University of Derby

School of Computing & Mathematics

A project completed as part of the requirements for the BSc (Hons) Computer Science

entitled:

Augmented Reality: Investigating how augmented reality effects the accuracy of object recognition

By Brendan Milton

In the years

2017 - 2018

1 ABSTRACT

This study set out to explore how augmented reality effects the accuracy of object recognition, by designing and developing an application capable of classifying objects from both image and augmented reality perspectives. The data generated by this application could then be analyzed to gauge the effects.

The literature review presented an awareness of relevant concepts and current augmented applications. Technologies paramount to the successful creation of the application were introduced for further practical research across the methodology.

The study successfully designed and developed the application, to test the accuracy of object recognition. Comparison data required for analysis was gathered by comparing objects recognized using the application in both a 2D and 3D space. To an extent this analysis offered some evidence that the 3D nature of augmented reality did impact the accurate recognition of objects. Unfortunately the results were inconclusive die to limited testing methods.

Although lacking conclusive results the study raises the importance of accuracy in augmented reality surrounding the medical, military and manufacturing fields. The study also presents some interesting insights into the importance of augmented specific machine learning models directed at future research.

ACKNOWLEDGEMENTS

First and foremost I would like to thank my cousin Sara, for proofreading my work and providing valuable feedback, the emotional support over the last four years and ultimately helping to make it all possible. I would also like to thank my research project supervisor Dave Voorhis for his enthusiasm towards the subject matter, and overall patient and understanding support throughout the final year.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. RATIONALE AND STRUCTURE

In the developed world, smartphones are now a fundamental tool used in everyday life, this importance leads to growth with the amount of users predicted by Cisco (2017) to be 2.87 billion by 2020. Smartphone applications share this growth, driving new and exciting technologies at an alarming rate.

These new technologies allow for a host of new features to become available. One such feature augmented reality described by Johnson (2016) of the University of Utah as augmenting the real world with additional computer generated information.

Although augmented reality will allow for many new applications, the technology will not be without challenges. A challenge presented by Rabbi and Ullah (2018) is scene visualization, a difficulty found as the computer must take in to account the scanned environment while placing 3D objects within the space.

Scanning the environment involves scanning objects already present in a scene, to complete this an established technology object recognition is used. Object recognition works very well with 2D images, and while constantly expanding is not perfected for the 3D environment that augmented reality requires, thus a problem is born. How does augmented reality effect the accuracy of object recognition?

To explore this problem in greater detail the structure of this study begins with a literature review to introduce the general body of research required to give an appropriate background into the fields of augmented reality and to a lesser extent machine learning. Background research is required to highlight an awareness of relevant concepts, researchers and related work. Only after attaining this awareness, can a practical and methodological approach be pursued to gain the relevant technical knowledge and planning required to build a suitable application. This application can then be used to gather the data needed to compare object recognition both in the realms of regular images (2D space) and

Augmented reality (3D Space). An overall hypothesis will clarify this study, followed by an aim to provide evidence for or against it and the objectives that will achieve this aim.

1.2. Hypothesis, Aim and Objectives

"Augmented reality will greatly impact object recognition accuracy."

To prove the above hypothesis the aim of this project is to investigate how augmented reality effects the accuracy of object recognition, by designing and developing an application capable of classifying objects from both image and augmented reality perspectives. The data generated by this application can then be analyzed to gauge the effects. In order to achieve this aim, the following objectives will be met:

1. A literature review will be conducted to gain an awareness of research concepts relevant to augmented reality and object recognition technologies.
2. A methodology will be undertaken in order to layout appropriate strategy for training a machine learning model required to classify objects, augmented reality application design/development, data generation, data analysis, ethical implications and limitations.
3. The resulting application will be deployed, with results being recorded to analyze, measure and display.
4. Conclusions will be made based on analysis, to provide evidence for or against the hypothesis along with recommendations on future research.

2. Literature Review

2.1. Introduction

By 2021 global spending on Augmented Reality related products and services will reach 20 billion while growing more than 100 percent each year (Shirer and

Torchia, 2017). Compton, (2017) states that industries including military, health and scientific research are expected to see the most substantial growth, consumer enthusiasm will drive expectations for AR in the workplace.

The drive behind augmented research has already led to an array of applications surrounding car repairs with BMW leading the pack (Levski, 2018). The idea is that a glance at an area of the engine or car will show potential issues and replacement parts. Similar advances in the medical field can already show surgeons an example 3D model of the human anatomy augmented over a patient to aid in procedures (Loughran, 2018).

Although these advances are very exciting accuracy will be paramount, these surgical applications present a certain danger. Should a computer decide to classify an object such as internal organ and present the location details with even a margin of error the results could be catastrophic. Although not presenting as much physical danger, an incorrect object classification from within a car could result in costly mistakes or even vehicle failure should the wrong part be replaced. Consideration of these factors presents room for many more augmented reality accuracy studies to take place.

In order to gain a more in depth awareness of relevant concepts and meet the requirements of objective one this research will focus specifically on augmented reality and to a lesser extent, technologies that are paramount to running it. Another such concept vital to this study "machine learning", described by tesla software engineer Lindell (2018) as "the heart of AR platforms". Machine learning is of particular importance to this augmented study due to the technology allowing people to manipulate 2D and 3D objects within a virtual space raising the importance of object recognition accuracy (Lindell, 2018).

This chapter will offer an overview of these topics defining, the technology, the history, current substantive knowledge, and an awareness of the most recent concepts. Overly technical information directly relating to design and development of the application and data generation will be introduced then covered in depth in the methodology chapter following the review.

2.2. AUGMENTED REALITY

Augmented Reality or AR, dates all the way back to the 1960s, but has only really been making headlines in recent year’s thanks in part to mobile games like Pokémon GO.

Technology Description

When looking through a phone screen, glasses, headset or another camera-equipped device, the technology overlays the unreal (virtual) over what is actually present in the space around the user.

While Pokémon GO is certainly one of the better-known examples, countless other uses have come to the masses in the past few years. For online shoppers, Perez (2018) presents the Wayfair app which lets users visualize shelves and other decor in their own home by viewing select products, pointing their device’s camera at the place where they are considering placing the item, and then “previewing” it in real-time. Although more technical applications like that found in the Wayfair app are in need of some further accuracy tweaking in order to get object angles and positioning right, augmented reality already has a very promising future.

For an awareness of AR’s importance a technology sometimes confused with AR, Virtual Reality or VR is often poised to have much more growth than AR but as multiple publications have pointed out, “augmented reality is where the money is.” (Armstrong, 2018). This is due to technology reach. The number of VR units in the is climbing though sits around 80 million, augmented reality tech is found in most smartphones suggesting AR units in the world peak at over three billion (Armstrong, 2018). For companies looking to invest in one of the two technologies, obviously the latter seems much more lucrative. While VR will require mass adoption through expensive and specialized equipment, AR-equipped platforms are already being unleashed thanks to the widespread usage of high-powered smartphones and tablets.

Now that multiple leading players have made a means for smaller developers to begin utilizing the power of AR development libraries available include Googles ARCore, Apple’s ARKit, and Snapchat Lens Studio. Making application development far more accessible these libraries are ensure a list of applications of AR will only continue to grow. While augmented reality is already in use on the small scale, this usage is expected to expand much further as people begin realizing the full potential of this idea.

Brief History

As of right now, a wide selection of photo-altering filters, games, basic retail applications and other casual uses for AR are available, but to really understand the full scope of possibilities, a summary of the original imagined uses of augmented reality type technology (Mekni and Lemieux, 2018).

Mekni and Lemieux (2018) describe major inspirations for AR coming from movies such as Terminator and Robocop where characters viewed the world with a graphical overlay. A particular scene shows the terminator identifying all sorts of 3D objects from motorbikes to people with what appears to be 100% accuracy. These science fiction applications might prove to be possible as the technology continues to evolve. As for 100% terminator style accuracy, in recent studies Schmalstieg (2015) describes major challenges with training models to recognize real world objects. In his presentations at the Graz University of technology, Schmalstieg states that models still need to be trained using 3D CAD representations of real objects and the accuracy is still quite low due to minor differences between CAD and real objects. Still, many suggestions that layers, tools and training methods eventually can be implemented to allow for technologies like these to be possible in the future. How did AR become possible? Answering this question requires a development timeline of AR back to 1968 when the first real milestone was made in its development.

In 1968, Ivan Sutherland created the first augmented/virtual reality system. It used an optical see-through head-mounted display that tracked with two different systems the first a mechanical tracker and second an ultrasonic tracker. Due to the limited processing power at that time, only very simple drawings could be displayed to the user in real-time (Arth et al., 2015).

Just four years later in 1972, another milestone in AR's development was reached when Alan Kay proposed the first conceptual tablet computer. He pushed the idea of the Dynabook, which was a personal computer for children. It had the format of a tablet paired with a mechanical keyboard. Decades before the iPad, the Dynabook is recognized as being the precursor to tablet computers. The following year, Motorola presented an idea for the first hand-held mobile phone. In 1982, the Grid Compass 1100 the first ever laptop was produced. It cost $10,000 and its weight of nearly 5kg. (Arth et al., 2015). In the 1990s, more steps were taken that would eventually bring things into alignment for the modern use of augmented reality. Like in 1992, when Tom Caudell and David Mizell coined the term "Augmented Reality" (Koller et al., 1997).

In 1993, the first smart-phone was introduced by Bellsouth and IBM. The phone's hardware made up of 1 Megabyte of memory, a black and white touch screen with a tiny resolution of 160 x 293 pixels. The IBM Simon as it was called had multiple functions a phone, pager, calculator, address book, fax machine, and e-mail device. Weighing at 500 grams and costing a staggering 900 USD (Arth et al., 2015).

More headway was made in the coming years as an outdoor navigation application for people with visual impairment was developed. This application used GPS receiver as part of a notebook including a head-worn compass. The application used data from a geographic database to provide navigational assistance using a virtual display. Additionally speech play back occurred at correct locations within the auditory space of the user (Arth et al., 2015).

By the mid-1990s, widespread use of the term "Augmented Reality" was occurring. However, it was not until 2005 with the release of an AR-Tennis game that it began to be implemented for entertainment purposes. Up to the point, much of the focus of AR has been on guidance systems visually impaired individuals.

Current Technology Applications

Since 2009, countless multi-national corporations have begun investing heavily in the power of AR. This is partially because of the realization its full potential has come to light.

Heightened usage and popularity is also due to the increased processing power of everyday computers, the lowered price of producing such technology, and the widespread usage of touch-screen smartphones.

While games are certainly stealing the spotlight as one of the more amusing applications for AR, most research and development has been focused on its more practical and productive applications.

Medical

Out of all the potential medical applications for AR, as introduced surgery is certainly a major focal point. Khor et al. (2016)
describes that with a visual overlay relaying critical information to surgeons during operations, AR can truly be a life-saving technology. AR is also gaining traction as an educational tool for medical students.

Traditionally an anatomical atlas was used in anatomy teachings, with practical experience gained in surgical training rooms. AR and VR are being used to combine these two methods and deliver a better appreciation of technical structures in virtual or real space, in an attempt to ease the transition from the learning environment to the clinical environment. The “Microsoft Kinect” (Gantenbein, 2018) is an example of this, used to produce an interactive ‘digital mirror’ to the user. This digital mirrored image is then augmented by anatomical datasets to visualize structures such as musculature, superimposed on the user’s own arm (Khor et al., 2016). Once a student has graduated and made their way into the operating theatre for themselves, they are also likely to begin seeing emerging, AR-driven technologies become commonplace in their day-to-day work.

Of equal importance are the introduction of electronic records to AR. Blood results, wound swabs, clinical notes, ward documentation, and medications can all be viewed paired with live vital signs using electronic health record systems such as drchrono. Upon the introduction of AR glasses which are not too far away according to Vuzix a company spearheading their commercial development (Vuzix, 2018). With a pair of AR glasses, the aforementioned electronic records will be viewed live as a patient is treated.

Military

The military is always working to improve the effectiveness of the strategies used for intelligence collection, enemy disarming, and tactical approaches. For all of these reasons, AR has certainly been a technology of significant focus since its early days.

Original military applications of AR technology used by US air force fighter pilots. The first application a system developed at Wright-Patterson air force base in the late 1960s named “Super Cockpit” (Livingston et al., 2018). This implementation of the modern display still used by fighter pilots to this day. The original applications used both virtual environment and transparent overlay, to enable the pilot to utilize system features in any lighting conditions. This application greatly assisted the pilot with limited cockpit visibility, low-altitude navigation, target acquisition accuracy, and weapons delivery. The system superimposed flight and target data into the pilot’s visual field and provided assistance sound cues.

This powerful system is in continuous development as many early versions pointed to the same key weakness: spatial reasoning is a very complex task. The human factors of these systems have to be considered in order to really be of help to the pilot and not cause “information overload” (Livingston et al., 2018), or simply making their high-pressure job more complex.

The right interface should take advantage of a person’s natural abilities, like reasoning in three dimensions, rather than making them adapt to the technology and have to use only two dimensions. Pilots are actually screened for high spatial reasoning abilities, so it makes sense that their view should be one inherently natural and 3D.

Manufacturing

The manufacturing industry certainly put a huge amount of focus on robotics technology to speed up processes and reduce the cost of human workers. But, there is also a large amount of money being put into AR research and how it can add value, reduce errors, and increase efficiency at operations across sectors. Two areas of particular interest introduced by Wild et al., (2014) are assembly and maintenance of equipment.

AR techniques were used to reduce task completion times and assembly tasks errors. Feiner and Henderson demonstrated that AR can help users remain more visually focused on current work while moving between maintenance tasks at a higher pace. Follow on studies also showed that AR can reduce errors in tasks where workers exhibit more mental activity. (Wild et al., 2014)

Multiple companies are already at work to help businesses reap these benefits, such as “AIDIMA”, a not for profit research firm working to enhance the competitiveness of furniture and packing industries in Spain. They are currently serving more than 700 businesses, and they aren’t alone. Similar Companies, “TELL-ME” and “Agusta Westland” are also working to help grow knowledge about AR applications and challenges using the technology within the manufacturing sector.

Educational

As NJIT puts it, “Augmented reality has the potential to revolutionize learning in primary and secondary schools more than any other technology has done in the recent past” (Cortez, 2017).

From AR applications that change the way children look at books and puzzles to science-based AR coloring pages, and even interactive homework. Imagine that, next time your child gets stuck solving a take-home math problem they can just pull out their phone and scan a code to see a video of their teacher walking them through the problem.

Student-made book reviews attached to hard copies in the library, the incorporation of AR into yearbooks so students can truly showcase their stories and personalities in interactive form, and especially applications like lab safety are all emerging ways that AR is being implemented into education everywhere.

Tourism

With some of the first applications for AR being that of interactive museum guides, the implementation of AR in modern-day tourism comes very naturally. Some example applications focus on educating, like “Tuscany+” giving the user history and information of surroundings, while others focus on informing users behind an entertaining storyline.

Urban Sleuth designed a similar tourism city adventure. In which users participate with the aim to compete against others or in teams while carrying out missions or solving mysteries while travelling around the city. Through the application by blending the real with the virtual, offered missions are designed so that participants can discover neighborhoods and historical monuments, among other interesting locations (Kounavis, Kasimati and Zamani, 2012).

“Street Museum” (Zhang, 2018) is another application working to bring modern technology into the tourism industry by informing and entertaining visitors while helping bring history to life. With this app designed specifically for the museum of London, users are offered the chance to visualize the city of London at various points in history (Kounavis, Kasimati and Zamani, 2012).

2.3. MACHINE LEARNING

If any time is spent looking into research surrounding AR and VR, it’s highly likely the topic of Machine Learning, or ML will appear.

Technology Description

Machine learning is a way programs and applications can learn from data that’s been passed to applications be it photos, live video streams, music or raw data and then make predictions based on that information.

A ML algorithm will classify the information that comes from a given set of data and then take that information to analyze patterns and observe it. With the ML model, the technology will make a prediction based on prior training.

Machine Learning is different from simply writing a program that takes X and does Y. With Machine Learning, a computer is able to actually learn new skills without being explicitly programmed to do them. While this requires a base model or foundation of some sort for this machine to learn things from in a similar way a child is given a basic vocabulary to start forming sentences, ML enables computers to achieve incredible knowledge.

Using neural networks, a ML model is able to learn what something is without being specifically programmed to know it. Currently, Machine Learning is being implemented at all sorts of levels to improve the efficiency of systems and even recognize risks and challenges before they arise. There are countless potential uses for this technology, like with calculating risk for insurance companies or even finding missed opportunities in a small to medium business advertising model.

Machine Learning can be used to save money, speed up completion time, and much more depending on its particular usage within any given sector. It's also at the core of self-driving cars and even everyday tools, like your ability to find particular photos from your ever-growing digital album.

Current Technology Applications

Many Machine learning applications of today are already in widespread use. Image and Object recognition for example, are some of the most popular uses. Automated tagging, sorting, and "search for similar image" tools are other prime examples of the practical uses for machine learning.

However, there are other contexts for machine learning that can be just as beneficial. With IBM Watson already being employed at corporations around the world to help maximize efficiency and improve processes, its clear machine learning also has a lot of value outside of the entertainment applications we currently think about it in. It’s already in place and being used to save money, improve products, and even increase safety (IBM, 2018).

The future for machine learning is vast, and likely will include applications that we haven’t yet considered.

2.4. CONCLUSIONS

As part of the study objectives, a literature review was conducted to gain an awareness of concepts relevant to augmented reality and object recognition technologies. In concluding this chapter a summary is offered to give relevance to the overall literature by highlighting industries where the accuracy of identifying objects through AR could be paramount to safety. Of the areas researched where accuracy may have the largest impact, medical, military and manufacturing the following assumptions can be made. Medical AR accuracy for training may not be so serious, however for surgical uses it could be the difference between life and death. For military fighter pilot AR applications, should any weapons systems rely on object recognition to identify targets lack of accuracy could be deadly. Manufacturing applications mentioned to keep workers more visually stimulated if used with any hardware that could cause harm, accuracy for injury prevention would be very important. Education and tourism present less threat when it comes to accuracy however it could still be important to keep users content with the operation of the software.

Technologies identified in the literature paramount to creating an application for testing the accuracy of augmented reality. Machine learning described by Lindell (2018) as the main technology that allows object recognition to work in a 2D and 3D space will play a major role in data generation. For the augmented reality application to be tested a machine learning model must be created to generate the required data for the study. The next chapter details the plan process for research and practical methods.

3. METHODOLOGY

3.1. INTRODUCTION

This chapter justifies the appropriate selection of research and practical methods required to create an application, effectively generate data for analysis and create an awareness of limitations in order to fulfil the project aim.

Although this chapter lays out the proposed methods in research and application design, the development chapter will outline how the application was actually achieved. In doing this choice of development methods will be verified and validated.

Quantitative research will be used for the study as data will be collected, then require statistical graphs to be analysed. The data will be collected via a handheld application stored on a mobile phone, then data can then be analysed to see if augmented reality has an impact on the accuracy of object recognition.

In order to create the required application, a sound waterfall development approach will be undertaken as laid out Powell-Morse (2016). This approach will include the following phases: requirements, design, coding, testing and operation. *Table 1* below shows the planned breakdown of these phases. For completion the table will be revisited in the next chapter to briefly validate and justify the method choices.

Application	**Proposed Waterfall Approach**
Requirements	*As machine learning was identified as the heart of AR applications an object recognition model must be created to allow for image classification. To achieve the project aim the application must be portable and allow a view for an image and augmented reality view, hence access to a camera is paramount. Requirements will be covered in subsections 3.3 and 3.4 of this chapter.*
Design	*Architecture and design will be explained and documented in*

	subsection 3.4 along with application wireframe's and flowcharts.
Coding	*Required coding platforms, languages and libraries will be explained in subsection 3.4*
Testing	*Appropriate testing will be explained in section 3.4*
Operation	*Operation Design will be explained in section 3.4. Actual operation of the application will be explained in the development, results and analysis chapter as this will produce the required data for analysis.*

Table 1 Proposed Waterfall Approach

3.2. MACHINE LEARNING DATASET

As a major requirement to generate comparison data, an underlying dataset will have to be trained and tested in order to meet the aim.

After extensive research the "Cifar-100 Dataset" collected by (Krizhevsky, 2009). Looks to be the most suitable option for the model. The Cifar-100 dataset appears suitable due to it containing roughly 600 images of 20 classes of objects. The data is already split into files of training and testing images, making it perfect for training and testing for the model.

3.3. APPLICATION ARCHITECTURE AND DESIGN

The application will be a very simple layout with a landing page and two subsequent pages. A button labelled "Object Recognition A-OR" will lead to a section titled A-OR (Application Object Recognition) where the custom trained machine learning algorithm will be able to recognize objects from a photo with a degree of certainty. The second section to be accessible by an alternate button labelled "Augmented Recognition A-AR" leads to the section titled A-AR (Application Augmented Reality) will perform the machine learning algorithm on a live augmented reality feed by placing an augmented labelled of the object name, the result can then be compared to the A-OR results to test the accuracy to achieve the aim of the experiment.

As the application will need to be created to allow the machine learning model to integrate with the application for testing. The required process can now explained in depth as to how the required technologies will be used to create the application.

IOS development will be chosen to develop the application given that an Iphone and Macbook were available to the study. This includes development using Xcode, an apple Ide in the swift programming language. To integrate the machine learning model to be created in python with Keras library, a conversion must take place as IOS applications can only interact with coreML. The coreML conversion of a trained keras model looks to be quite a simple process as mentioned in apple development documentation (Apple, 2018). To convert the trained “Cifar-100” dataset with a single object image input and add classification labels from existing files the relevant code samples can be seen below in *Figures 1* and *2* as directed by Apple (2018).

Converting a model with a single image input.

```
>>> coreml_model = coremltools.converters.keras.convert(model, input_names =
... 'image', image_input_names = 'image')
```

Figure 1 (Apple, 2018) "Converting a model"

Class labels for classifiers can also come from a file on disk.

```
>>> coreml_model = coremltools.converters.keras.convert(model, input_names =
... 'image', image_input_names = 'image', class_labels = 'labels.txt')
```

Figure 2 (Apple, 2018) "Class labels for classifiers"

The A-OR Section of the application will be quite straightforward due to only needing to read a picture and then apply the newly converted coreML model. The A-AR section however will require a couple of extra frameworks.

The vision framework which will aid with general feature tracking of objects when applying the coreML model as described on the apple developer documentation (https://developer.apple.com/documentation/vision). Finally the

ARkit framework to aid in augmented reality development and tracking for placing the AR Labels over objects. (Apple Developer, 2018)

The application mock layout can be seen below in *Figure 3* showing the landing page, A-OR and A-AR pages.

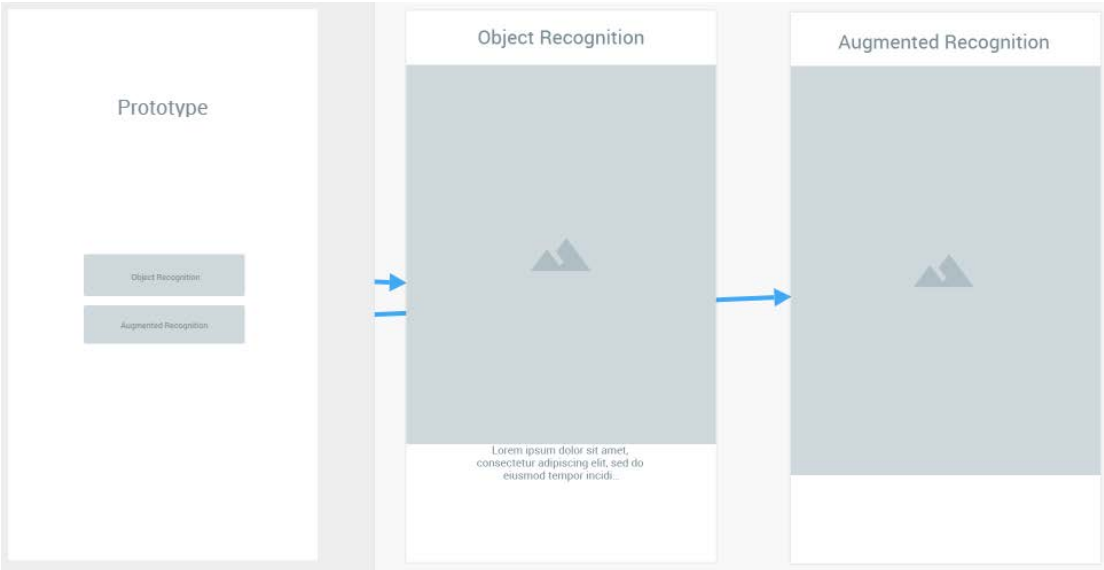


Figure 3 Augmented Application Mock Layout

An operation flow chart of the application can be seen below in *Figure 4* which displays how to interact with the application and the resulting requests and actions after pressing a certain button.

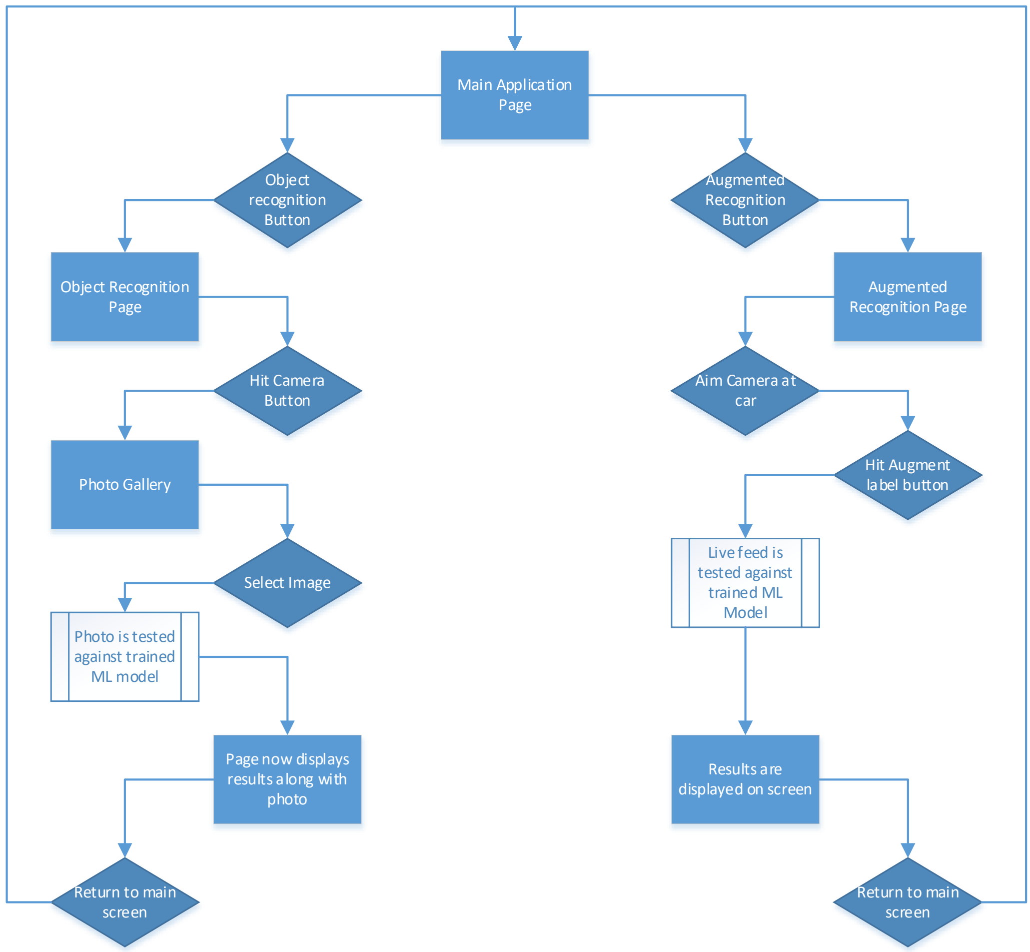


Figure 4 Augmented Application Operation Flow Chart

3.4. Data Generation

As stated in objective two after creation the application must be tested in the field. To gather results of both application sections A-OR and A-AR. The testing will be controlled in that each object will first be subjected to A-OR section by taking a picture of it and applying the machine learning algorithm, classifying the object. This process is followed up by applying the A-AR section placing a prediction label over the object. The resulting results will then be compared, analyzed and graphed in the Results and Analysis section. Ideally sampling will include 100 object for comparison.

3.5. Data Analysis

Both application sections A-OR and A-AR will have prediction data extracted from them for analysis, the resulting prediction will be placed into a comparison table. Comparison insights can then be outlined within the table to check if a match is present. The table will be presented in a format as seen below in *Table 2*.

Object Recognition **(Using A-OR)**	**Augmented Recognition Match** **(Using A-AR)**	**Augmented Recognition Angle Match** **(Using A-AR)**
Letter Box	*Match*	*No Match*

Table 2 Example comparison insights table

The application will produce three different results to be placed into the table above. The first column the data extracted from the A-OR section, in the example this is a “Letter Box”. The next section A-AR holds the data suggesting whether or not the “Augmented Recognition” section of the application produced the same prediction. Finally the “Augmented Angle Match” suggests whether the Object was a match when holding the A-AR at an angle. Upon completion of each comparison, the data will be collated into graphical analysis using software.

3.6. ETHICAL IMPLICATIONS

In accordance with the University of Derby's ethical guidelines. There will be no concerns regarding permission to take photographs as each object will be in public areas or in a location where permission is granted.

3.7. LIMITATIONS

Limitations can be present in even the smallest of experiments, given the size of this task there will likely be difficulties in development. Development limitations could be present through all stages, potentially within the machine learning model. Programming limitations as a result of limited development experience with chosen technologies, in addition it seems clear that there are quite a few different factors and libraries that will make up the application as explained in this section. This may impact the ability to effectively construct a suitable application for testing. If the application cannot be created the experiment fails. Should the application reach the testing phase data generation methods chosen may create a difficult time for analysis and restrict results. As important to realize as these limitations are, it is still believed at this stage that an appropriate application to achieve the aim can be made.

3.8. CONCLUSIONS

In concluding this section it can be seen that in attempt to meet objective two, an in depth methodology was undertaken to attain the base information including both the advanced theory and practical knowledge to train a machine learning model to recognize specific objects from images. An application was designed, along with the introduction of the appropriate technologies needed to develop it. A plan was outlined as to how to generate the data needed to achieve the aim of the study and finally perceivable limitations were stated to give a realistic view of the approach. The application can now be developed and tested in the field. The resulting outcome along with the data will be discussed in the results and analysis section to follow.

4. DEVELOPMENT

To validate and verify sound design methods chosen, this section covers the application development that took place and how it was developed. As part of objective two a machine learning model was created to classify objects, an augmented reality application was designed and developed. The following subsections explain how this was achieved.

4.1. MACHINE LEARNING MODEL DEVELOPMENT

The development of the machine learning model followed the method laid out in the previous section for the most part. As mentioned the "Cifar-100 Object Dataset" (Krizhevsky, 2009) was used. In practice the only deviation from the methodology was found in the exact training and testing method of the data. The model was then successfully trained using Keras library as planned the code used to complete these steps can be seen in *Appendix 1-Table 6* labelled "Object_Recognition_Train.py". With the created model testing the dataset to recognize objects took place, the relevant code can be seen in *Appendix 1-Table 7* labelled "Object_Recognition_Test.py". Finally the code to convert the dataset for integration with the application using CoreMl can be seen in *Appendix 1-Table 8* labelled "Object_Recognition_CoreML_Convert.py". With the creation, training and converting of the model completed objective three was achieved. This allowed for integration into the application for testing.

4.2. APPLICATION DEVELOPMENT

The development of the application followed the methodology with very little deviation. Required technologies described were used. These included Swift as the programming language, Xcode as the IDE, CoreML, Vision Framework and ARKit. The application pages created as planned can be seen below showing an example of functionality in *Table 3* Followed by an explanation of each application page and the functions used to create it.

Application Landing Page	Object Recognition Page (A-OR)	Augmented Reality Page (A-AR)

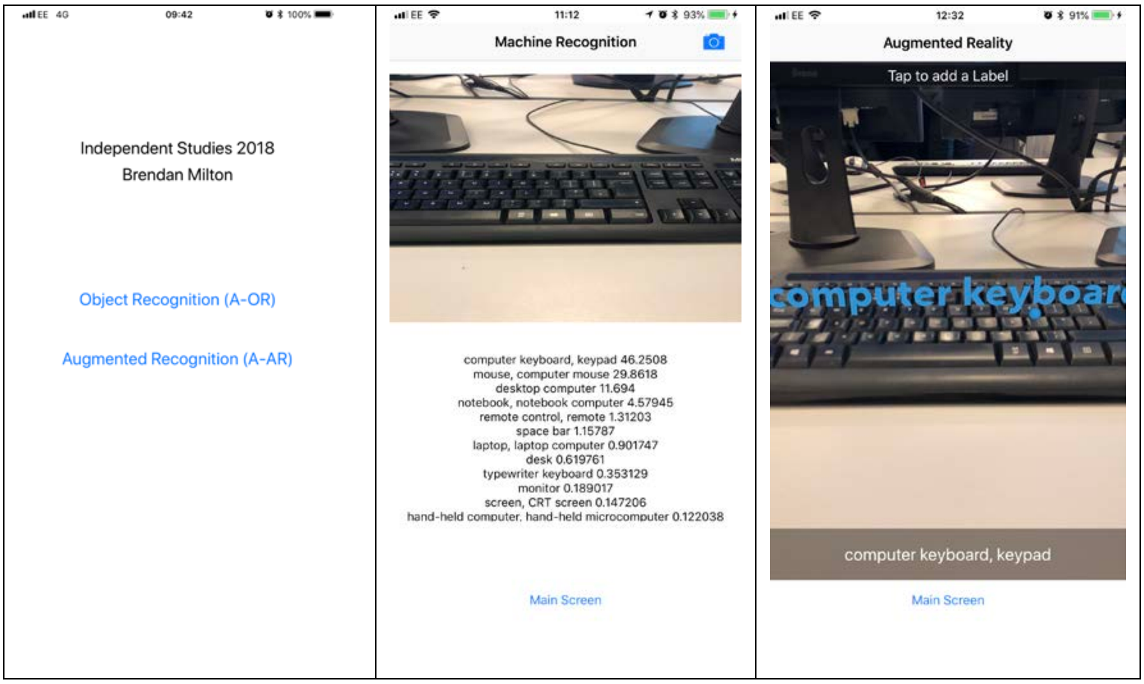


Table 3 Completed application example functionality

The application landing page is simple enough visually, it consists of a title label and two buttons. The buttons representing the functions "objectRecognitionBtnPressed()" and "arBtnPressed()" connecting to object recognition (A-OR) and augmented reality page (A-AR) respectively. The relevant code snippets can be seen in *Appendix 2-Table 9* under heading "ViewController.swift".

Object Recognition page (A-OR) visually consists of an image picker (camera icon), a label instructing page use and a main screen button directing user back to main page. A summary explanation of key functions as follows:

1. <u>openPhotoLibrary():</u>
   - Creates access to image gallery
2. <u>imagePickerController():</u>
   - Returns selected picture as a UIImage (Standard IOS image) for use with CoreML model.
3. <u>processImage():</u>
   - Process chosen image converting the UIImage to a CIImage (Required by CoreML).

- Create a vision model and request (Computer vision required for CoreML to recognise image).
- Display machine learning object recognition prediction.

4. mainScreenBtnPressed():
   - Returns user to the application landing page.

The relevant code snippets for this page can be seen in *Appendix 2-Table 10* under heading “ObjectRecognitionController.swift”.

Augmented Reality Page (A-AR) visually consists of an a label instructing page use, a live camera screen, a text label offering up to date predictions and a main screen button directing user back to main page. A summary explanation of key functions as follows:

1. viewDidLoad():
   - On A-AR start up load the scene (Live camera view), initialize the CoreML model,
2. viewAppear():
   - Create AR session() and run the session in the previously created scene.
3. viewDisappear():
   - Pause the session upon press of augmented label.
4. session():
   - Call AR session.
   - Check value of camera tracking state.
   - Limit tracking on process errors, make tracking unavailable and then resume.
5. tapHandler():
   - Centre tracking point for AR label.

- Use hit testing (ArKit method for interacting with the scene view).
- World transform (Allow for AR object to be placed) on hit point of screen to place AR label.
- Connect CoreML model to pass prediction to augmentedText().

6. augmentedText():
    - Settings and constraints required for AR text including:
        - Colour
        - Alignment
        - Geometry
    - All settings are required to allow for AR text to pivot and remain three dimensional in the scene view.
7. initializeModel():
    - Live scale the scene view and make a classification request to the CoreMl model ready to retrieve the prediction.
    - Error checking if unable to load model.

8. classificationCompletionHandler():
    - Complete classification request if no errors and a prediction is available.
9. visionRequest():
    - Apply a pixel buffer to the scene view in order to convert live UIImage to CIImage for use with CoreML model.
    - Constantly use vision framework to update view for each prediction request.
10. coreMlUpdate():

- As vision request is updated above, constantly update CoreML request to renew the prediction and place each new prediction in a queue.

11. <u>mainScreenBtnPressed():</u>

    - Return user to landing page.

The relevant code snippets can be seen in *Appendix 2-Table 11* under heading “ARRecognitionController.swift”.

Although most required functionality was completed to specification some bugs and issues are present. Firstly, although machine learning model was trained, not all object recognition attempts produce the correct item name. This does not impact the results, only really noticed within the sample pictures in Appendix 3. For unknown reasons the heading on the A-OR page could not be changed and still reads “Machine Learning” from the original prototype. For the same reason A-OR and A-AR could not be updated on the respective pages. A prediction percentage that can be seen on the A-OR page was attempted to be placed on the A-AR page, unfortunately this kept causing issues with the coreMLUpdate() function and would not go into the queue. This functionality was abandoned when discovered it would not affect the comparison results as expressed in the results and analysis chapter to follow.

5. RESULTS AND ANALYSIS

5.1. INTRODUCTION

This study aimed to identify how augmented reality effected the accuracy of object recognition by developing an application that could classify an object from both an image and live capture using augmented reality. With objective two completed, the resulting augmented reality application was deployed and tested on 44 objects to generate data for analysis. To complete objective three, clear presentation of data findings listed under results followed by analysis and finally analysis conclusions.

5.2. RESULTS

Data generated consisting of 44 objects. As outlined in methodology a table was used to collate the results, these results can be seen below in *Table 4*. The 44 objects are listed as classified using A-OR application on the application in the "Object Recognition" column, followed by the "Augmented Recognition" column where resulting matches are listed as collected by A-AR application on the application. Finally the third column holds the "Augmented Recognition Angle" matches also collected using the A-AR application. The comparison match simply states if when viewing the object in the A-AR application of the application, the object remained the same or not, then if the object remained the same when viewed through A-AR at an angle. All gathered data pictures can be found assigned to each object name under: *Appendix 3-Table 12* Full Data Comparison Results.

Test: Comparison Match

Object Recognition **(Using A-OR)**	**Augmented Recognition Match** **(Using A-AR)**	**Augmented Recognition Angle Match** **(Using A-AR)**
Letter Box	*match*	*match*
Fire Extinguisher	*no match*	*match*
Cash Machine	*match*	*match*
Trophy	*match*	*no match*
Motorbike	*no match*	*no match*
Printer	*match*	*no match*
Chair	*match*	*no match*
Sign	*no match*	*no match*
Large Pot-Plant	*no match*	*no match*

BMX Bike	*match*	*no match*
Shampoo Bottle	*no match*	*no match*
Shell	*match*	*no match*
Hand Towel	*match*	*no match*
Colour Lamp	*no match*	*no match*
Window Blind	*no match*	*no match*
Small Computer Mouse	*match*	*no match*
Candle Holder	*match*	*no match*
Circle Plant	*no match*	*match*
Scale	*match*	*no match*
Toy	*match*	*no match*
White Switch	*match*	*no match*
Spotlight Lamp	*no match*	*no match*
Detailed Pillow	*match*	*match*
Dog 1	*no match*	*no match*
Dark Lamp	*no match*	*no match*
Case	*match*	*no match*
Dog 2	*no match*	*no match*
Magnifying Glass	*match*	*no match*
Roses	*match*	*no match*
Small Pot-Plant	*match*	*no match*
Pillow	*match*	*no match*
Tv Remote	*match*	*match*

Round Switch	*match*	*match*
Candle	*no match*	*no match*
Mask	*match*	*no match*
Couch Chair	*match*	*match*
Banana	*match*	*match*
White Lamp	*match*	*no match*
Grey Switch	*match*	*no match*
Kettle	*match*	*no match*
Fan	*match*	*match*
Monitor	*match*	*no match*
Large Computer Mouse	*match*	*match*
Vending Machine	*match*	*no match*

Table 4 Application comparison match results

5.3. ANALYSIS

Test: Comparison Match Analysis

The generated data from the comparison match *Table 4* above when analysed led to four comparison scenarios. The comparison scenarios: Match/Match, Match/No-Match, No-Match/Match and No-Match/No-Match are presented in *Table 5* for explanation below.

Comparison Scenario	Scenario Explanation

Match – Match	*Using A-AR front on and at an Angle Matched A-OR prediction.*
Match – No Match	*Using A-AR front on matched A-OR prediction, however A-AR at an angle did not.*
No Match - Match	*Using A-AR front on did not match A-OR prediction, however A-AR at an angle matched.*
No Match – No Match	*Both A-AR front on and at an Angle did not match A-OR prediction.*

Table 5 Comparison Scenarios

The comparison results charted for analysis can be seen in *Figure 5* below. The chart shows the amount of times each comparison scenario appeared.

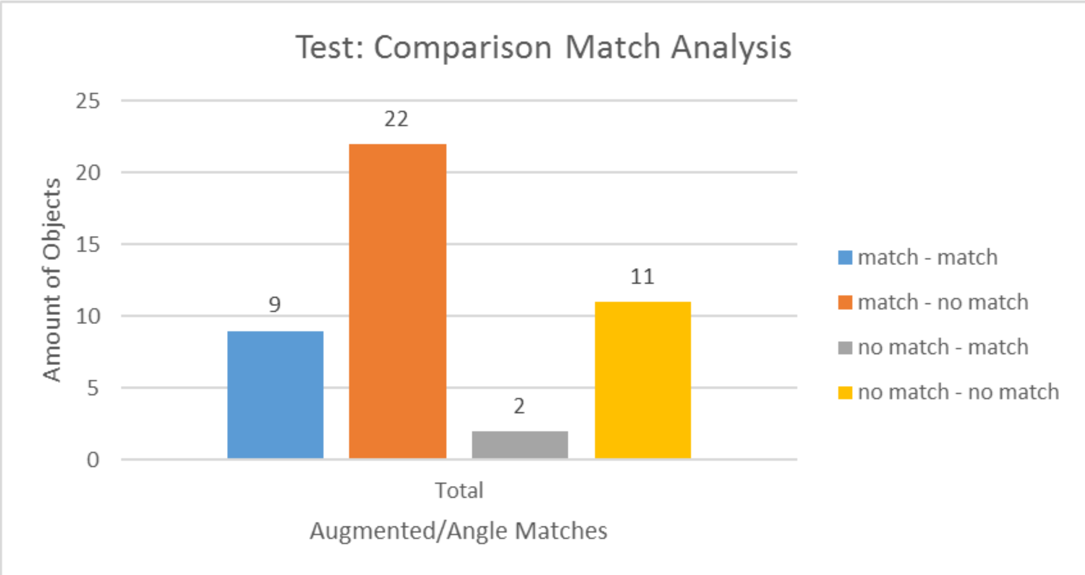


Figure 5 Comparison match analysis

Although results appear quite crude and sampling pool of 44 objects was smaller than intended in the methodology, some interesting trends present are discussed below.

5.4. Conclusions

In concluding this section, varying results of augmented recognition accuracy have been discovered when compared to object recognition using the developed application. Although crude, the data analysis still posed some insights that warranted further investigation.

The Match/No-Match scenario was by far the most common occurring in 50% of cases, meaning that when A-AR was applied front on it matched quite often. This result is not entirely a huge surprise given that using A-AR front on is not too dissimilar to A-OR, yet the second the angle of the camera is changed on the object no match is found. The angle effecting the prediction works in favour of the hypothesis suggesting that augmenting the vision at an angle does impact the prediction greatly.

The No-Match/No-Match scenario occurring in 25% of cases, meaning that A-AR did not match A-OR in either case. This result also enforces the hypothesis that augmented reality effects the prediction regardless of the angle.

The Match/Match scenario occurring in 20% of cases, meaning that A-AR both front on and at an angle matched the A-OR prediction. This result works against the hypothesis showing that augmented reality did not impact the prediction almost a quarter of the time.

Finally the No-Match/Match scenario occurring only in 5% of cases, meaning that A-AR front on did not match the A-OR prediction but at an angle did. This result is a bit of an anomaly as it is much harder to match at an angle.

Upon first glance the data suggests that augmented reality has an impact on object recognition by suggesting that the 3D space in which augmented reality needs to work impacts the prediction. The results leading to this conclusion are offered by the angle impacting the prediction in 50% of cases, as an augmented match occurred when viewing the object front on but not when viewing from an angle proving that 3D view made a difference. Unfortunately this conclusion needs to be taken with a grain of salt as the machine learning model trained for the predictions was based on the CIFAR-100 dataset, which in essence is a collection of 2D image objects.

6. DISCUSSION

This chapter discusses key issues encountered across the study. Ideally it would mostly have focused on issues surrounding the results. Unfortunately due to the crude and minimal scope of the data and analysis, resulting from issues with planning it will mainly discuss outcomes that impacted the study. These issues are outlined in the order in which they impacted the study, their focus on the literature review, methodology, development, results and application.

6.1. KEY ISSUES

Issue 1: Literature Review

The issues surrounding the literature review in the end impacted the entire study. Although the literature gathered offered enough to understand the overall subject matter with a history leading up to current applications of augmented reality and machine learning, gaps in the literature neglected the more focused research that the study required. Focused research that would have benefited the study includes: accuracy focused information targeting augmented reality, some alternative ways to test accuracy and current accuracy testing methods. In turn this impacted the results as a more suitable way to test the accuracy was not discovered, and appropriate methods for testing could not be implemented.

Issue 2: Methodology

The outlined methods were sufficient to train a machine learning model and create an augmented reality application for testing. Unfortunately without suitable literature offering options to properly quantify the research, the methodology struggled to outline appropriate data gathering and analysis methods. This issue directly impacted the results as it limited the data/scope leaving only very basic options for testing.

Issue 3: Development

As mentioned the chosen methods were enough to successfully design and develop the application for testing. Although the application was the most successful part of the project, without the practical research and appropriate methods for testing the function of the application only allowed for a very limited data generation method. In turn the limited amount a data generation.

Issue 4: Results

As the chosen methods limited the available data and analysis scope, the results were left inconclusive for the most part. Although it could be concluded that augmented reality had an effect on object recognition, as shown by the higher percentage of non-matches occurring when using the application view A-AR at an angle. This could not be properly tested due to the supporting literature and methodology allowing only for limited testing methods. As the results were inconclusive, no more than an educated guess can attributed to “how” augmented reality has an effect on object recognition. Which is to say due to the 2D vs 3D nature of the technology, machine learning model training is more of an impacting factor.

Issue 5: Application

As a design and build study it is important to explore how the application stacks up with existing systems. Given that there are no comparable small-scale applications made directly for recognizing objects, due to larger companies such as google offering inbuilt systems. It is written with confidence that the developed application would not stack up to a professional recognition app. Does that make this application useless? Interestingly there could be a market for it. As expressed this study had difficulties in part due to focus placed on two major topics: augmented reality and machine learning. As a flaw in this study it may have created a market for such an application. Many new machine learning models are being trained every day. It would be an awful lot of work for all of these data scientists to create a custom application to test them in the field each time. With a few improvements to allow custom models to be uploaded and used with this application actually making it useful.

7. CONCLUSIONS AND RECOMMENDATIONS

The aim of this study was to identify how augmented reality had an effect on the accuracy of object recognition, in an attempt to prove or disprove a hypothesis that augmented reality will greatly impact object recognition accuracy. To achieve the aim the study was broken up into a series of objectives, in the following paragraphs offer the final result of the objectives, aim, hypothesis, before recommendations can be made.

The objectives of the study were achieved as originally set out, although in practice some fundamental issues as highlighted in the discussion section became apparent. A literature review was conducted informing an awareness of concepts relevant to augmented reality, unfortunately the research content was far too broad and focused less on the focused accuracy research required. A methodology was undertaken, laying out an appropriate strategy to create an application to gather data. In retrospect this was not enough to appropriately analyse the data. An application was developed and designed to gather data and did so accordingly, unfortunately this also was not in a way specific enough to properly analyse the data. Finally there was just enough data to offer some conclusions which will be able to form recommendations for the application improvements and future research.

Given the objectives were achieved for the most part, the aim was also completed to an extent. The aim set out was to investigate "How" augmented reality effects object recognition. The actual results led to an insight suggesting that the 3D nature of augmented reality effected object recognition. Though to be more accurate the aim the study achieved is more likely in favour of "If" augmented reality effects object recognition. To which the study offers yes, but more an educated guess than a conclusive argument.

Finally taking the potential achievement of the aim into consideration, this does work in support of the hypothesis. More likely though “Augmented reality will impact object recognition” removing the word “Greatly” as there is little conclusive proof. Although plagued with issues the study still offered some vague conclusive findings, which can in turn offer some basic recommendations.

7.1. Recommendations

Application Improvements

As the application turned out quite basic and lacked real abilities to compare augmented reality and standard recognition, improvements would be welcomed. As the only conclusion discovered was that an angle effected the comparison more IE: 3D nature of augmented reality, it would be helpful to have a 3D machine learning model trained to better gauge results. In addition a direct results comparison would be fantastic, a database that registered a comparison that occurred multiple times whilst scanning. This direct results database could also compare 2D and 3D results to allow for further conclusions to be made. A nice feature could be allowing custom machine learning models to be uploaded may potentially open up the application to a niche market for machine learning researchers.

Future research

Although limitations resulted such as the lack of data causing analysis difficulties, this study still shows evidence that augmented reality does in fact have an effect on object recognition. With augmented recognition showing less accuracy in a 3D space, as the literature review showed the scope of applications being created even the small amounts of evidence produced in this study when applied to medical or military grade operations could have disastrous repercussions.

Considering these implications it would be interesting to see a study completed with much more data on a larger scale to provide a more conclusive insight.

Further study could result in even more guidelines for safety of future applications.

7.2. Personal Reflection

Upon completion of this paper I am unhappy overall with the study. I found it very difficult to find a project where I could create something that allowed me to work practically with augmented reality and machine learning at the same time. With the desire to make both of these large topics fit my time management was impacted with most time spent on required practical theory and programming to get the application built. In terms of the study that I ended up with, mistakes were made primarily due to writing the literature review before deciding on an accuracy focus, resulting in research being too generalized, vague and unrelated to specific object recognition and augmented reality concepts. As a result methods for data generation methods were floored, leaving vague analysis and conclusions as more of an educated guess than scientific fact. Without a strong analysis recommendations and future research have little validity.

Whilst I am unpleased with the overall study and the direction it took in an academic sense, there are still some positives. Although inconclusive objectives were completed and the aim was achieved offering a small amount of proof towards the hypothesis. Through the creation of the application I discovered a passion for augmented reality. The practical skills gained in using machine learning, augmented reality and IOS programming have already had a positive impact on future employment. As highlighted in key issues the application may actually have a scientific use. I understand where I went wrong and how it happened, if nothing else I can learn from these mistakes. Finally, the project has also been a wakeup call in terms of time management and finding an appropriate balance for research and practical implementation.

To conclude, I am still appreciative for the challenge this research project represented, and am confident that it poses an important question to further investigate this field of study regardless of the inconclusive results. This study has to a small degree indicated that 3D machine learning model appears to be a major factor in augmented reality accuracy and due to the potential danger augmented reality applications can pose in medical and military fields alone it should be researched even further.

8. REFERENCES

Apple (2018). *coremltools.converters.keras.convert — coremltools 0.8 documentation*. [online] Apple.github.io. Available at: https://apple.github.io/coremltools/generated/coremltools.converters.keras.convert.html [Accessed 4 May 2018].

Apple Developer (2018). *ARKit - Apple Developer*. [online] Developer.apple.com. Available at: https://developer.apple.com/arkit/ [Accessed 4 May 2018].

Armstrong, P. (2018). *2018 will be big for augmented reality – but 2021 will be bigger*. [online] Evening Standard. Available at: https://www.standard.co.uk/tech/2018-will-be-a-big-year-for-augmented-reality-but-2021-will-be-bigger-a3725101.html [Accessed 4 May 2018].

Arth, C., Grasset, R., Gruber, L. and Wegner, D. (2015). The History of Mobile Augmented Reality. pp.65-95.

Cisco (2017). Global Mobile Data Traffic Forecast Update. *Cisco Visual Networking Index*, [online] 2016-2021, pp.2-7. Available at: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html [Accessed 4 May 2018].

Compton, J. (2017). *Dell TechnologiesVoice: What Huge Growth In Augmented Reality Could Mean For Business*. [online] Forbes. Available at: https://www.forbes.com/sites/delltechnologies/2017/11/27/what-huge-growth-in-augmented-reality-could-mean-for-business/#562fb34eb4c4 [Accessed 4 May 2018].

Cortez, M. (2017). *How Will AR Transform Education? [#Infographic]*. [online] Technology Solutions That Drive Education. Available at:

https://edtechmagazine.com/k12/article/2017/10/how-will-ar-transform-education-infographic [Accessed 4 May 2018].

Hager, G. (2003). *Object Recognition Techniques*.

IBM (2018). *IBM Watson Machine Learning - IBM Watson and Cloud Platform Learning Center*. [online] IBM Watson and Cloud Platform Learning Center. Available at: https://developer.ibm.com/clouddataservices/docs/ibm-watson-machine-learning/ [Accessed 4 May 2018].

Johnson, D. (2016). *Augmented Reality*.

Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R. and Tuceryan, M. (1997). Real-time Vision-Based Camera Tracking for Augmented Reality Applications. *Virtual Reality Software and Technology*, 97, pp.1-5.

Kounavis, C., Kasimati, A. and Zamani, E. (2012). Enhancing the Tourism Experience through Mobile Augmented Reality: Challenges and Prospects. *International Journal of Engineering Business Management*, 4, p.10.

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images.

Levski, Y. (2018). *5 Examples of Amazing Augmented Reality Car Repair Apps - AppReal-VR*. [online] Appreal-vr.com. Available at: https://appreal-vr.com/blog/augmented-reality-car-repair/ [Accessed 4 May 2018].

Livingston, M., Rosenblum, L., Brown, D., Schmidt, G., Julier, S., Baillot, Y., Swan, E., Ai, Z. and Maassel, P. (2018). Military Applications of Augmented Reality. pp.13-25.

Loughran, J. (2018). *Surgeons use augmented reality to 'see inside' the bones of patients*. [online] Engineering and Technology. Available at: https://eandt.theiet.org/content/articles/2018/01/surgeons-use-augmented-reality-to-see-inside-the-bones-of-patients/ [Accessed 4 May 2018].

Mekni, M. and Lemieux, A. (2018). Augmented Reality: Applications, Challenges and Future Trends. *Applied Computational Science*, pp.205-210.

Powell-Morse, A. (2016). *Waterfall Model: What Is It and When Should You Use It?*. [online] Airbrake Blog. Available at: https://airbrake.io/blog/sdlc/waterfall-model [Accessed 4 May 2018].

Rabbi, I. and Ullah, S. (2018). A survey of augmented reality challenges and tracking. pp.34-36.

Schmalstieg, D. (2015). *Trends and Challenges of Augmented Reality*.

Shirer, M. and Torchia, M. (2017). *Worldwide Spending on Augmented and Virtual Reality Forecast to Reach $17.8 Billion in 2018, According to IDC*. [online] IDC: The premier global market intelligence company. Available at: https://www.idc.com/getdoc.jsp?containerId=prUS43248817 [Accessed 4 May 2018].

Vuzix (2018). *Vuzix Blade™ | Augment Reality (AR) Smart Glasses for the Consumer*. [online] Vuzix.com. Available at: https://www.vuzix.com/Products/Blade-Smart-Glasses [Accessed 4 May 2018].

Wild, F., Perey, C., Helin, K., Davies, P. and Ryan, P. (2014). *Mini-Proceedings of the Workshop on Advanced Manufacturing with Augmented Reality*. 1st ed. [ebook] Munich, pp.6-9. Available at: https://pdfs.semanticscholar.org/6e8e/e12c8eb2e2fe4ea935cde138a73fefa441e0.pdf [Accessed 2 May 2018].

Zhang, M. (2018). *Museum of London Releases Augmented Reality App for Historical Photos*. [online] PetaPixel. Available at: https://petapixel.com/2010/05/24/museum-of-london-releases-augmented-reality-app-for-historical-photos/ [Accessed 4 May 2018].

9. APPENDICES

9.1. APPENDIX 1 MACHINE LEARNING MODEL CODE

Table 6 Object Recognition Model Code

Object_Recognition_Model.py

```
from PIL import Image

image_pathname = '/Desktop/Images/Objects.jpeg'

image = Image.open(image_pathname)

image.show()



display_image_pathname = input('Enter image pathname: ')

display_image = Image.open(display_image_pathname)

display_image.show()



from keras.datasets import cifar100



(X_train, y_train), (X_test, y_test) = cifar10.load_data()



index = int(input('Enter an image index: '))

display_image = X_train[index]

display_label = y_train[index][0]

from matplotlib import pyplot as plt
```

```
red_image = Image.fromarray(display_image)

red, green, blue = red_image.split()

plt.imshow(red, cmap="Reds")

plt.show()

from keras.utils import np_utils

new_X_train = X_train.astype('float32')

new_X_test = X_test.astype('float32')

new_X_train /= 255

new_X_test /= 255

new_Y_train = np_utils.to_categorical(y_train)

new_Y_test = np_utils.to_categorical(y_test)


from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers.convolutional import Conv2D, MaxPooling2D

from keras.optimizers import SGD

from keras.constraints import maxnorm


model = Sequential()

model.add(Conv2D(32, (3, 3), input_shape=(32, 32, 3), activation='relu',
padding='same', kernel_constraint=maxnorm(3)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())

model.add(Dense(512, activation='relu',
```

```
kernel_constraint=maxnorm(3)))

model.add(Dropout(0.5))

model.add(Dense(10, activation='softmax'))


model.compile(loss='categorical_crossentropy',
optimizer=SGD(lr=0.01), metrics=['accuracy'])

model.fit(new_X_train, new_Y_train, epochs=10, batch_size=32)


import h5py

model.save('Object_Recognition_Model.h5')
```

Table 7 Object Recognition Test

Object_Recognition_Test.py
from PIL import Image import numpy as np from keras.models import load_model model = load_model('Object_Recognition_Model.h5') input_path = input('Enter image file pathname: ') input_image = Image.open(input_path) input_image = input_image.resize((32, 32), resample=Image.LANCZOS)

```
image_array = np.array(input_image)

image_array = image_array.astype('float32')

image_array /= 255.0

image_array = image_array.reshape(1, 32, 32, 3)

answer = model.predict(image_array)

input_image.show()

print(labels[np.argmax(answer)])
```

Table 8 Object Recognition CoreML Conversion

Object_Recognition_CoreML_Convert.py

```
import coremltools

comreml_model = coremltools.convertors.keras.convert(model,
input_names=['image'], image_input_names='image')

coreml_model.author = 'Brendan Milton'

coreml_model.short_description = 'Predicts objects for use with
independent studies application'

coreml_model.save('/Users/BrendanMilton/Desktop/Object_Recognition
_model.mlmodel')
```

9.2. APPENDIX 2 APPLICATION CODE

Table 9 View Controller (Landing Page) Code

ViewController.swift

```
import UIKit

class ViewController: UIViewController,
UINavigationControllerDelegate, UIImagePickerControllerDelegate {



   @IBAction func objectRecognitionBtnPressed(_ sender: Any) {



       self.performSegue(withIdentifier: "ObjectRecognitionSegue",
sender: self)

   }



   @IBAction func arBtnPressed(_ sender: Any) {



       self.performSegue(withIdentifier:

           "ARViewSegue", sender: self)

   }

   override func viewDidLoad() {

       super.viewDidLoad()

       // Do any additional setup after loading the view, typically from a
nib.

   }



   override func didReceiveMemoryWarning() {
```

```swift
        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

}
```

Table 10 Object Recognition Controller (A-OR) Code

ObjectRecognitionController.swift

```swift
import UIKit

import CoreML

import Vision


class MachineRecognitionController: ViewController {


    // Image picker to grab image to compare ml library against

    private var imagePicker = UIImagePickerController()


    // Implement COREml library to model variable

    private var model = CarRecognition()


    @IBOutlet weak var photoImageView :UIImageView!
```

```
    @IBOutlet weak var descriptionTextView :UITextView!


    // Go back to main screen button

    @IBAction func mainScreenButtonPressed(_ sender: Any) {


        self.performSegue(withIdentifier: "MainScreenSegue", sender: self)

    }


    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from a
nib.


        // Image picker source type will be from photolibrary

        self.imagePicker.sourceType = .photoLibrary

        self.imagePicker.delegate = self

    }


    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }


    @objc func imagePickerController(_ picker:
UIImagePickerController, didFinishPickingMediaWithInfo info: [String
```

```
: Any]) {
        // Dismiss standard animation
        dismiss(animated: true, completion: nil)
        // Get access to picked image then return image to UI image
        guard let pickedImage =
info[UIImagePickerControllerOriginalImage] as? UIImage else {
            return
        }
        // Assign image to photo image view
        self.photoImageView.image = pickedImage
        // Pass image to process image function
        processImage(image :pickedImage)
    }
    // Process image to compare coreML library to image
    private func processImage(image :UIImage) {

        // Converts image to ciImage to pass into vision request handler
        guard let ciImage = CIImage(image :image) else {
            fatalError("Unable to create the ciImage object")
        }
        // Create vision model using coreML model
        guard let visionModel = try? VNCoreMLModel(for:
self.model.model) else {
            fatalError("Unable to create vision model")
        }
```

```
        // Create vision request: takes in vision model to return completion
block

        let visionRequest = VNCoreMLRequest(model: visionModel) {
request, error in

            // Stop process if error present

            if error != nil {

                return

            }

            // Show results including COREML confidence in
VNCLassification Observation

            guard let results = request.results as?
[VNClassificationObservation] else {

                return

            }



            // NOTE FOR FURTHER APPS: IDENTIFIER AND
CONFIDENCE displays labels

            // Display results by mapping over current text, Each item is
called and observation

            let classifications = results.map { observation in

                "\(observation.identifier) \(observation.confidence * 100)"

            }



            // Display results in description text view and join together with
seperator

            DispatchQueue.main.async {

                self.descriptionTextView.text =
```

```
classifications.joined(separator: "\n")
            }
        }
        // Vision request handler: pass in converted ciImage and photo orientation
        let visionRequestHandler = VNImageRequestHandler(ciImage: ciImage, orientation: .up, options: [:])


        // Invoke vision request array (in seperate queue)
        DispatchQueue.global(qos: .userInteractive).async {
            try! visionRequestHandler.perform([visionRequest])
        }


    @IBAction func openPhotoLibraryButtonPressed() {


        // Present image gallery on pressing button
        self.present(self.imagePicker, animated: true, completion: nil)
    }
}
```

Table 11 AR Recognition Controller (A-AR) Code

ARRecognitionController.swift

```
import UIKit

import SceneKit

import ARKit

import Vision


class ARViewController: ViewController, ARSCNViewDelegate  {


    @IBOutlet var sceneView: ARSCNView!

    @IBOutlet weak var statusLabel: UILabel!

    @IBOutlet weak var predictionLabel: UILabel!


    var currentPrediction = "Empty"

    var currentConfidence = "Empty"

    var visionRequests = [VNRequest]()


    // Use queue to make sure coreml requeusts are running smoothly
without effecting other processes

    let coreMLQueue = DispatchQueue(label: "com.rume.coremlqueue")

    override func viewDidLoad() {

        super.viewDidLoad()


        // Set the view's delegate

        sceneView.delegate = self
```

```
        // Create a new scene
        let scene = SCNScene()

        // Set the scene to the view
        sceneView.scene = scene

        // Automatically adjust lightning for ARobjects
        sceneView.autoenablesDefaultLighting = true
        initializeModel()
        coreMLUpdate()
    }
    // Following functions are in charge of running pausing and resuming
AR session
    override func viewWillAppear(_ animated: Bool) {
        super.viewWillAppear(animated)

        // Create a session configuration
        let configuration = ARWorldTrackingConfiguration()
        // Run the view's session
        sceneView.session.run(configuration)
    }
    // Pauses session
    override func viewWillDisappear(_ animated: Bool) {
        super.viewWillDisappear(animated)
```

```
        // Pause the view's session
        sceneView.session.pause()
    }

    // Gets called each time camera changes its tracking state
    // User indicates wether or not ready to recognize object
    func session(_ session: ARSession, cameraDidChangeTrackingState
camera: ARCamera) {

        // Checks value of camera.tracking state
        switch camera.trackingState {
        // Camera could still be initializing
        case .limited(let reason):
            statusLabel.text = "Tracking limited: \(reason)"
        // Camera issue could be halting tracking
        case .notAvailable:
            statusLabel.text = "Tracking unavailable"
        case .normal:
            statusLabel.text = "Tap to add a Label"
        }
    }
    override func touchesBegan(_ touches: Set<UITouch>, with event:
UIEvent?) {
        tapHandler()
```

```
    }


    //Handle user tapping on screen
    func tapHandler(){
        // Get centre point for AR label
        let center = CGPoint(x: sceneView.bounds.midX, y: sceneView.bounds.midY)
        // Feature point to place label
        let hitTestResults = sceneView.hitTest(center, types: [.featurePoint])


        if let closestPoint = hitTestResults.first {
            let transform = closestPoint.worldTransform
            let worldPosition = SCNVector3Make(transform.columns.3.x, transform.columns.3.y, transform.columns.3.z)


            // Create prediction text
            let node = augmentedText(for: currentPrediction)
            sceneView.scene.rootNode.addChildNode(node)
            node.position = worldPosition
        }
    }


func augmentedText(for string: String) -> SCNNode {
```

```
let text = SCNText(string: string, extrusionDepth: 0.01)
let font = UIFont(name: "AvenirNext-Bold", size: 0.15)
text.font = font
text.alignmentMode = kCAAlignmentCenter
text.firstMaterial?.diffuse.contents = UIColor(red: 52/255, green: 152/255, blue: 219/255, alpha : 1.0)
// let confidence = prediction.confidence
text.firstMaterial?.specular.contents = UIColor.white
text.firstMaterial?.isDoubleSided = true

let textNode = SCNNode(geometry: text)
let bounds = text.boundingBox

// Pivot and center when label placed
textNode.pivot = SCNMatrix4MakeTranslation((bounds.max.x - bounds.min.x)/2, bounds.min.y, 0.005)

textNode.scale = SCNVector3Make(0.2, 0.2, 0.2)

let sphere = SCNSphere(radius: 0.005)
sphere.firstMaterial?.diffuse.contents = UIColor(red: 52/255, green: 152/255, blue: 219/255, alpha : 1.0)
let sphereNode = SCNNode(geometry: sphere)

// Label contraints rotation
```

```
        let billBoardConstraint = SCNBillboardConstraint()
        billBoardConstraint.freeAxes = SCNBillboardAxis.Y


        let parentNode = SCNNode()
        parentNode.addChildNode(textNode)
        parentNode.addChildNode(sphereNode)
        parentNode.constraints = [billBoardConstraint]
        return parentNode
    }
    // Initialize coreml model
    func initializeModel() {
        guard let model = try? VNCoreMLModel(for:
CarRecognition().model) else {
            print("Could not load model")
            return
        }
        // Completion handler
        let classificationRequest = VNCoreMLRequest(model: model,
completionHandler: classificationCompletionHandler)
        // Crop photo and scale image from center to pass correct format to
Carrecognition model
        classificationRequest.imageCropAndScaleOption =
VNImageCropAndScaleOption.centerCrop
        visionRequests = [classificationRequest
   }
```

```
    // get vision request result to work with
    func classificationCompletionHandler(request: VNRequest, error: Error?) {
        // Check error
        if error != nil {
            // print error and return so not to continue
            print(error?.localizedDescription as Any)
            return
        }
        // Access results of vision request
        guard let results = request.results else {
            print("No results found")
            return
        }

        // If not nil prediction will contain a classification
        if let prediction = results.first as? VNClassificationObservation {

            // ****** SHOW IN LIVE LABEL USE IN AR LABEL
            // Obtain prediction information for label
            let object = prediction.identifier
            let confidence = prediction.confidence
            currentPrediction = object
            currentConfidence = "\(confidence)"
            DispatchQueue.main.async {
```

```
                self.predictionLabel.text = self.currentPrediction //
self.currentConfidence
            }
            /* let classifications = results.map { observation in
                "\(observation.identifier) \(observation.confidence * 100)"
            } *
        }
    }
    // Continuously called from coreml update to check latest image
    func visionRequest() {
        // Create vision framework request
        let pixelBuffer = sceneView.session.currentFrame?.capturedImage
        if pixelBuffer == nil {
            return
        }
        let image = CIImage(cvPixelBuffer: pixelBuffer!)
        // Perform requests with vision framework
        let imageRequestHandler = VNImageRequestHandler(ciImage:
image, options: [:])
        do {
            // Pass vision request that contains model
            try imageRequestHandler.perform(self.visionRequests)
        } catch {
            print(error)
        }
```

```
    }

    // APP DIFFERENCE due to augmented reality constant image
changing model must update to keep up

    // Constantly update to offer predictions based on what is seen

    func coreMLUpdate() {

        // Runs in custom queue

        coreMLQueue.async {


            self.visionRequest()

            // calls itself to continually run

            self.coreMLUpdate()

        }

    }


    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

    @IBAction func MainScreenPressedButton(_ sender: Any) {


        self.performSegue(withIdentifier: "MainScreenSegue4", sender:
self)

    }

}
```

9.3. Appendix 3 Full Data Comparison Results

Table 12 Full Data Comparison Results

<table>
<tr><th>Object</th><th>Object Recognition

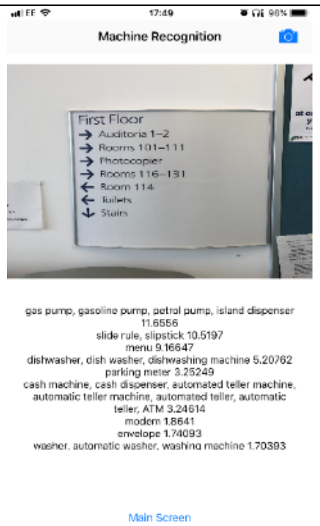
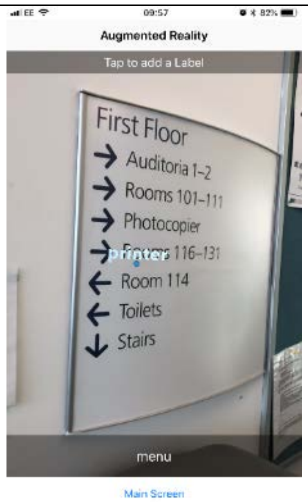
(Using A-OR)</th><th>Augmented Recognition Match

(Using A-AR)</th><th>Augmented Recognition Angle Match

(Using A-AR)</th></tr>
<tr><td>Letter Box</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Letter Box</td><td>match</td><td>match</td></tr>
<tr><td>Large Computer Mouse</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Large Computer Mouse</td><td>match</td><td>match</td></tr>
</table>

Fire Extinguisher			
	Fire Extinguisher	*no match*	*match*
Cash Machine			
	Cash Machine	*match*	*match*
Trophy			

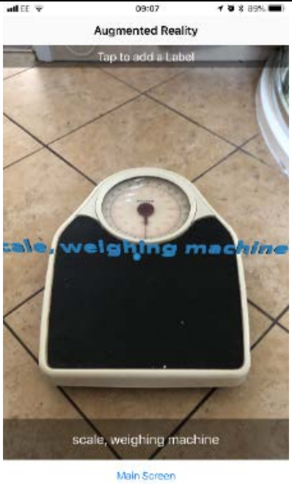
	Trophy	*match*	*no match*
Motorbike			
	Motorbike	*no match*	*no match*
Printer			
	Printer	*match*	*no match*

Chair			
	Chair	*match*	*no match*
Sign			
	Sign	*no match*	*no match*
Large Pot-Plant			
	Large Pot-Plant	*no match*	*no match*

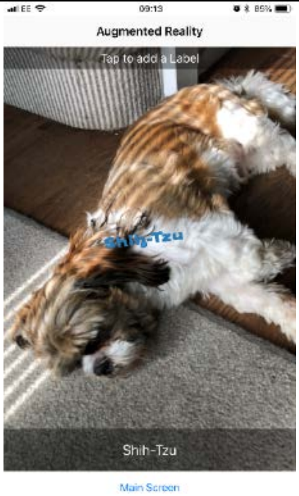
<table>
<tr>
<td>BMX Bike</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BMX Bike</td>
<td>match</td>
<td>no match</td>
</tr>
<tr>
<td>Shampoo Bottle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Shampoo Bottle</td>
<td>no match</td>
<td>no match</td>
</tr>
<tr>
<td>Shell</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

	Shell	*match*	*no match*
Hand Towel			
	Hand Towel	*match*	*no match*
Colour Lamp			
	Colour Lamp	*no match*	*no match*
Window Blind			

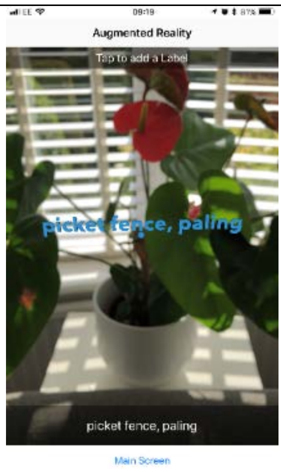
<table>
<tr><td></td><td>Window Blind</td><td>no match</td><td>no match</td></tr>
<tr><td>Small Computer Mouse</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Small Computer Mouse</td><td>match</td><td>no match</td></tr>
<tr><td>Candle Holder</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Candle Holder</td><td>match</td><td>no match</td></tr>
<tr><td>Circle Plant</td><td>
</td><td>
</td><td>
</td></tr>
</table>

<table>
<tr><td></td><td>Circle Plant</td><td>no match</td><td>match</td></tr>
<tr><td>Scale</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Scale</td><td>match</td><td>no match</td></tr>
<tr><td>Toy</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Toy</td><td>match</td><td>no match</td></tr>
</table>

<table>
<tr>
<td>White
Switch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>White Switch</td>
<td>match</td>
<td>no match</td>
</tr>
<tr>
<td>Spotlight
Lamp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spotlight Lamp</td>
<td>no match</td>
<td>no match</td>
</tr>
<tr>
<td>Detailed
Pillow</td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>Detailed Pillow</td><td>match</td><td>match</td></tr>
<tr><td>Dog 1</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Dog</td><td>no match</td><td>no match</td></tr>
<tr><td>Dark Lamp</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Dark Lamp</td><td>no match</td><td>no match</td></tr>
</table>

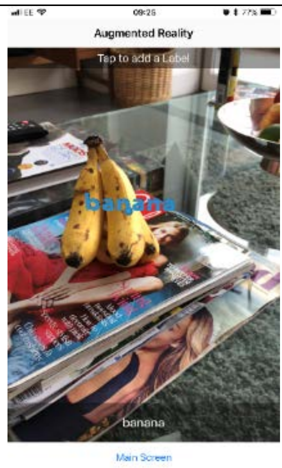
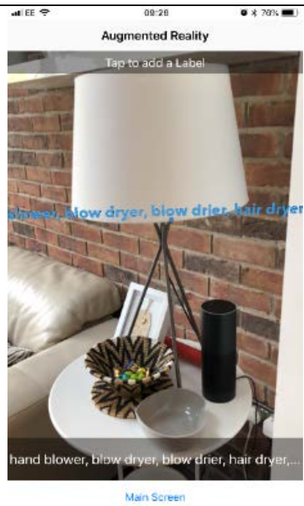
Case			
	Case	*match*	*no match*
Dog 2			
	Dog	*no match*	*no match*
Magnifying Glass			

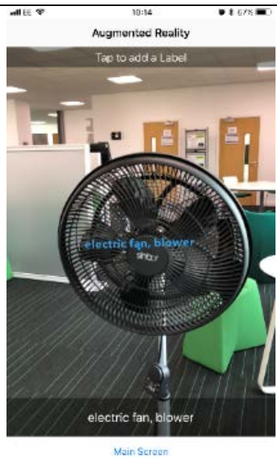
<table>
<tr><td></td><td>Magnifying Glass</td><td>match</td><td>no match</td></tr>
<tr><td>Roses</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Roses</td><td>match</td><td>no match</td></tr>
<tr><td>Small Pot-Plant</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Small Pot-Plant</td><td>match</td><td>no match</td></tr>
<tr><td>Pillow</td><td>
</td><td>
</td><td>
</td></tr>
</table>

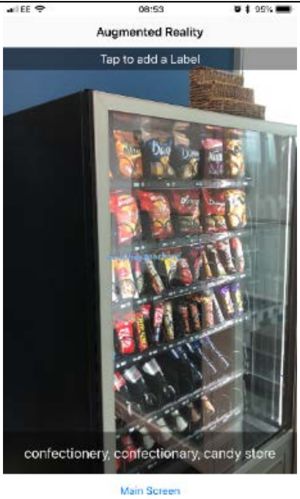
<table>
<tr><td></td><td>Pillow</td><td>match</td><td>no match</td></tr>
<tr><td>Tv Remote</td><td></td><td></td><td></td></tr>
<tr><td></td><td>Tv Remote</td><td>match</td><td>match</td></tr>
<tr><td>Round Switch</td><td></td><td></td><td></td></tr>
<tr><td></td><td>Round Switch</td><td>match</td><td>match</td></tr>
<tr><td>Candle</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><td></td><td>Candle</td><td>no match</td><td>no match</td></tr>
<tr><td>Mask</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Mask</td><td>match</td><td>no match</td></tr>
<tr><td>Couch

Chair</td><td>
</td><td>
</td><td>
</td></tr>
<tr><td></td><td>Couch Chair</td><td>match</td><td>match</td></tr>
</table>

Banana			
	Banana	*match*	*match*
White Lamp			
	White Lamp	*match*	*no match*
Grey Switch			

	Grey Switch	*match*	*no match*
Kettle			
	Kettle	*match*	*no match*
Fan			
	Fan	*match*	*match*
Monitor			
	Monitor	*match*	*no match*

<table>
<tr>
<td>Vending Machine</td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td></td>
<td>Vending Machine</td>
<td>match</td>
<td>no match</td>
</tr>
</table>