**Acknowledgement**

Firstly, I would like to acknowledge my supervisor Steve Cassidy, for supporting me and providing valuable insight throughout this endeavour. I would also like to specifically acknowledge my partner for the support provided and my little brother for making me laugh throughout this project. Finally, I would also like to acknowledge my family and friends, who reminded me to persevere throughout this project.

**Abstract**

The popularity and advent of implementing Machine Learning into various systems and industries is a developing area with more interest growing each day. The purpose of this paper is to explore this interest in on-device Machine Learning for FAIMS. FAIMS is an application that simplifies the data collection processes in STEM fields. The interest is concerned with the computer vision problems of image classification and object counting in image samples collected using the application. By the end of this paper, a proposed on-device Machine Learning workflow for FAIMS will be presented. The steps involved in creating this workflow include a survey of the current on-device Machine Learning environment. The existing approaches are analysed, and a tailored testing method and preliminary design are proposed to highlight the process for solving these on-device Machine Learning problems.

The rise and rapid growth of research surrounding Machine Learning and its potential use cases have captivated many professionals from various disciplines. These use cases range from many walks of life, such as medicine to commerce. Each target domain has a unique problem that requires a particular solution to solve its issue. For instance, a team of medical practitioners' relevant use case is being able to diagnose and detect cancer in their patients so that they can be quickly classified into high or low risk categories \cite{kourou\_machine\_2015}. The aim of ML is for a machine or computer to incorporate learning processes similar to the human nature of learning and solve unseen problems based on the training received \cite{carbonell\_1\_1983}. Fortunately, with the significant research and development in ML, there are appropriate ML models and algorithms designed to effectively generate the desired results for a variety of problems \cite{jordan\_machine\_2015}. The current research into ML has broken down into three key areas these being:

\begin{itemize}

\item Task Orientated Studies

\item Cognitive Simulation

\item Theoretical Analysis

\end{itemize}

Versatility in ML systems is achieved as hardware performance has significantly improved over the past decade, allowing for more exceptional opportunities for usability in dedicated workstations and edge devices. The ability to apply ML to various devices, like mobile devices, is becoming an ordinary reality due to the target use case being addressed. It's also important to highlight that depending on the project's complexity, issues and constraints, implementing ML into a system may not be viable as it could not generate the desired accuracy results or may exceed the cost constraint for the project.

\section{FAIMS}

FAIMS is an existing cross-platform application designed to simplify the data collection processes for various science disciplines. The application is robust because it can account for various data expected from the different fields whilst being offline \cite{ballsun-stanton\_faims\_2018}. The application recently underwent a re-design into the FAIMS 3.0 version to adapt its usability for multiple platforms whilst retaining the core functionality provided by previous versions of FAIMS. The vision of a multi-platform system is achieved with the web application being designed in the React framework using JavaScript and the native applications being created by wrapping the web code in native application code\cite{ballsun-stanton\_faims3\_2021}. These generated apps provide the benefits of creating a multi-platform solution whilst not requiring specific teams to develop for targeted platforms. Additionally, native APIs can be targeted through access to the native code allowing for a robust user experience on each platform \cite{xanthopoulos\_comparative\_2013}.

\subsection{FAIMS Use Case}

As already discussed, ML has a variety of use cases. This report aims to provide a workflow for solving a particular type of ML problem for FAIMS. This problem is taking the existing system's ability to upload image data from the field, manually label it, note information about the objects present in the capture, count how many objects are in the sample, and adapt it so that these processes will be automated. This issue will be automated using an ML model to classify and detect/count objects in the image sample. The labelled image and counted objects will be entered into the notebook depending on the model's prediction. This adaptation will benefit the core functionality of the FAIMS application, making field data collection more efficient. Ideally, implementing this new feature into the existing infrastructure would require using the ML feature in an offline environment.

\subsection{On-device Machine Learning}

The hardware ability of edge devices has impressively progressed, allowing ML to expand from the constraints of only functioning on powerful workstations. On-device ML exemplifies this sentiment as it is an alternative way of conducting typical machine learning on devices that have less performance than workstations and lack the feasibility to update hardware to improve performance. On-device ML can be accomplished through many means due to various options and methods to approach the issue. For a robust workflow to be presented to FAIMS, a selection of standard and tested designs will be examined.

\begin{itemize}

\item{Server-Side Design}

\item{Client and Server Design}

\item{Client-Side Design}

\end{itemize}

The client and server design will be used as the architecture for the FAIMS implementation due to being the most robust and tested approach that satisfies the problem's constraints. This paper will also present the exploration of alternative designs to provide a comparative analysis and benchmark the effectiveness of each design.

\section{Current FAIMS Application Infrastructure}

As stated, FAIMS is a cross-platform application with various interfacing sub-systems that allow it to function. Key areas of the application infrastructure need to be addressed to ensure the proposed workflow will properly handle this existing design, incorporating it into the choices made for the method.

\subsection{Ionic Capacitor}

FAIMS uses ionic's capacitor framework to simplify the process of building platform specific applications by generating native builds from the JS code used for the web component of FAIMS. This process is achieved by capacitor compiling the JS code into its native platform builds like IOS or AOS. With generated platform versions, this allows for native APIs to be easily targeted and used \cite{ballsun-stanton\_faims3\_2021}.

\subsection{CouchDB}

The database for FAIMS, which is responsible for managing the application's data, specifically survey data found in a created notebook. Version management of the data is also conducted.

\subsection{Web Application}

This version of the FAIMS system is a web application programmed in JS and utilises the React framework for implementing web components.

\subsection{Mobile Application}

The mobile version is a platform native build of the FAIMS system generated from the web application through capacitor. Native code can be accessed and modified to implement platform specific changes and target device APIs.

\section{Purpose and Structure of Paper}

This paper intends to examine any related literature that provides research on the image classification and object counting on-devices problem. Moreover, after reviewing the relevant literature, a solution will be proposed to implement a highly accurate image classification and object counting model, which will be trained and retrained on the cloud and loaded to edge devices for inference. Inference is how the model is executed. During the review stage of this paper, it will explore the various and current solutions to solving this problem. This review will assist in understanding the approaches taken and seeing the lessons in adapting a solution for FAIMS. Whilst not providing a revolutionary solution to this commonly researched problem, it will attempt to provide a unique, specific solution for FAIMS, which will clarify approaches and research already conducted. Through the transfer learning process, a pre-trained model will be evaluated, as seen in later section of this report and retrained with a relevant data set, allowing it to perform its specified task accurately. Any unseen data that the model will classify inaccurately will be sent to the cloud to retrain the model to ensure accurate predictions and classifications will be made. This paper will follow a structure of; initially, an in-depth analysis of the currently existing literature provided in chapter \textbf{\ref{chap:LitReview}}, continued by a high-level methodology to answer the research question in chapter \textbf{\ref{chap:singleuser}}, a more in-depth workflow solution for image classification and object counting in the FAIMS application is provided in chapter \textbf{\ref{chap:design}}, proceeded by a project plan highlighting the steps to completing the project in \textbf{\ref{ chap:plan }}, followed by an analysis and discussion of the results in chapters \textbf{\ref{chap:Results}} and \textbf{\ref{ chap:Discussion }}, and finally the conclusions derived from this paper in chapter \textbf{\ref{chap:Conclusions}}.