...
└── utils.py
```

4.1 Starter code

We have provided starter code in `MLab_Onboarding_Project_Fall_2023.ipynb`. The starter code provides a good framework to approach this problem; however, you do not need to use this starter code if you wish.

4.2 Data

We have the training images and labels available in `bird_data.zip`. However, because of how these files are stored on the Google Colab machines, it is extremely slow to load all of the images individually. Therefore, we have compressed the training images, labels, and class names into a single HDF5 file, `bird_data.hdf5`. Code for loading this file has been pre-implemented for you in `utils.py`.

However, in practice we often do not compress the entire dataset into a single file. In order for you to get a better grasp of how to load data directly from the images, please implement the `BirdDatasetSmall` dataset object in the Colab notebook. This dataset loads `small_data`, a mini-version of the dataset (featuring three other bird species) image-by-image. **Please note that this is a deliverable for the project, not an optional component!**

5 Implementation

You must implement your model in **PyTorch** using the methods we have described in the workshops. More implementation hints may be found in the Project Tips section.

Please **do not** use external data (e.g. other images of birds). You are, however, encouraged to apply data augmentation techniques to the existing dataset!

Please **do not** use pretrained weights (e.g. models downloaded from online) for your model.

6 Evaluation

At the conclusion of the project, you will submit:

- Your **code** as a Colab notebook[1]. The code will not be directly evaluated, but will be used to run the model on a comprehensive **test set** which you will not be able to access until the project concludes. The code should contain the following:
  1. Implementation of the `BirdDatasetSmall` dataset. We will use the tests in the notebook to make sure that your implementation is reasonable.
  2. Definition of your `BirdModel` class.
  3. Your code for loading data, training, and evaluating your model. We will not be running this, but we do want to see how you do it!
  4. Implementation of the `predict` function. This function should accept a single **string parameter** which is a path to a `.jpg` image and returns a single integer from 0-19 corresponding to the numerical label of the predicted class.
- Your final **model**. Specifically, please use the "Saving model weights" cell in the notebook to save the model, and upload this file to Gradescope along with your code. You may assume that the test images will be of the exact same resolution and form as the training images.
- A written **report** describing your model (as a PDF). The report does not need to be comprehensive (i.e., with background information and references), and should only contain the following sections:
  1. **Architecture**. What model class did you choose, and why? Did you experiment with different architectures? How did they perform?
  2. **Training**. How did you train your model? What losses did you use and why? What hyperparameters did you tune and what results did you see? How can you explain your choice of hyperparameters?
  3. **Results**. Did you validate or test your model? Explain your choice of validation and test set. Report and describe strategies to reduce overfitting, if present. Did you conduct any error analysis? Can you hypothesize an explanation for the errors?

Please submit these on Gradescope by **11:59PM PT on Monday, November 28** (we will send out a class enrollment link later).

The final evaluation criterion will be the **accuracy** between the prediction and ground truth over all the test images.

6.1 Awards

A **winning team**, or multiple winning teams, will be selected and recognized after the project concludes. The winner(s) will be selected with $\frac{2}{3}$ consideration to the performance of the model and $\frac{1}{3}$ to the quality of the report. Additionally, all teams will be asked to briefly present their model and/or paper to the other teams.

7 Timeline

- **Wednesday, October 26, 23:59 PT**: Final teams released.
- **Monday, November 28, 23:59 PT**: Code, model, and report due. Project closes.
- **Tuesday, November 29, 19:30 PT**: Present your work at the regular workshop time.
- **Tuesday, December 6, 19:30 PT**: Announcement of winners.

[1] See the Project Tips section for details on how to download the notebook as a file

8 Project Tips

An updated appendix with tips for the project can be found at `https://www.overleaf.com/read/gjtzmpykxhbc`.

9 Getting Help

Please reach out to us with any questions! The entire ACMLab leadership team is available to provide help and guidance. We encourage you to post general questions that may be of interest to all teams in **#mlab-general**. You can also DM **@nivsiyer**, **@pliu1**, **@tejasn9**, and/or **@kuhegbu** with specific questions about the project specification, or any administrator of MLab with technical questions. Further, we will be doing weekly office hours if you would like to have in-person help outside of the project workday — stay tuned to **#mlab-general** for updates. Finally,

Have fun!

Acknowledgements

Tejas Narayanan came up with the idea of the project and built the framework code. Patrick Liu and Tejas Narayanan wrote this project specification document. Niveditha Iyer, Patrick Liu, Tejas Narayanan, and Kelechi Uhegbu are all responsible for the specification in its current form.