MscThesis

University of Turku
Department of Computing
Master of Science Thesis
Laboratory Name
February 2025
Botond Ortutay

UNIVERSITY OF TURKU
Department of Computing

Botond Ortutay: MscThesis

Master of Science Thesis, 26 p.
Laboratory Name
February 2025

Keywords: tähän, lista, avainsanoista

TURUN YLIOPISTO
Tietotekniikan laitos

Botond Ortutay: MscThesis

Pro gradu -tutkielma, 26 s.
Labran nimi
Helmikuu 2025

Asiasanat: here, a, list, of, keywords

Contents

List of Figures

1 Introduction

1.1 The goals of this thesis

This thesis explores the usability and functionalities of an Augmented Reality (AR) experience and its interoperability with modern Deep Learning (DL) based computer vision (CV). This is explored with a prototype, where a camera is used to capture an image feed, information is extracted from it using CV-algorithms and finally AR is used to communicate this information with the user. This thesis explores the creation of such a system and examines its usability and potential real-world usefulness.

1.2 Research Questions

The thesis aims to answer the following Research Questions:

RQ1: What are the technological challenges in combining advanced computer vision algorithms with an AR user interface?

RQ2: Can a system with a backend computer vision system and an AR user interface be used in a cooking environment?

RQ3: Can such a system provide satisfactory user experience?

1.3 Methodology Overview

The work covered in this thesis mostly consist of the following three phases: firstly I conduct a literature review in chapter 3. The purpose of this is to learn about the technologies involved in this project, to find out what perceived challenges were found by other people working with these technologies and to search for projects similar to ours, conducted by the scientific community. Starting with a literature review should also provide a firm scientific basis to the later phases. In this phase we aim to answer **RQ1**.

The second phase of the work concerns architecture design and prototype development. All relevant technological challenges found during the literature review are collected to chapter 4.2 for further analysis. Based on all these findings we then propose an architecture for a prototype in chapter 4.3. The prototype is then implemented and the whole development process, the technologies used, as well as anything notable that happens during development is described in chapter 5. The goal of this phase is to build an actual prototype and thus answer **RQ2**.

The third phase of the work is an empirical usability study conducted on the developed prototype. Here the finished prototype is given out to test subjects to measure how well the system performs. Usability is measured both through asking the opinions of the test subjects through a questionnaire and through measurements made by the prototype software. All the collected data and the questionnaire used can be found in the attachments. This phase of work is more thoroughly described in the chapters 6 and 7. In this part of the work we aim to answer **RQ3**.

2 Background

2.1 Justification for RQ1 (come up with a better title later)

RQ1: technological challenges in combining CV & AR

- Introduce CV AR
- Why do we want to combine these techs?
  - Traditional AR tracking: built to detect specific things in environment
  - CV: larger computer model used for image analysis could theoretically be used to perform this task
    - Solves some problems but creates others
      - This is what we aim to map out

2.2 AR in a nutshell (come up with a better title later)

AR is a technology that can be defined many ways. Monroy Reyes, Vergara Villegas, Miranda Bojórquez, *et al.* [1] define it as a real-time view of the real-world environment that has been enhanced by adding computer generated information on top of

it. Ghasemi, Jeong, Choi, *et al.* [2] call AR an extended version of the real world, on top of which digital content is overlaid acting as a bridge between the real and virtual worlds. Minaee, Liang, and Yan [3] define AR as an interactive experience where objects of the real world are enhanced by information added by a computer system. I would personally define AR as software that superimposes information on top of the real world to create interactive experiences. This information is often represented as 3D-objects known as augmentations.[4][5]

AR experiences cannot be created without certain hardware components. First, a computing unit is needed for generating the augmentations and calculating their locations. Second, a projection surface is needed on which to display the virtual content to the users.[2] Additionally sensors or cameras are needed to observe the real world. Input devices are also needed to let the users interact with the AR-software.[1][3]

There are multiple different ways to achieve AR. The two most obvious categories are marker-based AR and markerless AR.[4] In marker-based AR, physical markers, such as barcodes or QR-codes are placed in the real-world environment, and these tell the software the location, where to add the augmentations.[4][1] In practical applications these markers often need to be attached to real-world objects[2] which may be unattractive in certain applications.

An alternative to this is markerless AR. Markerless AR doesn't use pre-defined markers, instead it collects data from sensors such as a camera or a GPS and uses advanced algorithms to determine where to render the augmentations.[4] Typically these algorithms are computer vision based[4] but in the future deep-learning based methods might also be used[2]. Different kinds of computer vision algorithms that can be used to achieve markerless AR are further discussed in 3.1. To render augmentations on top of the real world without using markers, markerless AR needs to in essence obtain a spatial map of its environment[2] and detect flat surfaces.[2]

Markerless AR can of course in itself be achieved in many different ways. Estrada, Paheding, Yang, *et al.* [4] list four in particular:

Location-based AR is a form of markerless AR where AR content is locked to geographical areas.[6] In these kinds of software the user's location is obtained through GPS, sometimes also using a digital compass or other sensors.[6] Location-based AR softwares typically use Simultaneous Localization and Mapping (SLAM) algorithms to map the local area, keep track of the user's position within it and show the augmentations.[4][7][3]

Another way of achieving markerless AR is known as superimposition-based AR. Here advanced object detection algorithms are used to detect an object from a camera feed and then an augmentation is superimposed on top of it.[4] A third way is outlining-based AR. In outlining-based AR, the objects which determine the positions of the augmentations are tracked by detecting their shape, using algorithms that can recognize certain contours or forms.[4]

Estrada, Paheding, Yang, *et al.* [4] also mentions projection-based AR (also known as projection mapping or spatial AR) as a fourth kind of markerless AR. Projection-based mapping is a type of AR where a projector is used to project the augmentation onto a surface rather than using a screen with a camera feed to show the augmentations.[4] This type of AR could use any type of CV-algorithm to calculate the location of the augmentations. For example Kim, Seo, and Han [8] created an application where animations could be projected onto a drawing surface. This project used various CV-algorithms to calculate augmentation positions and facilitate interactivity, such as geometric shape detection and hand gesture detection.[8]

3 Literature review

3.1 Computer Vision (CV)

Computer Vision is a field of study where cameras and computer systems are used to extract information from the real world.[9] It is a fairly old field dating back to the 1960s. Early CV utilized technologies like pattern recognition, part-based algorithms, matching techniques and statistical classifiers to analyze images.[2] Nowadays Deep Learning (DL) based methods are often used to perform CV tasks.[2] Deep Learning is a sub-field field of Machine Learning (ML) where algorithms modeling the structure and function of the human brain, known as deep neural networks are used to complete various tasks.[4] A big difference in using traditional CV-methods and DL-based methods to perform CV tasks is, that in early CV practicioners themselves had to define algorithms fit for the task at hand whereas in DL-based methods lots of data is fed to the deep neural networks (models) which will then analyze that data and learn patterns that they then use to perform the task.[2]

DL-based methods are often based on Convolutional Neural Networks (CNN). A CNN takes an image, divides it into pieces, analyses the pieces through convolution and pooling and then performs object classification through this analysis.[2] Region-based Convolutional Neural Networks (RCNN) are an evolution of this technology. An RCNN first analyses the image using another method to define Regions of interest, then applies a CNN to the detected regions of interest.[2][4] You Only

Look Once (YOLO) is an even more advanced DL-based CV-technique. YOLO divides the input image into S*S grids and searches each grid for the object. It is considerably more light-weight than region-based methods like RCNN.[2][4] A key task within CV, that is very relevant in AR applications is object detection. Object detection has been a popular research topic in this past decade both within the fields of CV and DL and on top of being relevant for AR, it also has potential in heavily researched applications such as Autonomous driving and robotics.[4][2] Object detection aims to find one or several pre-defined objects, given an image frame.[2] According to Ghasemi, Jeong, Choi, *et al.* [2] object detection can be viewed as a task consisting of the following sub-tasks:

Object localization: determining the exact location of the object within the frame

Object classification: determining which kind of object has been found, when using object detection to detect multiple different kinds of objects

The tracking of markers in Marker-Based AR can be thought of as an example of object detection. However this is typically not done with DL-algorithms, but with older CV-techniques such as statistical classifiers.[2] For example Zhang, Fronz, and Navab [10] compare different tracking systems and while they don't describe each of their internal workings in detail, they build their own tracking solution to measure the tracking accuarcy of the compared system. To build this they used the openCV-libraries built in corner detection algorithm which is a traditional CV-algorithm from 1988, developed long before the existance of DL-based methods.[11] While DL-based object detection methods require more computation, both Ghasemi, Jeong, Choi, *et al.* [2] and Minaee, Liang, and Yan [3] showed, that there do exist DL-based object detection methods that are suitable for running on low powered hardware, such as mobile phones or AR-glasses, at a high enough framerate, that they could be used to create markerless AR experiences on these devices. Ghasemi, Jeong, Choi, *et al.*

[2] names the CNN-based MobileNet v2, and the YOLO-based tiny YOLO v2 as suitable for real-time AR use on mobile hardware. In 3.2 we look at an example from Estrada, Paheding, Yang, *et al.* [4] where a DL-based algorithm is used to perfor object tracking.

In a very fascinating article Minaee, Liang, and Yan [3] looked through examples of the usage of modern DL-techniques in several AR applications. They found that DL was used in AR applications for use cases such as tracking, user pose estimation, traditional pattern recognition and geometrical application. But on top of these DL-based methods were also used to generate augmentations for scene reconstruction, human reconstruction and face and body transformations. Essentially here they used DL-based image-to-image transformations to manipulate the input image by adding augmentations. This technique was typically used in shopping apps to let the users try on clothing or makeup. Minaee, Liang, and Yan [3] mention several datasets that can be used to train the DL-models used in these apps. For example Deepfashion1[12] is a high quality dataset of over 800 000 pieces of clothing which can be used to train a DL-model to track pieces of clothing. Minaee, Liang, and Yan [3] also mention several AR applications powered by DL-algoritgms. For example VITON[13] is an AR application which lets the user choose a piece of clothing and then shows the user in that piece of clothing, and Deep Localized Makeup Transfer Network[14] is an AR application that can recommend makeup products to users and show them how'd they look with that makeup on.

3.2 Prototypes Similar to Ours

Before designing and describing our own software architecture it is worthwhile to look at applications other people have developed that solve a problem similar to ours. A great example of such an application is the work done by Pylvänäinen, Solis, Toivola, *et al.* [15].

The work conducted by Pylvänäinen, Solis, Toivola, *et al.* [15] started from a simple observation: They coulnd't find any mobile apps for microscopy education that incorporated AR and Virtual Reality (VR) features and step-by-step guidance. They then sent out a needs assesment survey to students to map out demand for such a software and found that 70% of the respondents showed interest in using such an app in their microscopy studies.[15] They then outlined goals for such an app and started to develop it. Pylvänäinen, Solis, Toivola, *et al.* [15] state that their application should:

1. Be a useful tool in teaching microscopy
2. Help its users to operate a microscope
3. Be a helpful tool with troubleshooting microscopy related issues
4. Be a tool that could be used to revive microscopy knowledge after a long pause in practicing microscopy skills

The app developed by Pylvänäinen, Solis, Toivola, *et al.* [15] consists of three sections: "Teach me microscopy", "Help me at the microscope" and "Help me to troubleshoot". What's relevant to this thesis is under "Help me at the microscope". In that section there is the option to view a 3D-model of a specific microscope commonly used in laboratories (Leica DM RXA microscope). Interactive step-by-step tutorials are also available for this microscope on various things, such as microscopy parts, setting optimal Köhler alignment and focusing the microscope on the sample. These tutorials are also usable outside the virtual microscope in the real world as the "Help me at the microscope"-section also acts as a marker-based AR-environment.[15] AR-markers put on the microscopes can also help students find different parts of the microscope and learn of their functions. This system can also be used to integrate microscope-specific information into this AR-environment.[15]

Pylvänäinen, Solis, Toivola, *et al.* [15] also conducted a questionnaire-based usability study on the app they developed, and found that using the app during microscopy education increased the students' confidence at later using the same microscope independently, without assistance. Furthermore 64% of the students reported that the app definitely helped them at learning microscopy and 90% of them reported that the app helped them recall microscopy skills later.[15]

Another quite similar project is the work carried out by Estrada, Paheding, Yang, *et al.* [4]. Their work was built on a simple goal: To enable students to have a better experience when learning how to use electrical engineering laboratory equipment. They aimed to do this by offering the students AR-based tutorials for various electrical engineering lab equipment, and by using Deep Learning (DL) methods to detect such equipment in the laboratory.[4] The long term goal of this project was to create a framework that could be later used to easily develop interactive smartphone apps for different laboratory devices integrating this concept.[4]

Essentially Estrada, Paheding, Yang, *et al.* [4] developed a superimposition-based AR-app with an integrated DL-model to be used for object detection. This application can be used for a template to create any AR-based tutorial for any device. Creating a tutorial for a device using this framework consists of three steps of work:

1. Training the DL-model to recognize the device
2. Creating an Augmented Reality User Interface (AR-UI) of the device, consisting of a 3D-model and User Interface (UI) panels
3. Creating step-by-step instructions that can be displayed using the AR-UI defined in the first step.

On top of designing and defining such a framework, Estrada, Paheding, Yang, *et al.* [4] actually developed an application using it. In this application the DL-model

was trained to detect various types of multimeters, oscilloscopes, wave generators and power supplies. They also created AR-UIs and step-by-step tutorials for using real multimeters.[4]

The application logic of the app by Estrada, Paheding, Yang, *et al.* [4] is as follows: First the DL-model detects (so classifies and localizes) the equipment. Next if a tutorial is available for that equipment, the UI notifies the user of this. If the user decides to view the tutorial, the AR-based tutorial gets loaded and the AR-UI gets superimposed on top of the real object. Then UI-panels are used to display the tutorial content.

Stepping outside of the laboratory setting, Van Gestel, Van Aerschot, Frantz, *et al.* [5] developed an AR-application for helping orthopedic procedures. Namely they wanted to create a tool that would act as a new real-time AR-based safety solution and guidance technique in the use of power tools in surgery. Prior to this there did exist other camera and AR-based surgery systems but Van Gestel, Van Aerschot, Frantz, *et al.* [5] noted them to be physically too large, expensive and time-consuming, which was said to limit their usefulness in assisting with performing surgeries. Van Gestel, Van Aerschot, Frantz, *et al.* [5] wanted to develop a solution that could run on a head-mounted display (HMD) so that a surgeon could use it while working with his/her hands. However one problem was that HMDs typically couldn't do accurate enough tracking for surgical use.[5]

Van Gestel, Van Aerschot, Frantz, *et al.* [5] don't describe the structure and logic of the software built for their task very deeply. They do mention building it for Microsoft's HoloLens headset, and circumventing the poor performance of its camera's tracking ability by using the built in infrared sensor instead.[5] They measured the infrared sensor's tracking accuracy to be below 1mm, accurate enough for surgical work.[5] Their software technically uses marker-based AR, however it must be noted that the markers they use are not physical markers, rather markers registered by an

infrared-tracked stylus at the key positions to the surgical operation.[5] These virtual markers were then used to show the users an AR-based guidance vector representing the desired drilling direction when performing the surgery. The vector would also change color to represent whether the current drilling direction was correct or not. If the surgeon was drilling in the right direction the vector would be green, otherwise the vector would be orange or red.[5]

This software was then tested by letting 18 people perform mock surgeries on wooden models.[5] Three surgery guidance techniques were compared: freehand surgery without guidance, proprioception-guided surgery and surgery using the new AR-tool.[5] The mock surgeries were quite simple: the wooden bone-models had defined entry and exit points between which the surgeons had to drill, with parts of the models being covered by a cloth to better simulate real conditions.[5] The success of mock surgeries was examined by measuring the distances and angles between the desired exit point and the actual exit points drilled by the surgeons.[5] They also performed statistical analysis on the drilling session results, considering the angle between the actual and planned exit points a variable dependent on the experience level of the surgeon, the guidance technique used and the desired drilling direction.[5] They found that the AR guidance tool they developed improved the surgeons' output regardless of experience level.[5] The AR tool was an especially useful tool when performing oblique, complex and angled drilling paths. With these drilling paths the difference between AR-guidance and traditional guidance was even more pronounced.[5]

Another prototype was developed by Monroy Reyes, Vergara Villegas, Miranda Bojórquez, *et al.* [1]. Their aim was to develop an AR system to aid novice users in using milling and lathe machines in a school manufacturing laboratory, and use this as a basis to measure the acceptance rate and performance of such a system in the field of education.[1] Since they designed a tool to aid with the physical operation

of industrial machinery, it was important to them that their tool was hands-free as the users would need to operate the industrial machinery simultaneously.[1]

Monroy Reyes, Vergara Villegas, Miranda Bojórquez, *et al.* [1] emphasized that their system was a Mobile Augmented Reality (MAR) system, so essentially it was technically an application built for the Android Operating System. Contentwise their application included tutorials for a milling and a lathe machine which would guide the users in tool setup, working material setup, machinery setup and starting the machines in question.[1] The AR-elements in this app included 3D-models of the machines themselves as well as additional tools, such as spanners and Allen wrenches, as well as text instructions with descriptions on how to perform the basic tasks, labels for helping the user in locating machinery components and tools, 3D arrows to indicate flow direction, and real time videos of task explanations performed by experts.[1]

Since the two-hand requirement made it undesirable to run the application on a smartphone, two AR-devices were chosen to run the application in the experimental phase. Firstly ORA-1 AR glasses, which are optical see-through glasses where the real world is observed through transparent mirrors placed in front of the eyes of the user, and the VR-PRO AR HMD, which is just a HMD with video see-through.[1] Their AR solution was simply Marker-based MAR with two kinds of markers: Frame markers (FM), which are traditional AR-markers, so frame patterns with encoded data in the frame, and Item Targets (IT) which are real world objects that the system tries to match to a 2D image using traditional CV-algorithms such as edge detection and corner detection.[1] The purpose of the markers in this system is to help the system detect the machinery, and provide the system with placement information for the augmentations and explanation videos.[1]

Monroy Reyes, Vergara Villegas, Miranda Bojórquez, *et al.* [1] describe the flow of the application as follows: first the user needs to start the system by scanning

the main menu marker. Then they will enter the main menu. From here the user can either start the application or get system help information. The system help information gives the users info on how to use the app. Once the application is started, the user can scan the device markers, which then lets the user scan the individual lesson markers and multimedia markers for that device. This lets the user access the interactive tutorials and educational videos which actually help him/her to learn the operation of that machine.[1]

This application was then tested by 16 students and teachers in the university manufacturing laboratory for an experiment.[1] In the experiment the subjects were first given a general introduction to AR, after which they were introduced to the AR-system developed for this study. Then they were asked to complete the lessons included in the AR-systems on one of the two hardware options that was available. Finally the participants were asked to complete a survey to gather results.[1] The survey was a 10-question survey with a Likert-scale scoring-system. The first five questions related to acceptance metrics (satisfaction, precision, understandability, explanativeness, attractiveness). The next three questions related to performance metrics (interface, speed, marker system). This was followed by a generic "Have you used mobile AR systems before?"-type of question. The last question was an oper answer field where they asked for feedback and comments.[1]

For each question they calculated a score by doing a Likert-scale to point conversion where "very bad" became 1, "regular" became 3, and "very good" became 5, and then calculating the average from all the answers. The scores for the different metrics were: satisfaction: 4; precision: 4; understandability: 4,5; explanativeness: 4; attractiveness: 4,5; interface: 3,5; speed: 4 and marker system: 4,5. Out of these interface was the weakest element, with only 50% of the respondents rating it as good or very good.[1] According to the feedback from the open answer field the teachers and the lab staff participating in this study were interested in using the

AR-system developed for this experiment as an educational tool later. The students also saw it as an appealing way to learn machine operation basics.[1] With this it is pretty clear to say that the system developed for this project has potential to be a real-life teaching-learning tool.

A somewhat similar tool was also developed by Lin and Lee [16]. With it they aim to use AR-simulation as a practical aid to deepen students' understanding of CNC machine operation. This should help the students to learn operate CNC machines independently while also helping to solve resource limitation problems in education. Oftentimes workshops in vocational education only have a limited number of CNC machines when compared to the demand, as well as insufficient space for operation. On top of there are also a limited amount of processing materials and the wearing of the machines themselves during operation is also a real issue.[16] All these problems lead to an outcome where students don't get enough practice on CNC machining.[16]

There are also issues with traditional education methods that make learning proper CNC machine operations difficult. For example it is hard for students to understand, how the physical size and movement capabilities of CNC machines relate to the workpiece models they create using modeling software.[16] Different three dimensional and complicated manufacturing processes and methods, such as turning over processing and mold hole processing among others, are extremely difficult to properly showcase without using three dimensional visualization.[16] Furthermore it is difficult for students to understand how the instructions sent to the CNC machine actually affect the workpiece itself during manufacturing.[16]

To solve these problems Lin and Lee [16] proposed using AR to simulate CNC machine operation and the machining process.[16] They built an AR application where the built-in virtual imaging technology can superimpose virtual 3D machining workpieces on the physical CNC machine using mobile hardware, such as tablets, as displays.[16] While Lin and Lee [16] do not discuss the workings or contents of their

application in-depth, they still disclose that they used AR-markers corresponding to different operation processes. They placed these markers on the CNC machine itself so that their system could be used in the real manufacturing workshop setting.[16] Lin and Lee [16] also disclose that their application includes pre-made animations for different manufacturing processes supported by the CNC machine.

Lin and Lee [16] used their application to conduct a ten participant study consisting of tool-aided CNC machine operation, followed by filling out System Usability Scale-type questionnaire (SUS). The participants had prior CNC machine experience, but were still novice level students.[16] These students used the app to simulate operating a CNC machine to manufacture an example furniture piece. The SUS consisted of 20 questions in total each scored 1-7, where 1 is high difficulty and 7 is low difficulty. The questions were of five different evaluation dimensions: "Understanding of Machining Procedures", "Time of Operation", "Accuracy of Operation", "Sequence and Information of the CNC Operation Steps" and "Understanding of the Interface Knowledge of the CNC Cutting Machine Center".[16] Each of these categories acquired a score of 4,86; 4,8; 5,52; 4,98 and 5,6 respectively.[16]

The results of the experiment showed that the AR system helped the students to understand the relationship between the workpieces and the CNC machining methods.[16] Lin and Lee [16] also listed the following advantages they found in using AR-based system for learning CNC machine operations:

1. Students understood and were able to complete necessary steps of using CNC machine for making furniture.

2. Students understood the relationship between the virtual representations of processing objects and the real-world procedure.

3. The system reduced the work needed to be done by the teachers in teaching operating the CNC machine to students.

4. Students were able to practice CNC machine operation more as they weren't limited by machine availability anymore. This also decreased material waste and safety risks.

5. Students were able to preview the interaction between models done by them and the cutting path of the CNC machine.

4 Architecture Description

4.1 Problem Description

In order to empirically examine the use of an AR application with a modern CV and DL-based tracking solution in a cooking environment, as defined in **RQ2**, a suitable prototype needs to be developed. In this chapter we aim to define this prototype: exactly what it'll need to do and what kind of challenges we might face during development. In chapter 5 we further describe the steps taken during development and the finished prototype.

The application developed for this thesis aims to guide the user through the recipe defined in listing 5.1. In it Computer Vision is used to track bowls, spoons, knives, apples, oranges and bananas. In chapter 6, the application will be used to conduct a usability test, where subjects are asked to prepare the recipe with the help of the application and their user experience is measured in a survey.

The goal would be to offer real-time guidance for all of the actions involved in making the fruit salad as defined in listing 5.1. AR-augmentations would need to be created for each of the actions involved. Examples of these actions are chopping the fruits and placing them in the bowl.

There is a case to be made here for using more advanced CV and DL-based methods for tracking. Most obvious among these is the simple fact that there are lots of things to track. Just in the case of our simple fruit salad, one needs to track three

different ingredients, which physically change shape as they are chopped into pieces during the preparation process. On top of the ingredients, different kitchenware such as a bowl and at least one knife needs to be tracked. It is pretty clear that such an application with this many different trackable objects and changing states would be extremely difficult to implement using traditional AR tracking methodologies, thus CV and DL need to be used.

Cooking was chosen as the example use-case because it is a real world problem where offering AR-based guidance would be useful and where using advanced CV-methods in tracking is justified. In 3.2 we also see prototypes developed by other teams using a similar step-by-step tutorial format for their AR-content. Both Pylvänäinen, Solis, Toivola, *et al.* [15] and Monroy Reyes, Vergara Villegas, Miranda Bojórquez, *et al.* [1] use this kind of approach to teach instrument use in laboratories, but their applications use traditional tracking methods that are developed for their specific instruments. If an easily retrainable modern CV-based tracking method is used, like in our application, one could theoretically just retrain the tracking module, and use it in a completely different environment, with any step-by-step tutorial, making the potential of this sort of application huge.

In order for our prototype to be usable in a kitchen environment it needs to be able to run on a mobile phone. However the prototype has to be easily portable to other platforms in the future. For this reason it will be developed as a web app. Testing the application will be performed on a mobile device.

4.2 Perceived Challenges

The first challenge in building the prototype is defining a recipe and deciding what to track. If there are too many things to track the complexity increases beyond the scope of a thesis. However if there are not enough things to track then the prototype will fail to demonstrate the potential of CV and AR in solving real world problems

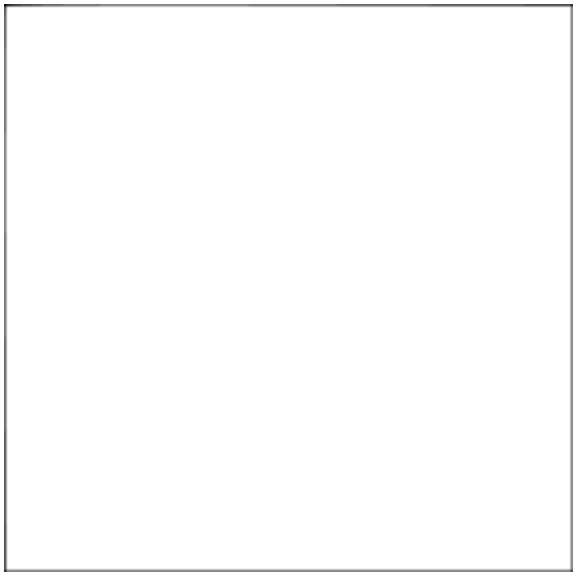

Figure 4.1: Visual Representation of the Proposed Architecture

and thus be useless.

4.3 Proposed Architecture

5 Implementation process

5.1 Defining the recipe

As already discussed in 4.1, in this work we're implementing an AR application that uses a CV and DL-based tracking solution to aid in cooking. To achieve this goal, one needs to train a DL-based object detection system, capable of detecting different ingredients and kitchen appliances used during the cooking process. To conduct this training process one needs to first collect a relevant dataset of images and annotations (location data within the image) of all object types we need to track. In essence this would mean collecting thousands of pictures, examining them by hand and creating appropriate annotation metadata for each image.

To avoid this costly data creation stage, we decided to utilize a pre-existing dataset. We use the 2017 version of Microsoft's Common Objects in COntext (COCO, MS-COCO) dataset, developed by Lin, Maire, Belongie, *et al.* [17]. This dataset contains annotated data for more than 90 object types, including apples, bananas, oranges, bowls, knives and spoons. With these specific object categories in mind, we've defined the following recipe, where each of the steps should be trackable using a DL-system trained with data from COCO:

```
Chop apples with a knife
Place apples to bowl
Chop oranges with a knife
```

```
Place oranges to bowl
Chop bananas with a knife
Place bananas to bowl
Stir apples, oranges and bananas in a bowl with a spoon
```

Listing 5.1: The recipe

5.2 Collecting the data

To download relevant images and annotations from the MS-COCO dataset[17], we used the Python library fiftyone developed by Moore and Corso [18]. This library allowed us to only download such images from COCO, that matched the object categories we were interested in. The images still had annotations from other categories too, so we wrote a Python script to trim all the unneeded annotations off of the downloaded data.

5.3 Training the CV module

Ghasemi, Jeong, Choi, *et al.* [2] mention that SSD MobileNet v2 is a suitable neural network to build a tracking solution for a mobile AR-application, due to it being lightweight and optimized for low-power devices. What's more Estrada, Paheding, Yang, *et al.* [4] actually built their DL-based AR-tracking solution using it. For these reasons we also wanted to use SSD MobileNet v2 to power our AR-application's tracking. To train an SSD MobileNet v2 neural network with our data we used the TensorFlow machine learning framework.[19]

Using the TensorFlow framework to train a neural network, requires the training data to be in a special .tfrecord data format.[20] We used an official TensorFlow tutorial by Patlolla, Neeli, Daoust, *et al.* [20] to package the image and annotation

data downloaded from COCO[17] into this data format.

More on CV training later:

- tf config files
- docker containerizarion
- trained model to tf.js conversion (this might be in 5.4)

5.4 Application backend

- How is the CV module integrated into the mobile app?
- How does the app track the steps of the recipe

5.5 AR-UI

How are augmentations rendered?

6 (USABILITY)

Justification for RQ3 here

RQ3: user experience

- Why measure user experience?
  - We want to demo the usefulness of these two techs working together
    - An app can only be useful if it's used
      - An app will only ever be used if it has good UX

7 (FEASIBILITY)

8 Conclusion and summary

8.1 Overview of Results

Research questions should be mentioned here again!

8.2 Answering Research Questions

8.3 Summary

References

[1] A. Monroy Reyes, O. O. Vergara Villegas, E. Miranda Bojórquez, V. G. Cruz Sánchez, and M. Nandayapa, "A mobile augmented reality system to support machinery operations in scholar environments", *Computer Applications in Engineering Education*, vol. 24, no. 6, pp. 967–981, 2016. DOI: https://doi.org/10.1002/cae.21772. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.21772. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.21772.

[2] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, "Deep learning-based object detection in augmented reality: A systematic review", *Computers in Industry*, vol. 139, p. 103 661, 2022, ISSN: 0166-3615. DOI: https://doi.org/10.1016/j.compind.2022.103661. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361522000586.

[3] S. Minaee, X. Liang, and S. Yan, *Modern augmented reality: Applications, trends, and future directions*, 2022. arXiv: 2202.09450 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2202.09450.

[4] J. Estrada, S. Paheding, X. Yang, and Q. Niyaz, "Deep-learning-incorporated augmented reality application for engineering lab training", *Applied Sciences*, vol. 12, no. 10, 2022, ISSN: 2076-3417. DOI: 10.3390/app12105159. [Online]. Available: https://www.mdpi.com/2076-3417/12/10/5159.

[5] F. Van Gestel, F. Van Aerschot, T. Frantz, *et al.*, "Augmented reality guidance improves accuracy of orthopedic drilling procedures", *Scientific Reports*, vol. 14, no. 1, p. 25 269, Oct. 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-76132-3. [Online]. Available: https://www.nature.com/articles/s41598-024-76132-3#citeas.

[6] B. Batuwanthudawa and K. Jayasena, "Real- time location based augmented reality advertising platform", in *2020 2nd International Conference on Advancements in Computing (ICAC)*, vol. 1, 2020, pp. 174–179. DOI: 10.1109/ICAC51239.2020.9357261.

[7] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii", *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006. DOI: 10.1109/MRA.2006.1678144.

[8] J. Kim, J. Seo, and T.-D. Han, "Ar lamp: Interactions on projection-based augmented reality for interactive learning", in *Proceedings of the 19th International Conference on Intelligent User Interfaces*, ser. IUI '14, Haifa, Israel: Association for Computing Machinery, 2014, pp. 353–358, ISBN: 9781450321846. DOI: 10.1145/2557500.2557505. [Online]. Available: https://doi.org/10.1145/2557500.2557505.

[9] R. Klette, *Concise computer vision* (Undergraduate Topics in Computer Science), en, 2014th ed. Guildford, England: Springer, Jan. 2014.

[10] X. Zhang, S. Fronz, and N. Navab, "Visual marker detection and decoding in ar systems: A comparative study", in *Proceedings. International Symposium on Mixed and Augmented Reality*, 2002, pp. 97–106. DOI: 10.1109/ISMAR.2002.1115078.

[11] *Opencv: Harris corner detection — docs.opencv.org*, https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html, [Accessed 27-05-2025].

[12] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, *Deepfashion: Powering robust clothes recognition and retrieval with rich annotations*, 2016. [Online]. Available: https://liuziwei7.github.io/projects/DeepFashion.html.

[13] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "Viton: An image-based virtual try-on network", *CoRR*, vol. abs/1711.08447, 2017. arXiv: 1711.08447. [Online]. Available: http://arxiv.org/abs/1711.08447.

[14] S. Liu, X. Ou, R. Qian, W. Wang, and X. Cao, "Makeup like a superstar: Deep localized makeup transfer network", *CoRR*, vol. abs/1604.07102, 2016. arXiv: 1604.07102. [Online]. Available: http://arxiv.org/abs/1604.07102.

[15] J. Pylvänäinen, J. Solis, D. Toivola, and P. Kankaanpää, "Supporting microscopy learning with ocul-ar, a virtual and augmented reality-powered mobile application", *CEUR Workshop Proceedings*, vol. 3393, pp. 28–37, May 2023, Technology-Enhanced Learning in Laboratories workshop (TELL) ; Conference date: 27-04-2023, ISSN: 1613-0073.

[16] Y.-T. Lin and I.-J. Lee, "Development of an augmented reality system achieving in cnc machine operation simulations in furniture trial teaching course", in *Virtual, Augmented and Mixed Reality. Industrial and Everyday Life Applications*, J. Y. C. Chen and G. Fragomeni, Eds., Cham: Springer International Publishing, 2020, pp. 121–135, ISBN: 978-3-030-49698-2.

[17] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1405.0312.

[18] B. E. Moore and J. J. Corso, "Fiftyone", *GitHub. Note: https://github.com/voxel51/fiftyone*, 2020.

[19] M. Abadi, A. Agarwal, P. Barham, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[20] L. R. Patlolla, S. S. K. Neeli, M. Daoust, and bharatjetti, *Object detection with model garden : Tensorflow core*, Feb. 2025. [Online]. Available: `https://www.tensorflow.org/tfmodels/vision/object_detection`.