**COMP4092 Logbook**

**1.0 Week 4 – Semester 2**

**Research/ Work**

**1.1 Prior to 15/08/2022**

* After confidently putting Steve Cassidy’s project as my number 1 preference, I did some very basic initial research into tensorflow lite and its relevance towards the project.
  + It seemed that its purpose was to provide mobile devices (android/ios) with the ability to implement machine learning models for various purposes.
  + The project description highlighted that the application would need a machine learning model that could tag photographs taken on the device.
    - Based on this description the problem seems like a vision based/ image classification problem.

**1.2 15/08/2022**

* With the project being confirmed as Steve Cassidy’s project, there was no hesitation to contact Steve to see what further steps can be taken to start exploring the problem.
* Steve promptly replied pointing me to begin some initial research on ML models for:
  + Vison problems
  + Mobile devices
* Steve also suggested some initial research into the FAIMS application as they are the industry partner that will benefit from my developments.
* I found an academic paper which explained what the FAIMS mobile application is, it’s a native android application which is supported by an Ubuntu server. I assume that the front-end is managed by Java/ Kotlin code for UI design and the backend handles all the data.
* The purpose of the application seems to be to provide the user (archaeologists, biologists, geologists, generally field workers) with the ability to efficiently capture data that would satisfy the requirements of standard fieldwork collection.
* It seems like the purpose of this application is to make the process of collecting data in the field efficient compared to having to manually write the data down.
* I refreshed my knowledge on the applications of machine learning as I hadn’t studied the area for a while.
* It’s clear that there is a lot of use cases for machine learning and that it can be applicable to this problem and mobile applications in general.
* From my brief initial research, it seems that there is a lot of tools provided for implementing machine learning into a mobile application.
  + Amazon
    - Amazon Rekognition
  + Google
    - Tensorflow
    - Google vision API
    - Mobile vision API
  + Salesforce
    - Einstein Vision
  + Clarifai
* This is only my initial findings, I’m sure there are plenty more libraries and tools which can offer varying results and need to be evaluated to determine the best fitting tool.

**1.3 17/08/2022**

* Unfortunately, the planned meeting was postponed, however Steve directed me to the next steps of research for the project for the week.
* Questions for the project are starting to now form as I have a clear direction of what I’m meant to be exploring

**2.0 Week 5 – Semester 2**

**Research/Work**

* Researched the image classification problem
  + Questions
    - What is an image classification problem?
    - What is the structure of an image classification task?
    - Data requirements?
* Structure of an image classification task
  + Pre-processing of the images
    - Preparing the data before processing it in the image classification model
    - A useful process to ensure better results from the model
  + Data augmentation
    - Post-classification
    - Making use of limited data
      * Modify the limited image data with transformations
    - Techniques include
      * Image flipping
      * Grey scaling of the image
      * Gaussian blurring
    - Data augmentation techniques will benefit smaller projects with limited access to data
* Different networks and image classification techniques
  + CNN
  + ANN
  + SVM
  + Decision trees
  + K-nearest neighbour
  + It seems like CNN is the best approach for image classification problems
    - Tested approaches
* Accuracy questions
  + How accurate does the system have to be?
  + How much data is needed?
* Conducted extensive research into the image classification problem.

**3.0 Week 6 – Semester 2**

**Research/ Work**

* **Tensorflow lite vs Tensorflow.js**

Graphical user interface, text, application, table

Description automatically generated

* Tensorflow lite
  + You can’t train new or existing model
  + Create your model convert it to tensorflow lite and deploy it to firebase > then can use the model on ios and android natively
  + Since its focus is on native code > gets benefit of mobile device hardware
* Tensorflow.js
  + You can train new or existing model
  + Node.js powered
  + Open source
  + Can allow for a deploying of python ml model directly into JavaScript
  + Cost
    - No server needed for inference
  + Privacy
    - No data is sent to the server
  + Runs on the client side
  + Flexible
    - Can run on backend, frontend, mobile devices, react, react native, IOT devices
  + Easy to use
  + Don’t train the model on the device
    - Train somewhere else i.e. tensorflow then import model into device?
  + Better for web applications
  + Uses webGl
    - Don’t think it would work with wrapping it in ionic capacitor as it wouldn’t make use of devices GPU, only the webGL or other web API’s to render components.
  + Tensorflow.js for React Native
    - Does use devices GPU – but still not as effective as I THINK as Tesnorflow lite
* Inference
* Epochs
  + Iterations
  + The more epochs/iterations, the more accurate the results will be
* Image Classification – Specifically on mobile devices how much training data to get/ will need
  + CNN
  + Training an existing model
    - transfer learning
  + Image Classification on Mobile Devices

**Meeting Notes**

* React web application
* Compile it into a anroid app or ios app using electron
* Wrap it using ionic capacitor
* Faims
  + Allows you to collect survey data
  + Create a notebook
    - Campus survey
      * Can attach a photo
      * Notes on photo
      * Location
      * Asset number
    - Gen
    - Archaeology
    - Want to record details of the dig
  + Generate a unique id for each record
  + Synchronises with backend
    - Backend
      * Conductor
        + Main things it deals with

Login

Does it with data central

* + - * + CatchDB

Manages database

Does version management of the data

* + - * Notifies of conflicts
  + User can create their own notebook with adjustable fields
  + Can be used offline and collect data – when reconnected data is synched to server

**Questions/ Thoughts**

* Is implementing tensorflow lite into the native ios, android os better vs tensorflow.js wrapped in ionic capacitor?
* Incorporating into faims a take photo button which uses ml learning to identify the image and provide a certainty response
* Training the model in python or can it be done in Tensorflow.js?
* Academic Readings
  + Image classification – specifically on mobile devices how much training data to get/ will need?

**4.0 Week 7 – Semester 2**

**Research/ Work**

* Looked into training a pre-trained ML model to solve a different problem
* Began developing demo using a tensorflow sequential model which was trained on a flower dataset
* Setup Github repository
* Setup latex/ overleaf structure
* Sent Github repository link and overleaf link

**Meeting Notes**

* Tensorflow lite and capacitor? How to integrate?
* Evaluating both alternatives to see if js application was ok or lite application for our task
  + Product implantations in both and compare performance
* Part of planning
  + eurali
  + Evaluating options is a core component
  + Results
    - Tensorflow.js is good here
    - Tensorflow lite is good here
    - Keep options open
    - Make recommendations about an application that work for FAIMS
    - Pin down image classification task – data sets that a prototype can be tested on
    - Whats the workflow?
      * Capture data using Faims and then use that data for training the
      * Slick workflow that a non-technical could instigate and manage – ideal and point towards
    - How much data do we need?
  + Phase 1
    - Training model
  + Phase 2
    - Evaluation execution engines
    - Find ANY/All relevant implementations
      * Part of thesis is evaluating/ comparing these different implementations
  + Phase 3
    - Prototype implementation
* Scope out what tools needed for
* Cloud tools for training model? – not training model on device
  + Google colab free GPU use
  + What kind of cloud solutions for training?
    - Upload the models to the device
    - Not retraining the model on the device
* Surveying the current background of machine learning on mobile devices/ edge devices to understand the current situation and determine how the project fits into the current field
  + Find 5 years older + work as they are tried and tested methods, not recent works as they are too unreliable/experimental – inception stuff
* General structure of chapters you want to include
  + Introduction chapter
  + Literature review chapter – euraling we are talking about – research plan
    - 2 or 3 content chapters would have main theme – machine learning , integration on mobile application, eural networks for ML, overall tools for mobile machine learning
  + Things about capacitor, mobile applications, machine learning implementation for mobile applications, ml models, performance
* Overall plan
  + Finish writing by week 12 – submit by week 13
    - 6-7 weeks to write
    - Next 4 weeks
      * Exploring things – things we found
    - Final 4 weeks
      * Refining thesis – has plan
    - Should have a exploratory demonstration

**Questions/ Thoughts**

* How to integrate tensorflow lite and capacitor?
* What is workflow going to look like?
* Since we’re using CNN’s are Deep Neural Networks the same?
  + CNNs are a type of deep neural network

**5.0 Week Mid-Sem 1 – Semester 2**

**Research/ Work**

* Taking notes on sourced literature
* Sourcing literature
* Working on demo of the sequential model trained on flower dataset
* Working on demo of TensorFlow MobileNetV3Large model being converted to tensorflowjs and deployed in a react native web application

**Meeting Notes**

N/A

**Questions/ Thoughts**

* Need to consider how to integrate both tensoflow lite and tensorflow js into the workflow?
  + They are both different ways of converting/ using tensorflow models
  + This system will be deployed in IOS, AOS and web application

**6.0 Week Mid-Sem 2 – Semester 2**

**Research/ Work**

* Continued working on demos
  + Finished the sequential model training demo
  + Still working on react native web application demo with MobileNetV3Large
* Structured thesis.
* Experimental/ New Approaches vs Tested/ Reliable Approaches
  + New Approaches
    - On-device ML
      * Whilst reduces inference due to device handling model (training, using) potentially limits accuracy of model due to restrictions is model size due to hardware/device performance
        + Less users due to imposed limitations
        + Greater costs associated with better hardware
      * More secure as data is handled on device
  + Reliable Approaches - (less secure, greater inference)
    - Server-side model
      * Server
        + Model is trained
        + Model is utilised and performs classification on server and results are sent to device
      * Client
        + Sends requests
        + Receives responses
    - Client-server model
      * Server
        + Model is trained
        + Model is re-trained with new data from client and then sends updated model
      * Client
        + Model is loaded onto device - with updates

**Meeting Notes**

N/A

**Questions/ Thoughts**

* Steve wanted a tried and tested design
  + What is the best design that encompasses this requirement?
  + A pure client side is probably too experimental and provides another layer of questions and constraints

**7.0 Week 8 – Semester 2**

**Research/ Work**

* Finalised demos
* Continued writing thesis
  + Structured the paper
* Finished intro to paper
* Continued working on background/ literature review
* Explored provided public datasets

**Meeting Notes**

* Explore 3 models not what is the best model - we are trying to see if this is what is currently being done
* Explore the data sets that were sent they were implemented with certain models
* Explore testing and training the selected models
* Write more by next Thursday - Steve will review
  + Overleaf outline
  + Good outline of 1 of the chapters

**Questions/ Thoughts**

* How will my paper be structured?
* Which datasets should I use?
* Do datasets have affinities to certain models?

**8.0 Week 9 – Semester 2**

**Research/ Work**

* Fished thesis introduction
* Continued working on background/literature review
* Explored object counting models
  + - [https://github.com/Tudor67/Object-Counting](https://protect-au.mimecast.com/s/D7kTC71R63ClRxEgh8NaFh?domain=aus01.safelinks.protection.outlook.com)
    - [https://paperswithcode.com/dataset/rsoc](https://protect-au.mimecast.com/s/j7d-C91W8rCQOgR0I3hDkh?domain=aus01.safelinks.protection.outlook.com)

**Meeting Notes**

* Object counting instead image classification
* UI
  + User can edit the items selected for counting
  + Model says 10 objects there's actually 8
* Vertical sprints
  + Elaboration on what has been explored
  + Elaboration on image classification
  + Elaboration on object classification
  + Elaboration on tfjs vs TensorFlow lite
* Next week
  + What exploring has been done
  + Explore the counting problem
    - More useful
    - Counting dataset
      * Could also
  + Dataset
    - Maybe provided by CSIRO but at the moment want a chosen dataset
    - Crowd dataset
    - Traffic and congestion dataset
      * Trancos
      * Tudor67/objectcounting
    - In a image there will be less than 200 objects to count
  + How much seed data is needed to train the model?
    - Collecting my own data?
    - Label
    - Start with 200 take 20 away then take a picture

**Questions/ Thoughts**

* What are the processes involved with collecting my own data?
* How will the ui elements interact with the ML system?
* Using pytorch vs TensorFlow

**9.0 Week 10 – Semester 2**

**Research/ Work**

* Continued working on background/ literature review
* Began methodology
* Began preliminary design
* Restructured paper based on feedback from Kate and Ansgar
* Structured paper
  + Finalised structure
    - Intro
    - Background/literature review
    - Methodology
      * Research question
        + Questions that answer research question
    - Preliminary design
      * Current design choice subject to change
    - Project plan
      * Plan for implementing design
    - Results
    - Discussion
    - Conclusion

**Meeting Notes**

* Look at what they have done and see what they have done
  + Find code to do object counting that is not too complicated
  + Take it and deploy
* Not tweaking the architecture is it better
* Not taking the best model finding a model that will perform well and be implemented easily on mobile
* Target 4 candidates for image classifcaition and object project/ models - describe them, explain strengths about them
* Perceived dfficulties - review
* Find something that has code, is tested and stable for, tackling both problems 2 different workflows
  + Try counting 1st and then image classification
* User experience with the object counting - how will it appear, how many images should they supply, annotation of the images, can they modify the counting result

**Questions/ Thoughts**

* Can a seamless workflow for both object counting and image classification be found?

**10.0 Week 11 – Semester 2**

**Research/ Work**

* Finalised background/ literature review
* Continued working on methodology
* Continued working on preliminary design
* Began work on project plan
* Began work on project requirements

**Meeting Notes**

* Collect the data through Faims - being a user of FAIMS
* Implement a form field through ML exentesion to faims
  + Fork faims app build in the ML capability
  + Plugin for integration

* Results Section
  + Building tests
  + Testing tests
  + Highlight problems

**Questions/ Thoughts**

* What kind of tests does my system need to account for?
* How will integrating my work with FAIMS look like?
* Best to be considerate about FAIMS whilst designing my preliminary project

**11.0 Week 12 – Semester 2**

**Research/ Work**

* Finalised methodology
* Continued working on preliminary design
* Continued working on project plan
* Continued working on project requirements
* Began editing

**Meeting Notes**

* Reviewed thesis
  + Is coming along well!

**Questions/ Thoughts**

* Preliminary design has highlighted system choices
  + Client and server architecture
  + Chosen mobilenet models for both image classification and object counting
* Continue to work on thesis

**12.0 Week 13 – Semester 2**

**Research/ Work**

* Finalised preliminary design
* Finalised project plan
* Finalised project requirements
* Finalised other components of thesis
* Edited thesis
* Finished Thesis part A

**Meeting Notes**

* Steve reviewed thesis and noted areas for improvement
  + Adressed feedback
* Steve noted areas in thesis that needed to be added
  + Added sections to thesis

**Questions/ Thoughts**

* Preliminary design highlighted system choices
  + Great preliminary results from testing data augmenetation to fix overfitting on Sequential demo
    - This approach will be tested and used on other models in future developments
  + Approach to using epochs and how they will look in system
  + The work environment I have setup and will use to complete the project
  + Built system designs with diagrams
  + Implementation for AOS, IOS and web application