FAUNA: Flora & Animals Universal Application

Report

Joshua Wilkinson

South East Technological University

Date: 19/04/2024

Student ID: C00262503

Abstract

This is the last report for the FYP, the project report. This report demonstrates that I have learned many new skills during the development of my project, FAUNA. There will also be the identification of any errors, and mistakes that were made during the development process.

Table of Contents

General Issues

Problems encountered and how they were resolved

Model training

I needed to find a way to train my convolutional neural networks, and when using my CPU, it would take a long time. The solution was using a GPU (Graphics Processing Unit). GPU processors can execute an AI algorithm much more quickly than a CPU. This means that an AI application or neural network will learn and react several times faster on a FPGA or GPU compared to a CPU.

Data preprocessing

Gathering data is usually pretty tedious, and often takes up a lot more time than you would expect. The data type being worked with was images. Initially, I found a snake dataset on Kaggle, but it was not labeled in a nice way (1,2,3,4, ect.), so I had to rename them to the actual snakes for clarity (135 classes, for training and testing).

Loading a model into Android Studio

When I started development, I was immediately faced with the issue of loading a model into android studio in order to make predictions. I had to export the models to tflite files (supported by android studio). I also learned that models with “metadata” have access to exclusive libraries, which I wasn’t able to use with my custom models. After a while, I was able to solve the problem by loading the rgb bytes into a bytebuffer array, and feeding that into the model.

Creating a map

Later on in the development of the app, I needed to create a visual representation of the different sightings around the world. To do that, I added Google's Firebase database to the android studio project, and used the longitude and latitude of the user's last location, uploaded that with the name of the classification to the database. From there, I created an RMD (R markdown) script to grab that data, clean it, convert to html, and present it on my website. It works well, however, I need to manually update the RMD file on my website, which is also accessible in the app. I also needed to track the coordinates of where a picture was taken. You can get a phone's location easily in Android Studio, but I also needed to store the data somewhere. I had heard of Google's Firebase, but had never used it before. It has great support in android studio, and is ideal for mobile devices. It was only a few lines of code in the end to upload to the database.

What I achieved

For the entire duration of this project, I managed to create a phone app, and judging directly from my specification document, I have implemented all of the functional requirements specified there. I was able to create my own CNN models, apply that to a phone's camera, and use OpenAI api tools to avoid using a database for storing descriptive information about the pictures. While I didn't use a database for storing descriptions, I used Google's Firebase to store location data that is applied to a map on my website, displaying all of the various sightings across the world, from people using my app. Overall, the aim of the project was to see if I was able to achieve these things.

What I did not achieve

Unfortunately, I was not able to finish the iOS version of the app. At the moment I am writing this, it is exclusively compatible with Android OS. Also, I feel that with more time, I would be able to add more categories to the machine learning side of the project (such as mushrooms) and improve the validation accuracy (how a machine learning model performs on new data) of each model.

What I learned

There were a good few things that I set out to learn for this project, that I didn't really know before, so here's a list:

- Kotlin
- Android Studio IDE
- Android Multi Platform Development
- Tensorflow
- Keras
- CNNs
- OpenAI api
- Data Preprocessing
- Firebase
- R
- Open Source Maps
- Anaconda

What I would do differently if starting again

If I were to start again, I would try to add more comments to my code explaining what the important lines and functions do, especially since there are many lines of code. Also, I could have done a better job at creating neater code; improving readability is very important. More of a focus on UX would have been nice too. But, I suppose that the biggest thing would be that I would spend more time on improving the features, such as the machine learning models. Although, I could certainly improve them overtime, now that I have the app developed.

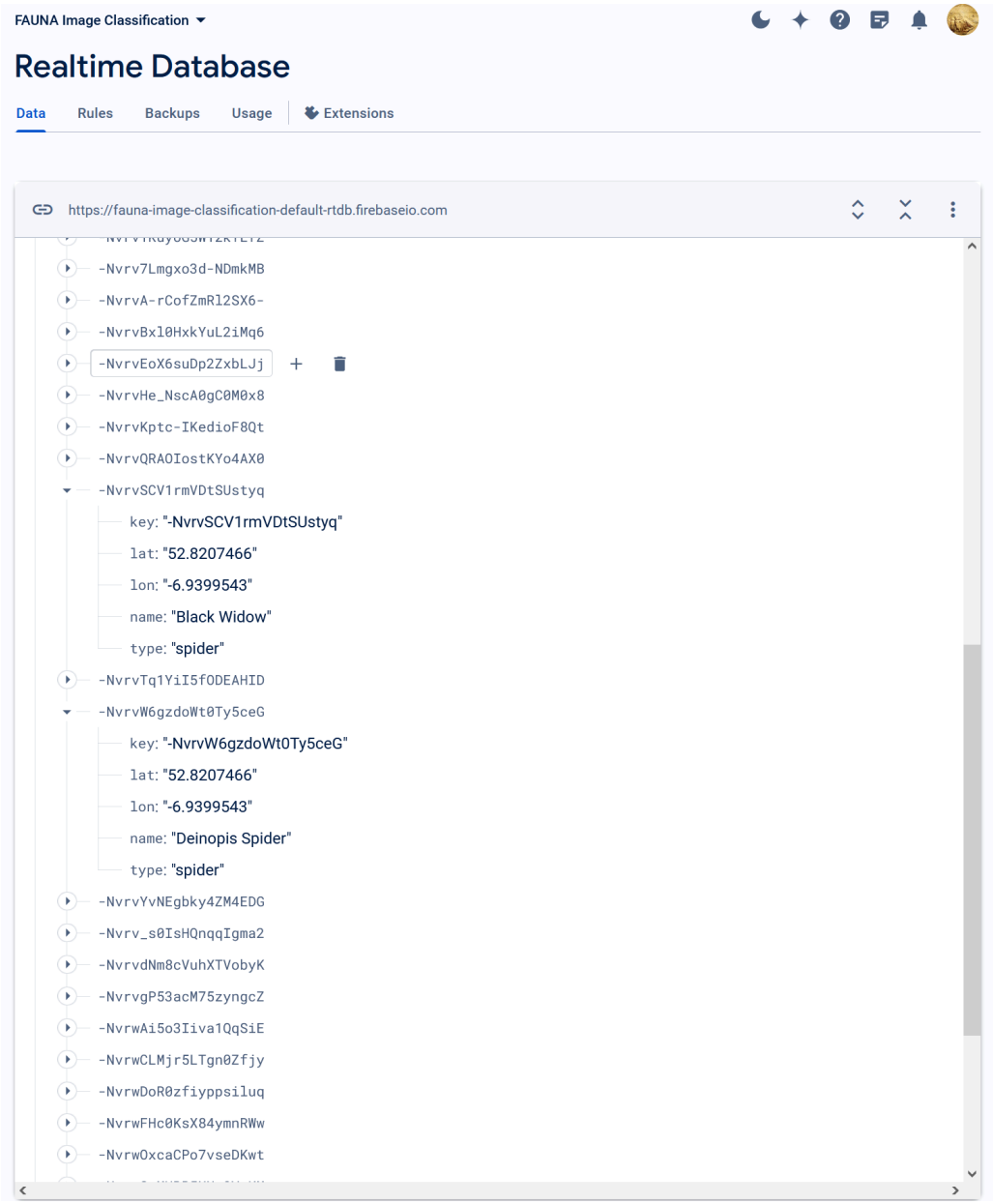
Technical Issues

Differences from my earlier design & additional research

The functional requirement had changed a lot by the second iteration, with some different functional requirements, such as having a map feature. However, there was one feature I didn't get around to, which was finding a way of taking the images that the user takes, and improving the machine learning models.

When I decided to do a map, I did some additional research into the different ways I could go about that. One was using Google Maps api, that's supported by android, but that wasn't free. So I wanted to use a free alternative. The programming language R had a library I could use (mapview), so I went with that.

Data structures (files/databases)



Google's Firebase cloud service

Firebase is structured using JSON. I used it to store location data for the map.

Testing

The IDE (Integrated Development Environment) that I used was Android Studio. In Android Studio, when something goes wrong in your app, it gets written down in the log (logcat). You can also trigger custom logs, when debugging. For example, when I was trying to get the OpenAI api working, the server requests would come back with a new error each time, until I eventually got it right.

References

- Google, Firebase; https://firebase.google.com/