

Homecoming

Task 3A - Creating and training models to classify Animals and Habitats

This document explains the problem to be solved in Task 3A and where to go next after you finish reading this document.

Problem Statement:

In this sub-task, you have to build two models to classify Animals and Habitats from the given arena images.

Given an arena image, your program/model should output the name of the Animal or Habitat.

Objective:

For every image classification task that you get, it's not common to have 1000s of labelled images. But you still want to achieve good accuracy. What will you do then?

We know that a CNN learns features hierarchically. This means that the first few layers of the network identify only general features like edges. The next few layers learn shapes. The next few build upon the previous ones until your network is able to classify a single image into a category. This phenomenon can be exploited. We can take a good, pre-trained neural network and only train it partially on some new data.

Not surprisingly, it is able to classify images into new categories. This is called Transfer Learning. Transfer Learning is applicable to a lot of things other than just CNN.

In this task, you'll fine tune a pre-trained model to work on your data. It's up to you to train a single model for both Habitats and Animals or different models.

NOTE: You are only allowed to use libraries that we have installed and used in the previous tasks. Tensorflow or Keras should NOT be used.

Dataset: You can download the datasets for this task from following links:

1. Animals: [Animals Dataset](#)
2. Habitats: [Habitats Dataset](#)

You will only use these datasets to train your model.

TIP: You can use the *dataset.py* that we built in **Task 1B** for data loading.

Code Structure:

From now on, you will develop things with only little supervision and mentoring. But we still expect you to follow good programming standards and writing reusable, least redundant code wherever possible.

1. Since we are NOT giving you templates or anything it will have a great weightage from now on.
2. If you write tests now, it is even better.
3. Faster and smaller models in size will be ranked better.

Inside your project folder, you'll have a **main.py** which will be the single most important entry point in your program.

Your **main.py** should expose these **parameters**. You can use argparse.

Name and Argument	Usage	Description
Animal image path (-a)	-a ./animal.jpg	Will output animal name from the image
Habitat image path (-h)	-h ./habitat.png	Will output habitat name from the image
Animal model path (--amod)	--amod ./animal-model.pth	Use a particular trained animal model. This is not compulsory and can be default path to your model.
Habitat model path (--hmod)	--hmod ./habitat-model.pth	Use a particular trained habitat model. This is not compulsory and can be default path to your model.

The model paths are necessary if we tell you to upload models on drive and provide a link to us. We will then download the models separately and use your **main.py** that way. So you'll only be uploading your code to the portal and separately submitting the models as drive links.

NOTE: If we download and paste the models directly in your project folder (alongside **main.py**), the **main.py** should use those models by default.

Resources: Check *Resources.pdf* to know how to make a model and other things. That's all.

Best Wishes!