Linh Tran

An augmented reality and machine learning iOS educational application

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme

Thesis

November 2018

Metropolia

Author Title Number of Pages Date	Linh Tran An augmented reality and machine learning iOS educational application 37 pages 19 November 2018
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Professional Major	Media Engineering
Instructors	Kari Salo (supervisor)

The objective of this paper is to provide better understanding about augmented reality as well as machine learning technology. The study demonstrates the popularity and importance of these technologies at the time this paper is written. ARKit and Core ML are mainly discussed throughout the study. However, a few alternatives are also briefly examined. Furthermore, the study shows comprehensive information about how to train a machine learning model for iOS application. Together with thorough gained knowledge, a practical trailing goal is to create an appealing educational iOS application based on prevailing technologies.

The research is carried out with extensive explanations of the technologies. In addition, the paper provides a broad plan of how to create iOS application utilised ARKit and Core ML . Besides, details about how to train Core ML models that are compatible for iOS are discussed thoroughly.

During the time that this project was carried out, Camera Translator, an iOS application was under construction. The app is developed in Swift with support of CoreML and ARKit. Camera Translator was created as the final product of this project. The application targets youngsters with the age of six and above. With help of this application, for example a field trip to zoo would would be quite exciting to pupils. It is not only providing advanced new experience but also implicitly helping users to gain better knowledge about animals.

Camera Translator brings brand-new sense to zoo visitors by offering modern touch as well as informative observation. With supported functionalities, application is expected to gain wide-spread usage. What is more, improvements about information presentation as well as recognition accuracy are considered for further development.

Keywords	Augmented reality, machine learning, Turi Create, ARKit, Core ML

Contents

List of Abbreviations

AR	Augmented Reality
ML	Machine Learning
AI	Artificial Intelligence
GPS	Global Positioning System
API	Application Program Interface
VIO	Visual-Inertial Odometry
IMU	Inertial Measurement Unit
CPU	Central Processing Unit
GPU	Graphics Processing Unit
WWDC	(Apple) Worldwide Developers Conference

1. Introduction

In recent years, augmented reality has been hailed all around the world. AR has been making its transformation from a futuristic new technology to one of the most lucrative technology battle fields where many big tech companies want to a share of. Regarding AR software frameworks, early 2017 marked the release of ARKit by Apple and AR-Core just few months after by Google. It would definitely be fair to mention Pokemon-Go as the phenomenon of the era. In fact, AR will not only be dominant in gaming but also it will be used in different sectors such as education, entertainment, news and other media kind of media. Having that said, augmented reality is no longer just gazing-technology but it is becoming more acknowledged by education field due to the new features brings to learning.

If we see augmented reality as something important in the present time, then artificial intelligence or machine learning will be relevant in the future. Just having a quick search on Google at the moment of time, Google could give us about 313.000.000 results within 0,32 seconds. Together with the result, Risto Siilasmaa who is the Chairman of the Board in Nokia and founder of F-Secure Corporation, believes that all organisations should have at least basic understanding of machine learning and ML is the modern technology that could create enormous value to the society [1]. Machine learning or AI are being integrated into various types of fields including educations, entertainment, and car manufactures.

Being an engineering student and a software developer, the author of this thesis has a great interest in these new technologies and hopes to benefit them to create an application that would serve educational purposes. Thus, this research aims to evaluate the power of current augmented reality available as well as machine learning technologies which were chosen for development. Moreover, this study focuses on steps and factors that are required when building a new attracting educational iOS application.

2. Background

2.1. Augmented reality definition

Augmented reality is the integration of digital information and physical world [2]. The technology definition could be translated as superimposing digital objects onto real world. This creates an alternative method of how we interact with the world [2]. AR does not replace any objects of the environments but it overlays computer generated data on top of existing scenes continuously and implicitly.

Augmented reality delivers digital objects to real world to enhance the sense and interaction of human beings with surroundings. Thus, AR goal is to help users gain maximum experiences and perception with virtual objects in physical environment in real time [3].

2.2. Machine learning definition

Machine learning is truly just data and math [1]. Machine learning is usually mentioned along side with artificial intelligence. Thus, it is essential to understand what AI is first.

AI is indeed just a jargon. Artificial intelligence is prominently thought of intelligence that a machine exhibits [4]. AI is an advancing group of innovations that empower computers to mimic cognitive procedures, for example, components of human reasoning as well as non-human types of cognition [4]. AI is a system which it would do everything like human such as thinking and giving decisions rationally. During 20th century, Alan Mathison Turing who was a computer scientist and a logician, was the pioneer in artificial intelligence field [5]. Alan Turing famous definition about the artificial intelligence is that AI is the science of creating machines that would have ability to do jobs that require human intelligence [6]. He believed that a machine containing a scanner and boundless memory would move through image by image, forward and backward in memory, interpret what it discovers and composes further image [5]. The scanner is designed with programmed instructions which has the ability to store images in the memory [5]. The Turing's concept implies the likelihood of a machine that could programmatically operate, change or enhance on its own [5]. Later, Alan Turing's origination is wide known as Turing machine [5]. Every advanced computers nowadays are basically originated from Turing machines.

Artificial intelligence contains a sub-discipline so called machine learning which also has a sub-discipline of deep learning [4]. Machine learning is the perception in which the system contains generic algorithms that provides engaging things of input sets of data with none instructed pieces of code [7]. Rather than having developer creating a specific program, generic algorithms take inputted data to compose own logic program.

For instance, one sort of algorithms would be grouping which is demonstrated in the below figure. It could sort the data to be into a same group based on shared features of inputted data. Similarly, the same generic algorithm which contains no differences in structure, is not only able to analyse hand writings but also could be used to filter spam emails [7]. It is clearly the exact pieces of code but it generates different logic outcomes when distinctive sets of data are fetched into, as shown in the figure below.

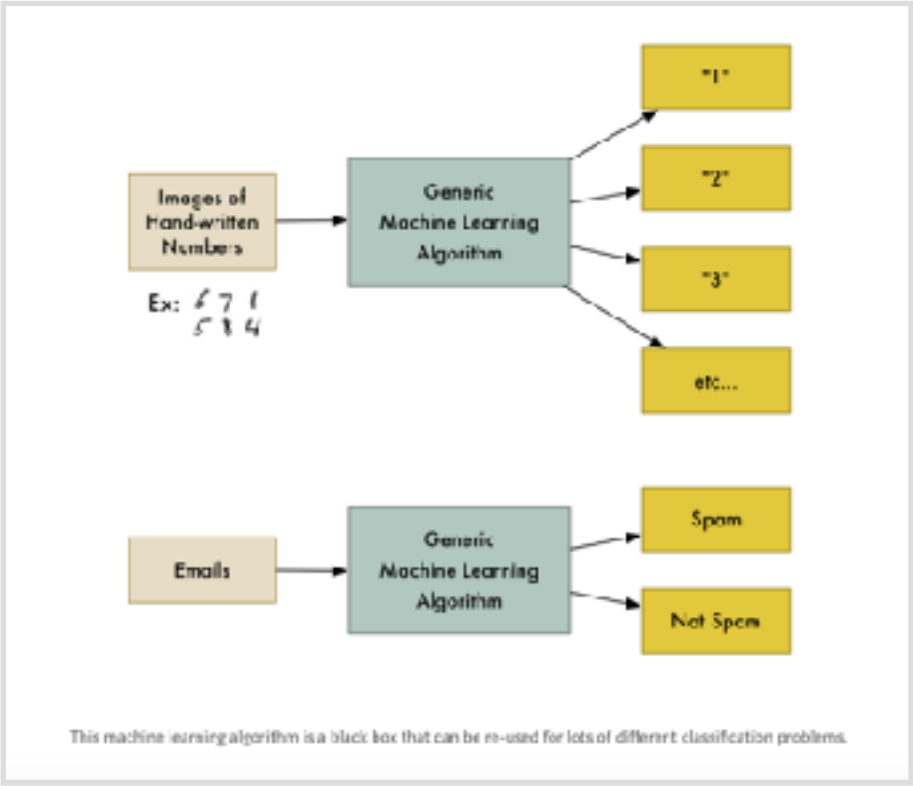


Figure 1: Generic algorithm example [7]

Machine learning algorithms are usually categorised as one of two fundamental group which are supervised and unsupervised learning [7]. There are also several other scenarios of machine learning algorithms but they will not be studied in this paper.

Supervised learning is an input-output mapping machine learning task based on trained set of input-output data [8]. In supervised learning, the output is result from the execution of functions that operate through labelled training data samples. To put it differently, the algorithm is aware of that goal is to figure out the connections between input and output which are predefined parameters [8]. In this relationship, an input is known as a vector and supervisory signal is the matched output [8]. Moreover, a supervised learning algorithm has the capability of composing inferred function that could also be used for new pairs of input-output data. For instance, face recognition in several applications such as Apple face ID or Facebook tagging features are great examples of applying supervised learning into practice. Given these points, the performance and accuracy of the applications using supervised learning algorithm undoubtedly depend on the size of data collections of the system [8]. In other words, supervised learning would be highly accurate if it contains vast training data elements.

Different from above introduced learning algorithm, unsupervised learning loads unlabelled training data to produce predictions for every undiscovered points that the system would encounter [9]. Due to none of the labelled data exist, it is challenging to get the significant measurement of the system's performance [9]. Taken an example of a clothing company having a giant database of customer information, there are plenty of difficulties that the company would try to resolve when reviewing database and trying to make profit out of it. The question would undoubtedly be how they could sell more. The answer for this is unsupervised learning. In other words, unsupervised learning would aid the company to determine market segments based on what it learns from the huge dataset of company's customers. Thus, the clothing company would sell more efficiently based on segmented market.

Ultimately, machine learning is not a software but it is the value that we get from the algorithms that we put there to train data [10]. The result of ML is depending on how human beings teach teach the system as well as how prosperous the data we provide to is [1]. More importantly, machine learning is indisputably an invention from human brainpower. In addition, as humans train the ML structure, there is a significant tendency that we shift our own biases to it [1]. As a result, if ML system generates an error, it is most likely because humans have transferred their own biases [10]. Therefore, it is recommended that developers always are careful with the algorithms taught to the machines.

2.3. Overview of existing applications

2.3.1. Augmented reality based applications

Augmented reality has already acquired a remarkable footprint in mobile development in the recent years. If AR in over a decade ago was solely a figment of human's imagination then at present time, AR applications are the striking and drawn a great deal of attentions from mobile users. Due to a fact that AR technology is strongly based on GPS and geolocation, augmented reality creates its own set of potential growing fields which can be listed as digital marketing, gaming, travel guide and education [11]. The figure below is an example of AR based application that can be used to boost up sales.

Figure 2: Ikea place application [12]

Digital marketing will have a brand new approach to the customers when establishing the connection of augmented reality. AR creates the robust and dynamic to the content of marketing application that buyers could take advantages while making decision on purchasing products [11]. With the release of iOS 11 and ARKit from Apple, the giant Swedish home merchandise IKEA has become the pioneer with regards to applying new innovation to enhances company's products and general retail experiences [12]. IKEA app provides a collections of over 2000 products including sofas, chairs, tables and so on [12]. In figure 2, the user sees an item through camera view with exact same

scale that the object has in real life. IKEA app would certainly enhance the furniture-purchasing experiences as well as reduce stresses for every customer while making the selection on suitable home appliances.

Figure 3: PokemonGo game on iOS [13]

In the world of gaming, it would be almost impossible to ignore the buildup of augmented reality. AR has flipped around the whole gaming industry with the arrivals of Pokemon Go as shown in figure 3 and AR Soccer. AR helps to games to be more appealing and also increases the numbers regarding retentions. [11]

Augmented reality implementation on travelling apps has a great potential. With the integration, it creates support to street signs translations in order to help travellers not to feel lost. Wikitude and Layar are the most popular AR programs that help to coordinate present reality components with inquiries to generate better outcomes. [11]

Education can be significantly reformed by AR integration. AR would be beneficial in making 3D pictures, historical stories remake, universal structure demonstration and significantly more in physical environment [11]. Augmented reality grants a blend of content with digital substances so that in this way giving undergraduates a more fascinating and immersive involvement and a more profound comprehension of subjects.

2.3.2. Machine learning based applications

By the time of writing this paper, the number of machine learning applications using Core ML are very limited. They often carry one main functionality which is real life recognition. Some limited amount of applications show up when searching in the store with key word of Core ML or machine learning. Some examples worth noticing are Dog Identify which identify dog breeds with camera lens using machine learning or DeepVision for Core ML which is a try out for new machine learning technology. As only limited results were found, it proves that the idea of machine learning based on mobile applications is still primitive.

3. Overview of technologies

3.1. AR Technologies

3.1.1. ARKit overview

Apple introduced ARKit which is a mobile framework for creating augmented reality applications for iOS. With ARKit, developers are not only provided from a straightforward interface to the impressive list of features but also powerful API. Moreover, ARKit is designed to support countless existing iOS devices including iPhone 6S and most of iOS 11 devices [14]. ARKit carries several core features that could be divided into three main layers in which tracking comes first.

Tracking is considered as the main functionality of the framework. By providing world tracking in ARKit, it is now able track user device's position with relation to physical environment in real time. Visual inertial odometry which using camera photos and in addition with motion information from the devices view are used in order to determine the location as well as orientation of the devices [15].

Furthermore, ARKit amazed people by its simple setup requirement of which there is no prerequisite knowledge, external configuration as well as extra sensors that are not built in the devices. Together with world tracking, scene understanding is the second core feature in the framework. Scene understand is the capability of detecting charac-

teristics and properties of the gadgets' surroundings. Scene understanding contains an ability such as plane detection. Plane detection could be explained as the ability to verify surfaces or plain in the real environment [15]. The room floor or a table are good example in this case.

Rendering is the last layer of ARKit. The framework supports simple integration with various kind of renderer. ARKit provides steady steam of camera pictures with motion data and also scene understanding that can be imported into renderer. If developers are familiar with SpriteKit or SceneKit, custom AR, which already contains most of the rendering, views are provided. Thus, ARKit is quite simple to begin development. Similarly, for those who is trying to achieve custom rendering, metal template is available in Xcode which enables ARKit integration into custom renderer [15].

3.1.2. Highlights about ARKit

The biggest advantage that ARKit owns is gigantic Apple's ecosystem with an estimated number of 500 million Apple devices that have capacity to run AR applications at the end of 2017 [16]. In addition, this number is expected to reach about 850 million by the end of 2020 [16]. ARKit delivery to the market indicates the tectonic move of Apple. Similarly as Google puts virtual reality in the hands of a large number of individuals with remarkable Google Cardboard, Apple has done the same strategic move for augmented reality expansion. While VR and AR idealists would not think about these gadgets as being genuine delegates of those technologies, it is true that no other frameworks or AR headset would gain this rapid reach. As a result, businesses doing AR cannot give ignore to ARKit.

Alongside with the market reach, developers are thrilled with technology used behind the framework. ARKit utilises local tracking system known as Visual-Inertial Odometry with simple 2D plain detection [17]. Cameras help to keep track of objects' orientation and layout and ARKit leverages the geometry and lightings of the scenes that are captured by the cameras [17]. Utilising this data, developers could add illustrations which stay settled on specific surfaces such as floor, ceiling, and table as the point of view of cameras while moving. VIO system keeps track of object's position in real time as it recalculates the object's illustration every frame stimulating on camera. These calculations are executed twice and at the same time [18]. The object's position is tracked by coordinating a point in reality to a pixel on camera sensor [18]. In the same way, the object's stand is watched by Inertial system including accelerometer and gyroscope which are known as Inertial Measurement Unit. The result is then passed through a

filter before being updated to the display by the ARKit. VIO gains that enormous power because IMU readings capability are roughly 1000 times every second and are accelerated based on user motion [18]. Given these points, VIO uses the information from camera and CoreMotion in order to make devices to gain superior comprehension of how it is moving inside the space with high precision and no extra calibration [17]. Thus, this gives support to keep motion tracking steady while objects moving fast and also creates a realistic feel of objects staying at the same position in the space.

The fundamental element that make ARKit so robust is scene understanding. Scene understanding includes plane detection. There is a compulsory requirement of having a surface to put the substance on, otherwise it would look like it is hovering around the space [18]. The optical system together with an algorithm will evaluate three points to define whether there is a real ground to place to object on. These points known as point cloud are utilised for optical tracking [18]. These points are also usually referred as sparse point clouds which utilise significantly less memory and processing time. Additionally, inertial system makes sure that the optical system could keep running normally with just a few points to track [18]. In addition, ARKit carries an advantage with scene understanding, namely light estimation. When browsing AR applications, it becomes obvious that lightings greatly contributes to the realistic of the content. With ARKit, camera sensor helps calculating the amount of omni-light [17]. From this result, the framework applies the adjustment for the lightings which would enhance the quality of virtual content. AR algorithm would be executed within milliseconds period of time with supported device's chips. For this reason, the more advanced AR algorithm operates, the less CPU power will be used, which allows to render better visual quality [17]. Last but not the least, in order to virtual objects at targeted location, scene understanding provides hit testing functionality. Hit testing helps to collect intersection points between physical world geology so that it would be possible to put virtual substances into real world.

What is more, ARKit provides support for Metal, SceneKit as well as third party 3D engines such as Unity and Unreal Engine which create a noteworthy visual precision [19]. Besides, ARKit requires A9 and up chipset to run on. As a result, scene understanding would be executed instantly and developers as well as 3D designers could construct detailed virtual objects to add onto applications. Consequently, this would make a better hardware performance with ARKit.

To summarise, ARKit is an advanced tool which enables developers to build augmented reality iOS applications. First, with world tracking, it generates the devices' relative position in relation with starting stage. Second, scene understanding supports to place

virtual objects into physical world. Scene understanding comes with plane detection as well as hit testing capability which find exact coordinates for the object. Furthermore, to enhance the fidelity of the virtual content, ARKit comes along with light estimation depending on camera view. Additionally, it is possible to integrate custom augmented reality content to integrate into the app using SpriteKit and SceneKit. Developers could also combine custom rendering engine a Metal template. Ultimately, ARKit is the key to open the intriguing world of augmented world.

3.1.3. Other AR technologies

There are plenty of prevailing frameworks that bring that possibility to create AR applications before ARKit. This paper will include a brief observation on the powers that those main applications could provide compared to ARKit.

First of all, Google turned down its popular product Tango to announce the ARCore last year [16]. It is a brand new technology that could be main competitor to ARKit. The announcement came just a few weeks after the ARKit release date. The giant tech company revealed that ARCore does not require any special hardware assets to run on and almost all of the recent Androids phone could operate with ARCore [16]. Some existing mobile phones such as Google Pixel or Samsung Galaxy 8 could run any ARCore applications if its operating system runs on Nougat or newer versions [16]. ARCore owns one significant advantage that ARKit could not has, that is it obtains all the successful technical factors of Tango as it has been developed for around 4 years. Yet, most of the augmented ability would be quite similar on both ARKit and ARCore. The distinction could only be recognised by mobile developers who have the direct connection to the frameworks. In order to get to know more about what ARCore is able to do, this document will go through the main features briefly. The first core feature of ARCore is motion tracking [20]. ARCore collects IMU data from the device's sensor to combine with necessary marks of the encompassing space to determine device's orientation as well as location even when it is moving. Together with motion tracking, ARCore has light estimation. The feature works as expected as in other frameworks. Ambient lighting data is gathered so as to ARCore could make the enhancement appropriately to the virtual objects. This process would help to adjust the visuality to look realistic. Furthermore, with ARCore virtual objects could be positioned accurately aided by anchoring objects feature [20].

Secondly, together with ARCore, Vuforia is a prominent SDK in the field of AR. Vuforia is almost the same as ARKit. However, there is one key thing that makes Vuforia slight-

ly easier for developers is that the SDK is compatible with iOS, Android and Windows devices [21]. Furthermore, Vuforia does not require the latest hardware to operate as needed in ARKit. In addition, besides the vastly usage of 2D marker in AR development, Vuforia SDK introduces objects recognition ability [22]. What makes this feature unique is that it enables the devices to identify and keep track of complex objects. The objects could be various such as toys, vehicles and many other merchandise products [22]. These objects could also be used as markers in the applications in order to connect with virtual objects. As a result, rich experiences are brought to the end users. However, Vuforia does come with a price for commercial distribution. It starts at $499 to get the classic offer of the software license [23]. Nevertheless, Vuforia is substantially dependent on printed marker to actually start off the augmented implementation. In other words, it is required to have printed photo to see the virtual AR models. Despite the fact that Vuforia SDK has 3D object recognition, the utilisation of 2D marker could turn out to be considerably restrictive when having the exhibition of the models out of the pre-defined environment [22]. The printed images likewise restrict the extent of the AR representation to the bounds of printed papers, and in the same manner keep from having consistent walkthrough shows without huge scale modern printed materials [22].

Overall, throughout the quick overview of ARKit and its alternatives, it is definitely important to identify the features that an application would build so that suitable SDK will be chosen. However in this project, ARKit is the best option to build the application both financially and technically.

3.2. Machine learning technologies

3.2.1. Core ML overview

Releasing at the same time with ARKit, Apple also introduced Core ML which is the machine learning framework. Perhaps this is the debut of the framework but the usage of it has already been wide spread in significants Apple's applications. Taking photos app as an example, the framework enables facial recognition and scene recognition [19]. Autocomplete or word prediction and smart responses use machine learning technology as well. Likewise, in Apple watch, it is applying the same technology to create handwriting recognition as well as smart responses [19].

Core ML is made to support iOS, macOS, watchOS and tvOS. It runs on iOS 11 and onwards. Core ML carries the power of artificial intelligence that let developers to cre-

ate various types of applications that anticipates users' needs such as facial as well as voice recognition based applications. Developers are now having a chances to integrate their own customised machine learnings models along side with trained models into the applications [19].

What is more, a quick recapitulation might be great for the decision making when choosing machine learning implementation method. Core ML is introduced to be the easies way to add machine learning to iOS apps. In fact, it is so easy that just a drag-and-drop in order to add pre-trained models into the app. In addition to the quick and simple setup, Core ML works completely on the user's device. All predictions are made locally based on imported pre-trained models. Moreover, Core ML supports various models from Caffe, Karas, scikit-learn and some others [24]. This is also can be seen as a minus point of Core ML. The reason is that, at the moment Core ML does not support any models from TensorFlow which the most prevalent machine learning engine for machine learning [24]. Lastly, Core ML is only available for iOS 11 and above with recently made devices due to hardware requirements

3.2.2. Highlights about Core ML

In WWDC 2017, Apple introduces the glittering ARKit framework. However, the highlight was truly about Core ML since it showed Apple biggest interest was artificial intelligence known as machine learning. It is widely believed that artificial intelligence is the software of the future. Core ML contains several huge advantages that are worth mentioning.

First of all, it is about user privacy [19]. This topic is now bigger than ever. The reason is that by the time of this paper is written, the giant software company Facebook has involved in a data scandal named Cambridge Analytical. In few words, this scandal was about collecting 87 million Facebook users' personal information and third party company known as Cambridge Analytical [25]. The company has obtained the data collection and created influences on political voters which would generate possible impacts for those politicians that hired the company service. Thus, what related to this event in this framework is that users' data such as personal location, images or text messages would not be sent to any cloud storage or server. This is because Core ML enables developers to create app that would work in offline mode.

What is more, all the predictions are done in the device locally and users would not need to pay additional fees to use this feature. Equally important, app owners or devel-

opers will not have to spend extra on server to run predictions. In addition, with the ability to run locally, this makes all apps stay independently from internet connection which would contribute significantly to the application performance.

3.2.3. Core ML and other machine learning technologies

At present, machine learning engineers have several ways to implement the technology into iOS applications. All the methods share the same fact which is only offering prediction ability. Most of the applications prepare own pre-trained data and then import it to the system. The traditional ways to train models is through cloud or offline. After having the training models ready to be used, developers could choose from Core ML, cloud service or third party APIs to start the app implementation process [26].

3.2.3.1. Cloud services

Artificial intelligence or machine learning particularly has been getting a great deal of interest from giant tech firms. Those companies include IBM, Google, Microsoft and Amazon and some others provide developers various options to work with machine learning such as IBM Watson, Google Cloud Vision, Microsoft Azure Cognitive Services, Amazon Rekognition and Clarifai [24]. All of these tools share a common advantage which is straightforward installation into the application. Furthermore, all cloud services make sure that developers would not have to worry about training the models because pre-trained models are available to be used. On the other hand, these services are not free of charge. Developers could start the trial with free tiers and after that it is approximately $1 for every 1000 requests made [27]. This is a significant aspect that affects developer's decision about whether choosing the cloud services or not. It is due to the fact that not every developer or company has a solid financial support for to acquire the commercial license of the cloud services. Likewise, when working with cloud service, it is clearly that developers are not able to perform addition modifications locally because predictions and results are made by posting and getting network requests to cloud's servers [27]. Overall, if there are available models provided by those services, it is assuredly that developers shall consider using the service because usually the supported models are well managed and maintained by the providers.

3.2.3.2. Local training

In the first place, TensorFlow would probably be named due to its popularity in the market. The tool is developed by the giant tech, Google [28]. TensorFlow is genuinely simple to export models and import into application. The significant advantage that developers get from the tool is a numerous collection of pre-trained models [26]. With the diverse in pre-trained models, machine learning engineers have better chances to choose ones that fit their goals most. Conversely, when running on iOS applications, TensorFlow does not utilise GPU but only CPU of the device [26]. With this in mind, most of the basic models would work fine but it would be problematic when comes to deep learning [26]. For the current support for iOS, it does not contain all capabilities which would result a problem when including graph in the app. Additionally, TensorFlow API is written in C++ so it would be necessary to know some understandings about Objective-C to start the development [26]. However, TensorFlow just released a new Swift version for iOS [29]. Even-though, it is at its early stage and actually unstable, Swift developers shall be cheerful and patient until more changes come to the API so it gets more reliable. Last but not the least, the library could possibly take up to 40 MB in the app bundle [26]. Altogether, TensorFlow is a great option for machine learning engineers looking for a tool to start their development.

Comparatively, Caffe is a well-known deep learning framework which is used by researchers. The framework offers plenty of pre-trained models under its own format. The Caffe framework shares a significant commons for both advantages and disadvantages of TensorFlow.

As a matter of fact, Turi Create stands out from the list of technologies offering local training. Firstly, Turi Create genuinely has a straightforward setup and high compatibility for Core ML model exportation. Additionally, Apple has acquired Turi in 2016 and set Turi Create to be open source the year later [30]. The generous act from Apple would help developers significantly in training process because not only they could freely customise their models but they would not concern about the cost to generate models [31]. Considered aforementioned attributes about Turi Create and other technologies, Turi Create has been chosen for training Core ML model used in iOS application.

4. Application's specifications

4.1. Application's concept

The main idea of the application is about to putting the most mentioned technology of 2017 into practice. The software will take advantage of two brand new machine learning frameworks and one augmented reality framework which were released by Apple in the fall of 2017. The list of frameworks includes Core ML and ARKit. The result of this work is a mobile applications which only runs on iOS devices like iPhones or iPads. The application goals are simply about:

- Utilise ARKit and Core ML in the application.
- Create more engaging and attractive educational content for application.
- Translate application's content into target languages.
- Open up a new approach on how teaching and learning experiences.

4.2. Applications' target users

The potential target users of the applications are youngsters above the age of six. The reason why the application requires users that are over six years old is because from this age youngsters will start learning English in school. Besides traditional way of learning a foreign language particularly English in this case, the app creates a brand new approach on how lecturers teach a new language to their students. In addition, primary students are indeed the most promising target group but other groups of users are also appreciated to check out the applications.

Together with the first target group, the application also aim to serve elderly people with age above 60 years old that are retired and want to explore modern technology as well as having side hobby of learning new languages. It is a fact that this group of people will have plenty of free time after retiring from the workplace. Thus, the application creates the environment for them to try out new technology as well as let them learn new languages while having fun using the app. This idea is not only to kill time but also to prepare themselves to have better understandings of different languages when travelling.

4.3. Application's functionalities

The application consists of three core features. The first capability that application offers is image recognition. With the utilisation of Core ML, users would be able to request predictions about particular species based on their photos collection. In the same way of taking advantages from Core ML, second feature enables end users to interact with what actually happens on the camera in real time. The result of interaction would be demonstrated by augmented reality. Lastly, results collected from sessions in feature one and two would construct a list where later user would browse to gain further information about animals.

4.4. Technical requirements

The target platform of the app is iOS. To develop iOS applications, it is required to have a computer which runs OS X operating system such as MacBook or iMac. Operating system also is required to be Sierra 10.13 or above on OS X and 11.0 or later on iOS to be able to start the development. The reasons behind this specifications of development tools are because ARKit and Core ML frameworks run on Apple devices that have A9, A10 and A11 processors [14]. The following devices use these processors:

- iPhone 7, 7 plus.
- iPhone 8, 8 plus.
- iPhone X, XS, XSMax
- 9.7-inch iPad Pro (2016 model).
- 9.7 -inch iPad (2017 model).
- 10.5 -inch iPad Pro.
- 12.9 -inch iPad Pro (2016 model).
- 12.9 -inch iPad Pro (2017 model).

Listing 1. List of devices could run the operations

In addition, Xcode is needed to be installed as the developer tool. The latest version of Xcode which is number 9 is desired due to frameworks' specifications [32]. Objective-C and Swift are two options for the programming languages. This project will choose latest Swift version 4.0 as the programming language due to creator's preference. When the application is at ready state, Apple developer program is needed in order to distribute the app to app store [33].

5. Camera translator: The application implementation

One of the application objectives is to create an intelligent application by utilising the benefits of machine learning. Thus, to achieve this goal, the application requires to have a trained Core ML model in order to be imported and later used with Core ML framework. In other words, before starting iOS development, it is necessary to acquire a Core ML model by training. Local training was carried out in order to obtain an operative model that is capable to work well with application targeted objects. After model is generated, it would be transferred to the iOS application to serve Core ML related tasks.

5.1. Training Core ML model with Turi Create

The objective of Turi Create is to enable developers to add experiences to user. For instance, an athlete wants to take a picture of his meal by selecting distinctive food to perceive the amount of calories that he is devouring. Similarly, a person could take control of a lamp with straightforward gestures. Likewise, an application to recognise the breed of a dog in real life. These aforementioned examples share the same fact. They utilise machine learning to achieve the result. Above experiences would indeed need rather little data to make. Every model in the examples was created with Turi Create which comes with Core ML model type.

Turi Create is written in Python that gives app developers ability to apply machine learning solutions. Turi Create is available for Mac and Linux users [34]. More importantly, Turi Create is open source which has a repository on GitHub [34]. The machine learning platform contains several outstanding features. They are:

- Recommender
- Image Classification
- Object Detection
- Style Transfer
- Activity Classification
- Classifiers
- Regression
- Clustering
- Text Classifier

Listing 2: Turi Create ML tasks [34]

This paper contains only details for image classification because of it was selected as the core machine learning task of the application. However other tasks of the platform could be found from its official webpage. Taking an image from a set of sample image collection, the objective of image classifier would be assigning the picture to a pre-defined label [35]. In the recent reports, deep learnings have proved to have superb results on the problem [35]. However, it faces several difficulties to do training such as extraordinary sensitivity as well as lengthy training period [35]. In other words, the time spent on testing various model settings could be months and the result might not be worth it. However, Turi Create has come up with image classifier solution which is intended to reduce these agonies, and giving it a higher possibility to make brilliant images classifier model [35].

In Camera Translator application, Turi Create version 5.0 is used. MacOS 0.13 and above is required to run version 5.0. In addition, under this release, Turi Create automatically uses GPUs for image classification task. More importantly, several system prerequisites need to be verified. The training machine used Python, for example 2.7, x86_64 for architecture and 16 GB of RAM and 4 GB was the minimum requirement. For this application, local machine used Anaconda to create virtual environment. Anaconda is an open source distribution built of R and Python programming languages for developers to develop, automate and manage machine learning as well as artificial intelligence from developments to productions [36]. Up to this stage, the laptop is ready to install Turi Create in the virtual environment.

```
conda create -n venv python=2.7 anaconda
source activate venv
```

Figure 4: Command to install virtual environment [34]

Figure 4 shows steps to create virtual environment in local machine. After the environment is created, Turi Create is installed using command demonstrated in below figure.

```
(venv) pip install -U turicreate
```

Figure 5: Install Turi Create [34]

After getting the machine ready to train model, the next step is to prepare animals images to fetch to the system. Several methods are used in order to acquire enough amount of images for training in this application. Google images and image-net.org are among those and they contribute substantial part in the collection. The collection contains 59 species that are currently inhabiting at Korkeasaari zoo in Helsinki. The

species names are collected manually by fetching data from the official website of Korkeasaari zoo. The size of the dataset is over 1 GB. Each species contains at least 100 photos and up to over 1000 picture for the largest folder. The list of species that are used to train model in the application is shown in figure 6.

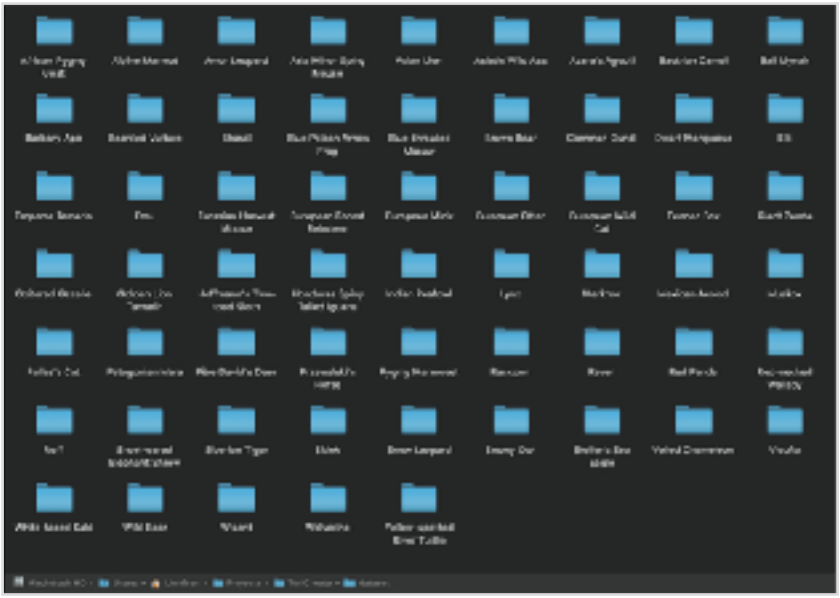


Figure 6: List of species that are used for training model in application

```
import turicreate as tc
import os
# 1. Load images
data = tc.image_analysis.load_images('dataset/', with_path=True)
# 2. From the path-name, create a label column
# Note: This is to extract folder with more than two classes
data['label'] = data['path'].apply(lambda path: os.path.basename(os.-
path.split(path)[0]))
# 3. Save the data for future use
data.save('animals.sframe')
# 4. Explore interactively
data.explore()
# 5. Load the data
data = tc.SFrame('animals.sframe')
# 6. Make a train-test split
train_data, test_data = data.random_split(0.8)
# 7. Create the model
```

```
model = tc.image_classifier.create(train_data, target='label',max_it-
erations=50, model='resnet-50')
# 8. Save predictions to an SArray
predictions = model.predict(test_data)
# 9. Evaluate the model and save the results into a dictionary
metrics = model.evaluate(test_data)
print(metrics['accuracy'])
# 10. Save the model for later use in Turi Create
model.save('animal.model')
# 11. Export for use in Core ML
model.export_coreml('AnimalImageClassifier.mlmodel')
```

Listing 3: Steps to train Core ML model with Turi Create

By the time, all prerequisites are ready for the procedure to move on to training a Core ML model. From listing 3, it is clear that in order to generate a model for this project it has to go through 11 steps. First and foremost, it needs to fetch images into the system. To be clear, in this figure, it indicates that the images collection folder named dataset is located at the same directory as well as level of the TuriCreate.py file. Next, from the sub-folders of dataset, it creates a label column according to the path name. Third step is to save the extracted data into SFrame format. Tables are clear configuration to utilise when gathering data in preparation for complex data analysis [37]. SFrame is invented for this purpose to work with Turi Create. SFrame is intended to expand to datasets significantly larger than memory space [37]. After that, it saves generated SFrame at step 3 and interactive exploration at following step. In a moment, data will be loaded again from the SFrame file that was saved previously. At step 6, the loaded SFrame data is split into two randomly by a decimal parameter ranging from 0 to 1 [38]. The fraction is used to divide the data into two parts in order to create training and testing data groups. In addition, it is possible to set an exact value for the separation by passing true for exact parameter [38]. In this Camera Translator application, 0.8 was set for the value of random split which means that 80 percent of the original data is used for training and the rest is used for testing. Then, an image classifier model is created by calling create function at step 7. The function takes training set from previous split result as input data. Second parameter is so called target and this parameter would point to the column of labels created at the beginning [39]. Third requirement to input into the function is a model type. This is a text value that presents a pre-trained model so as to start image classifier process [39]. Several values are available to choose for this spot. It would depend on the scale and usage of applications to select

this value. Camera Translator used resnet-50 as the pre-trained model. With this type of model, the result for Core ML model would be greater than 90 MB [39]. Other options are squeeze net_v1.1 and VisionFeaturePrint_Screen that would generate a Core ML model of an approximate size of 4.9 MB and 41 KB respectively [39]. Last parameter used in the model creation is the value for maximum iterations named max_iterations. By default, the value is 10 but it is suggested that the higher the value is the more accurate model it would generate. Thus, it is worth considering that when the accuracy result is low, the value of passes should be greater [39]. In the next phase, Turi Create provides capability to gives predictions over given dataset. The function utilises logistic regression model to construct predictions [40]. The predictions could be created as class names, probabilities that the objective result is correct [40]. After predictions are made, the procedure continues its way to evaluation stage where it evaluates the recent generated results [41]. The input of this evaluation function is the test data that were created aside with train data earlier. As a result, the function output is a collection of evaluation values where each of those has key name of accuracy and a number indicating evaluation grade [41]. Steps 10 and 11 do saving model and exporting Core ML model for iOS implementation.

```
# Use all GPUs (default)
turicreate.config.set_num_gpus(-1)

# Use only 1 GPU
turicreate.config.set_num_gpus(1)

# Use CPU
turicreate.config.set_num_gpus(0)
```

Figure 7: Sample code for utilising GPUs for training [42]

Turi Create provides an advanced option for users to utilise GPUs power while training CoreML model [42]. This would make training an image classifier model swifter. Since the machine, used for training model in this project, runs on macOS 10.14 which is higher than the minimum requirement of 10.13 so Turi Create would automatically exploit power from GPUs for training process [42]. Figure 7 presents a sample code showing how to utilise GPUs for training with Turi Create.

5.2. Demonstration of image classifier model training

Figure 8. Screenshot of virtual environment activation and start of training.

```
[LinhTran@Linhs-MacBook-Pro-2 TuriCreate $ source activate venv
[(venv) LinhTran@Linhs-MacBook-Pro-2 TuriCreate $ python TuriCreate.py
```

Figure 8 indicates that virtual environment needs to be activated before running Turi Create model training.

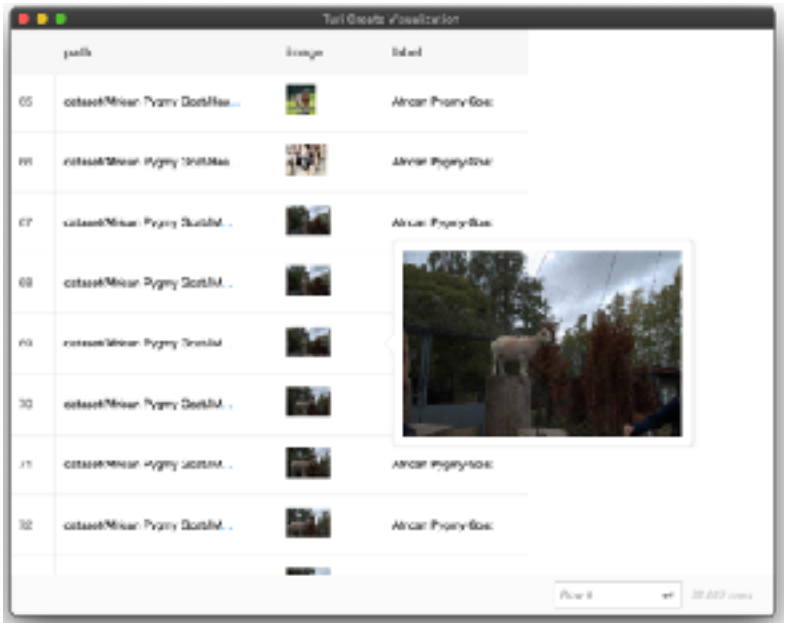


Figure 9: Screenshot of visualisation when running the explore function with SFrame data.

Figure 9 clearly indicates the each folder or image would link to a label which is the name of the class or the animal source folder.

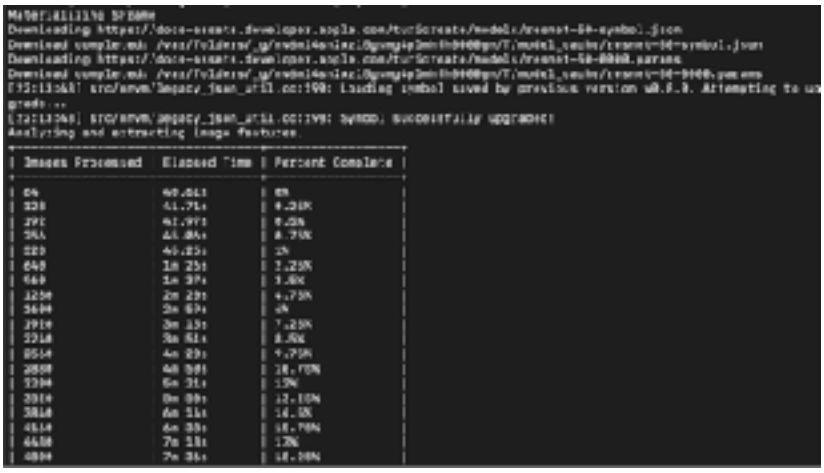


Figure 10: Screenshot of analysing and extracting image features.

Figure 10 shows the analysing and extracting image features processes. In addition, it indicates the chosen resnet-50 trained model is downloaded from developer source.

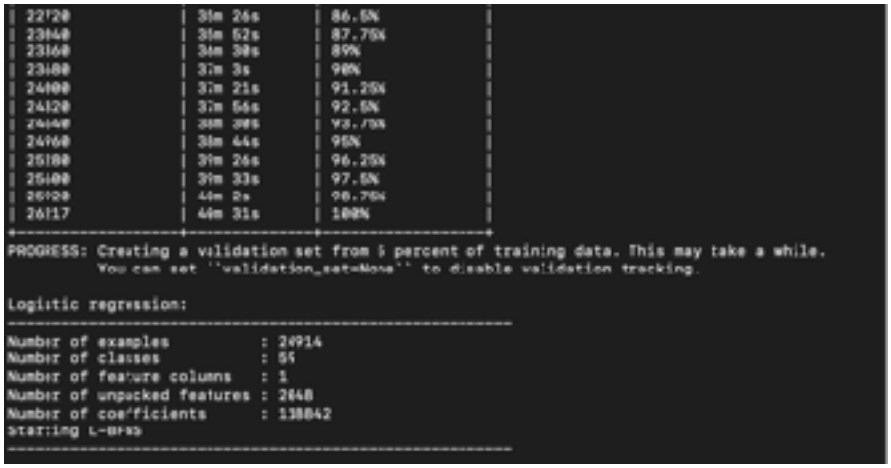


Figure 11: Screenshot of image classifier model creation statistics.

Analysing images and creating image classifier models is a hard job, as can be seen in Figure 11 has proved that. In order to go through training data of 59 classes, it takes 40 minutes and 31 seconds. Besides, various numbers are listed in the statistic such as total number of images is 24914 and amount for feature columns is 1. However, there is a note named progress containing message about validation set. Validation_set is one the image classifier creation function. By default, this value is set to be auto which could be understood as extra metrics calculation for measuring generalisation performance of the model [39]. If this argument is set to none, it would have additional computation otherwise the validation dataset needs to be similar as training data [39].

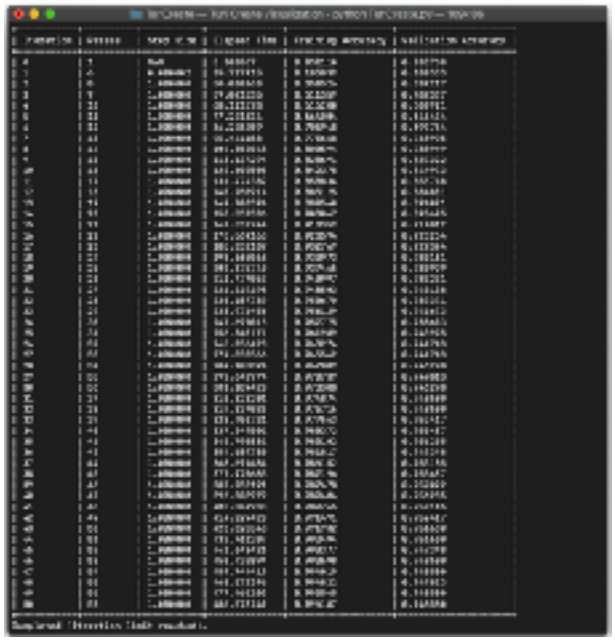


Figure 12: Screenshot of iteration statistics.

Figure 12 contains two important numbers that are training accuracy and validation accuracy. It can be seen that throughout iteration cycles, the accuracy of both training and validation has risen up significantly. Certainly the higher values those attributes have, the better chances the final model would have when testing on application.

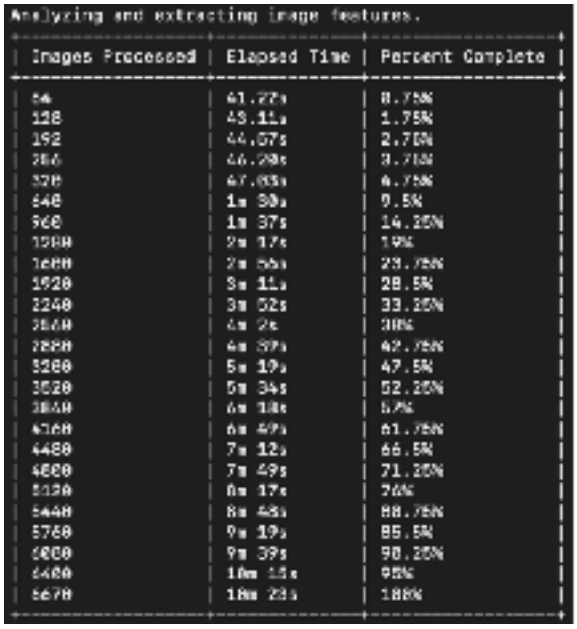


Figure 13: Screenshot of analysing and extracting image feature for Core ML model exportation

Figure 13 demonstrates the final step before Core ML model gets exported. With 59 classes, it takes totally 10 minutes and 23 seconds for analysis and feature exportation.

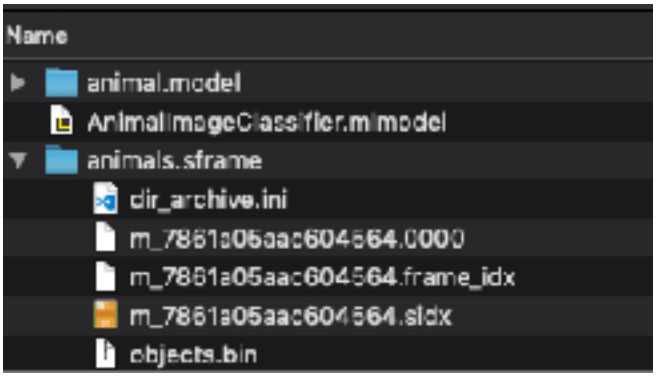


Figure 14: Screenshot of exported Core ML image classifier model together with SFrame data

It can be seen from figure 14 that the Core ML model is named as AnimalImageClassifier with mlmodel extension. The size of the model is about 94.7 MB. Up to this stage, training a Core ML model with Turi Create has finished and it is ready to move to iOS implementation.

5.3. iOS application implementation

First and foremost, the iOS application will be live in app store under the name of Camera Translator. In order to release Camera Translator to the store, an Apple developer program membership has to be acquired earlier [43]. Application details need to be fulfilled in app store information section from app store connect page. With paid membership being acquired, it is ready to start practical implementations. Camera Translator contains three core features. Each of them will be thoroughly examined below.

The application is constructed from tab bar controller where it connects three main features in the app. The tab bar is illustrated as UIViewController in diagram shown in appendix 1. Every feature has one class managing all operations. For example, as demonstrated in appendix 1, ImageRecognitionViewController is responsible for first feature of the app which does image recognition. ObjectDetectingViewController and AnimalRecordsTableViewController operate camera translator and history features respectively. Beside these three classes, multiples supporting classes as well as set of methods are utilised in order to make the app operate as expected. Taking ImageRecognitionViewController as an example, the class conforms methods of UIImagePickerControllerDelegate to create image selection functionality. Similarly, in ObjectDetectingViewController, it used ARSKViewDelegate methods to perform AR related tasks. Last but not the least, AnimalRecordsTableViewController consists of various delegation methods such as UITableViewDelegate and UITableViewDataSource to construct the list view. A custom class named RecordedAnimalsManager is initiated to manage creating and getting encountered animal records. Other sub-classes which are responsible for background and user interface would not be studied completely in this document. The paper endeavours to explain comprehensively how core features are built.

5.3.1. Image recognition - First feature

The first introduced feature has a clear self-explanatory name. The idea behind this component is to enable end users to select images taken earlier inside the application. With selected photo, application would give predictions based on trained data collection.

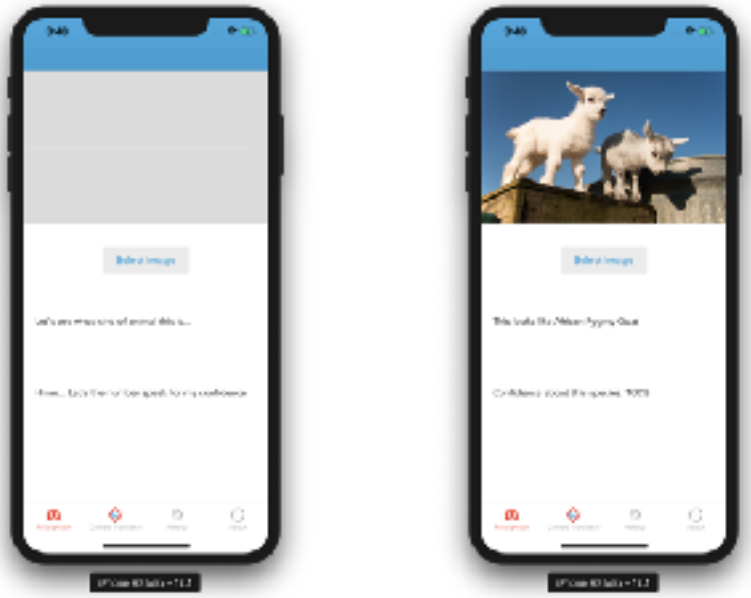


Figure 15: Screenshot of image recognition feature

Above figure is the before and after an image is loaded into the application. By tapping on select image button on the screen, users' photo library would show up as the source of selection. In the example demonstrated by figure 15, it can be seen that if her a photo is selected and loaded into Camera Translator, the result of sample photo turned to be African pygmy goat. Besides, in order to prove its confidence of prediction, a value in percentage is also given out for confident measurement. In this case, the target is surely confirmed as African pygmy goat. To understand better about the steps of converting an input image into a prediction result, below figure would help to demonstrate the process.

Figure 16: Screenshot of main function used for processing images

Figure 16 demonstrates processing images pulled out from photo library. As shown, the function takes one input which is a photo with type of CIImage. Following steps are extracting Core ML model from classifier and dispatch a Core ML request. Detail about these steps will be explained in the next part of this paper.

This was not the only successful testing case to prove that this feature would work properly under production. A great amount of tests were drawn during the development period as well as testing phases. Results gathered from these tests were carefully studied in order to make improvements for recognition.

5.3.2. Camera translator - Second feature

Second feature is named after application name. It might have indicated its role playing in the application. This feature is certainly most complex and extraordinary component that exists in the app. Both ARKit and Core ML frameworks are utilised in the second feature. Under camera translator feature, it is decided that the application would take advantage of built in camera of the phone in order to capture live image session. However, during development period, there were several methods taken out in order to capture live session. Among various solutions, AVCaptureOutput from AVKit and ARSKView from ARKit were selected to use for development and production. The reasons would be explained along with summary of how this feature is constructed. When opening second feature from tab bar at the bottom of the screen, phone camera would

open up and show live session on the screen. Below feature will demonstrate how the view in real life.

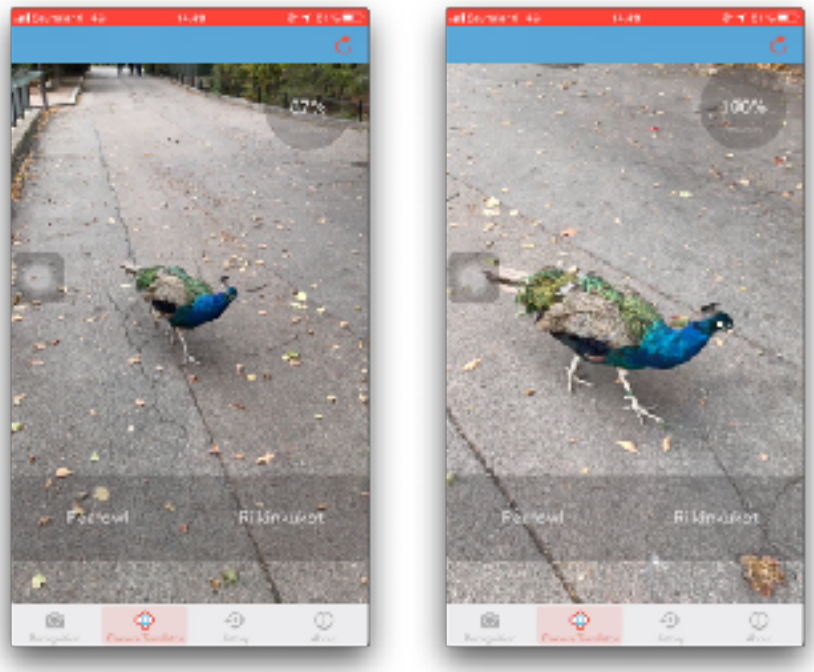
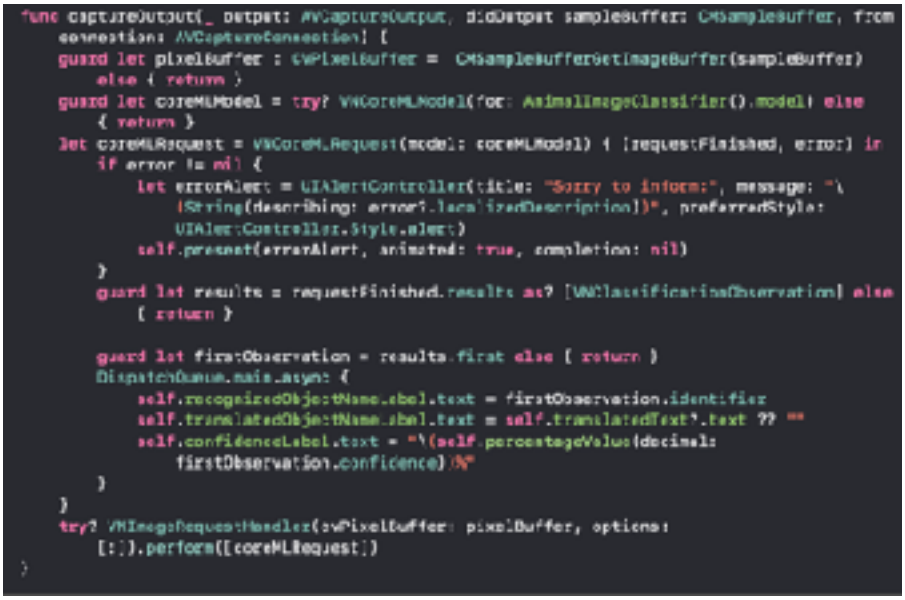


Figure 17: Screenshot live session captured using AVKit

Figure 17 is in fact cut from a screen recorded video from a field trip to Korkeasaari zoo during last phase of development. As demonstrated in figure 17, second feature opened up in a live camera view where it actually contains a few components. From top to bottom, it can be seen that there is reloading icon on top right corner of the screen. This is an action button which is responsible for restart the entire running session on this view. Prediction confidence number is laid below with a descriptive label named accuracy. The result of the prediction is constructed by two labels. The first is in English and the other is in Finnish. All the texts are placed on a transparent grey background in order to maximise the area of live camera view.

As aforementioned, the live camera session in Figure 17 is from AVKit approach. The reason behind this approach was purely about the productivity of testing during the field trip. With AVKit framework, it enables the application to capture every frame from the camera view and later transporting to a Core ML function to process the frame.



```
func captureOutput(_ output: AVCaptureOutput, didOutput sampleBuffer: CMSampleBuffer, from
    connection: AVCaptureConnection) {
    guard let pixelBuffer : CVPixelBuffer = CMSampleBufferGetImageBuffer(sampleBuffer)
        else { return }
    guard let coreMLModel = try? VNCoreMLModel(for: AnimalImageClassifier().model) else
        { return }
    let coreMLRequest = VNCoreMLRequest(model: coreMLModel) { [requestFinished, error] in
        if error != nil {
            let errorAlert = UIAlertController(title: "Sorry to inform:", message: "\
                (String(describing: error?.localizedDescription))", preferredStyle:
                UIAlertController.Style.alert)
            self.present(errorAlert, animated: true, completion: nil)
        }
        guard let results = requestFinished.results as? [VNClassificationObservation] else
            { return }

        guard let firstObservation = results.first else { return }
        DispatchQueue.main.async {
            self.recognizedObjectName.abel.text = firstObservation.identifier
            self.translatedObjectName.abel.text = self.translatedText?.text ?? ""
            self.confidenceLabel.text = "\(self.percentageValue(decimal:
                firstObservation.confidence))%"
        }
    }
    try? VNImageRequestHandler(cvPixelBuffer: pixelBuffer, options:
        [:]).perform([coreMLRequest])
}
```

Figure 18: Screenshot of AVKit implementation in the application

The above figure shown in detail how live session captured by built in camera is being processed. Firstly, taking advantage one of AVKit framework methods, a CMSample-Buffer data is returned from the captured data. It is then converted into a CVPixelBuffer instance type. Secondly, an instance of Core ML model is extracted from AnimalIm-ageClassifier class which was the result from Turi Create training. Extracted model is later used as input to create a VNCoreMLRequest instance which is dedicated for im-ages analysis using Core ML model [44]. The request has a completion handler where it contains a result and an error. The result in this case is an array of VNClassification-Observation type which include the information of classification generated from analy-sis request. On the other hand, error would consist of request failure response where it explained why the request did not succeed. If the error variable is not empty, applica-tion would prompt an alert to user indicating that the session has failed with an expla-nation text. Surely, the application is designed to run through the prediction successful-ly. Thus, when the array of classifications is produced, the first observation is taken as the final prediction because the most accurate classification would be listed in front of others. With produced classification, the name of the predicted species is extracted from identifier attribute and accuracy value is generated from confidence element. By the time, necessary ingredients are collected and ready to show on the screen. Main thread is now intercepted in order to make the needed update. However, before all of this could happen, the Core ML request needs to be dispatched. The very last piece of code in figure 18 has fired the action.

Given these points, logic behind Figure 17 will be well explained. However, there was not any justification for the difference in confidence value between the two examples in Figure 17. Taking consideration that these screenshots are from Korkeasaari testing trip, images were produced in real time with movements and various environmental and physical impacts such as lightings and distance between target to camera in this case. Thus, from left to right, it could be noticed that the peafowl on the scene was moving closer toward the camera. As a result, better frames were captured by camera. This also means that more details were collected from the subject such as shape, colours and size. Consequently, the higher accuracy that application would give to classify the subject. Given these point, difference between accuracy value in Figure 17 is now reasonable. Utilising AVKit helps to minimise a great amount of time testing on individual species. Since with this implementation, it allows application to run image analysis continuously without having user to make interaction with the app. Even-though current operation returns rather valuable result, it was not chosen as for release state.

Figure 19: Screenshot of second feature at its final stage

As shown above, figure 19 demonstrates the final view of the second feature in the application. From the view, it can be seen that there is only one change between development and production stage. Notably, a floating text indicating name of species is now added. Although the rest of the view looks similar to the prior stage, the live session is now integrated with ARKit.

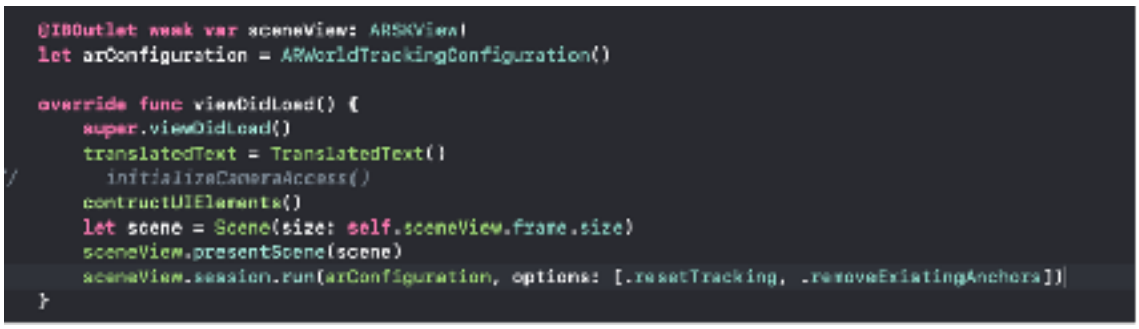


Figure 20: Screenshot for initialisation of AR scene view

As shown in figure 20, an IBOutlet variable is created with type of ARSKView which would be responsible for AR session. In addition, an instance type of SKScene is created in order to display floating text nodes. Other user interface elements such as name and accuracy labels are produced by constructUIElements() function. Later, AR session is created and operated by ARWorldTrackingConfiguration setting along with options of reseting all tracking as well as removing all existing anchors.

Figure 21: Screenshot for ARSKView delegate methods

After AR session is configured, ARSKView delegate methods are obligatory added to view controller. These methods from figure 21 would construct nodes that is responsible for showing hovering text shown in figure 19. Appearance of the text would also be configured at this stage.

Figure 22: Screenshot for handling user interaction while using second feature

With these given points, there is one last preparation to make. In this feature, user would have the ability to tap on the screen and the application would return a co-respondent result. In this case, if user tapped on a particular area on the screen, this area would be sent for further investigation. For example, figure 19 has demonstrated that a tap was made where the raccoon stays. A frame generated from the tap action is now used us input for image analysis. After that, similar processes would take place for creating and dispatching Core ML request. However, an additional operation is added. In order to display the hovering text, an ARAnchor instance is initialised. The anchor is created based on the location of the tap action. Thus, when the result is ready, it would be shown up on the exact location where user has tapped on. Lastly, after every prediction is made, an instance type of Animal class is produced. This instance holds four values that would be used later in the third feature. These values are name of species in English, Finnish, confidence or accuracy value and the date of prediction creation.

5.3.3. History - Last feature

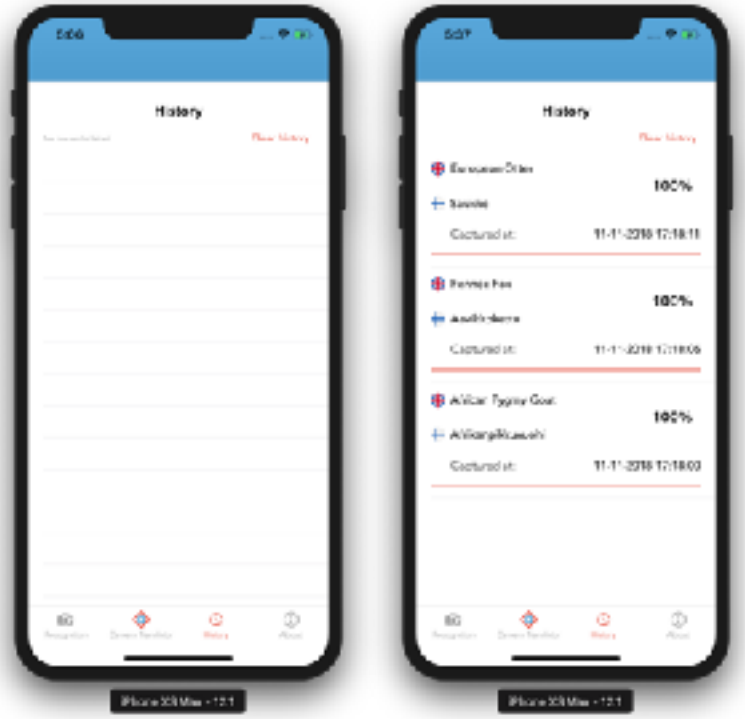


Figure 23: Screenshot of list view of encountered animal species

The last feature of Camera Translator is basically a list. The view is constructed from single cell item. Each cell represents for a record containing information about encountered animal as well as captured date. Obviously, when the application first started, the list would be empty and a text indication of non records exist. After a few tests, the list would be filled quickly as shown in the right of figure 23. In addition, language flags are also added for better indication of species name. In detail, they are English and Finnish flags in figure 23. The intention of this list view is to allow end users to review which species they have found of all time. However, the list is in fact possible to be re-created by tapping on most right corner button which has a self explanatory name. List of recorded animals is not the only information that the History feature brings. Tapping on single item of the list would reveal additional information.

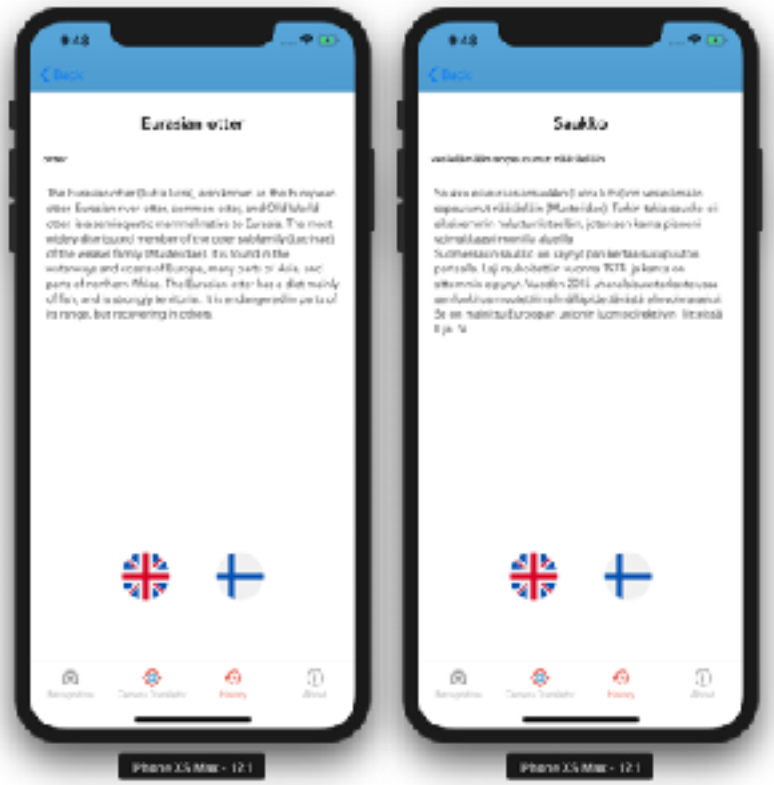


Figure 24: Screenshot of additional information of single item on History list view

In figure 24, it can be seen that the view is built mainly with text. On the top level, it shows the name of the species. Supplementary information is added below. Two flags are actually interactive items that would allow users to select preferred language. English and Finnish are supported up to this point. All additional details on this view are extracted from Wikipedia.

Lastly, the fourth item on the bottom tab bar is an information view where general details about features in the application are given.

6. Feedback from end users and discussion

6.1. Feedback from users

A demo day was carried out with pupils in Munkkiniemi elementary school or Munkkiniemen ala-asteen koulu in Finnish. Pupils were at age of 11 and could speak fluent English. A group of 13 pupils was gathered for the session. A brief introduction about the application was given to the students. After that four small groups were formed and each group was provided testing phones such as two iPhone 7, iPhone 7 Plus and iPhone XSMax. Printed photos were used as materials for testing. Ten different kinds of species were prepared and they were all inhabiting in Helsinki Korkeasaari zoo. Throughout the session, five distinctive animals were successfully experienced with the Camera Translator application. Most of the students expressed huge excitement during the show.

Question	Yes	No	Feature 1	Feature 2	Feature 3
1	13/13	0/13	X	X	X
2	7/13	6/13	X	X	X
3	12/13	1/13	X	X	X
4	X	X	5/13	5/13	2/13
5	13/13	0/13	X	X	X
6	12/13	0/13	X	X	X

Table 1: Collected results from questionnaire

A questionnaire described in appendix 2 was given to all attended pupils and the results are collected and being constructed systematically in table 1. As can be seen from table 1, the testing ended with spectacular result from questionnaire carried out by the students. From the collected statistics, all of pupils have used iPhone or iPad previously. About 55 percent of students answered that they have used similar kind of application before. The application has proved to have excellent user experience, over 93 percent of the students thought the app was easy to use. It was a fair result when prompting pupils about the most favourite feature of the app. Both first and second fea-

tures shared an equal number of over 39 percent. One student which equals to about 7 percent of students, preferred both features. As result, it would leave about 15 percent for the History feature. Surprisingly the students gave out an absolute 100 percent positive answer when being asked if the application was educational and informative about the animals. Lastly, Camera Translator received and outstanding figure of about 93 percent vote for recommendation when users visit Korkeasaari zoo. The rest of the group was indecisive about this question. With all numbers considered, the session was outstanding and obviously gave a significantly positive push to put the app to store as well as to continue with improvements and further developments.

6.2. Discussion

During development period, several approaches for training Core ML model were carried out. Create ML which is a newly introduced Apple framework for training machine learning model, was used. However, because the framework was on its early release state, it was not fully compatible and completed for developers to use. Besides Create ML, IBM Watson Services which is a cloud tool, was also selected for the training task. With IBM Watson Services, the training had undergone for a short period of time. It resulted in a rather successful trained model. However, cloud service always comes with a certain price. This aspect has a significant impact on the capability of the project. In other words, it is financially out reach for the project. Eventually, Turi Create was chosen for the training work.

7. Conclusion

The original intention for creating a new educational app in this project was achieved. With the help of this thesis project, principles of machine learning and augmented reality were studied. In practice, recently introduced Apple technologies for augmented reality and machine learning were utilised effectively in the application created in this project. Especially with machine learning tasks, the results were surprisingly accurate. Equally important, Turi Create played a crucial role behind the entire operation. Moreover, selecting Turi Create for training task seemed to give an easy option to scale the animals' collection freely. All the training work was done locally. However, collecting material for training could be improved. Due to the fact that only the author of this paper worked on this project, it was time-consuming to collect the images. The preparation process could have been carried out more effectively had there been a team of researchers.

Positive user feedback was collected at the final phase of this project and the goal of this study as well as practical project have been achieved. As a result of this study, a Camera Translator was successfully developed, and it is ready to be launched into the market.

References

1. Siilasmaa R. Why you should study AI and Machine Learning and how I did it [Internet]. Nokia Blog, 2017 Nov. Available from: https://www.nokia.com/blog/study-ai-machine-learning/

2. Kesim M, Ozarslan Y. Augmented reality in education: current technologies and the potential for education. Procedia - Social and Behavioural Sciences 47, 2012, p. 297 - 302

3. Azuma R. A Survey of Augmented Reality. In Presence: Teleoperators and Virtual Environments 6, 4; 1997 Aug, p. 355-385.

4. AP/Reuters. Facebook says up to 87m people affected in Cambridge Analytica data-mining scandal [Internet]. ABC news. Available from: https://www.abc.net.au/news/2018-04-05/facebook-raises-cambridge-analytica-estimates/9620652

5. Copeland B.J. Artificial intelligence [Internet]. Encyclopaedia Britannica. Available from: https://www.britannica.com/technology/artificial-intelligence#ref219087

6. A Brief Introduction to Artificial Intelligence [Internet]. Artificial Intelligence. Eduonix; 2018 Jan. Available from: https://blog.eduonix.com/artificial-intelligence/brief-introduction-artificial-intelligence/

7. Geitgey A. Machince Learning is Fun! [Internet]. A Medium Corporation; 2014 May. Available from: https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471

8. Patel H. Machine Learning Introduction: Supervised vz Unsupervised (Part1) [Internet]. Towards Data Science; 2017 May. Available from: https://towardsdatascience.com/machine-learning-introduction-supervised-vs-unsupervised-part-1-6058bce829d3

9. Mohri M, Rostamizadeh A, Talwalkar A. Foundation of Machine Learning |Internet]. In: Thomas D, editor. The United States of America. Massachusetts Institute of Technology; 2012 [cited 2018 Aug]. p. 7 - 8. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=3339482

10. Copeland M. What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?. Envidia [Internet]. 2016 Jul. Available from: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

11. Algoworks. Is Augmented Reality the future of apps?. Augmented Reality Developers Resources; 2017 Jun. Available from: https://www.appfutura.com/blog/is-augmented-reality-the-future-of-apps/

12. Lunden I. IKEA Place, the retailer's first ARKit app, creates lifelike pictures of furniture in your home. Iphone Event; 2017. Available from: https://techcrunch.com/2017/09/12/ikea-place-the-retailers-first-arkit-app-creates-lifelike-pictures-of-furniture-in-your-home/

13. Kumparak G. Pokemon GO gets a new and improved augmented reality mode (but only on iOS) [Image of internet]. 2017 Dec [Cited 2018 Aug]. Available from: https://techcrunch.com/2017/12/20/pokemon-go-gets-a-new-and-improved-augmented-reality-mode-but-only-on-ios/

14. Arkit [Internet]. Apple Developer Website. Arkit2; 2018 [cited 2018 May]. Available from: https://developer.apple.com/arkit/

15. Understanding World Tracking in ARKit [Internet]. Apple Official Developer Documentation [cited 2018 May]. Available from: https://developer.apple.com/documentation/arkit/about_augmented_reality_and_arkit

16. Pradhan A. Apple's ARKit - A game changer when it comes to augmented reality applications. Softweb Solution; 2017 Oct. Available from: https://www.softwebsolutions.com/resources/Apple-ARKit-for-augmented-reality-app-development.html

17. Ashwini A. What Is Apple ARKit and What Does It do? [Internet]. Cognitive clouds; 2017 Jun. Available from: https://www.cognitiveclouds.com/insights/what-is-apple-ar-kit-and-what-does-it-do/

18. Miesnieks M. Why is ARKit better than the alternatives for AR?. A Medium Corporation; 2017 Jul. Available from: https://medium.com/6d-ai/why-is-arkit-better-than-the-alternatives-af8871889d6a

19. Introducing Core ML [Video on Internet]. Apple Developer Website; 2017 [Cited 2018 April]. Available from: https://developer.apple.com/wwdc17/703

20. ARKit vs ARCore - The Key Difference [Internet]. Newgenapps; 2017 Sep [cited 2018 Aug]. Available from: https://www.newgenapps.com/blog/arkit-vs-arcore-the-key-differences

21. ARKit versus Vuforia [Internet]. Darf Design; 2017 Nov [cited 2018 Jul]. Available from: https://www.darfdesign.com/blog/arkit-versus-vuforia

22. Object Recognition [Internet]. Developer Portal [cited 2018 Jun]. Available from: https://library.vuforia.com/content/vuforia-library/en/articles/Training/Object-Recognition.html

23. Vuforia Developer Portal [Internet]. Develop Portal [cited 2018 Jun]. Available from: https://developer.vuforia.com/vui/pricing

24. Converting Trained Models to Core ML [Internet]. Apple Official Developer [cited 2018 Sep]. Available from: https://developer.apple.com/documentation/coreml/converting_trained_models_to_core_ml

25. Chessen M. What is Artificial Intelligence? Definitions for policy-makers and non-technical enthusiasts [Internet]. A Medium Corporation; 2017 Apr [cited 2018 Apr]. Available from: https://medium.com/artificial-intelligence-policy-laws-and-ethics/what-is-artificial-intelligence-definitions-for-policy-makers-and-laymen-826fd3e9-da3b

26. Hollemans M. Pros and Cons of iOS machine learning APIs [Internet]. Machine think blog; 2017 Jul [cited 2018 Aug]. Available from: http://machinethink.net/blog/machine-learning-apis/

27. Hollemans M. Machine learning on mobile: on the device or in the cloud [Internet]. Machine think blog; 2017 Feb [cited 2018 Jul]. Available from: http://machinethink.net/blog/machine-learning-device-or-cloud/

28. TensorFlow releases [Internet]. TensorFlow. Available from: https://github.com/tensorflow/tensorflow/releases

29. Swift for TensorFlow [Internet]. TensorFlow; updated 2018 Oct. Available from: https://www.tensorflow.org/api_docs/swift/

30. Kumparak G. Apple acquires Turi, a machine learning company [Internet]. Techcrunch; 2016 [Cited 2018 Nov]. Available from: https://techcrunch.com/2016/08/05/apple-acquires-turi-a-machine-learning-company/

31. License [Internet]. Apple Github [Cited Nov 2018]. Available from: https://github.com/apple/turicreate/blob/master/LICENSE.md

32. Integrating a Core ML Model into Your App [Internet]. Apple Official Developer Documentation [cited 2018 May]. Available from: https://developer.apple.com/documentation/coreml/integrating_a_core_ml_model_into_your_app

33. App Store Review Guidelines [Internet]. Apple Developer Website. App Store; 2018 [cited 2018 May]. Available from: https://developer.apple.com/app-store/review/guidelines/

34. Apple/Turicrate [Internet]. GitHub [Cited 2018 May]. Available from: https://github.com/apple/turicreate

35. Image Classification [Internet]. Apple Github [Cited 2018 Jul]. Available from: https://apple.github.io/turicreate/docs/userguide/image_classifier/

36. The Most Popular Python Data Science Platform [Internet]. Anaconda [Cited 2018 Jul]. Available from: https://www.anaconda.com/what-is-anaconda/

37. Working with Tabular Data [Internet]. Apple Github [Cited 2018 Jul]. Available from: https://apple.github.io/turicreate/docs/userguide/sframe/tabular-data.html

38. Turicreate.Sframe.random_split [Internet]. Apple Gibhub. Turi Create API; updated 2018 Oct. Available from: https://apple.github.io/turicreate/docs/api/generated/turicreate.SFrame.random_split.html

39. Turicreate.image_classifier.split [Internet]. Apple Gibhub. Turi Create API. Application; updated 2018 Oct. Available from: https://apple.github.io/turicreate/docs/api/generated/turicreate.image_classifier.create.html#turicreate.image_classifier.create

40. Turicreate.image_classifier.ImageClassifier.predict [Internet]. AppleGibhub. Turi Create API; updated 2018 Oct. Available from: https://apple.github.io/turicreate/docs/api/generated/turicreate.image_classifier.ImageClassifier.predict.html

41. Turicreate.image_classifier.ImageClassifier.evaluate [Internet]. Apple Gibhub. Turi Create API; updated 2018 Oct. Available from: https://apple.github.io/turicreate/docs/api/generated/turicreate.image_classifier.ImageClassifier.evaluate.html

42. Advanced Usage [Internet]. Apple Github [Cited 2018 Oct]. Available from: https://apple.github.io/turicreate/docs/userguide/image_classifier/advanced-usage.html

43. Purchase and Activation [Internet]. Apple Developer Website. Support [Cited 2018 Sep]. Available from: https://developer.apple.com/support/purchase-activation/

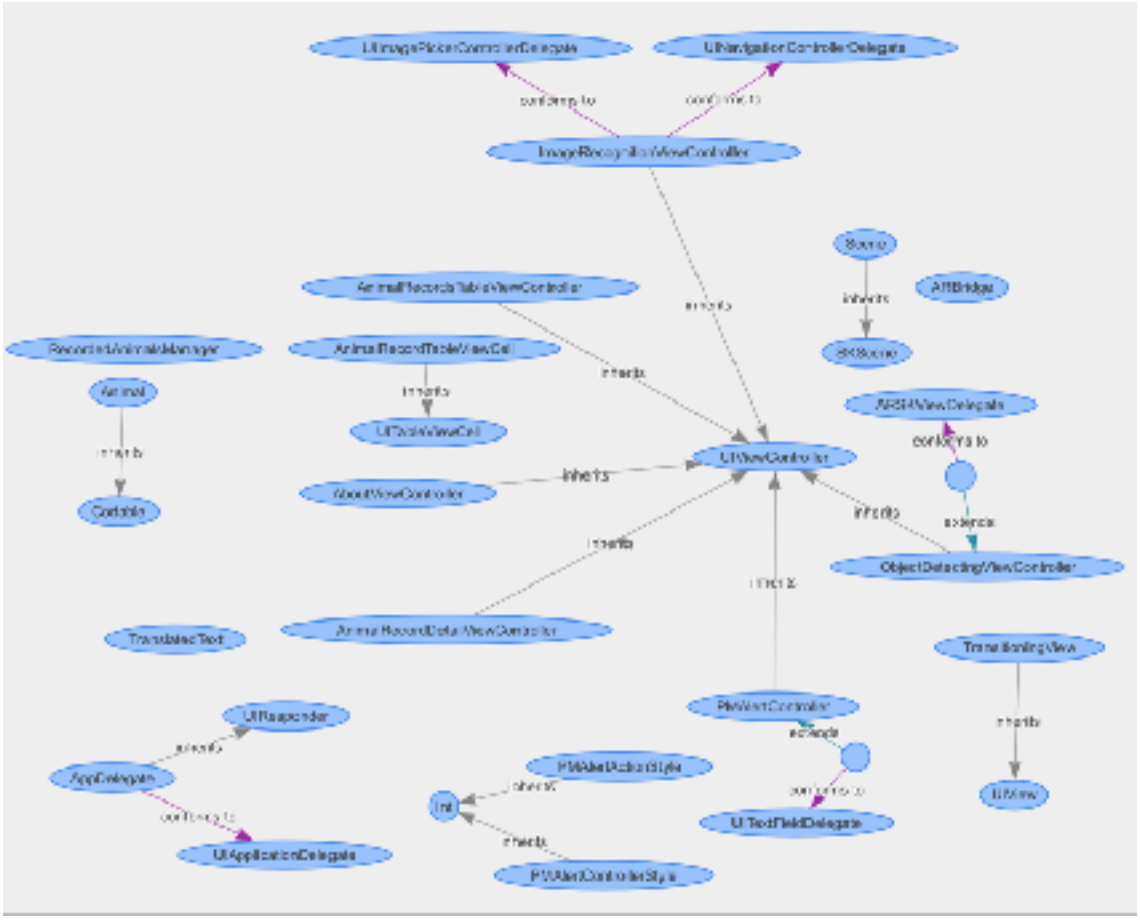
44. VNCoreMLRequest [Internet]. Apple Developer Documentation. Vision [Cited 2018 Sep]. Available from: https://developer.apple.com/documentation/vision/vncoreml-request

Appendix 1

1(2)

Appendices

Class diagram of application



Appendix 2
2 (2)

Questionnaire

1. Have you ever used iPhone/iPad before?

   Yes No

2. Have you ever used similar kind of mobile application before?

   Yes No

3. Is the application easy to use?

   Yes No

4. What is the most favourite feature that you like in the application?

   Image recognition(First feature)

   Live camera recognition(Second feature)

   History/Information (Third feature)

5. Would you think the application help to learn more about animals?

   Yes No

6. Would you use the application if you go to Korkeasaari?

   Yes No