Milestone 3 Report

Team Deinonychus – Patrick Creighton, Luka Rogic, Christopher Lee, Juntong Luo

Current State of Project

Building upon the base of our project laid out in previous milestones, which consisted of a basic file uploader and downloader application along with the exploration of machine learning models for object detection and facial recognition; the aim for this milestone was to connect these components together to produce an early working version of our project that is able to categorize and tag uploaded images and return them to the user. Currently, our application is able to input images uploaded by the user to our custom-trained ML model to classify and tag them. The user can then view all their uploaded images on our application's gallery page which displays the image along with the category the ML model classified the image as. A visual demonstration of our application can be found in the README of our project's repository.

Main Milestone Goals

From our proposal, our main goals for milestone 3 are: finalize our custom-trained ML models, connect the models to the application, and have our application design/layout mostly finalized. Relating these to the current state of our project, we have finished training all our models and connected the core ones to our application. The rotation correction model is an optional application feature that we started later on, which we will be connecting to the application for the next milestone. The application design is not quite finished, but the proposal goals did not take into account that the backend would have to be completed before the frontend. Thus, we have decided to focus on finishing up the backend in the current milestone, leaving the frontend for the next milestone.

Current Challenges / Bottlenecks

1. Uploading our application onto the Google App Engine for public use.
    - We are currently facing challenges with deploying our application with the added ML models. We are looking for a solution to this issue, but if we are unable to find one, we will have our application run locally as an executable instead.
2. Predicting with the custom object detection model as it sometimes misses obvious object classes in a photo.
    - Will begin debugging this issue by logging what objects are being detected in each image and comparing that to our notebook model, then if necessary, manually filtering bad samples from the training dataset and retraining the model.

Team Member Responsibilities

Christopher

- Added photo gallery and [With Juntong] implemented the machine learning tagger in the application, along with the ability to view tags from the gallery.
- For milestone 4: Add functionality to manually edit image tags in gallery; Improve appearance of application to increase usability and user experience.it run locally

Luka

- [With Patrick] Added code that maps objects detected in the image to their category tags.
- Finished the open images dataset builder by fixing the issue from M2 where it would not properly label all classes in an image (see the Open Images Downloader Finished notebook).
- Started training the model on classes from Tags and Objects.pdf with 2000 images for each class.
- For milestone 4: begin writing final report, finetune/experiment with the model and finalize the object category tags.

Juntong

- [With Christopher] Added code to display tags on the webpage.
- Added confidence levels of the face recognition model to the tagger. The tagger now stops tagging faces that it is not sure.
- For milestone 4: Connect rotation correction model to application.

Patrick

- Added confidence levels to the face recognition model.
- Developed and trained a low-light enhancement model.
    - Pictures that don't have a consistent brightness level are outputted with bright, neon spots, rendering it unusable -> label feature as experimental for the project deadline.
- Developed and trained a rotation correction model.
- For milestone 4: work on homepage frontend, add rotate and crop function to application.