

Feed Me: an in-situ Augmented Reality Annotation Tool for Computer Vision

Cedrick K. Ilo

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science Application

Nicholas F. Polys, Chair

Denis Gracanin

Joseph L. Gabbard

May 6, 2019

Blacksburg, Virginia

Keywords: Augmented Reality, 3D User Interface, Computer Vision Training

Copyright 2019, Cedrick K. Ilo

Feed Me: an in-situ Augmented Reality Annotation Tool for Computer Vision

Cedrick K. Ilo

(ABSTRACT)

The power of today's technology has enabled the combination of Computer Vision (CV) and Augmented Reality (AR) to allow users to interface with digital artifacts between indoor and outdoor activities. For example, AR systems can feed images of the local environment to a trained neural network for object detection. However, sometimes these algorithms can misclassify an object. In these cases, users want to correct the model's misclassification by adding labels to unrecognized objects, or re-classifying recognized objects. Depending on the number of corrections, an in-situ annotation may be a tedious activity for the user. This research will focus on how in-situ AR annotation interfaces can be improved by providing different visual cues to the user, and their usability of the Feed Me voice and gesture interface.

Feed Me: an in-situ Augmented Reality Annotation Tool for Computer Vision

Cedrick K. Ilo

(GENERAL AUDIENCE ABSTRACT)

The power of today's technology has allowed the ability of new inventions such as computer vision and Augmented Reality to work together seamlessly. The reason why computer scientists rave so much about computer vision is that it can enable a computer to see the world as humans do. With the rising popularity of Niantic's Pokemon Go, Augmented Reality has become a new research area that researchers around the globe have taken part in to make it more stable and as useful as its next of kin virtual reality. For example, Augmented Reality can support users in gaining a better understanding of their environment by overlaying digital content into their field of view. Combining Computer Vision with Augmented Reality could aid the user further by detecting, registering, and tracking objects in the environment. However, sometimes a Computer Vision algorithm can falsely detect an object in a scene. In such cases, we wish to use Augmented Reality as a medium to update the Computer Vision's object detection algorithm in-situ, meaning in place. With this idea, a user will be able to annotate all the objects within the camera's view that were not detected by the object detection model and update any in-accurate classification of the objects. This research will primarily focus on visual feedback for in-situ annotation and the user experience of the Feed Me voice and gesture interface.

Dedication

I dedicate this thesis to my dear friend Jasmine Davis. You inspire me to become a better person daily. Without your friendship, none of this would have been possible.

Acknowledgments

I would first like to thank my thesis advisor Dr. Polys of the Computer Science Department and Advanced Research Computing center at Virginia Tech. His door was always open whenever I ran into a problem or had a question about my research or writing. His guidance consistently kept me on track in writing this paper. Dr. Polys allowed this thesis to be my own work of art: giving me the flexibility and creativity in designing the application, while keeping me cognizant of the scope and completion of this paper. I would like to acknowledge Dr. Gracanin of the Computer Science Department at Virginia Tech as the second reader of this thesis, and I am gratefully indebted to the stern and compassionate direction in helping along my path in forming a solid background. Though Dr. Gracanin only served as a committee chair, he didn't treat me as such. Dr. Gracanin, when he was able went out of his way to meet with me to help in any way he could. For that, I am forever grateful. I would like to acknowledge Dr. Gabbard of the Industrial and Systems Engineering Department at Virginia Tech as the third reader of this thesis, and I am grateful for your creative direction and knowledge of usability. Every time I was able to meet with you concerning the application development process you always gave me fresh ideas and insightful advice. I would also like to acknowledge Nicholas Barlow for his expert knowledge in Unity3D. Thank you for providing me with a lower level of understanding of how the platform works. Finally, I must express my very profound gratitude to my brother Cory Ilo for pushing me always to be better, to my mother for being my anchor to reality, to my dear friend Jasmine Davis for providing me with unfailing support and continuous encouragement throughout my years of study. And to my friends, I made here at Virginia Tech for the support through thick and thin. This accomplishment would not have been possible without any of them. Thank you.

Contents

- List of Figures ix

- List of Tables xii

- 1 Introduction 1**
 - 1.1 Motivation 3
 - 1.2 Problem Statement 6

- 2 Background 9**
 - 2.1 Computer Vision 9
 - 2.2 Annotation Tools 10
 - 2.3 Augmented Reality 13
 - 2.4 3D User Interfaces in AR 14
 - 2.5 Full Stack Development 17

- 3 Prototype 19**
 - 3.1 Design 19
 - 3.1.1 Activity Design 19
 - 3.1.2 Information Design 20

3.1.3	Interaction Design	21
3.2	Implementation	22
3.2.1	Front-end	23
3.2.2	CGI Container	30
3.2.3	Back-end	31
4	Experiment	32
4.1	Scope & Hypothesis	32
4.2	Methods	33
4.2.1	User Study	33
5	Results	43
5.1	Time to Completion	43
5.2	Number of Box Repositions	44
5.3	Number of Box Relabelings	45
5.4	NASA TLX	46
5.5	Exit Survey	47
6	Conclusion	52
6.1	Findings	52
6.2	Guidelines	54
6.3	Challenges & Future Work	55

Bibliography	57
Appendices	63
Appendix A VT IRB-18-1113	64
A.1 Authorization Letter	64
A.2 Experiment Hardware	64
A.3 Experiment Software	66

List of Figures

1.1	This figure is an example of the spatial mapping feature for Microsoft Hololens.	2
1.2	COCO dataset listing of the predicted labels the CV can detect within a picture.	6
1.3	Here is a picture from the COCO dataset where the algorithm has detected particular objects within the picture. Noticeably, there are a couple of objects in the scene that is not detected. Using AR, a user can take a picture(s) of this scene and then, using in-situ annotation, the user can assist the algorithm by annotating the unidentified object(s) in the scene with its associated label and save the new annotation from being used later in re-training the CV neural net.	8
2.1	Here is a picture of an implemented Vatic annotation tool that utilizes vatic.js to create an environment for high school students can detect fish in a pond.	12
3.1	Full stack communication pipeline.	23
3.2	When a user takes a photograph, the image is sent to the Feed Me back-end server to be processed by the ImageAI object detection algorithm.	24
3.3	When a user wishes to update an existing object, issue the voice command will send a voice request to the Feed Me back-end to turn the speech to text using Google's speech to text API.	26

3.4	Using the Feed Me system, this is an example of the user adding a new detection to the scene.	27
3.5	Using the Feed Me system, this is an example of the user ability to size the detection box.	28
3.6	Using the Feed Me system, this is an example	30
4.1	Example of a high/low-density scene with black detection boxes.	34
4.2	Example of a high/low-density scene with colored detection boxes.	35
4.3	This picture represents the ground truth capture area for the user experiment.	36
4.4	This picture represents a participant in a trial of a high-density setting.	39
4.5	This picture represents a participant in trial tasked to reposition a detection box.	41
4.6	This picture represents a participant in trial tasked to relabel a detection box.	42
5.1	Descriptive statistic for Time Completion.	44
5.2	Time to Completion for density x color.	45
5.3	Descriptive statistic for Number of box repositions.	46
5.4	Number of Box Repositions for density x color.	47
5.5	Descriptive statistic for Number of Box Relablings.	48
5.6	Number of Box Relabelings for density x color.	49
5.7	Mean error distribution of the NASA TLX. Error bars show Standard Deviation.	50

5.8 Mean error distribution of the Exit Survey. Error bars show Standard Deviation.	51
---	----

List of Tables

4.1	This table shows the counterbalancing mechanism used in the user experiment, where the density for each trail changes per prototype feedback. . . .	40
-----	---	----

List of Abbreviations

AR Augmented Reality

CV Computer Vision

DL Deep Learning

HUD Heads-up Display

IoT Internet of Things

ML Machine Learning

MR Mixed Reality

NLP Natural Language Processing

TF TensorFlow

VR Virtual Reality

NLP is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages.

AR is an interactive experience of a real-world environment where the objects that reside in the real-world are "augmented" by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory, and olfactory.

CV is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

MR sometimes referred to as hybrid reality, is the merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time. Mixed reality takes place not only in the physical world or the virtual world, but is a mix of reality and virtual reality, encompassing both Augmented Reality and augmented virtuality via immersive technology.

IoT is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

TF is a machine learning system that operates at large scale and in heterogeneous environments. Tensor-Flow uses dataflow graphs to represent computation, shared state, and the operations that mutate that state.

ML is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

DL is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks.

VR is an interactive computer-generated experience taking place within a simulated environment. It incorporates mainly auditory and visual feedback, but may also allow other types of sensory feedback.

HUD is any transparent display that presents data without requiring users to look away from their usual viewpoints. The origin of the name stems from a pilot being able to view information with the head positioned "up" and looking forward, instead of angled down looking at lower instruments.

Chapter 1

Introduction

In today's technology such as smartphones and smart tablets, companies and investors are using Augmented Reality (AR) in unique ways to connect their users to the data that they consume in their daily lives. AR is an interactive experience of a real-world environment where the digital objects that the user sees in the real-world are "augmented" by computer-generated perceptual information. AR encompasses multiple sensor modalities, including visual, auditory, haptic, somatosensory, and olfactory. The term "Augmented Reality" was first coined by Boeing's researcher Tom Caudell after the first ever head mounted display created by Ivan Sutherland in 1968. More recently, AR is a heavily researched area in computer science, and computer engineers are finding ways to make it more acceptable to the general public. The real question is what captivates users to use AR? Is it the input modalities the system enables the user or is it the 3D exploration aspect of AR? No matter the platform the user runs, the program should always give the user the ability to interact with data embodied and in context [26].

A current issue with AR is its ability to register or correct the alignment of the virtual world with the real one. By utilizing the Microsoft HoloLens's ability to spatial map the environment; this feature may negate some registration effects

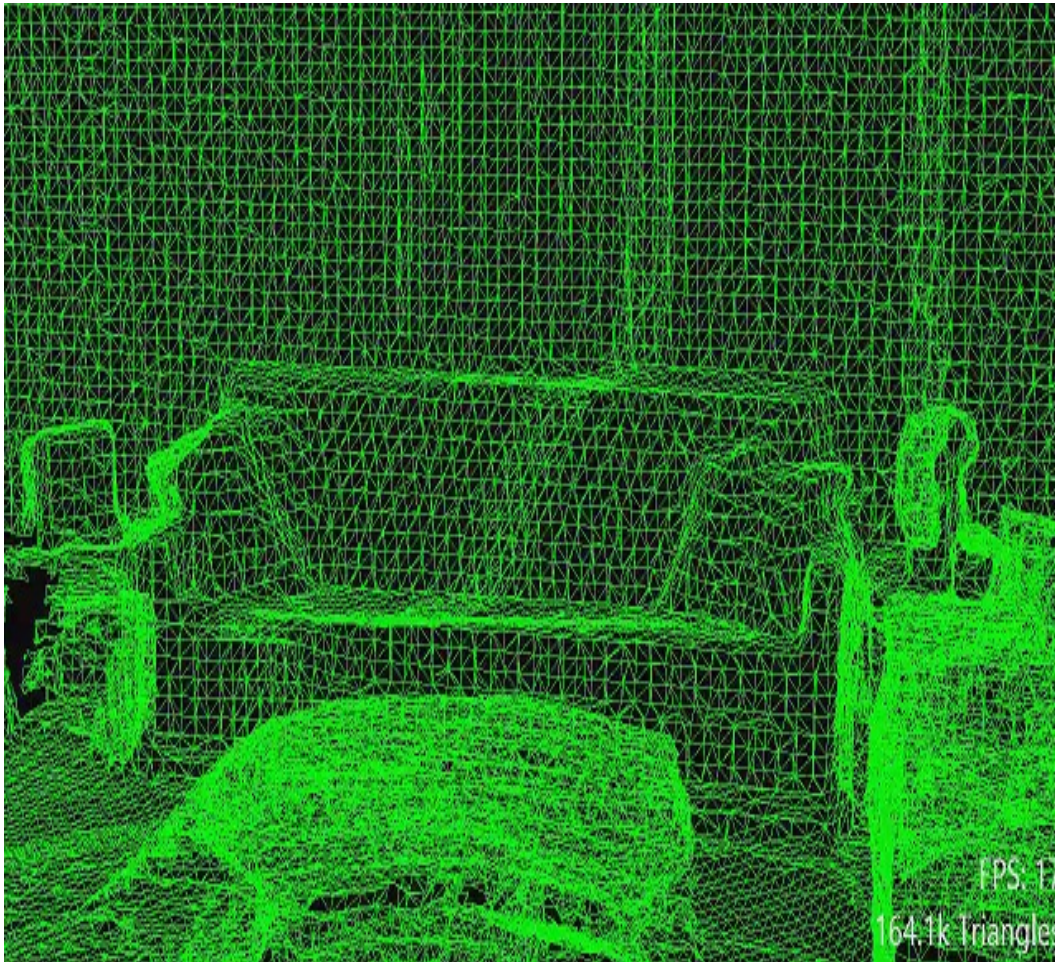


Figure 1.1: This figure is an example of the spatial mapping feature for Microsoft HoloLens.

a user may endure during registration. This spatial mapping feature provides a detailed representation of the real-world surrounding the Microsoft HoloLens and defines regions of space for in-situ spatial reference [1].

AR systems can also include features of Computer Vision(CV) for feature and object detection. CV is a subset of machine learning. CV allows the computer to gain a higher level of understanding from depth cameras and registration techniques.

An aspect of AR I find fascinating is the creative control of how digital information can be display to users. Concerning the real world environment, a proper AR user interface could create AR workstations that will become more flexible than 2D workstations. AR workstations will give users a more embodied interface with which to work with information. Current challenges with AR that delay widespread use of this technology include: registration, resolution, and applications. Although there are applications that use AR for maintenance in the industry and situational awareness in the military, AR is still being researched to make it useable across different demographics. It is imperative to test these new emerging technologies through time and usability benefit for all end users when implementing AR as a service [4].

1.1 Motivation

Movies of our generation seem to fortell the future development of new technologies. AR in popular movies is perceived as a hyper-reality. Hyper-reality introduces tons of information consumable by a user in a 3D 360-degree space. The information retrieved by the AR system could utilize communicative technologies like the Internet of Things (IoT) to make this happen. In this hyper-reality, AR is aware of surrounding objects and with context, has an understanding of its environment based on the user's orientation to effectively display this information without any disruption to the user's experience. For this to happen, the AR system needs to identify an object correctly, and portray this information to the user in a clear and informative way.

Coupling computer vision with Augmented Reality will indeed make this system possible. The power of current state-of-the-art computer vision algorithms is strong enough to detect objects of different categories. Current CV algorithms integrated into consumable products only have a high-level understanding of these objects. A high level of understanding of these objects group objects into separate categories but does not allow distinction between objects of the same category. The term disambiguate in this context means to distinguish between two or more objects. In this area, there is an untapped area of marketing and interaction designs that could come about if an AR system were able to disambiguate content from two objects of the same nature.

Imagine walking to a bus stop, and an AR system was able to filter the sign for a bus route that a user wants to take. This systems then enables the user to interface with the bus stop sign giving that user user-specific actions that are only associated with that bus route. There is a means to exploit this area by using AR as a medium to aid the neural net during object identification by adding a human in the loop. The human in the loop will aid the neural net in its disambiguation of an object thus providing training data which will improve detection performance in future scenes. This research presents an in-situ annotation system in AR that enables users to reclassify inaccurate detections of a CV algorithm by allowing the user to make changes to the detected objects in both three-dimensional space and two-dimensional image space.

In-situ annotation in AR refers to the medium in which the user annotate scene referenced images in the AR front-end. With AR allowing digital information in the virtual world to overlay the real-world, using in-situ annotation in AR would

give the user the ability to update any errors presented by the algorithm quickly. With the rising popularity of AR browsers, users can now to explore particular points of interest in a mixed reality setting. An example of a potential system that would benefit the use of in-situ AR annotation would be Google's recent development called Google Lens [19]. An image recognition mobile app, that when directed to a physical object can identify that object and retrieve relevant search information back to the user. Adding an AR layer of interactivity to this type of architecture could help users get more accurate results. A better experience using this technology will help users understand the capabilities of AR and how AR as a whole, can assist users with mundane activities.

This thesis aims to demonstrate and evaluate an AR user interface as an annotation tool to aid the disambiguation of objects from a CV algorithm in-situ. Our Feed Me AR annotation tool will allow users to edit labels of detection boxes and reposition detection boxes to test the usability of the system. This initial investigation would become the foundation of smarter interactions within AR. Ideally, the created system will have the capabilities to "feed" updated image or video information to the CV algorithm to make better detections in the next cycle. Accurate detections from the updated neural net will lead to not only smarter interaction designs within AR but also promote a more enriching user experience to the human user. By allowing humans to "correct" falsely labeled predictions from a CV algorithm, we could support more the research into the training of neural nets and CV object detection algorithms. A scenario in which using this system could be beneficial is for annotation datasets like COCO Dataset. (See Figure 1.2 and Figure 1.3).



Figure 1.2: COCO dataset listing of the predicted labels the CV can detect within a picture.

1.2 Problem Statement

The power of today's technology has enabled the combination of Computer Vision (CV) and Augmented Reality (AR) to allow users to interface with digital artifacts between indoor and outdoor activities. For example, AR systems can feed images of the local environment to a trained neural network for object detection. However, sometimes these algorithms can misclassify an object. In these cases, users want to correct the model's misclassification by adding labels

to unrecognized objects, or re-classifying recognized objects. Depending on the number of corrections, in-situ annotation may be a tedious activity for the user. This research will focus on how in-situ AR annotation interfaces can be improved by providing different visual cues to the user, and their usability of the Feed Me voice and gesture interface.

Chapter 2

Background

2.1 Computer Vision

In this project, we will be using TensorFlow (TF) as our backbone to optimize and train various deep learning algorithms for object detection within images. TensorFlow is a machine learning system that works at a larger scale for different environments. TF has the power to map nodes of data flow into a cluster within a machine across multiple computational devices. With this computational flexibility, we decided to use this ML library in our back-end [2].

To handle the object detection within the images, we will run a RetinaNet model in our back-end. Based on Lin et al.'s research, running a one-stage detector could be potentially faster. Depending on the focal loss for a dense object detection, we plan to use our in-situ method to correct any mistakes. Due to the RetinaNet model running faster, we will utilize its speed to increase our full stack pipeline [22].

Another object detection algorithm with strong performance in object detection is Redmon et al.'s You Only Look Once classifier(YOLO). YOLO is a new approach for real-time object detection that uses a single neural net that detects

boxes and probability scores from images in one evaluation. In the initial investigation to test the real-time detection, compared to a Fast R-CNN and DPM, YOLO made fewer background mistakes. Future implementations with the Feed Me system could involve real-time detection using video footage, and we plan to test YOLOv3 with our system [12, 32, 33, 36].

Actively running a deep neural network can be computationally intensive and memory intensive. In order to get the best performance for real-time object detection, we will look into applying a smaller version of YOLO known as Tiny-YOLO or YOLO-Lite to run locally on to the Microsoft Hololens device [21, 24, 30, 39, 40].

2.2 Annotation Tools

In Hanbury's survey, he stresses the point that in order to evaluate image annotations from an object detection algorithm, correctly annotated images with text describing the image is required. Hanbury evaluates three different annotation techniques: free text annotation, keyword annotation, and annotations based on ontologies. The experimental design for the Feed Me application will implement a free text annotation by the use of voice [13].

LabelMe is an online annotation tool that allows the sharing of images and their annotations. The functionality of this tool allows users to draw polygons, query images and browse the database. LabelMe was designed for object class recognition and to learn detailed information about the objects embedded in the

scene. LabelMe was meant to contain a database for high-quality labeling since most systems supply just captions of an object being present within a picture. Russel et al. use LabelMe as an online medium to gather ground truth labels for images by allowing the community to draw controls points over objects in the scene to identify the object. The LabelMe system uses crowdsourcing to allow other users to correct other labels if identified incorrectly. The storing mechanism of LabelMe stores its labeled results in an XML file format making data accessible to be used in other platforms. This central idea is the blueprint experimental design of the user interface for the Feed Me application [35].

In Kim et al.'s paper she introduces a framework that allows the annotation of real-world objects within an indoor location. The framework use case is to allow non-experts to capture room dimensions and annotate locations and objects within the scene to link virtual information to the real space represented by an approximated box. The framework gives the user the support of object-based space to space registration to have a multi-dimensional view of the objects generated by the system. The Feed Me experimental design will also focus on making interactions with the user interface to be efficient for all users no matter their technical background [17].

The experimental design for Feed Me utilizes the human perspective of annotations within a scene which would be similar to Vondrick et al.'s research for efficiently scaling up crowdsourced video annotations. Vondrick et al. designed a user interface known as Vatic, to annotate frames in a video sequence and ran a user study to evaluate the aspects of the system while simultaneously minimizing the cognitive task load on to the user. Vondrick et al.'s research focused on

Shot 2019-04-02 at 10.58.00 PM.png Shot 2019-04-02 at 10.bb
 Welcome To Trout Finder! Lets see if you can spot me!

1. Lets start off by selecting a video where I've been caught swimming!

pool10_1920.mp4

Video dimensions determined: 1920x1080

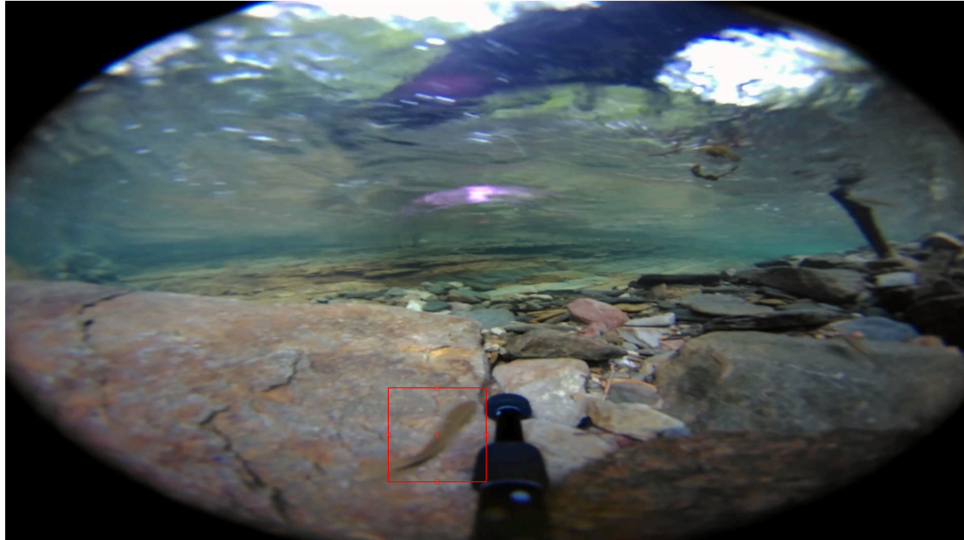
Extraction completed. 3644 frames captured.

2. To create a new bounding box, first click 'n' (for new), and then left click on two locations in the video corresponding to the corners of the box.

Tip: Use the spacebar to play/pause the video, and the left and right arrows to navigate frame by frame.

Tip: The visibility of each object can be toggled with its visibility checkbox under the video.

Tip: Zoom in with your browser to place the bounding boxes more accurately.



3. the vatic-compatible XML annotations file.

Figure 2.1: Here is a picture of an implemented Vatic annotation tool that utilizes vatic.js to create an environment for high school students can detect fish in a pond.

exploiting the specialized skill sets required for using such an annotation tool. We seek to find if our user interface is useful no matter the expertise of an individual as well(See Figure 2.1) [37].

2.3 Augmented Reality

Augmented Reality coupled with computer vision can provide valuable information for users in a variety of contexts. Computer vision based registration enables the AR system to locate objects in the user's perspective and accurately overlay virtual information on top of the view [14]. Though there have been reports of registration errors in AR that deals with system delay and tracker error [15], Lepetit et al.'s research uncovers that the use of camera registration for AR system can make registration more stable [20]. Frantz et al. also uses this idea of using the front-facing camera of the Microsoft HoloLens for neuronavigation for neurosurgery for registration using tracking data [9]. Since this experiment uses a Microsoft HoloLens to interact with our user interface, we take into account possible registration and tracking errors.

Julier et al.'s paper tries to tackle the problem of displaying tasked oriented information to a user by analyzing different information filtering processes and creating a hybrid of them to decrease irrelevant information. This system, in theory, would be used in the real world for complex mappings and information display. The information filtering approaches they looked at started with distance-based and visibility-based filtering. These approaches gave the user too much information that a user would need. To decrease distance-based filtering, they tried a spatial model interaction, which would be aware of the virtual environment that determines the object visibility and the ability to interact with it, which lead to a rule-based filtering approach that used the Knowledge-based Augmented Reality for Maintenance Assistance (KARMA) system. This system took into consideration the user's position and orientation, and inter-object oc-

clusion relationships to determine whether and how objects should be displayed, highlighted, or labeled. The KARMA system had some scalability concerns if more data came into play which could cause the system to be computationally expensive [16].

Based on these approaches they came up with a hybrid of the three-filtering process where the algorithm required the representation of the user's objects and goals to rank the objects to those goals to calculate the focus. The first step of their approach is initialization, given the knowledge of the user's objectives and goals, they calculate the focus for each object and is updated when a user's objective change. The second step is the culling that uses the spatial model of interaction to eliminate all objects that are not in the user's focus. Lastly, they refine the information by applying higher-order decision logic. The premise of this initial research is the future direction of the Feed Me application. Applying our retraining of detected objects with Julier et al.'s filtering process would create a design space for new interactions based on what an AR system sees in an environment [16].

2.4 3D User Interfaces in AR

Optical see through AR introduces several information design challenges: First, to match the perspective of digital imagery and real-world perspective, the second would be matching the lighting between the digital imagery and real world and third to sustain contrast between digital imagery and the real world to support legibility. Gabbard et al. provided some recommendations for outdoor AR;

while our study is indoors, we will follow their display guidelines, such as high color contrast text color for labels and color contrast distinction [11].

Today's computers are becoming more complex in today's technology, and it is imperative to take into account how humans interface with new technology. Interfacing with AR, in particular, has a multitude of interaction designs for a user to interface with the technology. In Rautaray and Agrawal's research, they provided an analysis of comparative methods concerning vision-based hand gesture interactions. Since this experiment uses a Microsoft HoloLens to deploy the Feed Me system, we will take into account how the user interfaces with the 3D UI using the HoloLens [31].

In order for the user to disambiguate effectively disambiguate objects efficiently, there should be a precise and naturalistic selection interaction. To address the issue of a precise selection of an object in a dense scene Mossel et al.'s research shows that a single selection interaction could increase performance in terms of speed and accuracy [27]. For the common task of selection, Feed Me will utilize a raycasting selection technique to aid users in the precise distinction of a detected object. Raycasting will be used as visual feedback for selection in the Feed Me user interface [3, 29, 38].

"FingARtips," is an AR and VR system that utilizes ARToolkit's computer vision tracking system coupled with a haptic feedback device on a user's hand for gesture-based manipulation of virtual objects in AR. The premise of Buchmann et al.'s research was to conduct a user study to measure the 3D input modalities on near objects to test the cognitive tasks of grabbing, pressing, dragging

and releasing virtual objects in their urban planning example. Buchmann et al. focused on near objects because they believed interactions would be more intuitive than distant objects. Buchmann et al. thought that haptic feedback was crucial in their experiment because it allowed users to feel the buildings they were moving in the augmented scene. To give users a better perspective of the object they were manipulating, Buchmann et al. included a button to switch the system between AR and VR. Essential takeaways from Buchmann et al.'s user study was they encountered many tracking issues which suggest finding a better alternative for marker-based tracking. The experimental design of Feed Me will test the usability of annotation manipulation in an AR HUD. As well as give the user the ability to see results of the CV algorithm to aid in positioning of detected objects [5].

An effective user interface should also have the capability for speech recognition. Enabling voice commands allow users to interact with the user interface naturally. Hao et al.'s research stated that signal quality is critical when using speech recognition in VR/AR systems [25]. In Liu et al.'s research, Liu et al. performed a technical evaluation of speech recognition using the Microsoft Hololens. Results of Liu et al.'s experiment showed 65 - 75 percent agreement rates for user-defined and system designed voice commands [23]. In Eckert et al.'s research, they used speech recognition for a selection interaction for object detection to support individuals with visual impairment or blindness. Results from their research showed that 83 percent of voice commands were detected correctly [7]. Also for an AR system using voice commands, commands should be precise and straightforward stated by Zimmer et al. in their user study [41]. In this experiment, our Feed Me system will use voice command to allow users

to interface with the UI more naturally.

In our user study, we plan to evaluate two different prototype designs to understand the perceptual costs and benefits to human performance better. As Gabard and Edward's research shows, user-based experimentation in Augmented Reality (AR) can give us insight on how to improve our application for future use. We will run a two factor, two-level within-subjects experiment [8, 10].

2.5 Full Stack Development

"Deep Decision" is a framework that ties the back-end (server) and front end (mobile device) to be able to understand when to use locally deep learning models (tiny non-computationally intensive models) to remote deep learning models (bigger more computationally expensive models) for machine learning and Augmented Reality to effectively switch between local and remote models. Jiasi et al. look at the complex interaction between model accuracy, video quality, battery constraints, network data usage, and network conditions to determine an optimal switching mechanism. Current machine learning algorithms for mobile devices perform poorly. For example, TensorFlow's Inception deep learning model can process about one video frame per second on a typical Android device which prevents real-time analysis. Typical processing times are approximately 600 ms, which is less than 1.7 frames per second. When the cloud is introduced to run deep learning, there becomes a problem on the network depending on its availability. Jiasi et al. propose a trade-off between accuracy and latency because network latencies are often much longer than the computational laten-

cies; it would be essential to optimize the offloading decision along with the local processing. Since neural nets are the state-of-the-art in computer vision for object detection, Jiasi et al. used a convolutional neural network (CNN) called YOLO for object detection in AR. Jiasi et al. used YOLO because it is optimized for processing video streams in real-time and has a feature that scales with the resolution of the given video data. In this experiment, we implement a full stack application using a deep learning library in our backend for object detection while using an AR device as our front-end. Considering the limitations using head-mounted displays, the future direction for the Feed Me full stack application will consider this constraint to make a faster experience for users [6].

In the idea of creating a full stack pipeline for in-situ annotation, we refer to Langlotz et al.'s paper called Sketching up the world: in situ authoring for mobile Augmented Reality. The premise of Langlotz et al.'s work was to enable inexperienced users to let them become the creators of their authoring content to be made available to the public. This idea is parallel to the Web 2.0 concept that the users are to become the content creators. Langlotz et al. provided five key factors that would that pushed AR to AR 2.0. These key factors consist of having a low-cost platform that combines AR display, tracking, and processing, as well as having the notion of mobility to realize AR in a global space, large scale AR tracking that works in real time and ease of use for creating content in the AR space and a back-end infrastructure for distribution of AR content and applications. The essential takeaways that we intended to use are tying the front-end AR application to a back-end service to store and retrieve in-situ data created by users of the Feed Me application [18].

Chapter 3

Prototype

3.1 Design

3.1.1 Activity Design

In order to make a useable system to handle the associated tasks for image annotation, we analyzed everyday tasks in existing annotation tools. For example:

- To get annotations, users will take a picture with the Microsoft Hololens to the CV object detection algorithm that sends back positions of the identified objects within the scene.
- Users may want to reposition detections boxes on the image plane reference using hand gestures.
- If the CV algorithm misclassifies a detected object within the image reference, a user then should be able to relabel the object with the corrected description of the detected object.
- There are times when the CV do not classify a specific region within the image plane. At these times a user should be able to create new detection boxes with its associated label.

- Users should have the ability to rescale new detection boxes as well as edit existing detection boxes.

In the user experiment, the user will focus on relabeling and repositioning activities.

3.1.2 Information Design

In order for the user to have constant access to the workspace in-situ, the Feed Me system user interface will draw objects in the Heads Up Display (HUD). Thus, with the Microsoft HoloLens, the information will display in front of the user's field of view relative to the user's head pose. As the user pivots their head in the environment, the information will be updated and repositioned back in front of the user's field of view. The rationale behind the HUD following the user is that it enables the user to reference the real world scene and the virtual world image reference at the same time. The advantage of this HUD design is that it will allow users to refer back to the real world scene to gather more information about the scene while the system is idle. The disadvantage of the HUD is that the image reference may occasionally occlude the real world scene when referring to objects.

At the top left area of the user interface, users presented with five buttons(+, Update, Delete, Server Results, Export). Interactions within the system that require a wait time activate loading animations; these animation provides feedback to the user to indicate that the information is in transportation. During the generation of detected objects, detection boxes overlay over on top of the

HUD workspace image plane reference. Feed Me also provides visual feedback on the current object manipulation mode informing the user if they can place a detection box or rescale a detection box.

The bounding boxes from object detections are drawn with any number of colors. Our first investigation will be into the following trade-off:

- Black Bounding Boxes
 - +: Black boxes are good for high contrast in a well-lit environment.
 - -: In a crowded scene, it may be difficult to distinguish bounding.
- Colored Bounding Boxes
 - (+): Different colors can distinguish bounding boxes in a crowded environment.
 - (-): Depending on the environment, colored boxes may blend into the background.

3.1.3 Interaction Design

Menu

There are multiple ways a user can interact with the Feed Me user interface. There are a total of five buttons, and ten voice commands that allow a user to get information about a new synthesized scene. Each button has its unique job as most annotation tools persist. The use of voice commands in this environment can achieve the same results as an associated button, as well as executing other features without having to press a button to get a desired result.

The selection process in the Feed Me user interface is simple, using the open source Hololens Tool-kit allows selection with either the hand or clicker to have similar characteristics a left-button mouse click. The selection mechanism enables the user to interact with all buttons in the user interface as well as manipulating the position of the detection boxes. The first click on a detection box in the scene latches the box to the user appointed head movement. By raycasting, a user can naturally place a detection box in any region on the image reference. The second click detaches the box from the user head movement and places the box in the desired location.

Voice Labeling

Using Feed Me's interface, specific actions can only activate by the use of voice commands. These specific voice commands relate to scaling the size of a detection box, adding new detection boxes, updating a JSON object in COCO dataset format, deleting an object and relabeling an object. The Microsoft Hololens microphone is always on to receive basic voice commands from a keyword dictionary. However, relabeling may include any arbitrary word. Therefore, a speech to text component will be required.

3.2 Implementation

Feed Me is a full-stack application. The Feed Me front-end handles all the actions associated with the manipulation of objects through disambiguation. The Feed Me back-end handles all of the computational processes (See Figure 3.1).

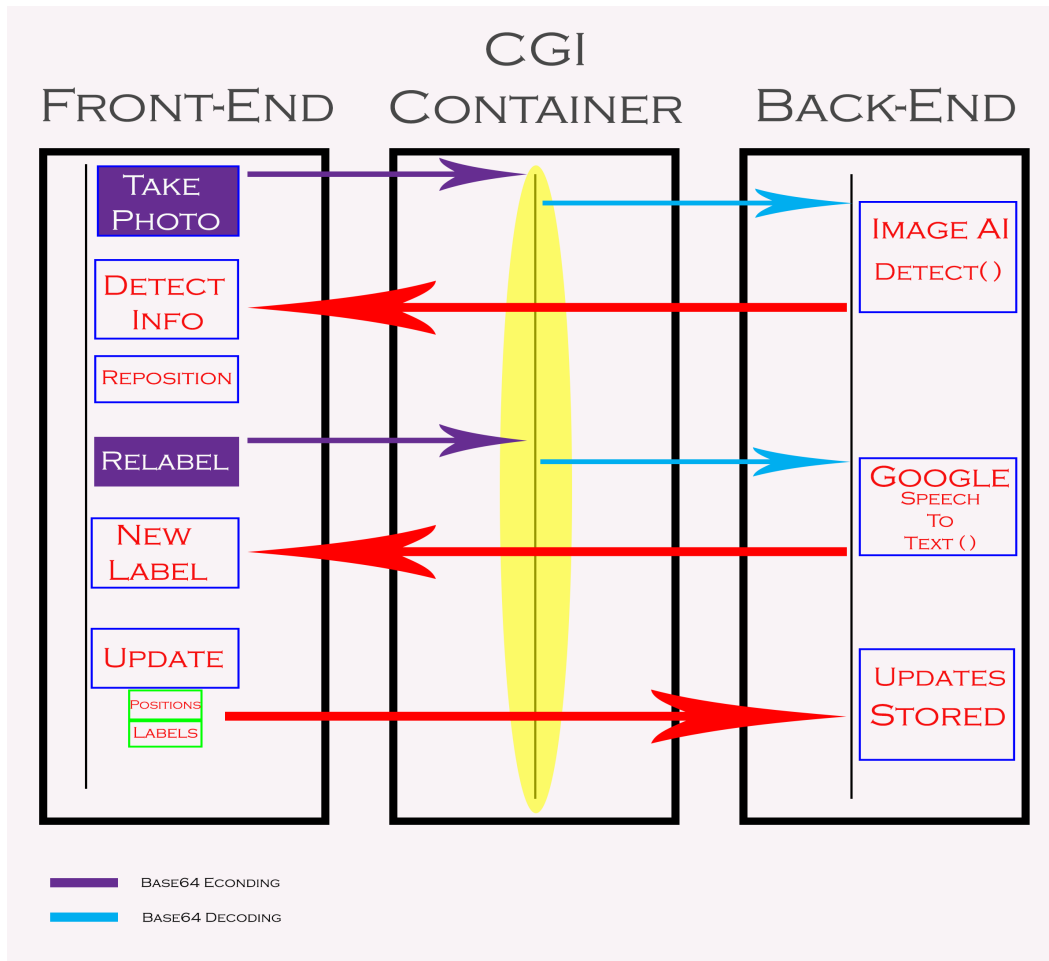


Figure 3.1: Full stack communication pipeline.

3.2.1 Front-end

In the act of disambiguation objects, a user would only interact with the front-end of the application. The Feed Me user interface presents users with several tasks that are somewhat similar to any two-dimensional annotation tool. The front-end user can take a photo, update an existing detection or add new detections to the scene the algorithm may have missed, see server results for alignment ground truth and export updated corrections.

“Take Photo”

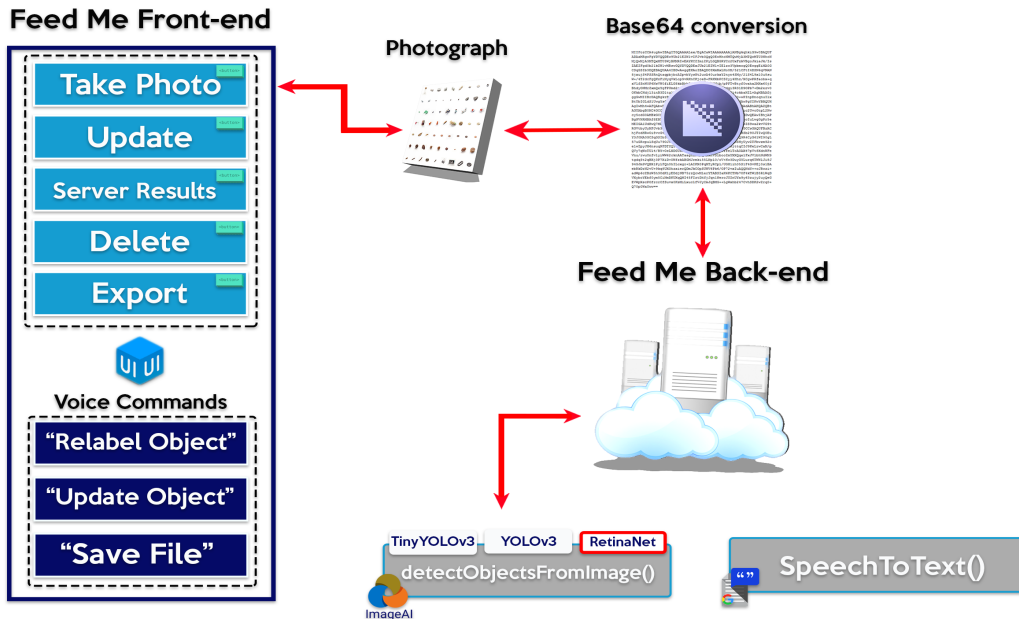


Figure 3.2: When a user takes a photograph, the image is sent to the Feed Me back-end server to be processed by the ImageAI object detection algorithm.

After the picture is stored using ImageAI libraries, we feed that image data to an object detection function that uses Retina Net to detect objects within the image. During the prediction of objects, we create and store the values of the detections into a correctly formatted JSON object in the COCO dataset format for future use. The detections are stored as followed: the category-id which states the name of the object, a score which indicates the confidence level of the algorithm, followed by a (x, y) pixel coordinate of the top left corner of the detected object as well as its corresponding bottom right (x, y) pixel

coordinate that creates a rectangular box around the object. After RetinaNet has identified objects in the photograph, the python CGI script sends that JSON object back to the user front end. Since the Unity engine pixel coordinate system is different from the Open CV libraries, we have to flip the y-coordinates for each detection. To make the detection boxes in the Unity engine, we have to transform the pixel coordinates from picture space to world space coordinates. First, we must normalize the pixels to a percentage of the picture resolution and convert the pixel coordinates from the top left origin to bottom right origin. Then we multiply by the picture to world transform to get world space coordinates and add the picture's positional offset. To scale the created detection boxes, we use the normalized width and height of the transformed coordinates to make each transform independent. Then finally we orient the detections to the image plane in the Feed Me interface(See Figure 3.2).

While the image is processed and detection boxes are in creation, the Feed Me applications provide visual feedback to let the user know that the information is loading. The detection boxes loads on to the image plane reference, displaying to the names of each detection box created by the Feed Me back-end server. Based on the objects detected presented by our neural net the algorithm states its observation. The Feed Me application can relabel any detected objects that were misclassified by the neural net. To relabel an object, the user would only have to focus on the object they wish to change and utter the voice command "Relabel Object"(See Figure 3.3). This speech intent listens for a four to five-second voice recording to send to the Feed Me back-end as an encoded base64 string. Once the Feed Me back-end receives the data using Google's natural language processing algorithm to turn speech to text, that information signals

“Relabel Object”

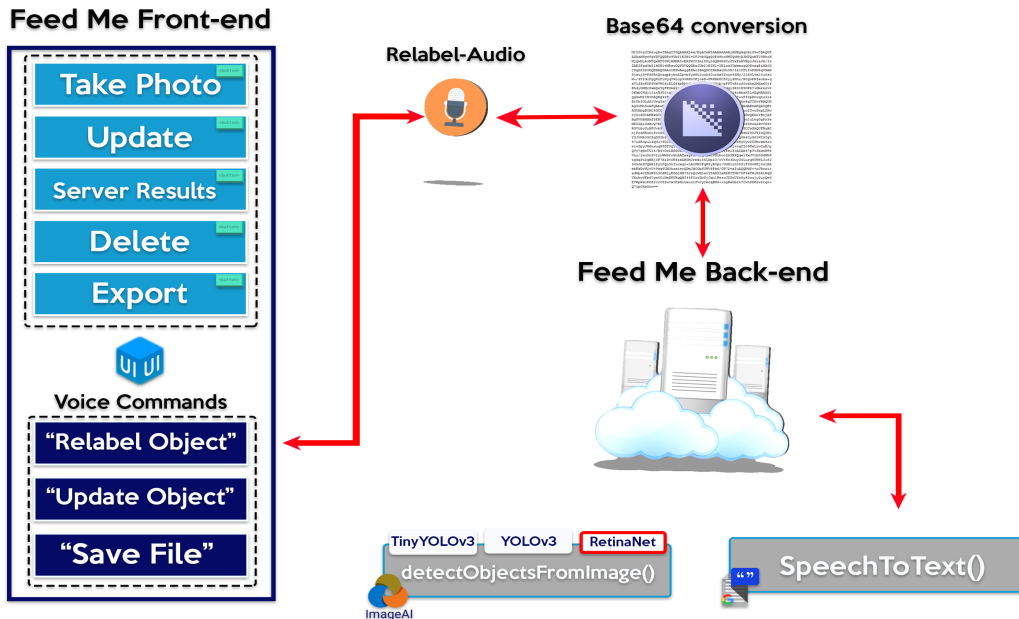


Figure 3.3: When a user wishes to update an existing object, issue the voice command will send a voice request to the Feed Me back-end to turn the speech to text using Google’s speech to text API.

back to the Feed Me front-end replacing the old label with a new detection from the human perspective. These new labels append to the JSON object that was created previously to be used for retraining later.

Using the Feed Me application a user is also able to make new detections on the real cases when the computer vision algorithm did not detect an object (See Figure 3.4). The point of having this feature is to give detailed information to our neural net to retrain it and therefore improve its ability to classify objects

New Detection.png New Detection.bb



Figure 3.4: Using the Feed Me system, this is an example of the user adding a new detection to the scene.

the next time they are seen. By clicking the update button, the user is asked to record audio of the name of the detected object they want to create for the scene. A user can also add a new detection box saying the voice command “Add Box.” Like the handling of re-labeling existing objects, adding a new detection box uses the same Google speech to text NLP algorithm to name the object. After the detection box has been named, the user can re-scale that object and place the detection box in the desired place. To re-scale a detection box, the user would have to issue the voice command “Scale On” to turn on the scaling

Detection.png Detection.bb

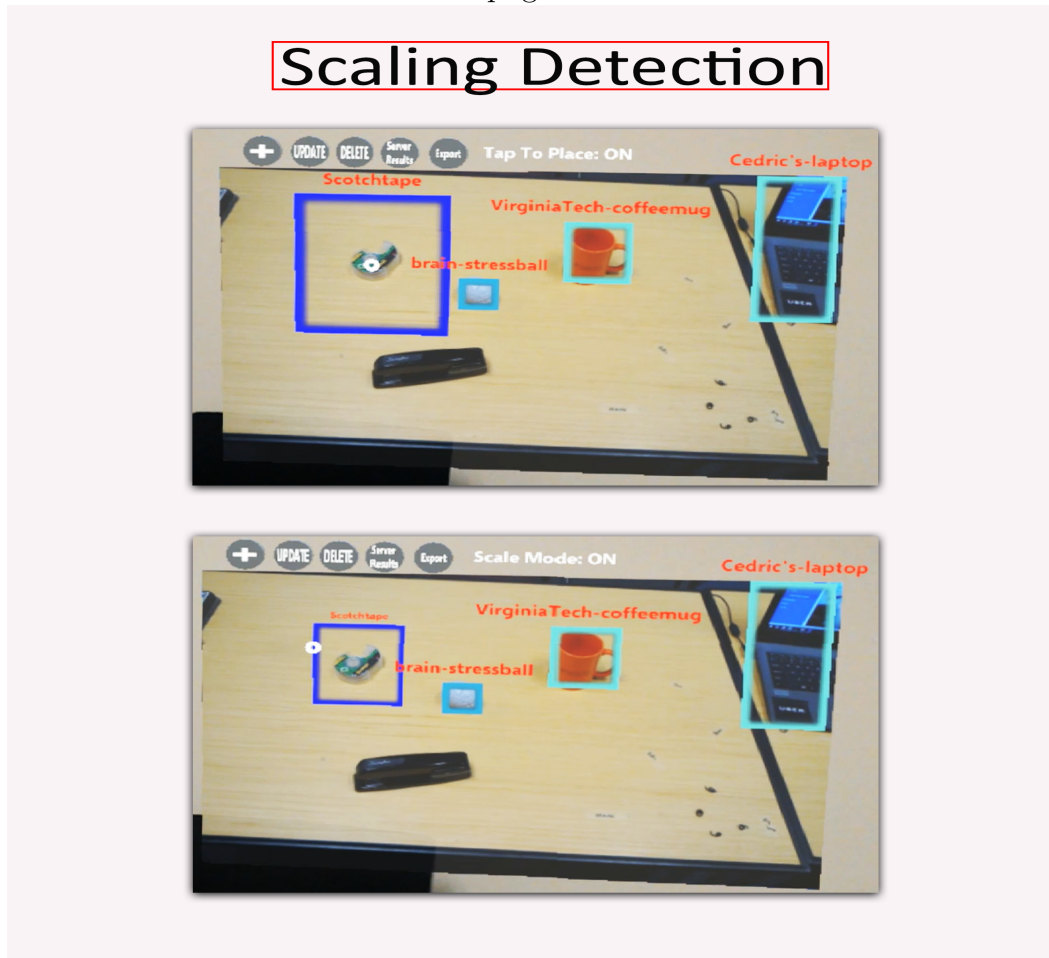


Figure 3.5: Using the Feed Me system, this is an example of the user ability to size the detection box.

function. When the user is satisfied with the size of the detection box, the user will once again issue the voice command “Scale Off” to turn off the function of scaling an object(See Figure 3.5). The Feed Me application provides visual feedback to let users know what mode is currently active in the application. Two modes are associated with detection boxes, and those are scaling of the objects and placement of the detection boxes. By default, the system keeps placement mode active when moving detected objects in the scene. The placement mode is only disabled when the user wishes to re-scale the object of their choice. At

that current time, the placement mode temporarily deactivates until the user is finished scaling the human detected object. Once the user is finished adding new detection boxes to the scene the user will then issue the voice command “Update Object.” This voice command will collect all new detection boxes and append all associated information of the human detected boxes to the JSON object we created from our Feed Me back-end.

Once the user is satisfied with all the new detection boxes and updated existing detection boxes, Feed Me allows the stored JSON object to be sent back to the Feed Me back-end to retrain the neural net. By clicking the “Export” button, the Feed Me front-end will convert the stored JSON object into a base64 string to send to the Feed Me back-end by CGI form post that decodes the object. There is also a voice command associated with the export button in cases where users find it easier to say the command rather than pressing the button.

The Feed Me front-end also allows the user the ability to see results from the Feed Me back-end. Potential users of this interface may wish to see these results to assist them in the positioning of detections boxes in the user interface. By clicking the image reference button on the front-end appends the new image reference over the existing image reference (See Figure 3.6). Once the user is satisfied with the assistance of this function, the user can delete the new image reference by issuing the voice command ”Delete Results.”

Lastly, there is a delete button that deletes all detected objects and the image reference from the scene. These are rare cases when a user takes a photograph, and the photograph the camera captures comes out blurry. These cases only

(5).png (5).bb

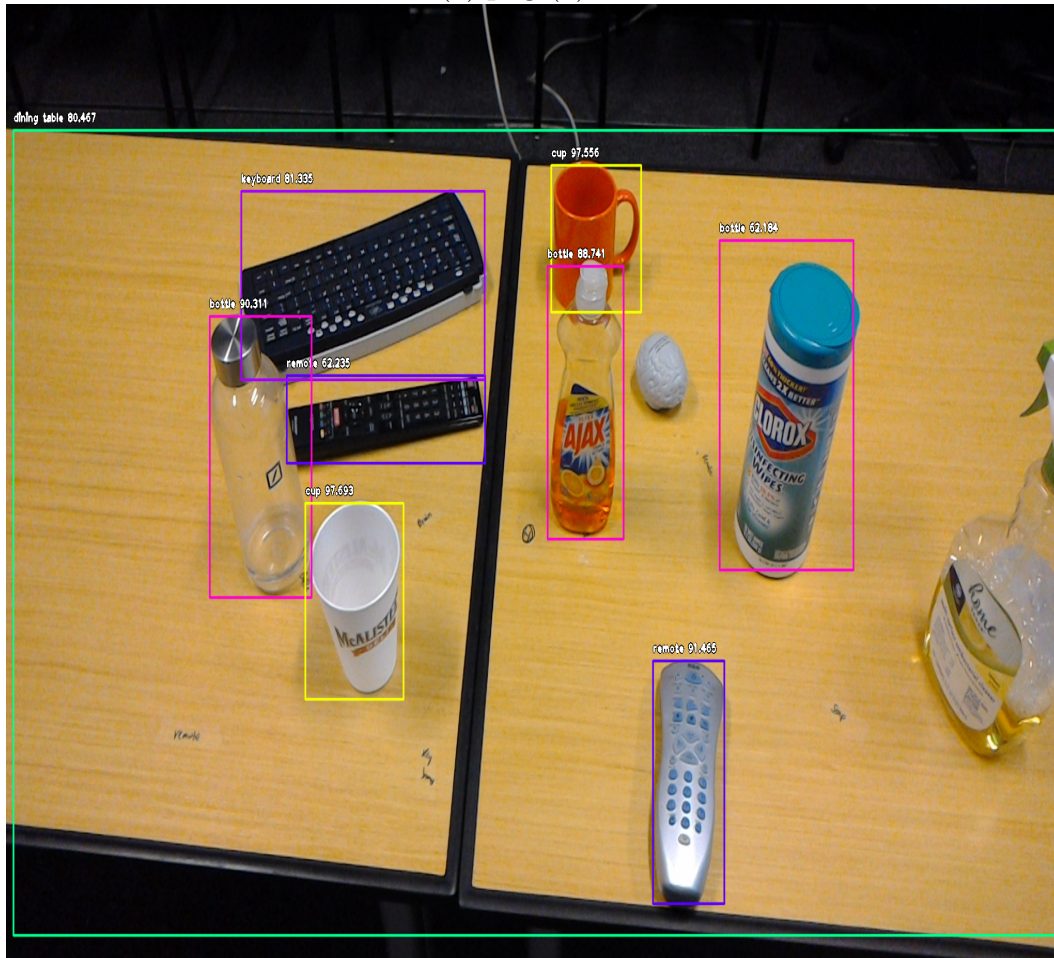


Figure 3.6: Using the Feed Me system, this is an example

occur when there is sudden movement while the user tries to take a photograph, which is why we give the user the chance to start over to recapture the scene.

3.2.2 CGI Container

When the Feed Me front-end wishes to communicate with the Feed Me back-end, the middle connected layer utilizes an implementation of CGI form posting written in python. CGI, also known as the Common Gateway Interface, is known

for its ability to communicate information between a server and a specialized CGI program. A CGI form acts like most World Wide Web forms where an internet user can interact dynamically with a server. CGI form posting has strict guidelines on information stored in a post. All data sent from the Feed Me front-end application must be base64 converted in order to be received by the Feed Me back-end.

3.2.3 Back-end

In the Feed Me back-end, this is where we run our neural net using a machine learning wrapper class known as Image AI. ImageAI is a python library built by John & Moses Olafenwa to build applications and systems with self-contained deep learning (DL) and CV capabilities [28]. ImageAI supports a list of state-of-the ML algorithms for image prediction, custom image prediction. Object detection, video detection, video object tracking, and image predictions training in the format of COCO dataset objects. ImageAI supports object detection, video detection and object tracking using RetinaNet, YOLOv3, and TinyYOLOv3 trained on COCO dataset. ImageAI uses Tensorflow as a backbone for its CV operations and also has CPU and GPU capabilities for CV operations. For the prototype Feed Me application, in the back-end we are using a CPU core implementation of ImageAI. Like any back-end, the Feed Me back-end can retrieve and return data to the Feed Me front-end where the user can take a photograph, to identify objects in the scene. The image reference is transported to the server in the form of a CGI form post [28].

Chapter 4

Experiment

4.1 Scope & Hypothesis

The scope of this experiment is to evaluate the ability a participant can edit existing labels created by our CV algorithm by using the Feed Me AR user interface to disambiguate objects within a scene. In our background, we mention that most annotation tools require a specific technical background and expertise to use the environment. With the Feed Me user interface, we believe that regardless of the user's technical background using our application will still be able to use and understand our environment. Our rationale for this claim is that we believe that by combining simple naturalistic interactions with the user interface would allow the user to make changes to a scene's annotation efficiently.

The initial hypothesis for this application is that users will have more difficulties disambiguating objects in high-density scenes than in low-density scenes. High-density scenes are scenes where there are multiple objects in the capture area. In these settings, it is likely that, it would be harder for the user to quickly traverse through each object to make changes to existing labels.

The second hypothesis for this application is that a user interfacing with colored detection boxes will identify objects faster than users interfacing with black detection boxes. The color scheme of the detection boxes may affect the user's ability to identify objects in high-density settings quickly, which will result in speed and efficiency improvements through each trial completion.

We planned a user study of selecting twenty to forty participants to evaluate the Feed Me full stack application. To test the broad usability of the Feed Me application, recruited participants should vary in age and technical backgrounds. The only requirement is that a participant of any gender is to be of the age of eighteen or older.

4.2 Methods

4.2.1 User Study

In experimenting, we will evaluate the Feed Me system per participant by controlling the arrangement of objects withing high-density and low density scenes. All users will be tested first with the black box prototype.(See Figure 4.1). Followed by a colored box prototype that changes the color of the detection boxes randomly(See Figure 4.2). Before starting trials, we provide a test trial to allow the user to maneuver in the environment to get a grasp on what they can do with the application.



Figure 4.1: Example of a high/low-density scene with black detection boxes.

Independent Variables

The independent variables in this experiment are the placement of the identifiable objects in each scene. Through each density setting, we switch each object's position rather than keep it in its previous position to see if the CV algorithm is still able to classify the object. This precaution adds variance to the output of the CV algorithm concerning the ground truth of what the user sees in the scene. In each trial, two random detected objects will purposely be misclassified and incorrectly positioned for the user to readjust. An incorrect positioning of a



Figure 4.2: Example of a high/low-density scene with colored detection boxes.

detection box can be determined by identifying a box offset by 200 pixels in the x coordinate and 100 pixels in the y coordinate near an object, where the size of the box fits the object perfectly. All incorrectly positioned detection boxes will have incorrect labels. These label names are randomly generated from a list of identifiable objects our neural net is able to detect.

Controlled Variables

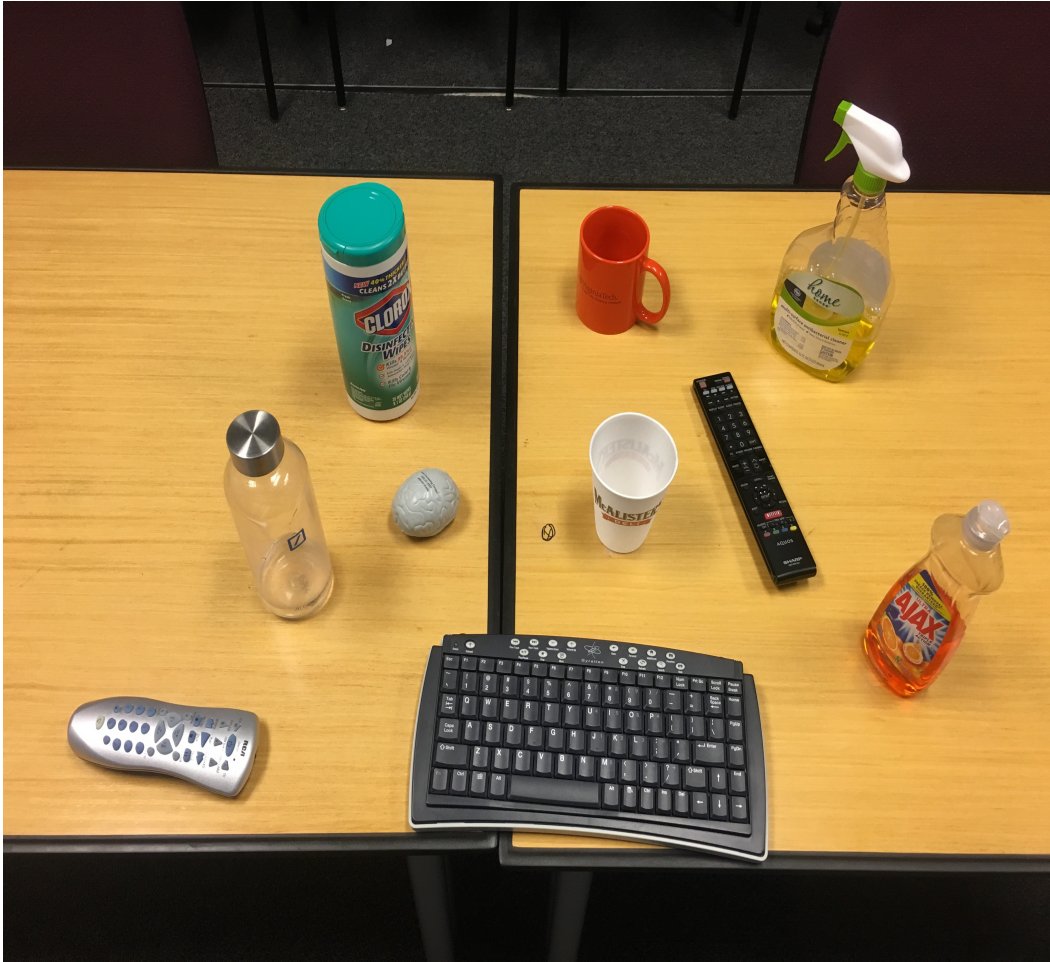


Figure 4.3: This picture represents the ground truth capture area for the user experiment.

The controlled variables in this experiment are the objects presented to the user and the CV algorithm. There are a maximum of ten objects, which were: a Bluetooth Gyration keyboard, a RCA TV remote, a Aquos Sharp smart TV remote, Clorox disinfectant wipes, a Home sense multipurpose cleaner spray bottle, Ajax dish soap, an orange Virginia Tech coffee mug, a McAllister's paper cup, a brain-shaped stress ball, and a Eco-friendly water bottle(See Figure 4.3).

Dependent Variables

The dependent variables in this experiment are: the time associated with completing each trial, the number of times a user has to relabel a detected object by the use of voice commands, and the number of times a user had to reposition a detected box. Another measure we recorded in this experiment was the position of each box that the user repositioned to compare against the CV's algorithm, which will be archived for later use.

The equipment that will be used for this experiment is as follows:

- Microsoft Hololens (Gen 1)
- Dell OptiPlex 7060

The software used to create the Feed Me front-end and back-end used for this experiment is as follows:

- Unity 2017.3.0f3 (64 bit)
- HoloToolkit-Unity-2017.2.1.0
- Google Cloud Speech API
- ImageAI
- Languages Used:
 - Python3
 - C Sharp

Procedure

The following procedure is what each participant followed during the user study. Upon arrival, the experiment is explained and informed consent is obtained. After signing a written consent, the participant has to fill out a background questionnaire that asks questions about their tiredness level and their expertise with using computers to rate their experience with any 3D user interaction devices such as the Microsoft HoloLens or Google Glass. When the participant is finished filling out the demographic and background questionnaire, they are given the Microsoft HoloLens and shown the various ways of selection within the HoloLens device. While the participant gets adjusted to wearing the HoloLens, the participant receives a brief instruction on the overall task flow of each trial. We then informed the participant of each voice command associated with each button, the specific function each button executes, and other voice commands not associated with a button.

After giving the participant a brief overview of the overall flow study, the participant received instruction on how to capture objects in the scene for each trial. In the testing environment, detailed markings are displayed to assist participants in where to look to take a correct photograph of the scene. Markings on the floor indicate the distance the participant needs to be from the scene capture area. Markings in the scene capture area indicate where a participant should focus their gaze when capturing the scene objects. When capturing a scene, the participant is informed to add some offset to the marker in the scene capture area. Add offset to the position of the HoloLens takes into account that the camera is positioned above their eyes (See Figure 4.4). The user is then given a test trial



Figure 4.4: This picture represents a participant in a trial of a high-density setting.

of using the Feed Me user interface to relabel and reposition detection boxes. When the user gives verbal affirmation stating they are comfortable using the interface, the user is then moved to the real trials of the experiment. The reason we wait for verbal consent of confidence using the interface is to ensure there are minimal learning effects or ordering effects gained during the trials.

There are eight trials total in the experiment. The user will be presented with four trials within a low-density setting and four trials within a high-density setting. A low-density setting refers to a scene with four objects in the scene

capture area. A high-density setting refers to a scene with ten objects in the scene capture area. Between the eight trials, a user will alternate between scenes with low-density and high-density. All users will start off with the black box prototype leading to the colored box prototype after completing two low density and two high-density alternating trials. After a participant captures a scene and detection boxes are visible, the participant is encouraged to count the number of detection boxes the CV algorithm produces. This allows users to validate the accuracy of the CV classification.

Table 4.1: This table shows the counterbalancing mechanism used in the user experiment, where the density for each trail changes per prototype feedback.

	Black Color Boxes				Colored Boxes			
User	1	2	3	4	5	6	7	8
N	HD	LD	HD	LD	HD	LD	HD	LD
N+1	LD	HD	LD	HD	LD	HD	LD	HD
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8

The participant then proceeds to reposition and relabel any detection box misclassified in the trial (See Figure 4.5 and Figure 4.6). When the participant is confident with their edits made in the trial, the participant is instructed to update the JSON file and send the file to the Feed Me back-end. The resulting file sent to the server contains information associated to the time elapsed during the trial, the number of times a detected object is repositioned and relabeled by a participant, and the relative user positioning of the objects in the scene. When the participant finishes all eight trials, the participant is instructed to take a NASA TLX survey that assesses the perceived workload of the Feed Me system and its support of in-situ annotation tasks. Finally, the participant is

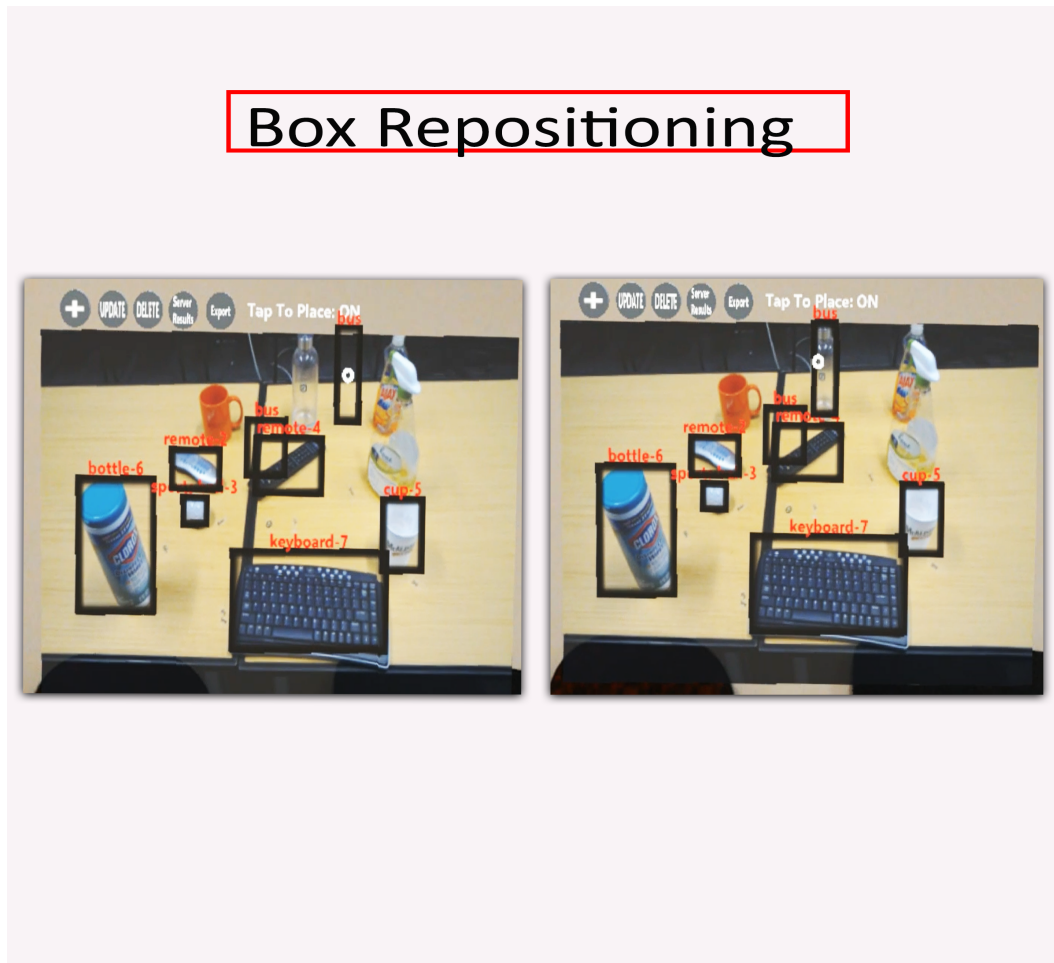


Figure 4.5: This picture represents a participant in trial tasked to reposition a detection box.

instructed to take a satisfaction survey that measures the satisfaction of the voice and repositioning capabilities for the Feed Me system.

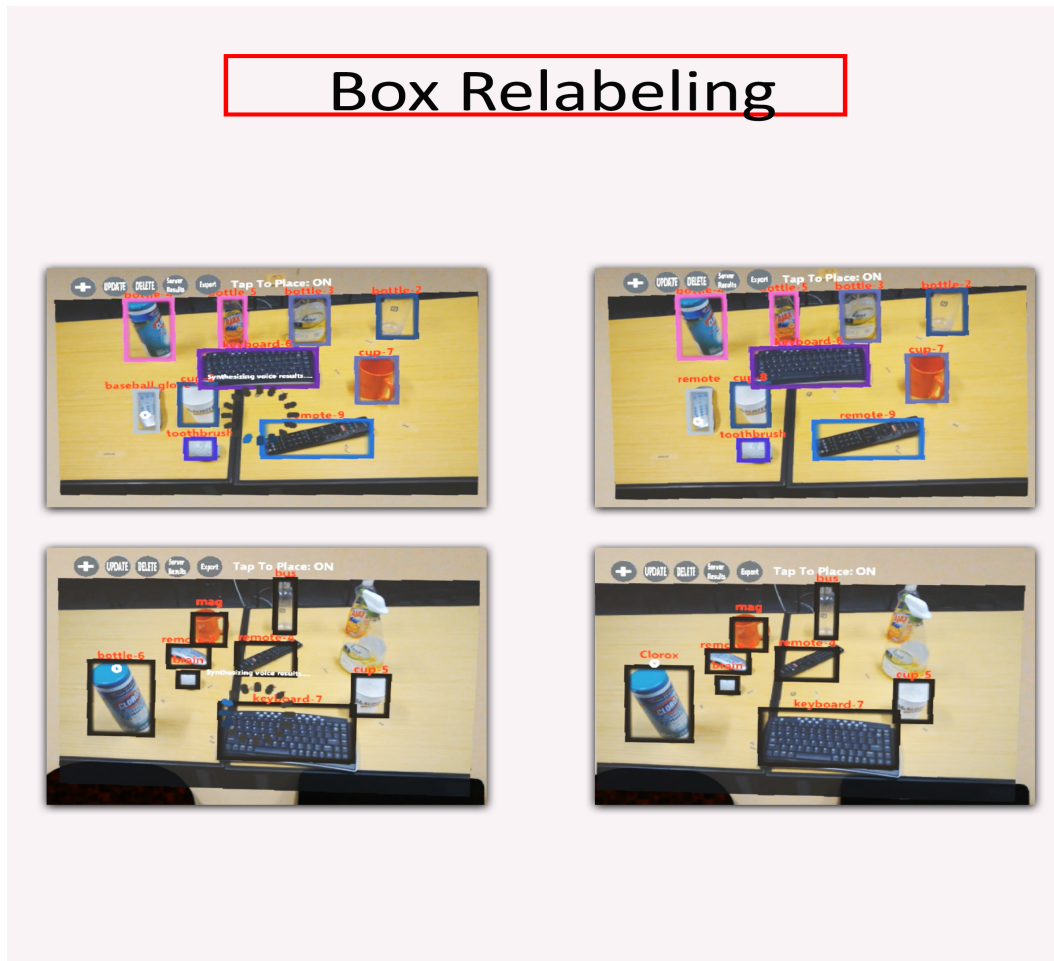


Figure 4.6: This picture represents a participant in trial tasked to relabel a detection box.

Chapter 5

Results

In this experiment there were 160 annotated image data collected from the user study. Based on the average variation of the user corrected labels, labels with the highest average would be used for future nureal net training. To measure the variables in the user experiment, we used a Two way ANOVA repeated measures to analyze the dataset referring to density and color (See Figure 5.1-5.6).

5.1 Time to Completion

The Two way ANOVA measure showed that the density of a scene in respect to time was significant. $F = 6.786^b, P = .013$

The Two way ANOVA measure also showed that the color of the detection boxes when identifying an object in a scene was of significant benefit to the user performance. $F = 22.133^b, P < .001$

The Two way ANOVA measure showed that there was no significant difference between the interaction between density and color concerning timely completion. $F = .003^b, P = .956$

Descriptive Statistics

	Mean	Std. Deviation	N
hd_b_time	134.056741	67.0969041	40
ld_b_time	102.435975	45.1015105	40
hd_c_time	110.765335	46.2236540	40
ld_c_time	79.8581061	26.5293159	40

Figure 5.1: Descriptive statistic for Time Completion.

5.2 Number of Box Repositions

The Two way ANOVA measure showed that the density of a scene in respect to box repositioning showed no significance. $F = .739^b, P = .395$

The Two way ANOVA measure also showed that the color of the detection boxes when repositioning an object in a scene was significant. $F = 5.340^b, P = .026$

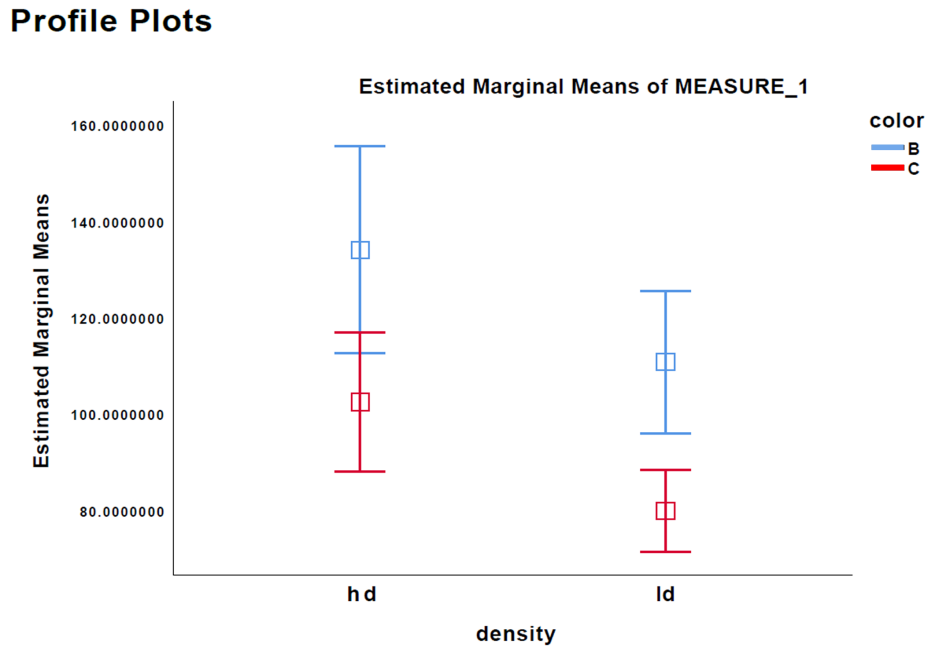


Figure 5.2: Time to Completion for density x color.

The Two way ANOVA measure showed that there was no significance between the interaction between density and color in respect to box repositioning. $F = .083^b, P = .775$

5.3 Number of Box Relabelings

The Two way ANOVA measure showed that the density of a scene in respect to box relabelling showed no significance. $F = .084^b, P = .773$

Descriptive Statistics

	Mean	Std. Deviation	N
hd_b_repo	4.48	3.486	40
ld_b_repo	3.45	1.739	40
hd_c_repo	4.23	3.711	40
ld_c_repo	2.98	1.860	40

Figure 5.3: Descriptive statistic for Number of box repositions.

The Two way ANOVA measure also showed that the color of the detection boxes when relabeling an object in a scene was significant. $F = 5.051^b, P = .030$

The Two way ANOVA measure showed that there was no significance between the interaction between density and color in respect to box repositioning. $F = .120^b, P = .731$

5.4 NASA TLX

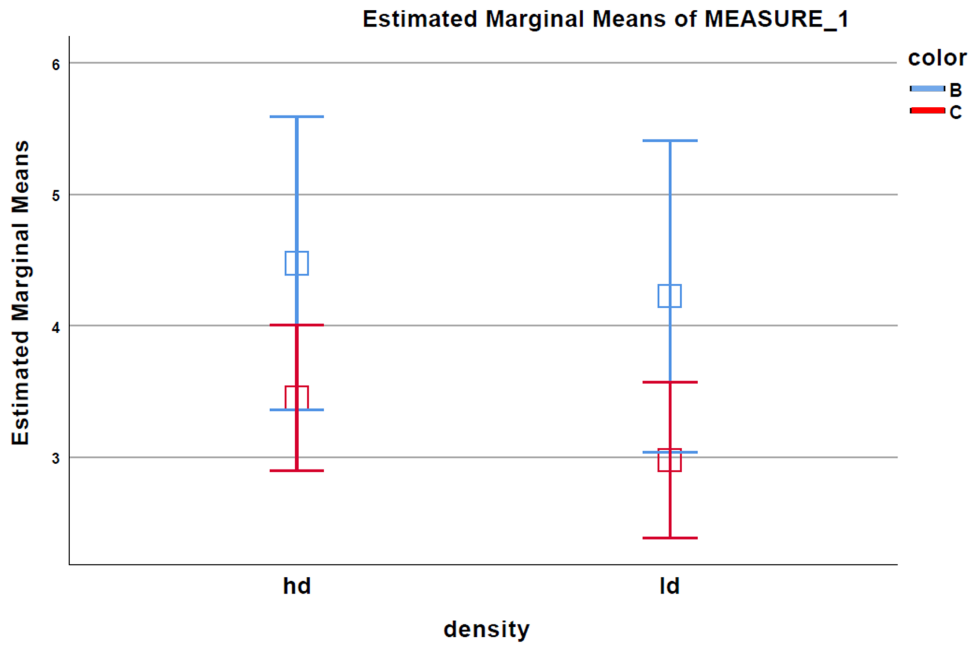


Figure 5.4: Number of Box Repositions for density x color.

To measure the usability of the Feed Me system, we utilized the NASA TLX protocol to understand how the user perceived task workload. Results below show the mean and standard deviation of the combined study participants (See Figure 5.7).

5.5 Exit Survey

Descriptive Statistics

	Mean	Std. Deviation	N
hd_b_relabel	4.00	3.219	40
ld_b_relabel	3.23	1.915	40
hd_c_relabel	4.00	2.717	40
ld_c_relabel	3.00	1.633	40

Figure 5.5: Descriptive statistic for Number of Box Relablings.

To measure the satisfaction of using voice commands and the user's ability to reposition boxes in AR, data results show the mean and standard deviation of the combined study participants (See Figure 5.8).

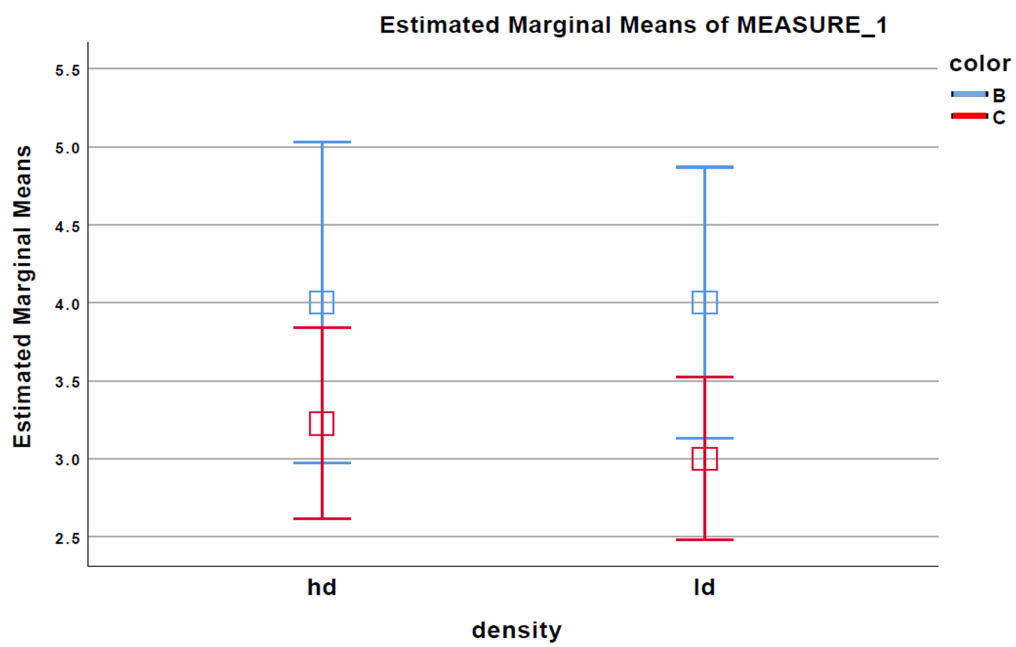


Figure 5.6: Number of Box Relabelings for density x color.

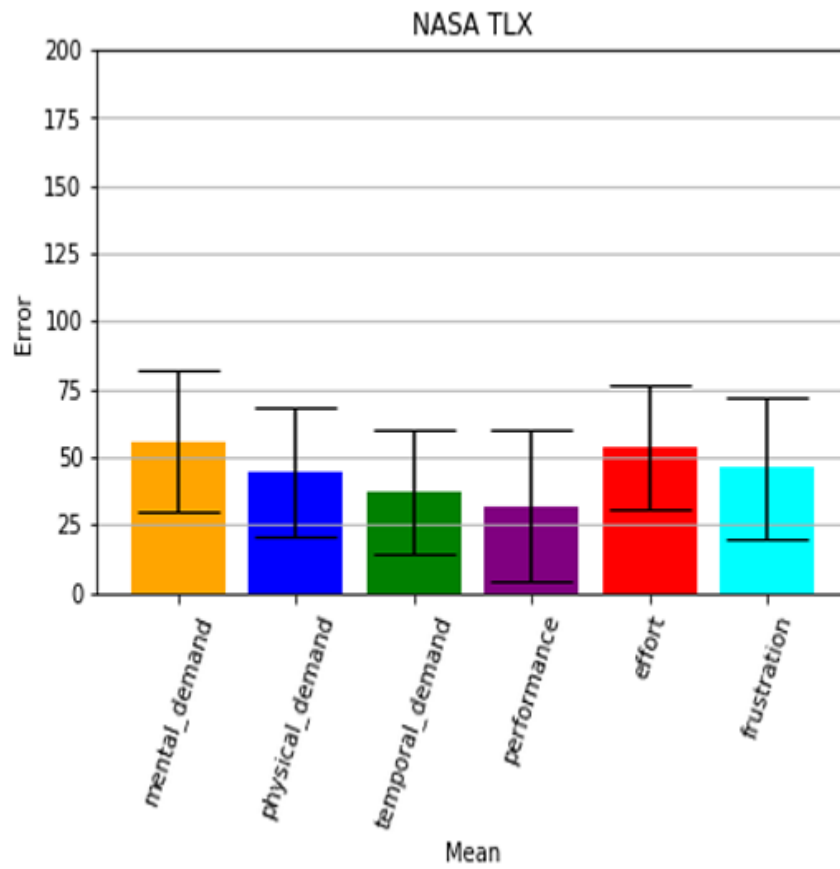


Figure 5.7: Mean error distribution of the NASA TLX. Error bars show Standard Deviation.

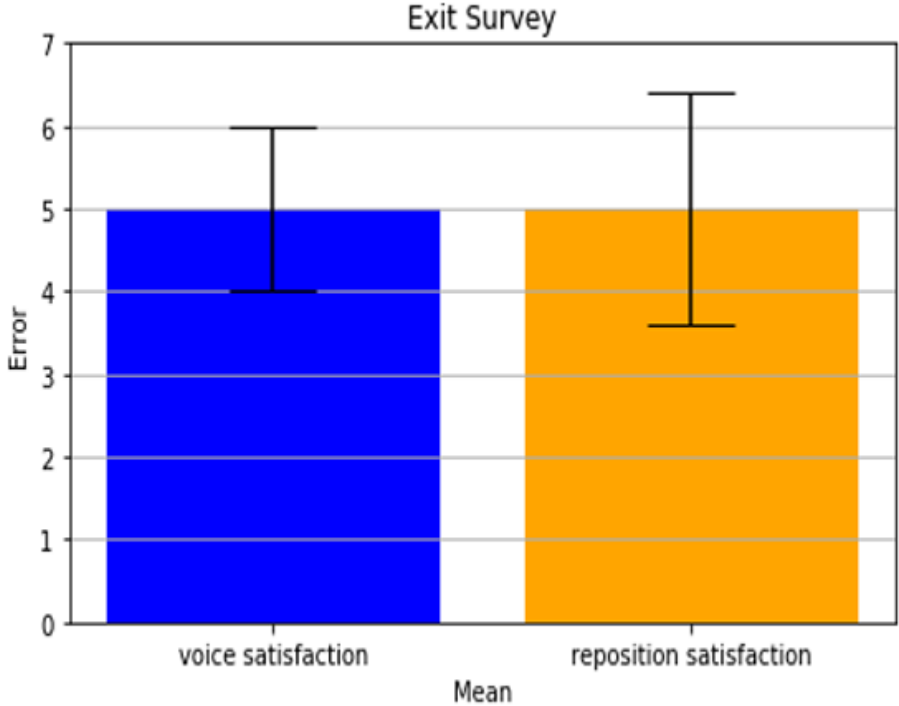


Figure 5.8: Mean error distribution of the Exit Survey. Error bars show Standard Deviation.

Chapter 6

Conclusion

6.1 Findings

As stated in our initial hypothesis, we believed that disambiguating objects in high-density settings will be more complicated than disambiguating objects in low-density settings. Measuring our results using the two way ANOVA repeated measures showed that there was a longer time to completion with increased density, thus proving our initial hypothesis. Box repositioning and relabeling performance did not show a significant difference depending on the object density in the scenes tested. Overall however, high-density scenes scored higher errors than low-density scenes by our dependent variables.

Our second hypothesis posited that the colored box scheme prototype performance would win out over the standard black box prototype. The two way ANOVA repeated measures showed significant favor toward the colored box in each of the dependent variables, thus proving our second hypothesis. Colored boxes helped users more efficiently disambiguate objects in terms of faster time to completion, number of repositions, and relabeling. Colored boxes marked lower error scores in both high-density and low-density in addition, than black boxes.

Utilizing the NASA TLX protocol, we calculated the means across NASA's multidimensional assessment tool to give a holistic overview of the usability of the Feed Me system. On average, Feed Me ranked between low to very low for each demand perspective of the NASA TLX protocol as well as the similar effects of effort and frustration level of using the system. As far as performance, the Feed Me system scored on the higher end of task efficiency. These results show promise towards wide-scale usability of this AR system.

The measured responses of the exit survey added to our observation of how useable this AR system could be to the general public. The use of voice commands and box repositioning ability both scored on the higher end of the scale of satisfaction. These averaged results show that this specific combination of interaction techniques is at least accessible to novice users.

Some interesting findings while experimenting with voices commands and use of voice showed that our system could be improved with additional feedback. Participants left comments like:

- " Voice recognition was responsive. I usually have trouble with Siri."
- " It was easy to capture photos with the voice command than using buttons."

Some participants whose first language was not English sometimes struggled to relabel objects. Participants that fell under this category' state that the system had a hard time recognizing words like bottle, cup, and mug. One participant left a comment saying "System should show pop up window for best matches for synthesized voice results." We will take into consideration adding a pop up box

for best word choice when designing the next iteration of the Feed Me interface.

Participants also left comments about the color of the detection box and how it helped them with repositioning boxes. One participant stated, "I don't know if it was just perception, but it seemed like the color boxes were easier to move than black boxes." Another participant stated, "Color boxes helped distinguish objects more when they are close together."

6.2 Guidelines

At a higher level, after conducting this experiment, future in-situ applications in AR should follow these guidelines:

- Use a color scheme to disambiguate objects's annotations.
- When drawing detection boxes in 3D, always draw bigger boxes first so that the smaller boxes on top can be selected.
- When enabling a speech to text functionality to an in-situ 3D annotation system, to overcome language barriers, the system must provide the user with a set of top possible recognized words.
- Finally, in respect to text legibility, the system should take into account: color, contrasting objects, font-size in relation to object, and background lighting.

6.3 Challenges & Future Work

3D User Interface

After evaluating the usability of the Feed Me System, future work should include more prototyping features and user evaluations. For example, the user should have the ability to have multiple selection interaction techniques rather than one selection interaction when picking a detection box. In the study the current interaction techniques allowed users to complete each task efficiently but failed to give users the choice how to interact with the system. For example, when scaling and repositioning boxes within the system the user could only interact with the box one dimensionally. When designing new interactions for the Feed Me system, interactions would still need to abide by the guidelines presented.

Outdoors

The ultimate goal for the Feed Me system is to enable its use for outdoor activities as well indoor activities. The next steps for the Feed Me system includes testing our colored box prototype outdoors and measure its performance in dynamic contrast lighting. This outdoors experiment will test the independent variables stated in Gabbard's experiment.

Contextual Information

During the experiment, we noticed that the detection algorithm labeled some detections incorrectly or sometimes not at all. In order to make our detection algorithm smarter in-situ, our system could take into account contextual infor-

mation. Similar to Julier et al.'s research, the Feed Me system would undergo a filtering process based on the images presented to our detection algorithm. The results from the user will improve the performance of the CV object detection. Also, the CV algorithm would be able to assign different high contrasting colors to objects in lighting conditions when the object is not distinguishable between figure and ground.

Faster Detection

The current implementation of our CV algorithm runs a CPU implementation of TensorFlow for deep learning, which gives a user a little wait time while the object detection service identifies objects in a scene. In order to make detections faster when identifying objects in real time, the Feed Me system will need to implement a GPU implementation of Tensor Flow. By running a GPU implementation of Tensorflow, would allow faster math computations, allowing the system to make faster detections. In this new GPU implementation, the system will switch from using image data to video data, allowing our system to use cache memory of previously seen objects.

Overall Performance

Lastly, future works will work on the overall performance of the Feed Me system concerning networking and processing. Similar to Jiasi et al.'s research for server communication, we will determine the tradeoffs of running a smaller neural net locally on the Hololens device than running the neural net from the server.

Bibliography

- [1] Mattzmsft. “spatial mapping - mixed reality.” mixed reality | microsoft docs. *Micorsoft*, 20, March 2018. URL docs.microsoft.com/en-us/windows/mixed-reality/spatial-mapping.
- [2] Martin Abadi et al. ”tensorflow: A system for large-scale machine learning.” usenix. *USENIX*, 1, January 1970. URL www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.
- [3] Ferran Argelaguet and Carlos Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, 2013.
- [4] Doug A Bowman, Jian Chen, Chadwick A Wingrave, John F Lucas, Andrew Ray, Nicholas F Polys, Qing Li, Yonca Haciahmetoglu, Ji-Sun Kim, Seonho Kim, et al. New directions in 3d user interfaces. *IJVR*, 5(2):3–14, 2006.
- [5] Volkert Buchmann et al. “gesture based direct manipulation in augmented reality.” fingartips. *ACM*, 15, June 2004. URL dl.acm.org/citation.cfm?id=988871.
- [6] Jiasi Chen et al. “deepdecision: A mobile deep learning framework for edge video analytics.” deepdecision: A mobile deep learning framework for edge video analytics - ieee conference publication. *IEEE*, 11, October 2018. URL ieeexplore.ieee.org/abstract/document/8485905.
- [7] Martin Eckert, Matthias Blex, Christoph M Friedrich, et al. Object detection featuring 3d audio localization for microsoft hololens. In *Proc. 11th Int. Joint Conf. on Biomedical Engineering Systems and Technologies*, volume 5, pages 555–561, 2018.

- [8] J. Edward, Swan Ii, and Joseph L. Gabbard. Survey of user-based experimentation in augmented reality. In Las Vegas, editor, *1st International Conference on Virtual Reality*, 2005.
- [9] Taylor Frantz, Bart Jansen, Johnny Duerinck, and Jef Vandemeulebroucke. Augmenting microsoft’s hololens with vuforia tracking for neuronavigation. *Healthcare technology letters*, 5(5):221–225, 2018.
- [10] Joe L Gabbard and J Edward Swan II. Usability engineering for augmented reality: Employing user-based studies to inform design. *IEEE Transactions on visualization and computer graphics*, 14(3):513–525, 2008.
- [11] Joseph L Gabbard, J Edward Swan, Jason Zedlitz, and Woodrow W Winchester. More than meets the eye: An engineering study to empirically examine the blending of real and virtual color spaces. In *2010 IEEE Virtual Reality Conference (VR)*, pages 79–86. IEEE, 2010.
- [12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [13] Allan Hanbury. “A Survey of Methods for Image Annotation.” *Journal of Visual Languages Computing*. Academic Press, 29 www.sciencedirect.com/science/article/pii/S1045926X08000037, 2008.
- [14] William A Hoff, Khoi Nguyen, and Torsten Lyon. Computer-vision-based registration techniques for augmented reality. In *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, volume 2904, pages 538–549. International Society for Optics and Photonics, 1996.

- [15] Richard L Holloway. Registration error analysis for augmented reality. *Presence: Teleoperators & Virtual Environments*, 6(4):413–432, 1997.
- [16] S. Juiler et al. “information filtering for mobile augmented reality.” information filtering for mobile augmented reality - iee conference publication. 2, August 2002. URL ieeexplore.ieee.org/abstract/document/880917.
- [17] Hyejin Kim et al. “imaf: in situ indoor modeling and annotation framework on mobile phones.” personal and ubiquitous computing, 2013.
- [18] Tobias Langlotz et al. “*Sketching up the World: in Situ Authoring for Mobile Augmented Reality.*” *Personal and Ubiquitous Computing*. Springer-Verlag, 6 dl.acm.org/citation.cfm?id=2425073, 2012.
- [19] Google Lens. *Google*. lens.google.com/.
- [20] Vincent Lepetit, Luca Vacchetti, Daniel Thalmann, and Pascal Fua. Fully automated and stable registration for augmented reality applications. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 93–102. IEEE, 2003.
- [21] Yuxi Li, Jiuwei Li, Weiyao Lin, and Jianguo Li. Tiny-dsod: Lightweight object detection for resource-restricted usages. *arXiv preprint arXiv:1807.11013*, 2018.
- [22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. pages 2980–2988, 2017.
- [23] Yang Liu, Haiwei Dong, Longyu Zhang, and Abdulmotaleb El Saddik. Technical evaluation of hololens for multimedia: a first look. *IEEE MultiMedia*, 25(4):8–18, 2018.

- [24] Jing Ma, Li Chen, and Zhiyong Gao. Hardware implementation and optimization of tiny-yolo network. In *International Forum on Digital TV and Wireless Multimedia Communications*, pages 224–234. Springer, 2017.
- [25] Rui Ma, Guocheng Liu, Qi Hao, and Cong Wang. Smart microphone array design for speech enhancement in financial vr and ar. In *2017 IEEE SENSORS*, pages 1–3. IEEE, 2017.
- [26] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
- [27] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. Drillsample: precise selection in dense handheld augmented reality environments. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*, page 10. ACM, 2013.
- [28] Moses Olafenwa and John Olafenwa. Olafenwamoses. “olafenwamoses/imageai.” github, January 2019. URL github.com/OlafenwaMoses/ImageAI.
- [29] Kasım Özacar, Juan David Hincapié-Ramos, Kazuki Takashima, and Yoshifumi Kitamura. 3d selection techniques for mobile augmented reality head-mounted displays. *Interacting with Computers*, 29(4):579–591, 2016.
- [30] Jonathan Pedoeem and Rachel Huang. Yolo-lite: A real-time object detection algorithm optimized for non-gpu computers. *arXiv preprint arXiv:1811.05588*, 2018.
- [31] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.
- [32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

- [33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [34] Azuma Ronald. “a survey of augmented reality.” mit press journals. 10: 1162, March 2006. URL [1997BytheMassachusettsInstituteofTechnology,13www.mitpressjournals.org/doi/abs//pres.1997.6.4.355](http://www.mitpressjournals.org/doi/abs//pres.1997.6.4.355).
- [35] Bryan C. Russell et al. “labelme: A database and web-based tool for image annotation.” springerlink. In *US, 31 link.springer.com/article/10.1007/s-0090-8*, pages 11263–007. 2007.
- [36] Mohammad Amin Sadeghi and David Forsyth. 30hz object detection with dpm v5. In *European Conference on Computer Vision*, pages 65–79. Springer, 2014.
- [37] Carl Vondrick et al. “Video Annotation Tool from Irvine. California.” Vatic - Video Annotation Tool - UC Irvine, Springer Netherlands www.cs.columbia.edu/~vondrick/-vatic/, 2012.
- [38] C Wingrave and D Bowman. Baseline factors for raycasting selection. In *Proceedings of HCI International*, pages 61–68. Citeseer, 2005.
- [39] Alexander Womg, Mohammad Javad Shafiee, Francis Li, and Brendan Chwyl. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 95–101. IEEE, 2018.
- [40] Bichen Wu, Forrest Iandola, Peter H Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for

- autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 129–137, 2017.
- [41] Christian Zimmer, Michael Bertram, Fabian Büntig, Daniel Drochert, and Christian Geiger. Mobile augmented reality illustrations that entertain and inform: Design and implementation issues with the hololens. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, page 23. ACM, 2017.

Appendices

Appendix A

VT IRB-18-1113

A.1 Authorization Letter

A.2 Experiment Hardware

- Microsoft Hololens (Gen 1)
 - Software
 - * Windows 10
 - * Windows Mixed Reality
 - * Understanding Capabilities
 - Spatial Sound
 - Gaze Tracking
 - Gesture Input
 - Voice Support
 - Spatial Mapping
 - Sensors
 - * 1 IMU (Accelerometer, gyroscope, and magnetometer)
 - * 4 environment sensors

- * 1 energy-efficient depth camera with a 120° x120° angle view
- * Four-microphone array
- * 1 ambient light sensor
- Processors
 - * Intel 32-bit (1GHz) with TPM 2.0 support
 - * Custom-built Microsoft Holographic Processing Unit (HPU 1.0)
- Memory
 - * 2GBRam
- Storage
 - * 64GB (flash memory)
- Power
 - * 2-3 hours of active use
 - * Up to 2 weeks on standby
- Dell OptiPlex 7060
 - CPU
 - * Intel Core™ i7-8700 CPU @ 3.20GHz x 12
 - Graphics
 - * Intel HD Graphics (Coffeelake 3x8 GT2)
 - OS
 - * Ubuntu 18.04.2 LTS
 - * 64-bit
 - Disk
 - * 1TB capacity

A.3 Experiment Software

- Unity 2017.3.0f3 (64 bit)
- HoloToolkit-Unity-2017.2.1.0
- Google Cloud Speech API
- ImageAI
 - RetinaNet
- Apache
- CGI
- Languages Used:
 - Python3
 - C Sharp

**Office of Research Compliance**

Institutional Review Board
North End Center, Suite 4120
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-3732 Fax 540/231-0959
email irb@vt.edu
website <http://www.irb.vt.edu>

MEMORANDUM

DATE: January 18, 2019
TO: Nicholas Fearing Polys, Cedrick K Ilo
FROM: Virginia Tech Institutional Review Board (FWA00000572, expires January 29, 2021)
PROTOCOL TITLE: Augmented Reality In-situ Annotation
IRB NUMBER: 18-1113

Dear Investigator(s):

RE: Protocol Submission for WIRB Review

The Virginia Tech Institutional Review Board (IRB) office screened this study and determined that it is ready for WIRB review.

Please download the "Instructions for the PI to Transfer the VT IRB Protocol to WIRB":

<https://secure.research.vt.edu/external/irb/wirb-submission-instructions.pdf>

Please go to <https://connexus.wcgclinical.com> to complete the protocol submission process to the WIRB.

ATTENTION:

* Nicholas Fearing Polys MUST BE LISTED AS THE PI ON THE WIRB SUBMISSION.

* All references to the VT IRB (including phone number and email address) MUST be removed from all study documents and replaced with Western IRB - (800) 562-4789, help@wirb.com.

*Special instructions, if any, are included on the top of the next page.

Invent the Future