Karlsruhe Institute of Technology

Designing User-adaptive Content for Mixed Reality Using Eye and Hand Tracking

---

Entwurf von benutzeranpassbarem Szeneninhalt für Mixed Reality mit Hilfe von Augen- und Handnachverfolgung

Master's Thesis

Jonas Heinle

1964256

at the Department of Informatics

Institut für Anthropomatik und Robotik (IAR)

Reviewer:	Prof. Dr.-Ing. Rainer Stiefelhagen
Second Reviewer:	Prof. Dr. Michael Beigl
Third Reviewer:	Prof. Dr. Alexander Maedche
Supervisor:	Dr. Peyman Toreini

22.05.2023

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

Affidavit

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, June 8, 2023

.....

(Jonas Heinle)

Dedicated to my beloved mother, Ursula.

Acknowledgment

With this thesis an awesome time comes to an end. But it only got so good as numerous people have spend advice, paid attention, proofread and generally spoken: have taken much time out of their day for me. For there were so many helpful people I'm not able to list everyone individually. So please don't feel neglected and let me say to everyone of you: Thank you so very much!

Nonetheless I would like in particular stress the excellent mentoring by Dr. Peyman Toreini, Tobias Ostertag and Dr. Anuja Hariharan I have received during my work. You have greatly helped me to put my research and scientific writing skills to a new level. I was able to learn more quickly through your advice as I was introduced to new fields and I had to adopt new techniques.

Abstract

With more and more powerful Mixed Reality (MR) hardware capabilities, MR applications are gaining ground in various industries such as construction, medicine, education etc. One of the main features of these tools is providing guidance by extending physical objects around the users with extra information. To reduce users' overload, such a real-world extension should map to users' tasks and intentions. However, currently MR applications are not aware of the users' intention and such lack of information influences the interaction quality between the user and the augmented world. This thesis exhibits interrelationships between users' currently fixated objects, their eye, head, hand movement and users' intentions while doing a task. Based on these findings this work provides user-adaptive guidance for improved interaction quality between MR environment and the user.

MR Anwendungen gewinnen durch immer leistungsfähigere Hardwarefunktionen in verschiedenen Branchen wie dem Baugewerbe, der Medizin, dem Bildungswesen usw. an Bedeutung. Eines der Hauptmerkmale dieser Tools ist die Bereitstellung von Orientierungshilfen, indem physische Objekte um den Benutzer herum mit zusätzlichen Informationen erweitert werden. Um die Überlastung der Benutzer zu verringern, sollte eine solche erweiterte Realität auf die Aufgaben und Absichten der Benutzer abgestimmt sein. Derzeit sind sich MR Anwendungen jedoch der Absichten der Benutzer nicht bewusst und ein solcher Informationsmangel beeinflusst die Interaktionsqualität zwischen dem Benutzer und der erweiterten Welt. Diese Arbeit zeigt Zusammenhänge zwischen den aktuell fixierten Objekten der Benutzer, ihren Augen-, Kopf- und Handbewegungen und den Absichten der Benutzer bei der Ausführung einer Aufgabe auf. Basierend auf diesen Erkenntnissen bietet diese Arbeit eine benutzeradaptive Anleitung für eine verbesserte Interaktionsqualität zwischen der MR Umgebung und dem Benutzer.

Contents

List of Figures

List of Tables

List of Abbreviations

$\chi(x)$ indicator function. 19

$\star$ Cross-correlation operator. 11

AP Average Precision. 14

AR Augmented Reality. 1, 8, V

AUC Area Under Curve. 43

BCE binary cross entropy. 15, 18, 20

CIoU Complete Intersection over Union. 15, 18

CNN Convolutional Neural Network. 8, 9, 11, 52, V

CRISP-DM Cross-industry standard process for data mining. vi, 3, 21, 22, 25

CSP Cross stage partial. 12, 16, 17, 38

CSV Comma-separated values. 23, 30, 53

CV Computer Vision. 4, 14

CVF Computer Vision Foundation. 16

DRAM Dynamic Random Access Memory. 36, 42

DTW Dynamic time warping. 10

FC Fully Connected. 16, 17

FLOPS Floating Point Operations per Second. 46, 47

FN False Negative. 13, 36

FOV Field of View. 48, 52, 55

FP False Positive. 13, 14, 29, 46

FPN Feature Pyramid Networks. 17, 18

FPR False Positive Rate. 14, 16

FPS Frames per second. 46, 47

GPU Graphics Processing Unit. 36, 42

HSV Hue Saturation Value. 42

i.i.d independent and identically distributed. 38

IOU Intersection over Union. vi, 14, 15, 19, 42, 46, 47, VIII

KNN k-nearest neighbors. 10, V

LR Learning Rate. 42

mAP mean average precision. 14, 46, 47, VIII

ML Machine learning. 3–5, 8–10, 21, 23, 36, 51, 55, V

MLP Multi Layer Perceptron. iv, vi, 8, 9, 11, 16, 21, 23, 37, 44, 52, V

MR Mixed Reality. iii, vii, 1, 2, 4–6, 8, 9, 23–25, 48, 49, 51, 55, V

MRTK Mixed Reality Toolkit. iv, 24, 25

MS-COCO Microsoft Common Objects in Context is a dataset for common computer vision tasks: https://cocodataset.org/. 14, 42

MSCOCO Microsoft Common Objects in Context. 29, 38

MSE Mean Squared Error. 15, 18

MTS multivariate time series. 10

NaN Not a Number. 21, 30, 44

NN Neural Nets. iv, vi, viii, 3, 10–23, 25, 28, 30, 36, 37, 42–44, 47, 48, 53

onnx Open Neural Network Exchange. 21, 25, 47

OvR One-vs-Rest. vi, 36, 43

PAN Path Aggregation Network. 16, 39, 41

PR Precision-Recall. vii, 14, 47, VII

ReLU rectified linear unit function. 12, 44

RGB Red Green Blue codification of an image. vii, 53

RL Reinforcement Learning. 8, 9, V

ROC Receiver operating characteristic. vi, 16, 43, 51

RQ Research Question. 2–5, 8, 9, 16, 21, 25, 27, 37, 48

S-UEQ Short User Experience Questionnaire. 49, 50, 52

SGD Stochastic gradient descent. 42

SiLU sigmoid linear unit function. 12

SLR Systematic Literature Review. vi, viii, 3, 4, 6–11, 16, 23, 52, 55, V

SPP Spatial pyramid pooling. 16, 18, 39

SVM Support Vector Machines. 8, 10, 22, V

TN True Negative. 13, 14

TNR True Negative Rate. 14

TP True Positive. 13, 36

TPR True Positive Rate. 13, 16, 36

TSC Time Series Classification. iv, 3, 9, 10, 34, 37, 52

UI User Interface. vi, 24, 25

UWP Universal Windows Platform. 25, 41, 47

VR Virtual reality. 8, V

WebRTC Web Real-Time Communication. 25

YOLACT You Only Look At CoefficienTs. vi, 16, 19, 41

YOLO You only look once. vi, 11, 13, 15, 16, 18–21, 29, 30, 39–41, 45, 46, 48, 49, 53

YUV Luminance(Y), chrominance component blue projection(U), chrominance component red projection(V) codification of an image. vii, 53

1. Introduction

Mixed Reality is increasingly gaining in popularity and relevance (Speicher et al., 2019). At work such as in manufacturing (Bottani and Vignali, 2019) it can substantially speed up the process. In learning context, it supports students to understand complex concepts (Moro, Birt, et al., 2021 Moro, Phelps, et al., 2021) or in health care it can augment the reality with e.g., useful information about organs for helping surgeons, etc. MR glasses represent the next step in immersive living and users interaction with mixed environment for increased hardware capabilities and off-the shelf already integrated technologies (in particular hand/eye/hand tracking as in HoloLens2). Furthermore traditional computer vision tasks like object detection (see Fang and Zhang, 2020) are gaining ground in applications for MR glasses. In the core of this technology, the user interaction with physical and augmented objects is important and considered as a research domain that needs further exploration (Speicher et al., 2019). Interaction techniques can be divided into implicit approaches which includes non scene changing techniques (for instance detecting objects) or explicit techniques which are considered as scene changing events based on user interactions (e.g. opening a context menu on an object in user focus). Designing efficient user interaction with augmented world is a challenge for users as the user only wants objects labeled based on their category (Gebhardt et al., 2019). However, the problem is that MR technology can detect objects around the users but user intention is not clear for this technology. Users are often interested in only specific parts of their surroundings rather than all. Therefore, MR applications need to know about users' intention and task before they can provide relevant information or extend the recognized objects with augmented information. Mainly, over fitting users with a lot of augmented options and information leads to bad user experience with MR technology, although they have great potential to support better-doing tasks.

In this research, we claim that one can substantially improve working processes by predicting the intention of the user to provide further guidance and aim to investigate that. However, best of our knowledge this capability is not investigated for MR yet.

Previous research by Holmqvist and Andersson, 2017, Duchowski, 2017 and Yin et al., 2018 shows that eye-tracking is a proxy for reading what is going on in users' minds. Furthermore, the body gestures and the hand positions can reveal what users want to do before they execute it. Therefore, in this research, we aim to use integrated eye-tracking technology on Hololens2 and the gesture tracking feature to predict users' intentions and design user-adaptive guidance based on that. To reach this goal, we focus on a specific use case (truck CAD model investigation in AR) and build thesis steps based on that.

Therefore, the main two **Research Questions (RQs)** for this thesis are as follows:

1. *How to detect user intentions in MR applications through eye, hand and head tracking features?*
2. *How to design user-adaptive guidance for MR applications based on detected user intentions?*

Overall, this thesis has the following steps:

- *Develop a system that identifies objects in MR using two methods: object detection, instance segmentation and investigate which method is most suitable for achieving user-adaptive guidance in the identified use case.*
- *Design and develop a system that detects user intentions via tracking users' eyes, head and hand positions.*
- *Conduct a user study to investigate if the developed user-adaptive guidance is satisfying.*

This work offers following core contributions:

1. *We provide a novel way of training jointly on head, eye and hand user input data for user intention prediction.*
2. *We investigate new ML approaches for user intent predictions and conclude their effectiveness with user studies for a specific use case.*

2. Related Work

The following chapter discovers and examines previous work relevant to the RQs from Section 1. The analysis of previous work will help to discover already used approaches and techniques for solving relevant tasks. For this I will conduct a Systematic Literature Review (SLR) in Section 2.1 following the guidelines provided by Brereton et al., 2007. Subsequently I can identify research gaps and having further discussions in Section 2.1.5.

Based on the findings of the SLR Section 2.3 introduces theoretical foundations on Time Series Classification (TSC) as parts of the RQ will be posed as such. Moreover an introduction to all relevant NNs approaches with their associated building blocks, activation functions and metrics will be given in Section 2.3.

I end this chapter by introducing CRISP-DM. This cross-industry standard data mining process guides the proceeding of building an appropriate ML model and its accompanying data collection steps.

2.1. Systematic Literature Review

With the aid of Figure 2.1 the following Section gives explanations of the SLR as described by Brereton et al., 2007.

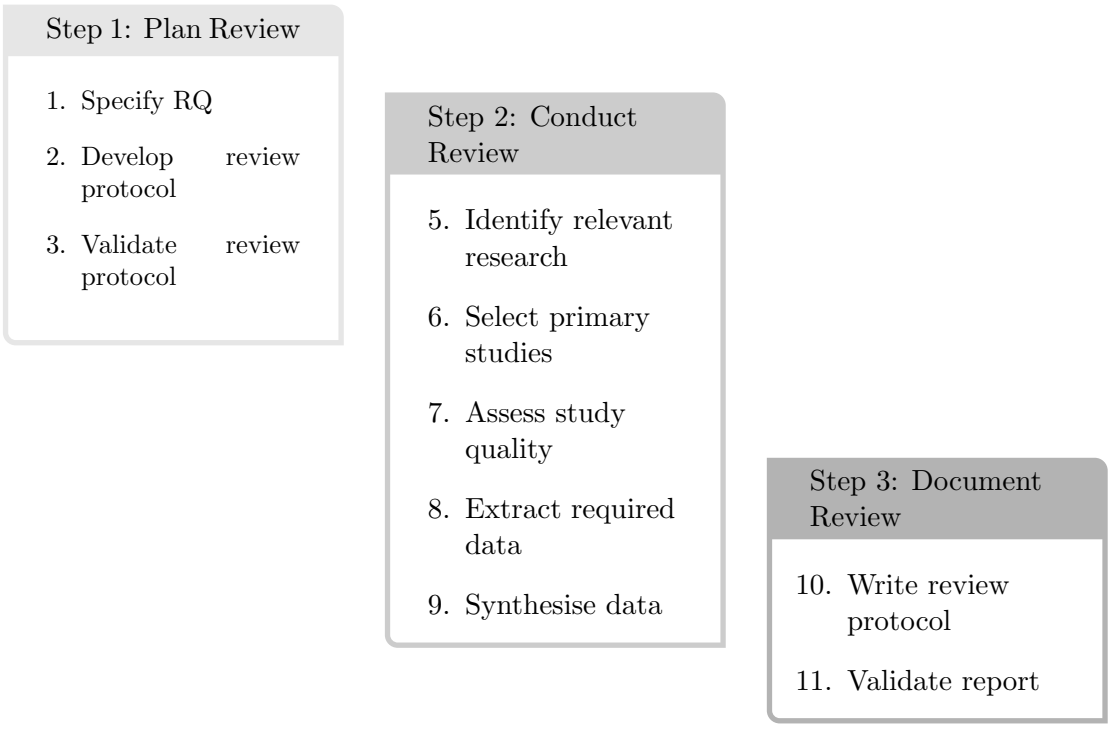


Figure 2.1.: All relevant steps for my SLR

The first step of the SLR will plan the review. This implies in the very beginning the specification of an appropriate RQ. This specification further allows to develop and validate a review protocol. As a product of the planning phase the review protocol then serves as a guideline for the execution phase. Conducting the SLR allows to identify all relevant research for the RQ. Under these found literature just high quality primary studies are extracted and further examined. This synthesised data is then documented as a final step.

2.1.1. Plan Review

The ultimate goal of this Section is to produce a review protocol. As described by Brereton et al., 2007 a review protocol aims to minimise bias by outlining as exact as possible how the SLR is conducted. This protocol itself should be reviewed for minimizing errors in the earliest stage as possible.

The absolute mandatory step of defining a RQ was already done in Section 1. This formulation is the core of the review protocol and helps to express the search strings in the next Section 2.1.2. For building search strings it is also helpful to visualize and list all relevant research topics. These research areas are depicted in Figure 2.2. The shaded area in this Figure is exactly the place where this work resides and comprises four major areas. One major research topic I identified for this work is ML as I want to process user/object data automatically with corresponding responses. The next significant research topic is about MR as the goal of this work is to extend users' reality. In the top of Figure 2.2 one can identify the next great research topic: user data tracking. The automated response to users entails extensive tracking. This work focuses on tracking users eye, hand and head with the help of the MR device HoloLens2 which in further details is introduced in Chapter 3. Last but not least I want to talk about the major research topic user guidance that stretches out over the bottom part of the Figure. As formulated in the RQ I want to provide user guidance based on the users' intention. This field can have numerous paraphrases. Furthermore I categorize the topic object detection/instance segmentation (see Section 2.3.4) as an intersection between the ML and user guidance domains. The reason is as follows: detecting and segmenting objects is a classic domain in CV and therefore a sort of ML. Detected/segmented objects of the users' visual field on the other hand do also contain important information for their guidance.

Regarding Figure 2.2 I want to stress the following: the equal sizes of all circles around the major research topics do mean equal importance. The spatial layout (top, bottom, right, left) however does not have a meaning of priority ordering. Links to appropriate Sections are only given if there are specially dedicated Sections for it. Otherwise they are covered by a superordinated Section.

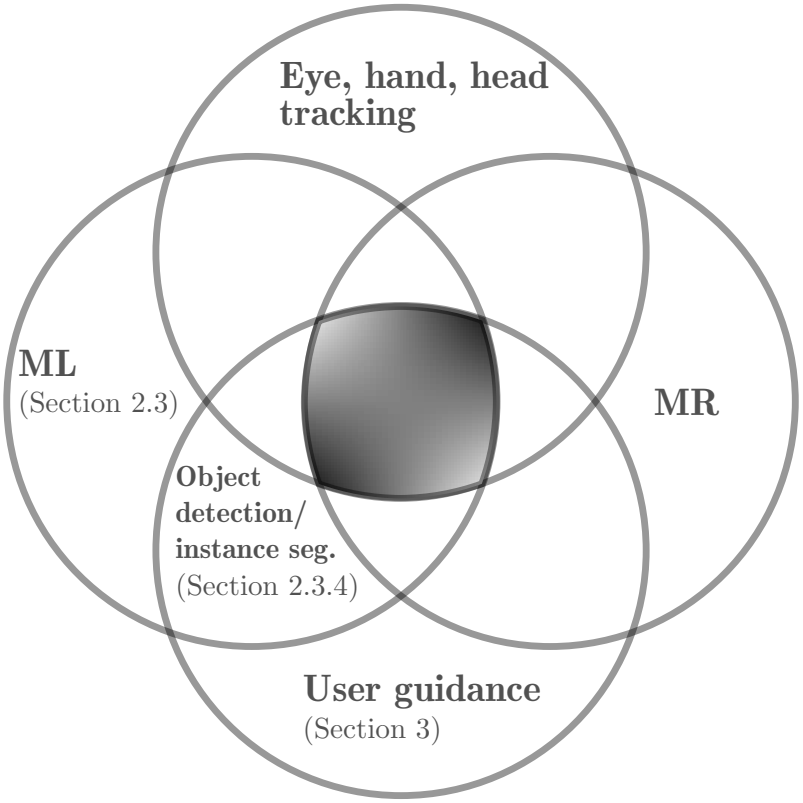


Figure 2.2.: Research topics

2.1.2. Develop Review Protocol

After outlining all relevant research areas and formulating the RQs I can subsequently develop the review protocol based on these information. The review protocol is the leading rule book for the second step: the actual execution of the review. Therefore I set the ACM Digital Library as the primary source of information. I will use their online search tool[1] for finding the most relevant research paper for me and my work. Hence I also use the ACM Digital Search syntax which consists of two main fields:

1. The query field which includes the actual query. A query has a sequence of search strings who are linked to each other with logical operators such as *AND,OR*. These logical operators determine the relationship of the strings. The asterix * allows to search for any number of characters where it is defined (f.e. with guid* I can allow to search for guide and guidance). If a word is placed in quotation marks the exact string will be searched for.

2. The filter field is responsible for omitting each intermediate result which does not obey the here listed criteria. Typical criteria among many others are f.e. the date (as I am only interested in state of the art results), conferences (as only top tier conferences should be considered)

[1] ACM. (2022). Acm digital library advanced search. https://dl.acm.org/search/advanced.

The initial full query syntax is as follows:

1.)	*"query":*	*AllField:("machine learning") AND AllField:("eye tracking" OR "hand tracking" OR "head tracking") AND AllField:(guid*) AND AllField:("Mixed Reality")*
	"filter:"	*Conference Collections: CHI: Conference on Human Factors in Computing Systems,E-Publication Date: (01/01/2013 TO 12/31/2023)*

Table 2.1.: Full query syntax for planned SLR

Table 2.1 reveals the complete query syntax I plan to use in the next step. I reject results older than 10 years. Also I am only interested in papers from the top tier CHI conference. The query field is made of the previously defined major research domains.

2.1.3. Conduct Review

As the review protocol has been defined previously I can now proceed in conducting the review. The search for papers was done with the query from Table 2.1. The idea the query consists of all major research domains (see Figure 2.2) resulted in to less papers. So this search string is to comprehensive. This first result can already be a first hint of a research gap which I explain in detail in Section 2.1.5. The absence of useful results brought me to rephrase the query by splitting it into two parts. They look as follows:

1.)	*"query":*	*AllField:("Mixed Reality") AND AllField:(guid*) AND AND AllField:(eye OR head OR hand) AND AllField:(track*)*
	"filter:"	*Conference Collections: CHI: Conference on Human Factors in Computing Systems,E-Publication Date: (01/01/2013 TO 12/31/2023)*
2.)	*"query":*	*AllField:("machine learning") AND AllField:("eye tracking" OR "hand tracking" OR "head tracking") AND AllField:(guid*)*
	"filter:"	*Conference Collections: CHI: Conference on Human Factors in Computing Systems,E-Publication Date: (01/01/2010 TO 12/31/2022),ACM Content: DL*

Table 2.2.: Revised version of the full query syntax for SLR

With this split enough relevant research papers could be found (see Figure 2.3). The main idea behind this split is as follows: both strings "machine learning" and "MR" are outsourced because this research fields can be considered separately. But they still have the same filters like the prior first query. And I am still only interested in results concerning guidance and user tracking.

SLR Overview

Steps	Search Results 1.) Query	2.) Query
1.) ACM Digital Library Advanced Search	324	152
2.) Analyze the titles, abstracts and keywords	46	32
3.) Analyze introduction and conclusion	13	12
4.) Analyze full content	7	7
5.) Forward & Backward Search	7	10

Table 2.3.: Individual steps of conducted SLR with search results

Table 2.3 gives a detailed listing of each individual step of the SLR alongside the remaining papers. In the very beginning there were a solid amount of papers (in contrast to the first search). But after analyzing the titles, abstracts and keywords duplicates and false positives could be removed which resulted in a set of literature reduced by a factor of ~ 6. In a subsequent step I examined all remaining papers additionally on their introduction and conclusion which had as a consequence that several other false positives were found and thrown out. I analyzed the 14 papers which were left by there full content and was able to exclude another set of papers. The very last step turned out to be precious. I was able to find very good papers by searching others who were cited or cite the paper respectively (named forward & backward search accordingly).

Based on the lasting 17 papers I build my previous work section which influences all further steps. But first I bring some structure into the results. Only with a viable structure I am able to conclude with found research gaps and further discussions in Section 2.1.5.

2.1.4. Document Review

In the third step of the SLR I want to analyze the 17 research papers found during the previous execution phase. The resulting papers are from the following sources and can be stated as very good conferences[2]:

- CHI: Conference on Human Factors in Computing Systems
- ACM Symposium on User Interface Software and Technology
- Proceedings of Computer Graphics International 2018
- ACM Symposium on Eye Tracking Research and Applications

[2]Google. (2023a). Top publications.

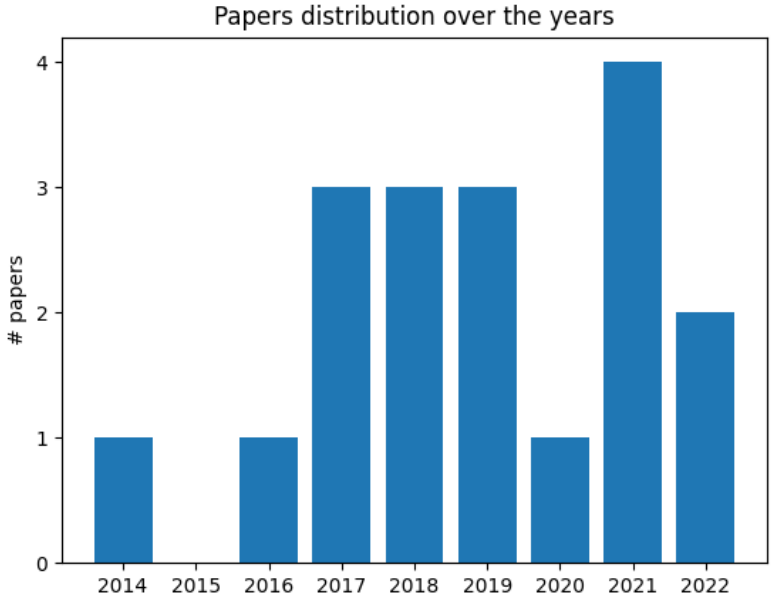


Figure 2.3.: Amount of relevant papers over the last few years

Regarding the content I ensured in the previous step that all papers do have much value for us. An other good metric is that they are contemporary. This is surveyed in Figure 2.3 which shows us that many found research is located in the last five years. I can infer that much work regarding my topic is currently done and therefore from high relevance.

I want to finish this Section by a classification of the found papers. The coding Table A.1 classifies each paper that survived the conducted SLR from Section 2.1.3 into the previously defined categories from Section 2.1.1 with additional subcategories. The subcategories were incrementally built by classifying each paper one after another. For the main research topic MR I was able to detect three subcategories: VR, AR and HoloLens2. The first two subcategories hint that a decent amount of recently relevant published works also deal with MR. This emphasizes my strive to follow the previously defined RQs. Regarding the last subcategory: as some works have also chosen the HoloLens2 for their work it is a viable option for me too.

If one focuses on the tracking capabilities in the resulting papers you can find a vast majority of papers using eye tracking. This reinforces my focus on eye tracking in this thesis. The minority of papers are using hand and/or head tracking for solving their task. This hints to a research gap and should be further investigated.

Now I want to put the focus on the research topic of ML. Figure A.1 makes clear there is no unique ML approach for answering the RQ. In a first step I divided all found approaches into two parts: traditional algorithms and more modern deep learning approaches. Much literature still uses very common techniques. Only the newer ones consider deep learning approaches. Among what is considered as deep learning the CNN and RL approaches are each seen only once. An MLP whereas is used by two research groups. The SVM is the

most used ML approach.

Based on these findings the following chapter will discuss the most auspicious techniques.

2.1.5. Discussion and Research Gaps

Based on the final report of my SLR I see the most promising approach for answering the RQ given by Pakdamanian et al., 2021. There are several reasons for it: First and foremost I see a conjunction of eye tracking, a state of the art ML method and visual user guidance. It is not exactly the same as what I formulated in the RQ though comes closest from all found literature. Their used ML method was also used by another research team by Karolus et al., 2017 and is therefore more promising than the lonesome RL technique used by Gebhardt et al., 2019 or the CNN approach by X. Wang et al., 2019. Moreover I do prefer a deep learning approach to traditional methods as they perform better on similar tasks as described by Pakdamanian et al., 2021.

Nonetheless is the work by Schwarz et al., 2014 important for my further actions. They offer valuable insights in training jointly on eye, hand and head user input data in MR. This combination is unique among the found literature.

Pakdamanian et al., 2021 pose the problem of user intention classification as a Time Series Classification (TSC) problem. Ismail Fawaz et al., 2019 demonstrates that not only they but numerous other researchers are using MLP for TSC. Hence I will adopt this approach and introduce these techniques in Section 2.2 and 2.3.

But all research mentioned until now don't mention object detection/instance segmentation. Hahn et al., 2018, Vazquez et al., 2017, Schwarz et al., 2014 and Koochaki and Najafizadeh, 2018 though are using some form of object detection for solving the task of user intention detection. This leads to the secondly formulated RQ of Section 1. The upcoming Section 2.3.4 is devoted to this topic.

In the past chapter I showed strong foundations where I can build up for answering the RQ. However does the past chapter also show research gaps which I want to answer in the next chapters. For example was it hard to find relevant previous work for the RQs (see Section 1) which hints to a research gap. No found approach is using instance segmentation for solving their task. There is also no approach that uses a MLP approach and learns jointly on head,hand,eye tracking data for user intention detection. Therefore I will provide a novel way of training jointly on head, eye and hand user input for user intention detection. The way I will pose the problem is a novel ML approach.

2.2. Time Series Classification

The SLR of Section 2.1 resulted in the proposal of formulating the user intention prediction as a TSC problem. Before I can now use this technique for the user guidance system (see Section 3) a strong theoretical background is needed. The following Section delivers exactly this.

I adopt the definitions of a time series by Ismail Fawaz et al., 2019 as follows:

Definition 2.2.1 (univariate time series X)**.** A univariate time series $X = (x_1, x_2, ..., x_T)$ is a sequence of values $x_i \in \mathbb{R}$.

Based on the knowledge about univariate time series a definition for multivariate time series can be provided:

Definition 2.2.2 (multivariate time series (MTS) $\hat{X}$)**.** A multivariate time series $\hat{X} = (X_1, X_2, ..., X_M)$ is a composition of M different univariate time series $X \in \mathbb{R}^T$.

Moreover these time series $\hat{X}$ can be classified:

Definition 2.2.3 (labeled time series dataset)**.** A set $S = (\hat{X}_1, Y_1), (\hat{X}_2, Y_2), ..., (\hat{X}_N, Y_N)$ is a labeled time series dataset of MTS $\hat{X}_i$ with their corresponding class label Y_i.

The concrete instantiation of definitions 2.2.1, 2.2.1 and 2.2.3 for the user (intention detection/guidance) system will be given in Section 3.3.4.1.
As you can see in definition 2.2.3 time series classification is a supervised ML technique which can be implemented in different ways:[3]

- Distance-based approaches (e.g. KNN,SVM,DTW) which uses a distance metric (e.g. Euclidian,Hamming,Manhattan,Minkowski) to classify samples
- Shapelet based approaches are time series subsequences which have in some sense maximum discriminative features.
- Model ensemble based approaches are collections of individual TSC classifiers. Majority vote over all classifiers gives final result.
- Dictionary based approaches with a key-value structure.
- Interval-based based approaches (e.g. time series forest) split the time series in individual intervals. Each interval is then used to separately train a classifier.
- Deep learning based approaches (see Section 2.3.3 for NN structures and Section 3.3.4.1 for concrete implementation details)

[3]IBM. (2023). Tsc at ibm. https://developer.ibm.com/learningpaths/get-started-time-series-classification-api/what-is-time-series-classification/.

As the previously found literature from Section 2.1 uses deep learning and the fact that most high accuracy time series classifier recently published are using deep learning or ensemble methods (see Guillaume et al., 2021) prompted me to introduce NNs in the following Section.

2.3. Neural Nets

The conducted SLR has proven the importance of NNs for the user guidance system as they provide state of the art performance in classification tasks. This serves as motivation for the upcoming Sections.

2.3.1. Building blocks

This Section now introduces all NN building blocks which are used in my user guidance system in Section 3. The YOLO approach (introduced later in Section 2.3.4.1) heavily relies on 2D-Convolution as defined in eq. 2.1. The syntax of all following equations conform the conventions of PyTorch[4]. Therefore all tensors are given with the shape: (N,C_{in},H,W) where N indicates the batch size, C_{in} the number of channels, H the input height and W the input width.

$$out(N_i, C_{out_j}) = \sum_{k=0}^{C_{in}-1} w_{C_{out_j},k} \star input(N_i, k) \tag{2.1}$$

$\star$ depicts the cross-correlation operator.

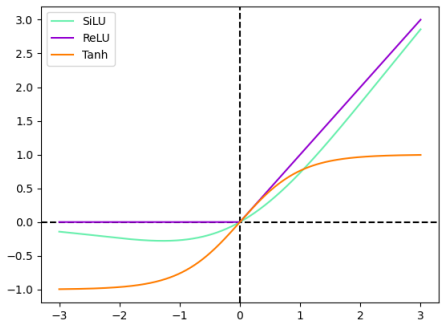
A Convolutional Neural Network (CNN) like YOLO also often uses a 2D-MaxPooling layer. They are simply defined as follows:

$$\begin{aligned} out(N_i, C_j, h, w) = \max_{m=0,\ldots,kH-1} \max_{n=0,\ldots,kW-1} input(N_i, C_j, \quad & stride[0] \cdot h + m, \\ & stride[1] \cdot w + n) \end{aligned} \tag{2.2}$$

with kH representing the kernel height and kW the kernel width.

The just now mentioned YOLO model and the later introduced MLP both NN approaches heavily rely on activation functions (all depicted in Figure 2.4).

[4]PyTorch. (2023). Pytorch. https://pytorch.org/.



$$ReLU = max(0, x) \tag{2.3}$$

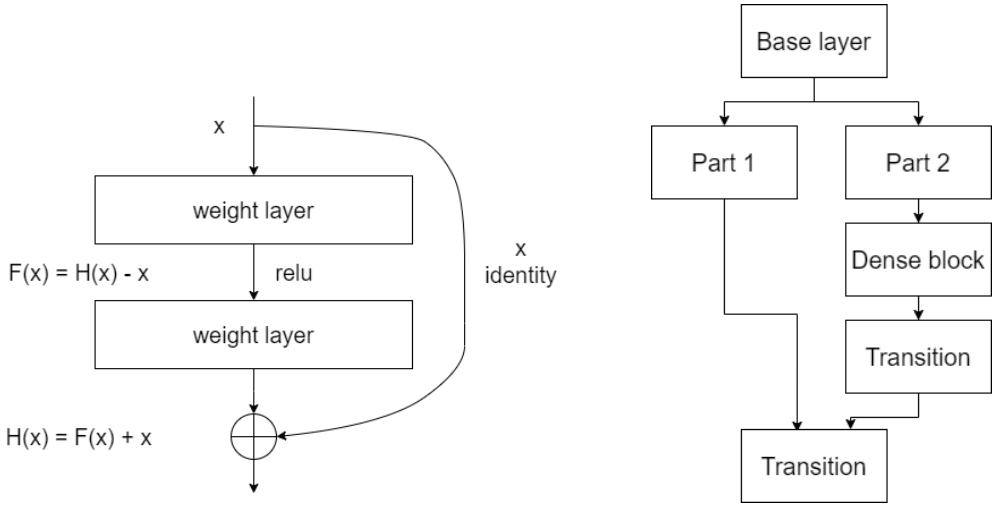
$$SiLU = x \cdot \sigma(x) \tag{2.4}$$

$$tanh(x) \tag{2.5}$$

Figure 2.4.: Activation functions I use for my NN

The ReLU function from eq. 2.3 is used in the user intention prediction module. The SiLU is a sort of enhancement of the ReLU function and is utilized in the user detection system for it yields better results in practice. The tanh in eq. 2.5 has the interesting property of allowing negative numbers in comparison to ReLU. This becomes important in a later chapter.

For further ease training of very deep and wide NN the investigated network structure in Figure 2.7 relies on residual connections as introduced by K. He et al., 2015. These connections enable the layers of a NN to directly fit a residual mapping $F(x) = H(x) - x$. The author K. He et al., 2015 assumes that these residual mappings are easier to optimize than the original.



(a) Residual connection

(b) CSP structure

Figure 2.5.: Building blocks for ease training of deep and wide NN

The implementation of the Darknet53 (see Figure 3.17) is not only making use of **residual connections** but also the **CSP** technique (see Figure 2.5b) as introduced by C.-Y. Wang et al., 2019 which consists of two main steps:

1. split feature map into two separate parts
2. merge the two parts through cross-stage hierarchy

Both steps can be seen in Figure 2.5b. The splitting of step 1 has as a consequence that the gradient flows through different paths and can result in large correlation difference in gradient information by switching concatenation and transition steps (resulting path: transition → concatenation → transition). This explains the reduced computation steps and lower inference time. One can spot both extensions in the blocks of the later implemented user guidance system C3 and C4 in Figure 3.16.

The following YOLO architecture makes use of **batch normalization** as described by Ioffe and Szegedy, 2015 to accelerate training by reducing internal covariance shift. This is achieved by recentering and rescaling the #m inputs $x = (x^{(1)}, ..., x^{(d)})$ of a mini-batch as follows:

$$\begin{aligned} \mu_B &= \frac{1}{m} \cdot \sum_{i=1}^{m} x_i \\ \sigma_B^2 &= \frac{1}{m} \cdot \sum_{i=1}^{m} (x_i - \mu_B)^2 \\ \hat{x}_i^{(k)} &= \frac{x_i^{(k)} - \mu_B^{(k)}}{\sqrt{(\sigma_B^{(k)})^2 + \epsilon}} \\ y_i &= \gamma \cdot \hat{x}_i + \beta \end{aligned} \tag{2.6}$$

γ and β in eq. 2.6 are considered as hyperparamters.

2.3.2. Metrics

This work uses common machine learning metrics as stated by Powers, 2020 to evaluate NN models. Firstly definitions for a binary classification scenario are given. The accuracy is defined as follows:

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

For the precision holds the following equation:

$$\frac{TP}{TP + FP} \tag{2.8}$$

For evaluating the sensitivity of NN models the recall(TPR) is defined by:

$$\frac{TP}{TP + FN} \tag{2.9}$$

Similar to that the specificity(TNR) of NN models can be calculated with following equation:

$$\frac{TN}{TN + FP} \tag{2.10}$$

And from this follows the False Positive Rate (FPR):

$$1 - TNR = \frac{FP}{FP + TN} \tag{2.11}$$

With both equations 2.8 and 2.9 the F1 score is defined accordingly:

$$2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.12}$$

The precision (see eq. 2.8) gives the confidence of the model in classifying samples. The model can have a very high precision and making a very good job in (almost) no falsely classifying samples as positives. But precision is almost always used in conjunction with the recall measure (see eq. 2.9) because it is a measure for how sensitive the model is for good predictions. The model could have a very high precision but concurrently doing very bad in recalling true positives. It is very interesting to see how much precision the model reaches for specific recall values and vice versa. Therefore a Precision-Recall (PR) curve is an important evaluation metric.

An other very important metric is the mean average precision (mAP). It is very popular in the CV community so e.g. used in the MS-COCO data sets and its challenges by Lin, Maire, et al., 2014. The Average Precision (AP) is defined as the area under the PR curve as follows[5]:

$$AP = \sum_{n} (R_n - R_{n-1}) \cdot P_n \tag{2.13}$$

One can can calculate the AP as shown in Equation 2.13 for each class. The mAP is the mean over all classes and can be calculated as in equation 2.14.

$$mAP = \frac{1}{\#classes} \cdot \sum_{c \in classes} AP \tag{2.14}$$

The mAP as in equation 2.14 can be measured at a fixed Intersection over Union (IOU) (noted as mAP@0.5 for an IOU of 0.5) and with different IOU levels spread over an intervall (noted as mAP@0.5:0.95). The IOU, also known as Jaccard index is described by Taha and Hanbury, 2015 as a widely used metric for image analysis. The IOU is used and calculated as in equation 2.15.

[5]Scikit-learn. (2022). Scikit learn post about precision, recall and f-measures. https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics.

$$IOU(A,B) = \frac{A \cap B}{A \cup B} \tag{2.15}$$

Figure 2.6.: The IOU 2.15 and one exemplary usage of ground truth bounding box A (red) and the predicted bounding box B (blue)

An extension to the IOU is given with the Complete Intersection over Union (CIoU)-loss by Zheng et al., 2019 as follows:

$$\begin{aligned} L_{CIoU} &= 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha \cdot v \\ \alpha &= \frac{v}{(1 - IoU) + v} \\ v &= \frac{4}{\pi^2} \cdot (\arctan(\frac{width^{gt}}{height^{gt}}) - \arctan(\frac{width}{height})^2 \end{aligned} \tag{2.16}$$

where b and b^{gt} are the center points of the predicted and ground truth bounding box respectively, c is the diagonal length of the smallest enclosing box covering the two boxes and v specifies the consistency of both aspect ratios. The used YOLO model favors this extended version instead of the original for it shows superior results in practice.

The models are also using the **binary cross entropy (BCE)**(see eq. 2.17, f.e. for calculating their loss) of a sample x_n and its corresponding label y_n:

$$y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n) \tag{2.17}$$

Other losses are calculated with the Mean Squared Error (MSE):

$$\frac{1}{N} \cdot (x_n - y_n)^2 \tag{2.18}$$

As the system is classifying samples the confusion matrices C[6] are important for visualizing classification results of NNs:

Definition 2.3.1 (confusion matrix)**.** A confusion matrix C is a two dimensional matrix. Let i specify an entry on the x-axis and j an entry on the y-axis then $C_{i,j}$ reveals how many samples known to be in group i are predicted to be in group j.

[6]scikit learn. (2023). Scikit learn entry about confusion matrix.

For measuring the quality of a classifier the ROC is also used:

Definition 2.3.2 (Receiver operating characteristic (ROC))**.** The ROC curve displays the TPR against the FPR. The goal is to push the curve to the upper left of the chart.

2.3.3. Multi Layer Perceptron

A MLP is a Fully Connected (FC) network architecture. This implies that neurons from layer l_i are connected to every neuron of layer l_{i-1} with $i \in [1, L]$. For every layer i and neuron j those connections are weighted with $\omega_{l_{ji}}$ respectively. As stated by Ismail Fawaz et al., 2019 the activations A_{l_i} of each layer i can be formulated as follows:

$$A_{l_i} = f(\omega_{l_i} \cdot \hat{X} + b) \tag{2.19}$$

As I pose the user intention detection task as a time series classification problem the input X in equation 2.19 will be a multivariate time series. Also f represents an activation (discussed earlier in Figure 2.4). The following Section 3.3.4.1 will show in Figure 3.15 the concrete MLP implementation for the user intention NN.

2.3.4. Object detection/instance segmentation

The SLR in Section 2.1 has shown the importance of object detection/instance segmentation for the RQ. This Section now examines which exact approaches will be used further. For this I searched in all conferences sponsored by the Computer Vision Foundation (CVF)[7] over the past seven years.

2.3.4.1. YOLO

The most promising approach for object detection is given by Redmon, Divvala, et al., 2015 with You only look once (YOLO) (and its subsequent work by Redmon and Farhadi, 2016 and Redmon and Farhadi, 2018). Not only for object detection but also for understanding You Only Look At CoefficienTs (YOLACT) by Bolya et al., 2019 in the next Subsection 2.3.4.1 it is important to have further investigations into this technique.
The used YOLO model for the user guidance system has the same building blocks as YOLOv4 by Bochkovskiy et al., 2020 namely (numbers indicate ordering of stages)

1. an input image as very first stage
2. the backbone CSPDarknet53 which is a Cross stage partial (CSP) extention as introduced by C.-Y. Wang et al., 2019 of the Darknet53-backbone (see Figure 2.7)
3. a neck consisting of a Path Aggregation Network (PAN)(first published by Liu et al., 2018 and a Spatial pyramid pooling (SPP)(see K. He et al., 2014)
4. a YOLOv3 head.

In the following each step will be analyzed as good as their interconnections.

[7] Foundation, C. V. (2023). The computer vision foundation. https://openaccess.thecvf.com/menu.

Darknet53-Backbone				
Module	#Filters	Size,Stride	Output	
Conv2D	32	3x3	256x256	
Conv2D	64	3x3,2	128x128	
Conv2D	32	1x1	-	
Conv2D	64	3x3	-	x1
Residual	-	-	128x128	
Conv2D	128	3x3,2	64x64	
Conv2D	64	1x1	-	
Conv2D	128	3x3	-	x2
Residual	-	-	64x64	
Conv2D	256	3x3,2	32x32	
Conv2D	128	1x1	-	
Conv2D	256	3x3	-	x8
Residual	-	-	32x32	
Conv2D	512	3x3,2	16x16	
Conv2D	256	1x1	-	
Conv2D	512	3x3	-	x8
Residual	-	-	16x16	
Conv2D	1024	3x3,2	8x8	
Conv2D	512	1x1	-	
Conv2D	1024	3x3	-	x4
Residual	-	-	8x8	

Figure 2.7.: Darknet53-Backbone

The **backbone** serves usually as feature extractor. For it can take on this role Redmon and Farhadi, 2018 pre-train their backbone on ImageNet (see Russakovsky et al., 2015 for more details). The input image (step 1) is consumed by the Darknet53-Backbone (step 2). As posed by Redmon and Farhadi, 2018 the FC network serves as feature extractor and consists of 53 convolutions in total with filter sizes of 3x3 or 1x1. Each highlighted block (black frame) consists of two 2D-Conv (calculated as in eq.2.1) and one earlier mentioned residual connection (see Figure 2.5a). These residual building blocks allow the Darknet to grow that deep in comparison to the predecessor Darknet-19 by Redmon and Farhadi, 2016. Regarding the number of filters one can moreover discover the typically structure of an backbone which serves as features extractor: the number of filters increase through the net whereas the resolution decreases.

The implementation in the user guidance system uses an altered CSP Darknet53 Backbone (see Figure 3.17). The CSP extension mitigates the problem of heavy inference computations (as described by C.-Y. Wang et al., 2019) in very deep and wide NN.

The **neck** is inserted between the backbone and the head. It gathers features from different levels of the backbone while walking several top-bottom and bottom-top paths in the feature hierarchy. It combines features from the top of the feature pyramid (low resolution and semantically strong) with features from the bottom (high resolution and semantically weak) resulting in a feature pyramid with strong semantic features across all scales as described by Lin, Dollár, et al., 2016. Or in other words: the top-bottom paths in the neck produce semantically strong but spatially more inaccurate results (as they introduce upsampling operations). This is alleviated by lateral connections to the backbone (bottom-top network) whose feature layers are less processed and therefore exhibit more precise localization information. Liu et al., 2018 enhanced the Feature Pyramid Networks (FPN) approach by introducing an additional bottom-up path augmentation. This augmentation propa-

gates low-level features which are located in the lower levels of a NN better through the net (as described by Zeiler and Fergus, 2013). These low level features (e.g. edges) are good transmitter of localization information. In comparison to FPN who consist of only one additional top-bottom path this approach significantly reduces the information path between lower layers and topmost features (factor of 10). An other part of the neck is the SPP layer which increases the receptive field and therefore is beneficial for us.
The **head** predicts classes and bounding boxes at 3 different scales. Therefore the head consists of three different branches (also the implementation in Figure 3.17). For making predictions YOLO uses a grid of NxN cells. According to Redmon, Divvala, et al., 2015 each individual cell predicts an objectness score $\hat{p}_i(c)$ which reflects the models' confidence that the box contains an object. Aside from that the model adds also a vector of class probabilities $\hat{C}_i$ and the bounding box coordinates which are calculated as described by Redmon and Farhadi, 2016:

$$\begin{aligned} b_x &= c_0 \cdot \sigma(t_x) + c_x, c_0 > 1.0 \\ b_y &= c_1 \cdot \sigma(t_y) + c_y, c_1 > 1.0 \\ b_w &= p_w \cdot e^{t_w} \\ b_h &= p_h \cdot e^{t_h} \end{aligned} \tag{2.20}$$

The coordinates t_x, t_y, t_w, t_h are predicted by the net. c_x and c_y represent the top left offset in regards to the image. The anchor box has dimension p_w and p_h. These are all outputs and are stacked together as a vector with the dimensions:

$$\begin{aligned} & (N * N * (\quad \#scales \quad * (\#bboxcoords \quad + objectnessscore \quad + \#classes))) \\ = \; & (N * N * (\quad 3 \quad * (4 \quad + 1 \quad + \#classes))) \end{aligned} \tag{2.21}$$

Regarding the calculated loss YOLO does the following:

Bounding box losses were calculated with the MSE as in eq. 2.18 in the original paper by Redmon, Divvala, et al., 2015. But as I use the newer YOLOv4 approach the CIoU loss (see eq. 2.16) instead of MSE. Remarks to the constants c_0 and c_1 in eq. 2.20: Bochkovskiy et al., 2020 added these values for mitigating the problem of grid sensitivity.

Now i can present the first high-level overview over the loss function:

- the above mentioned loss for the bounding boxes L_{loc} as Complete Intersection over Union (CIoU) loss (see eq. 2.16)
- loss for the the classification scores L_{cls} as BCE (see eq. 2.17)
- and the loss for the objectness scores L_{obj} as BCE:

And finally the more precise loss L_{YOLO} as follows:

$$
\begin{aligned}
& \lambda_1 \cdot L_{loc} && + \lambda_2 \cdot L_{cls} + \lambda_3 \cdot L_{obj} \\
= \; & \lambda_1 \cdot L_{CIoU} \\
& && - \lambda_2 \cdot (\sum_{i=0}^{NxN} \sum_{j=0}^{B} \chi_{ij}^{obj} [\hat{C}_i \cdot \log(C_i) + (1 - \hat{C}_i) \cdot \log(1 - C_i)] \\
& && - \lambda_{noobj} \cdot \sum_{i=0}^{NxN} \sum_{j=0}^{B} \chi_{ij}^{noobj} [\hat{C}_i \cdot \log(C_i) + (1 - \hat{C}_i) \cdot \log(1 - C_i)]) \\
& && + \lambda_3 \cdot (\sum_{i=0}^{NxN} \chi_i^{obj} \sum_{c \in classes} [\hat{p}_i(c) \cdot \log(p_i(c)) + \\
& && \qquad (1 - \hat{p}_i(c)) \cdot \log(1 - p_i(c))])
\end{aligned}
\tag{2.22}
$$

Variables with an hat (f.e. $\hat{p}_i(c)$ specify predicted values. $p_i(c)$ defines the class probabilities of a cell i regarding class c. C_i defines the objectness score of a cell i. B denotes the number of bounding boxes in each cell. $\chi(x)$ is the indicator function. χ_i^{obj} denotes if object appears in cell i and χ_{ij}^{obj} denotes that the j-th bounding box predictor in cell i is responsible for this prediction. An predictor is responsible for a prediction if he has the highest IOU among all predictors.

2.3.4.2. YOLACT

You Only Look At CoefficienTs (YOLACT) by Bolya et al., 2019 is an extension to the above mentioned and explained YOLO approach as it achieves to generate additional instance segmentation masks alongside the conventional output (see Section 2.3.4.1) by:

1. prototype masks generation
2. and subsequently predicting per protomask coefficients
3. finally linearly combine prototype maks by means of their coefficients

Note that the prototype masks span the whole image. Therefore an additional cropping step with the corresponding bounding box is needed. This approach don't need an localization step for they are rather learned implicitly. Out of the prototype masks one can get the resulting segmentation mask as stated by Bolya et al., 2019 with following equation:

$$M = \sigma(PC^{\mathrm{T}}) \tag{2.23}$$

Regarding the prediction of suitable coefficients in the second step it is important to mention that every coefficient must go through a tanh(x)-activation (see Figure 2.4) for allowing substractions.

YOLACT can easily be integrated to an one-stage object detector as e.g. YOLO by hook it into a backbone feature layer. In Figure 3.17 one can see how the protonet is integrated into

the whole instance segmentation system whereas Figure 3.18 gives an high level overview of the individual protonet (see Figure 2.8) itself.

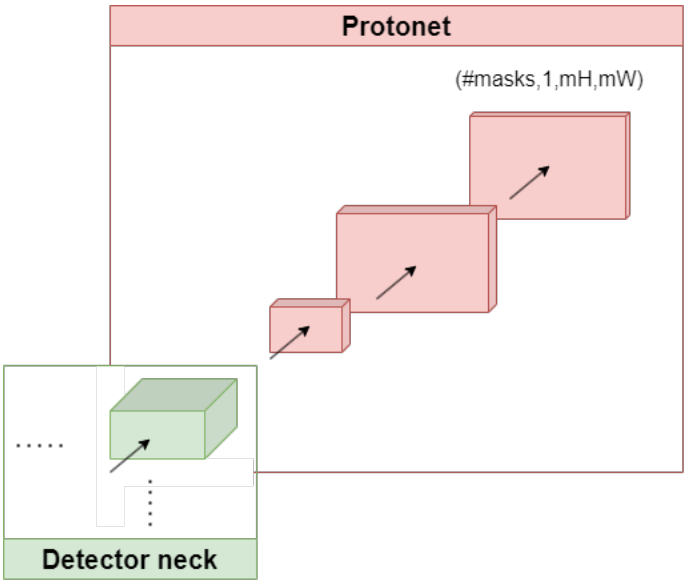


Figure 2.8.: Protonet(red) high level overview and feature backbone layer(green) from the neck

The protonet puts out a defined number of masks(#masks) with height mH and width mW.

The previously defined head furthermore is extended by an extra mask branch who is responsible to predict #masks coefficients per anchor box on the grid NxN.

Additionally to the already defined losses of the YOLO model from eq. 2.22 this model needs an additional mask loss L_{Mask}. Bolya et al., 2019 propose a pixel-wise BCE(see eq.2.17) loss between the calculated mask M of eq.2.23 and their ground truth M_{gt} which is adopted by the follow up work:

$$BCE(M, M_{gt}). \tag{2.24}$$

2.3.5. Cross-industry standard process for data mining (CRISP-DM)

As one has to process lots of user data it is useful to take advantage of a structured procedure in doing so. Shearer C., 2000 introduced CRISP-DM a long time ago. But it is still described nowadays by Mariscal et al., 2010 as one of the most used models in Data Mining. It defines precise procedures one have to follow in a data mining process. Figure 2.10 explains each step in detail. The CRISP-DM additionally is a dynamic model with the possibility to go back an forth. The transitions between the steps have the following meaning:

1. Business and data understanding are tightly bound together. Sometimes one has first to gain insights into the data before making business decisions.
2. Feature Engineering and Modeling are strongly connected to each other. So depends the model quality much on the quality of selected features.
3. One can always go back if in the process the business understanding is changing

Figure 2.9.: Transitions between steps in CRISP-DM

The Feature Engineering step is very crucial for the success of a project. Hence it is worth going into more detail about this phase. One will receive data from many inputs (eye, head, hand, object detection results). Bringing this data together and finding a common representation that can be fed into a machine learning method is very important and is the result of this phase.

For making a link to my project and to further explain why this method is so important I will shortly describe the implementation ideas of the above mentioned steps for the follow-up Section 3. In the very beginning the business understanding makes a connection to use cases and the previously defined RQ (see Section 1, e.g. defining user intentions). This helps us to expand all further steps. Then will be portrayed which kind of data is collected (e.g. head/hand/eye data from the HoloLens2, instance segmentation data) alongside its quality, dimension, etc.. These findings enable me to clean and transform all data samples to an usable form (e.g. throw away entries with NaN, make matrix multiplications, etc.). This data will be input to one of the ML models (e.g. MLP, YOLO from Section 2.3). They are described next with all information in detail e.g. the corresponding activation functions (see Figure 2.4) and metrics (see Section 2.3.2). The models will be trained accordingly and finally will be evaluated for proving their correctness. As a very last step the models are deployed on different devices (e.g. server,HoloLens2) with suitable formats (e.g. onnx).

Business Understanding

This is the initiation step. In the very beginning one have to define his goals and success criteria for the project alongside the frame conditions the project have to met. Furthermore one defines his data sources.

⇓⇑ (1.)

Data Understanding

In this step one wants to first collect and examine the data. A detailed description of the data with its quality must be an artifact of this phase.

⇓

Data Preparation (Feature Engineering)

This phase takes care of preparing, processing, editing, purifying the collected data appropriately before it is fed into a machine learning process.

⇑(3.)

⇓⇑ (2.)

Modeling

This step identifies appropriate models for the previously defined machine learning task (e.g. SVM, ADABoost, NN, etc.). After selecting a model one has also to define the set of hyperparameters for the selected methods.

⇓

Evaluation

This step is important to evaluate how good the selected model meets the business goals which were selected in the very beginning.

⇓

Deployment

The last step is the deployment of the new model. This involves the set up and actual execution of the model on the actual system.

Figure 2.10.: CRISP-DM overview

3. User guidance system

This Section attends to explain the implemented user guidance system. I draw connections whenever necessary to the detailed related work Section 2 with the conducted SLR. I make use of all found relevant literature to build the user guidance system. For motivating the following-up Sections I will firstly explain the general structure of the application:

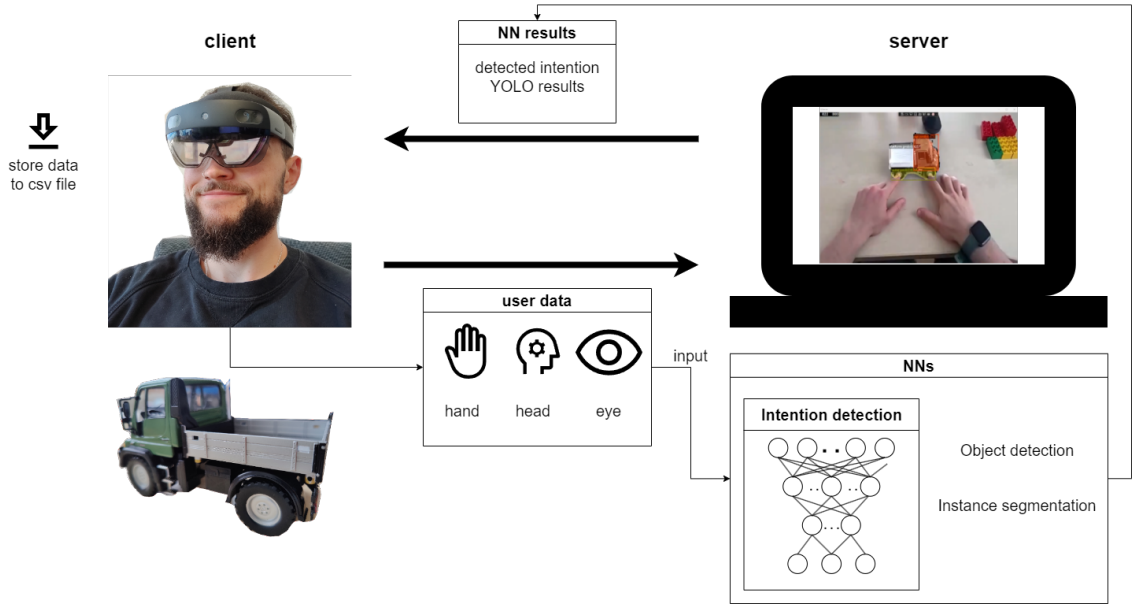


Figure 3.1.: User guidance application structure

In a higher-order view on the application structure (see Figure 3.1) one can see the underlying client-server structure. Numerous literature advocates to transfer computational intensive parts (for my application it's mainly the NNs) of the application (see the work by Bahri et al., 2019,Guo et al., 2021) to an server and rather transfer the results back to the HoloLens2. I expect to be able to run larger models with less inference time. An additional view into the tech specs[1] reinforced this design choice. With the help of this structure one is furthermore able to store all relevant data (user data + NN data) together in one CSV file. This data collection capability allows to train the user intention ML model (a MLP).

3.1. Tools and Technologies

The following Section discusses technologies and tools in use. It will also give well-founded reasons for how and why these technologies are used.

3.1.1. Microsoft HoloLens 2

Ungureanu et al., 2020 introduced the HoloLens2 as a research tool in MR with rich capabilities for user tracking. The following Figure 3.2 presents these capabilities used by our application:

[1]Microsoft. (2023). Microsoft hololens2. https://www.microsoft.com/de-de/hololens.

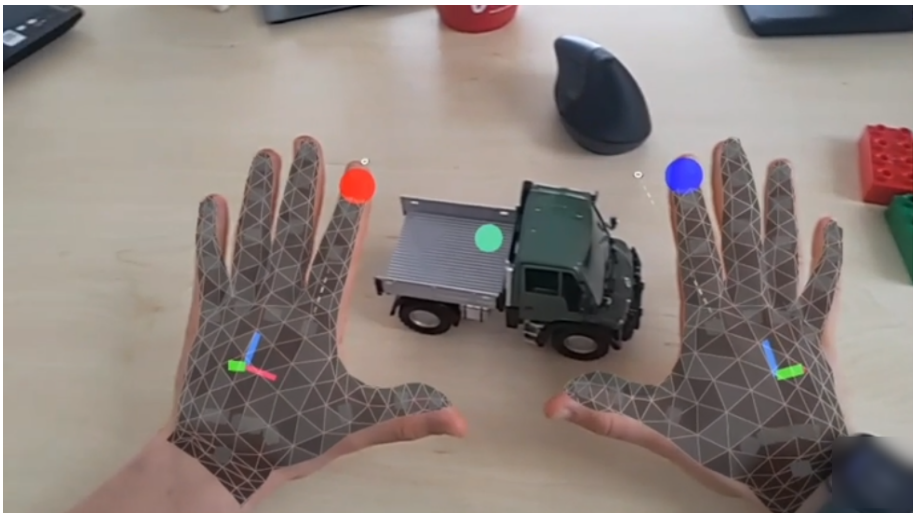


Figure 3.2.: Tracking capabilites of Hololens2: left index tip (red), right index tip (blue), eye hit position (green)

The green dot in Figure 3.2 visualizes the captured eye hit world position. The recorded index tip positions are visualized with red and blue accordingly. This data serves in addition to information about detected objects as input to the guidance system.

3.1.2. Mixed Reality Toolkit

The decision to use the HoloLens2 brought me also to utilize the Mixed Reality Toolkit (MRTK)[2]. The MRTK comes with a multiplicity of different building blocks for MR applications making the development of user guidance application easier. Figure 3.3 shows the application UI with building blocks from the MRTK:

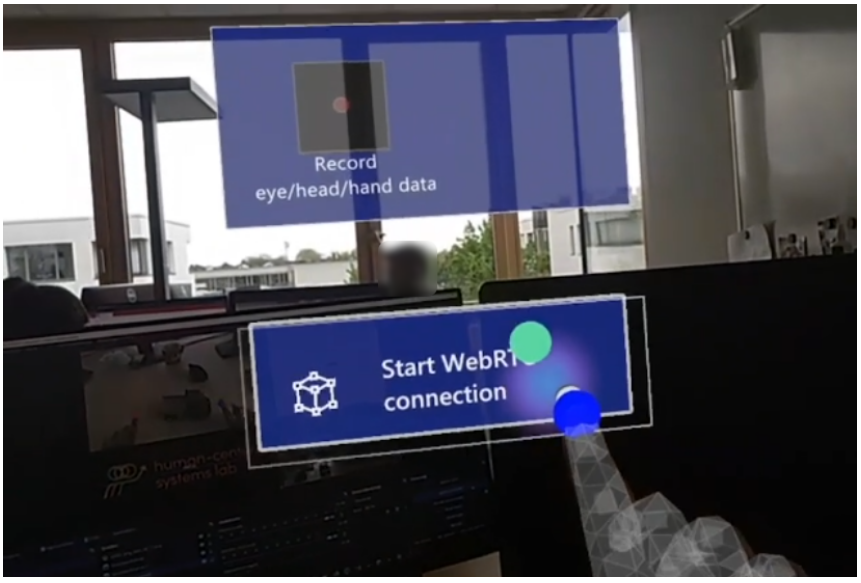


Figure 3.3.: UI design for the system

[2]Microsoft. (2022b). Mixedrealitytoolkit for unity. https://github.com/microsoft/MixedRealityToolkit-Unity.

The UI in Figure 3.3 enables the user to record all user data (eye/hand/head) alongside the detection results that are transmitted from the server to the HoloLens. For initializing the connection from the server to the client the user has to push the button named "Start WebRTC connection" which is further explained in the next Section.

3.1.3. Networking

In the beginning of this chapter I have illustrated the client/server architecture for the system. For establishing the connection between client and server a series of techniques are used. WebRTC[3] is standard for real-time communication. I use the Mixed Reality WebRTC[4] implementation for transmitting video and audio between client and server in real-time. Dead simple signalling for WebRTC[5] serves as message broker between server and client.

3.1.4. Server

The previously mentioned design decisions suggest the use of an Universal Windows Platform (UWP) application on server side. This again has as a consequence that NNs are exported to onnx for deployment.

3.1.5. Unity

The MRTK is only usable in a dedicated engine. Therefore I decided to use the Unity Engine[6] for implementing the application on HoloLens side.

3.2. Business Understanding

The following Section will implement the first step from the earlier defined CRISP-DM(see Section 2.3.5) model. As stated in the RQ from Section 1: the goal here is to first detect user intentions. For this purpose I conducted an exploratory study for investigating the three most important user intentions while a user interacts with a truck. One can find more detailed information about the truck in the following Section 3.3.1. The study was laid-out as a think aloud session. Each individual user was told to report me his thoughts and ideas while investigating the truck. Overall one could discover great user engagement (head/eye/hand movement) which supports my first formulated RQ. These first insights provide a good basis for the next step: collecting data and examine their meaningfulness.

[3]Google. (2023b). Webrtc. https://webrtc.org/?hl=de.
[4]Microsoft. (2022a). Mixedreality-webrtc. https://github.com/microsoft/MixedReality-WebRTC.
[5]bengreenier. (2022). Dead simple signalling for webrtc. https://github.com/bengreenier/node-dss.
[6]Unity Technologies. (2022). Unity cross-platform game engine. https://unity.com/de.

(a) User exploring the frame of the truck

(b) User exploring the cap interior of the truck

(c) User exploring the platform of the truck

Figure 3.4.: All three user intentions I want to detect

The list of the most frequently observed user intentions is as follows:

- the bottom part of the truck with its frame and wheels is of interest as in Figure 3.4a
- the cap with its interior as in Figure 3.4b
- the movable platform with its capability of loading things up as in Figure 3.4c

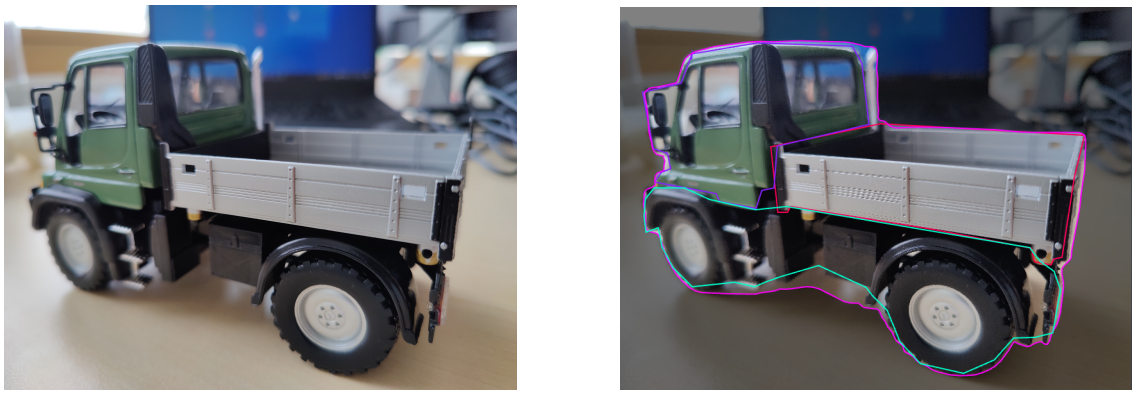
3.3. User intention detection

This chapter tackles the first part of the RQ which is about detecting the previously defined user intentions from Section 3.2.

3.3.1. Data Understanding

Before user intentions can be detected one has to provide clarity of the used data creation/collection procedure first.

As stated by the RQ and supported by the relations between truck parts and user intentions I collect data for training a corresponding object detection/instance segmentation model. As the system should work well on real trucks it is important to have a real truck model:



(a) truck

(b) Annotations: frame (green), platform (red), cap (purple)

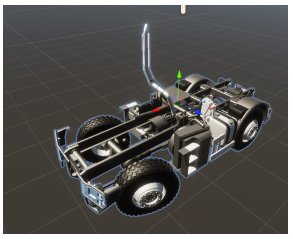
Figure 3.5.: Object detection/ instance segmentation data set creation

The real truck model can be seen in Figure 3.5a with its corresponding labeling in Figure 3.5b. The previous Section discovered three different major intentions. Each intention has a different truck part in its focus. The labeling is chosen accordingly.
But not only data from a real model is considered. Figure 3.6 shows the truck model imported as .fbx file in Unity with the different labeling classes. The underlying idea behind this is to enrich the data set which consists of solely real world images by synthetic data. There is a huge popularity in using synthetic data as described by Nikolenko, 2019 for augmenting a data set. As I have to build an data set with lots of samples it is very interesting to investigate the usage of synthetic data. This can alleviate the cumbersome work of manually creating the data set. With the provided 3D-Model and the Unity Perceptron Package[7] it is interesting to investigate the benefits of this technique. The created data set can be accessed for free[8].

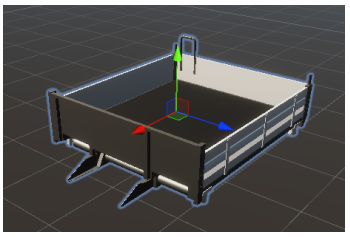
[7] Unity Technologies. (2020). Unity Perception package.
[8] TruckDataSet. (2023). Truck detection dataset [visited on 2023-02-08]. https://universe.roboflow.com/truckdataset/truck-detection-plrxm.

(a) truck

(b) 1.label: cap



(c) 2.label: frame



(d) 3.label: platform

Figure 3.6.: Synthetic data set with corresponding labels

Furthermore I adopt the following rules for creating an instance segmentation/ object detection data set as described by Yolov5, 2022. The first set of rules concern the pure number of samples:

1. rule: $\geq$ 1500 images per class recommended
2. rule: $\geq$ 10000 instances per class recommended

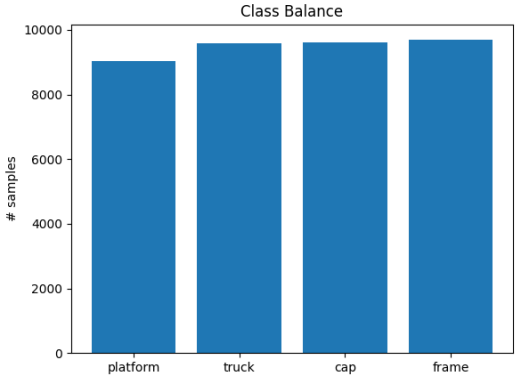


Figure 3.7.: Inter class balance of truck data set

Figure 3.7 visualizes the #classes in relation to their class membership. I do not only have enough class samples but also a sufficient inter class balance. This balance ensures the NN is not biased based on an imbalance.

The next set of rules is as follows:

3. The image variety must be representative in relation to the deployed environment. Therefore I took images from different times of day, different weather, lightning and angles.
4. Label accuracy. As I label the truck the corresponding polygon must enclose the object tightly.
5. Background images are images with no objects that are added to an existing data set to reduce FP. The truck data set has 2% background images (similar to MSCOCO)

All previous rules are related to the taking of images. After took a sufficient amount of images were taken I was able to proceed with the annotation step. The annotation step was conducted with the Roboflow[9] tool. After annotating the images they went through the following pipeline:

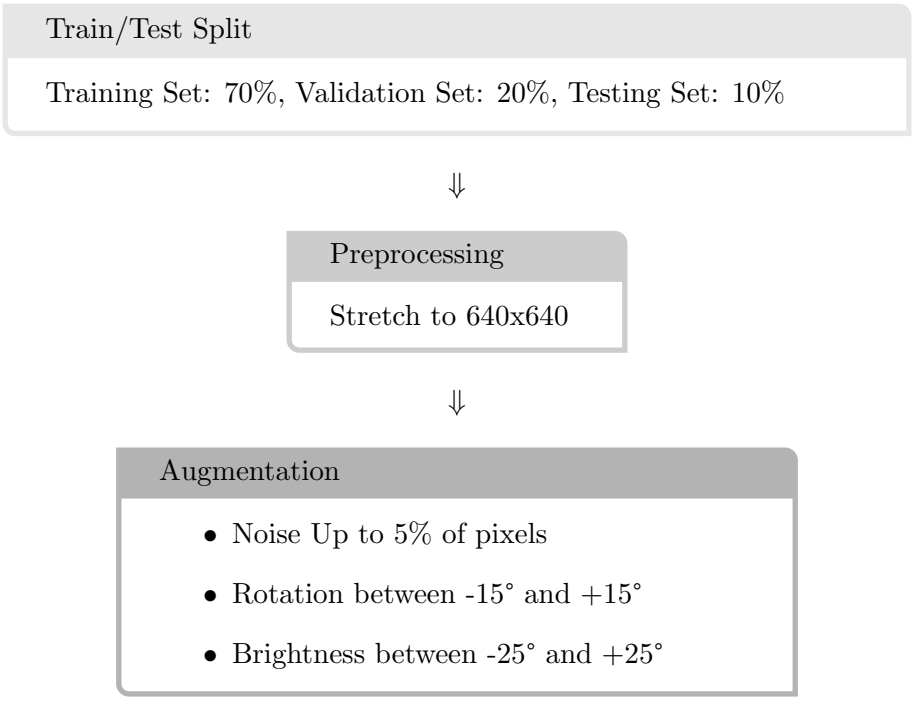


Figure 3.8.: Truck data set creation steps

Figure 3.8 about the data set creation procedure gives interesting insights. The first step guarantees that the YOLO model trains on one set, consecutively validates its training on another set and in the end uses a test set which the net has never seen during training for testing the final quality of the model. The pre-processing step is needed for the YOLO model only allows a single input size. In the very end some noise, rotation and brightness augmentation as described by Bochkovskiy et al., 2020 is added for enriching the data set.

[9]Roboflow. (2022). Roboflow [Create your own object detection/ instance segmentation data set]. https://docs.roboflow.com/quick-start.

For completing the data understanding a detailed list of all generated data from the YOLO net is further processed. The exact output vector was already developed in eq. 2.21. As this data will become input to the user guidance NN the exact value ranges and invalid values become very important. In Table 3.1 all data is listed alongside the type, value ranges and an indicator for falseness if there is one. If no entry is given in the value cell the corresponding data sample can take every possible value of its underlying type.

Name	Type	Values	Invalid value
class label	string	cap,frame,platform,truck	empty string
bounding box coordinates x_1, x_2, y_1,y_2	4 floats	$\{x_1, x_2, y_1, y_2 \in \mathbb{R}_0^+ \mid x_1, x_2 \leqslant screenWidth \wedge y_1, y_2 \leqslant screenHeight \wedge x_1 < x_2 \wedge y_1 < y_2\}$	-
probability	float	[0,1]	-
objectness	float	[0,1]	-
segmentation mask	byte[maskWidth· maskHeight]	-	null

Table 3.1.: Data output from my instance segmentation NN

After I gained knowledge about collected truck data I spent attention to the user data that will be collected. This user data was collected after the object detection/instance segmentation NN was trained and deployed on the server. This circumstance namely puts me in the position to concurrently save the object detection/instance segmentation results and the user data every $\sim 33ms$ into one big CSV file. The following table 3.2 shows a detailed list of every value that is stored within each sample. As described in the overall application overview in Figure 3.1 the values are stored in a CSV file (user data and segmentation data) whose name consists of user name, data and current detection. This explicit naming helps in ordering the CSV files later. The functionality to store these values in a CSV file is needed to collect training data for each individual intention. The explicit value listing is very necessary as the data has to be transformed in the next step. Without knowing how the data is arranged beforehand this transformation step is not feasible. One important insight I have gained is that the positions are given with reference to world space. Because of this I decided to also track the "worldtoCamera" and "projection" matrices attached to the HoloLens2 object in Unity. Both matrices applied in order will transform a world position to screen space as follows:

$$x_{screenSpace} = projectionMatrix \cdot worldToCameraMatrix \cdot x_{worldPosition} \tag{3.1}$$

A Vector3f is stored within three columns of the CSV file if all dimensions have valid data ($\neq NaN$). A Matrix4x4 is stored a list of 16 values separated with spaces. As these values are fed to a NN it is also important to know what values are already normalized. If values are not normalized an extra step is added later in the model.

If any listed data point has an invalid value the whole sample is omitted. This makes sense as any invalid value hints that some tracking capability is not ready yet. The other way which means to replace the invalid value is not viable for no reasonable auxiliary values can be given.

Name	Type	Values	Invalid value	Description
worldToCameraMatrix	Matrix4x4	-	empty string	-
projectionMatrix	Matrix4x4	-	empty string	-
Head origin	Vector3f	-	-	in world space
Head direction forward	Vector3f	normalized	-	in world space
Head direction to the right	Vector3f	normalized	-	in world space
Head movement direction	Vector3f	-	one of the entries is NaN	entry NaN indicates that eye tracking is currently not enabled or invalid
Head velocity	Vector3f	-	one of the entries is NaN	entry NaN indicates that eye tracking is currently not enabled or invalid
Eye origin	Vector3f	-	one of the entries is NaN	in world space, entry NaN indicates that either eye tracking is not valid or not enabled or there is no gaze target
Eye direction	Vector3f	-	one of the entries NaN	in world space, entry NaN indicates that either eye tracking is not valid or not enabled or there is no gaze target
Distance to target	float	-	one of the entries NaN	in world space, entry NaN indicates that either eye tracking is not valid or not enabled or there is no gaze target
Eye hit position	Vector3f	-	one of the entries NaN	in world space, entry NaN indicates that either eye tracking is not valid or not enabled or there is no gaze target
Eye hit normal	Vector3f	-	one of the entries NaN	entry NaN indicates that either eye tracking is not valid or not enabled or there is no gaze target
(Right/Left) Index Tip Position	Vector3f	-	(0,0,0)	in world space
(Right/Left) Index Tip Up	Vector3f	-	(0,1,0)	-
(Right/Left) Index Tip Right	Vector3f	-	(1,0,0)	-
(Right/Left) Index Tip Rotation	Quaternion	-	(0,0,0,1)	-

Table 3.2.: Overview of user data that is generated at fixed sampling rate

After gaining complete knowledge of the collected data with its proper shapes and values one can further proceed and examine the expressiveness of the data. In the very beginning (see Section 3.2) I defined the intentions which should be detected. Therefore the following explanations map the intentions to the collected data.

3.3.2. Intentions and user data

For the following examinations a collection of 160 samples ($\approx 30\frac{samoples}{s} \rightarrow 160$ cover a time window of $\approx 5s$) was chosen randomly.

For the first defined user intention (movement of the truck with eye focus on the bottom parts) as shown exemplary in Figure 3.4a) one can discover the following structure in the recorded user data:

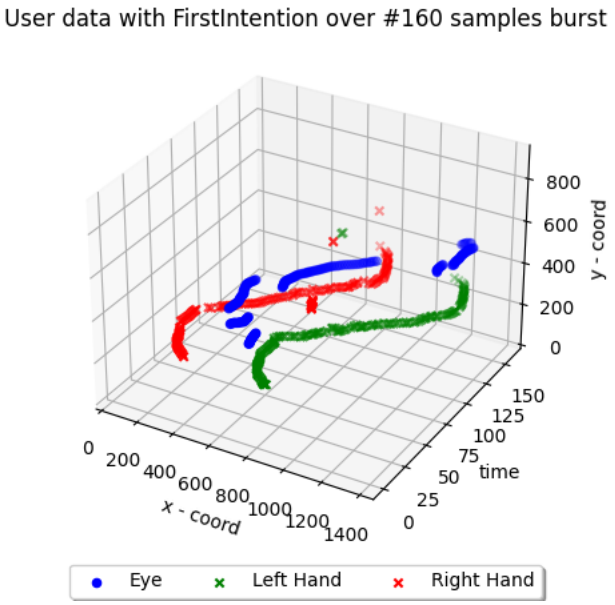


Figure 3.9.: Accumulated data points of first intention

Figure 3.9 reveals a sinusoidal structure in eye and hand data. This is not surprising since the user drags the truck from one side to the other in a constant speed. Some further remark on all collected data (all intentions are included): eye data shows jitter. When you follow the blue line which represents eye sample points in space over time you can detect cracks. Whereas the hand data has a very good interpolation. Both hand data lines, the red and the blue reveal very clearly a sinusoidal structure.

Figure 3.10 exhibits the connection from user data to the collected object data. With this Figure about the first intention the segmentation mask of the bottom part of the truck in conjunction with the user data is visualized. The Figure 3.10 shows what I observed previously: the user has the frame of the truck in focus while holding it on the left and right side.

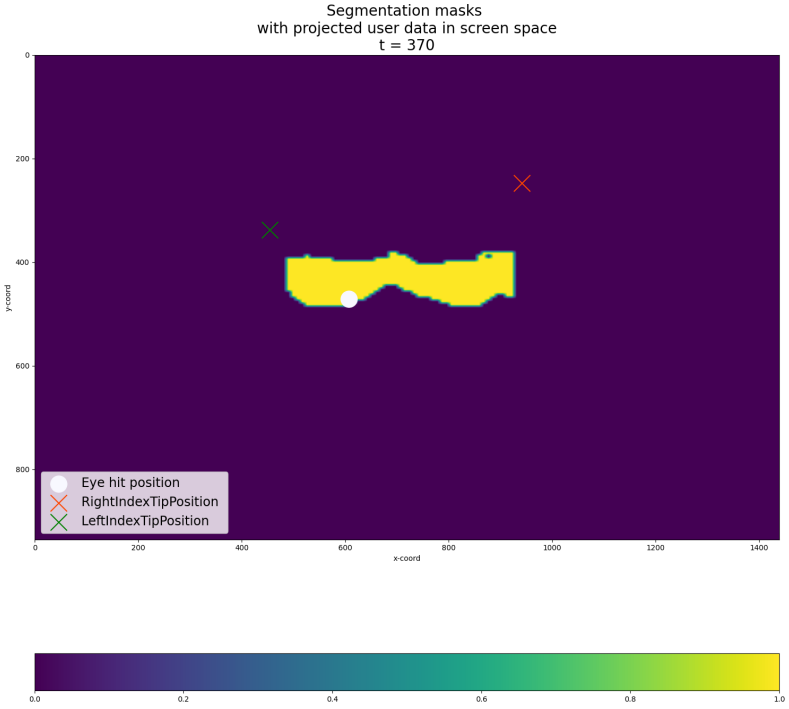


Figure 3.10.: Segmentation masks and user data of first intention

For the accumulated data points of the second intention as seen in Figure 3.11 I have discovered a different pattern. As stated in the business understanding Section 3.2 the second intention is about discovering the cap with its interior. The movement patterns of eye and hand data over time are significantly smoother. The user takes less abrupt movements in general.

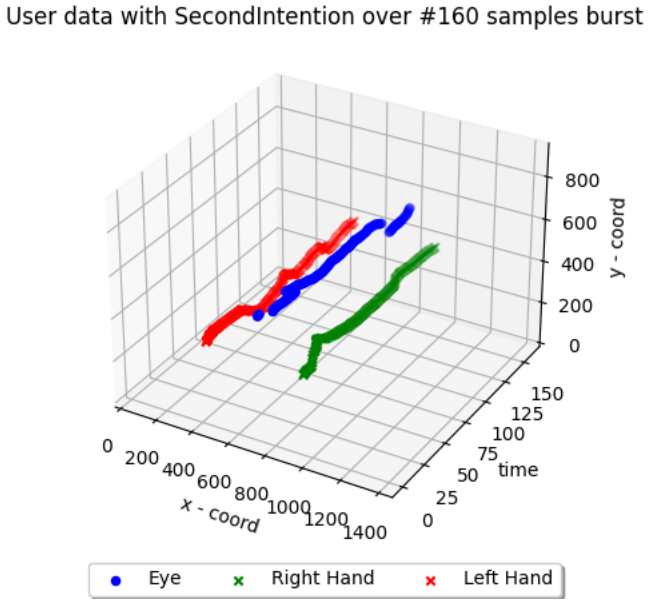


Figure 3.11.: Accumulated data points of second intentions

Now I draw once again the connections to the segmented objects. For the second intention I highlighted in Figure 3.12 the cap and user data accordingly. Regarding the user data one can discover that the hand data points are closer together than for the first intention. The eye focuses the object related to the intention.

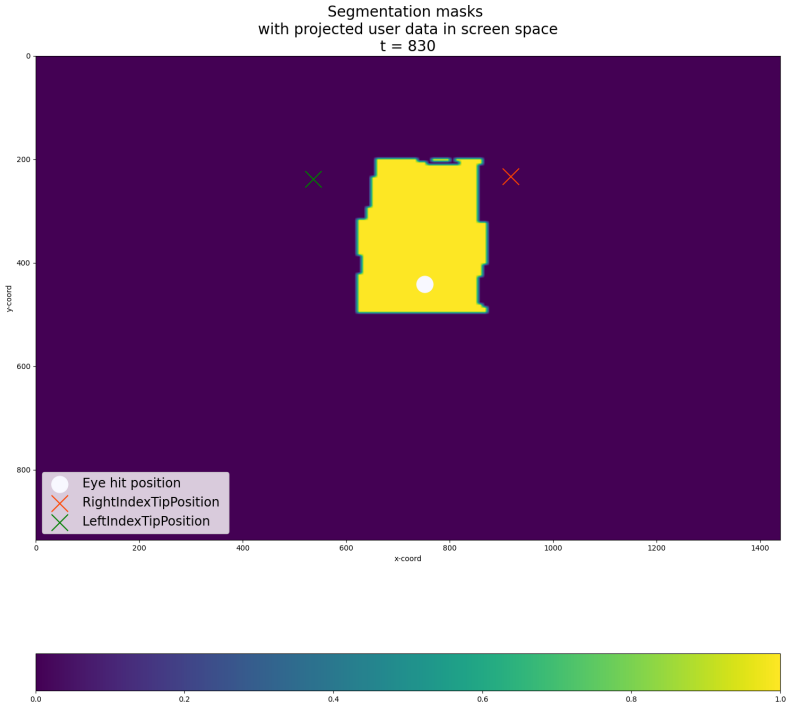


Figure 3.12.: Segmentation masks and user data of second intention

The accumulated data points of the third intention as seen in Figure 3.13 have a different look than both precursors. As the user loads up things upon the truck and moves the platform much movement into the y-axis can bee seen.

Last but not least does the collected data from the third intention reveal other patterns than both previous. Figure 3.13 shows movement in the vertical direction. This can be explained as follows: the user zooms in on the platform of the truck. The platform is adjustable for height and can be loaded. These movements own patterns with amplitude along the y-axis. I leave out the connection from object to user data for the third intention for it doesn't show significant new insights.

Let's draw a conclusion from these insights. The previously defined three different intentions exhibit good discriminative features. Their temporal structure shows different behaviour. Hence it is worth to proceed in classifying user intentions by user data using TSC (as defined in Section 2.2).

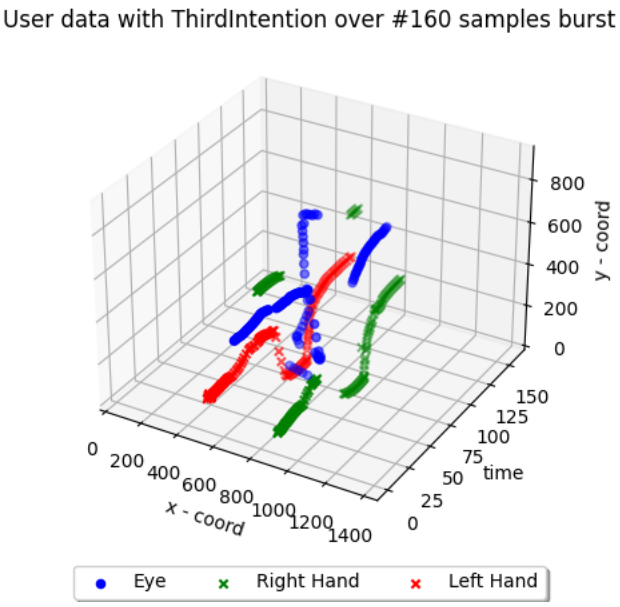


Figure 3.13.: Accumulated data points of third intention

But how does the user head movement contribute to its intention? To get a grasp on that I investigated the head movement from one user over all intentions:

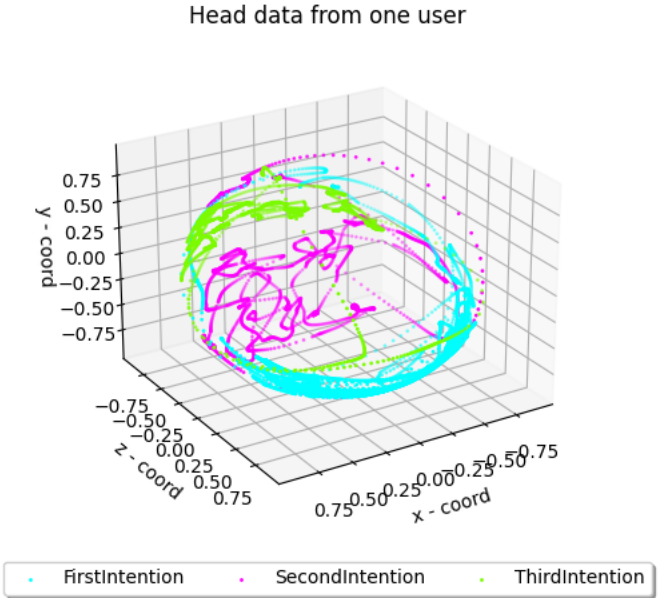


Figure 3.14.: Visualized head movement from one user

Figure 3.14 shows that each intention has its own head movement pattern and thus reinforces my preliminary made decision to use head data as input to classify user intentions.

3.3.3. Data preparation/feature engineering

All samples with at least one data point containing an invalid entry are omitted. The mapping of data point to its invalid value is shown in Table 3.1 and 3.2. There are f.e. very much invalid data points in the very beginning of the application because the tracking capabilities are not initialized completely. Furthermore I preprocess all data points representing a position in world space by transforming them into screen space (as depicted in eq. 3.1). The reason for this is as follows: the positions are fed into a NN which must not learn world positions. As the class labels also become input of a NN they are transformed to an ordinal scale.

3.3.4. Modeling

This Section gives information about the used ML models with their set of hyperparamters and metrics. For training two Graphics Processing Unit (GPU) were used to accelerate training:

- NVIDIA GeForce RTX 3080 Ti, 12054MiB
- NVIDIA GeForce RTX 2080 Ti, 11019MiB

in conjunction with 64GB of DRAM.

3.3.4.1. User Intention Prediction Module

The task is to detect and classify multiple intentions. The previously defined metrics in the previous work Section 2.3.2 however only consider the binary case. Therefore I have to extend these definitions to a multi class scenario. Hence I am introducing the OvR approach:

Definition 3.3.1 (One-vs-Rest (OvR))**.** With the concept of an OvR scenario one can transform a multi class into a binary class problem and therefore being able to use metrics normally only suited for binary scenarios. This is obtained in the following way: For each class $c_j, j \in \{0, \#classes - 1\}$ build a new class that consist of all samples unequal to class c_j and treat it as individual class.

Moreover I introduce the two concepts of micro-/macro averaging for multi class scenarios. As the created user data set is not perfectly balanced regarding its class distribution it is important to evaluate the micro average in addition to the macro average. The macro average represents the arithmetic mean whereas the micro average attaches importance to class imbalances. Equation 3.2 calculates the $TPR_{micro-averaged}$ in contrast to equation 3.3 the $TPR_{macro-averaged}$

$$\frac{\sum_c TP_c}{\sum_c (TP_c + FN_c)} \tag{3.2}$$

$$\frac{\sum_c TPR_c}{\#classes} \tag{3.3}$$

In Section 2.2 the MLP were identified as best suitable model for answering the RQ. As input to the NN a time series of user data is given. As defined in 2.2.3 a TSC data set is instantiated as follows:

$$
\begin{aligned}
S = \{ \quad & (\hat{X}_{FirstIntention}, Y_{FirstIntention}), (\hat{X}_{SecondIntention}, Y_{SecondIntention}), \\
& (\hat{X}_{ThirdIntention}, Y_{ThirdIntention})\} \\
\hat{X} = \{ \quad & X_{eyeHitPosX}, X_{eyeHitPosY}, \\
& X_{headMovementX}, X_{headMovementY}, X_{headMovementZ}, \\
& X_{leftIndexTipPosX}, X_{leftIndexTipPosY}, \\
& X_{rightIndexTipPosX}, X_{rightIndexTipPosY}, \\
& X_{classLabels1stDetection}, X_{classLabels2ndDetection}, \\
& X_{classLabels3thDetection}, X_{classLabels4thDetection}\}
\end{aligned}
\tag{3.4}
$$

The collected data is rolled out as in eq. 3.4 before it serves as input to the MLP:

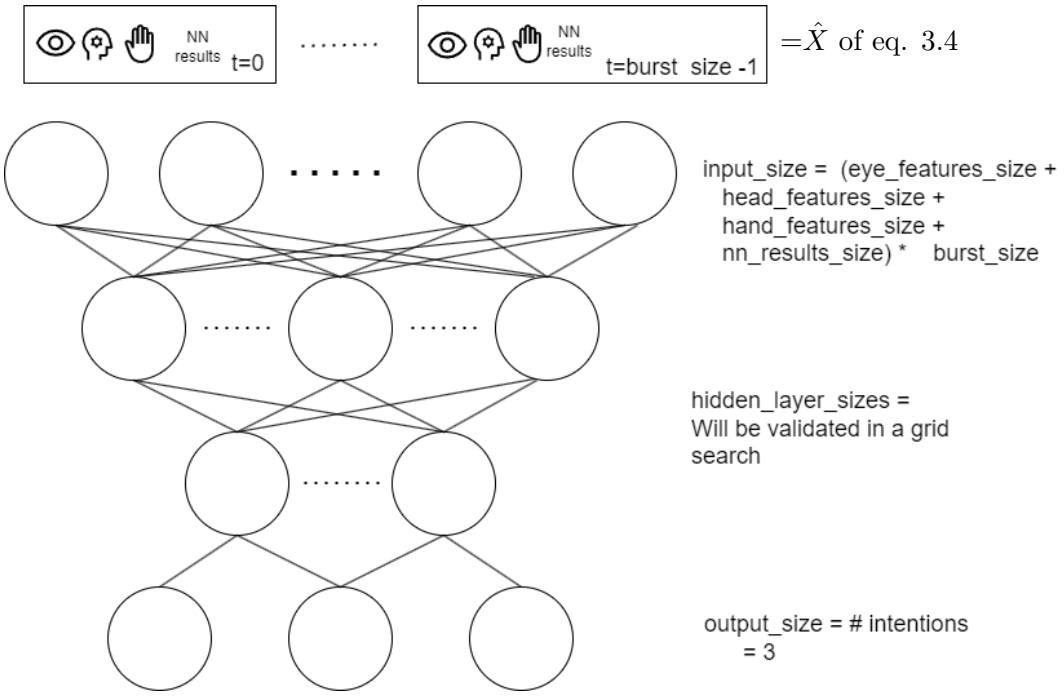


Figure 3.15.: High level overview of implemented MLP

In Figure 3.15 one can see two hyperparameters of the model: the burst size (size of a sliding window) and the hidden layer sizes. As stated by Ismail Fawaz et al., 2019 a softmax (see eq. 3.5) for multiclass classification is used.

$$
\hat{Y}_j(X) = \frac{e^{A_{L-1} * w_j + b_j}}{\sum_{k=1}^{K} e^{A_{L-1} * w_k + b_k}} \tag{3.5}
$$

With the softmax equation 3.5 one can build up the categorical cross entropy as loss of the model:

$$L(X) = -\sum_{j=1}^{K} Y_j \log \hat{Y}_j \tag{3.6}$$

With this preliminaries the following statements are made for the training of the model: As it is recommended by many authors[10] KFold-Cross Validation with K=5 is used. Also cross-validation iterators for grouped data are used which means there is an user id for each sample. As samples are drawn from different users the i.i.d assumption is broken. For mitigating the problem not to depend on individual persons all samples from validation fold come from groups that are not in training folds. For the grid search the following grid is defined (see Table 3.3):

hidden layer sizes	[(23, 14, 8), (64, 23, 14, 8), (14, 8), (128, 64, 23, 14, 8), (70)]
burst sizes	[90, 120, 150, 180, 210, 240, 270]
activation	['relu']
learning rate	[0.001]
alpha	[0.0001, 0.05]

Table 3.3.: Regarding Grid Search: hyperparameters with their possible values

3.3.4.2. Object Detection/Segmentation Module

The YOLO implementation used for the application has a changed bounding box calculation compared to the original formulation (see eq. 2.20).

$$\begin{aligned} b_x &= (2 \cdot \sigma(t_x) - 0.5) + c_x \\ b_y &= (2 \cdot \sigma(t_y) - 0.5) + c_y \\ b_w &= p_w \cdot (2 \cdot \sigma(t_w))^2 \\ b_h &= p_h \cdot (2 \cdot \sigma(t_h))^2 \end{aligned} \tag{3.7}$$

Furthermore it uses a reimplementation of the famous Darknet53BackBone (see Figure 2.7) as described in Figure 3.17. The backbone was pre-trained on the MSCOCO data set by Lin, Maire, et al., 2014. For better understanding I first explain smaller building blocks of the net in Figure 3.16. As described in the previous work section one can clearly see the CSP extension in the beginning of the blocks. The input gets divided in the very beginning and concatenated in the end as in Figure 2.5b. Not only CSP but also residuals serve as further improvement. As in Figure 2.5a the building block C4 uses a skip connection with a corresponding bottleneck.

[10] Scikit learn. (2023). Scikit cross validation.

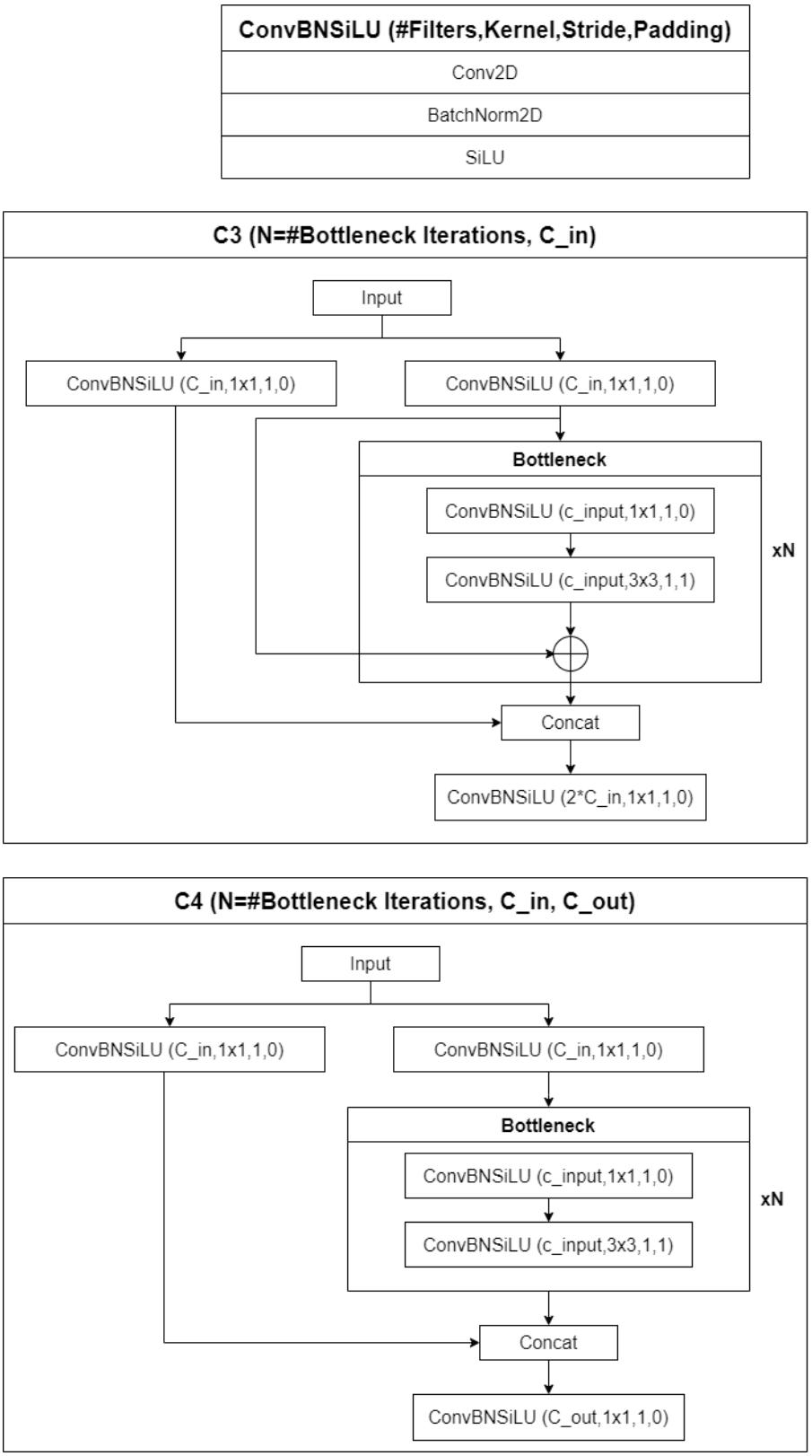


Figure 3.16.: Building blocks for new Darknet53 implementation

Those building blocks are used by the YOLO structure as seen in Figure 3.17. Explanations about the individual modules (SPP,Neck,Backbone,PAN) are given in Section 2.3.4.1.

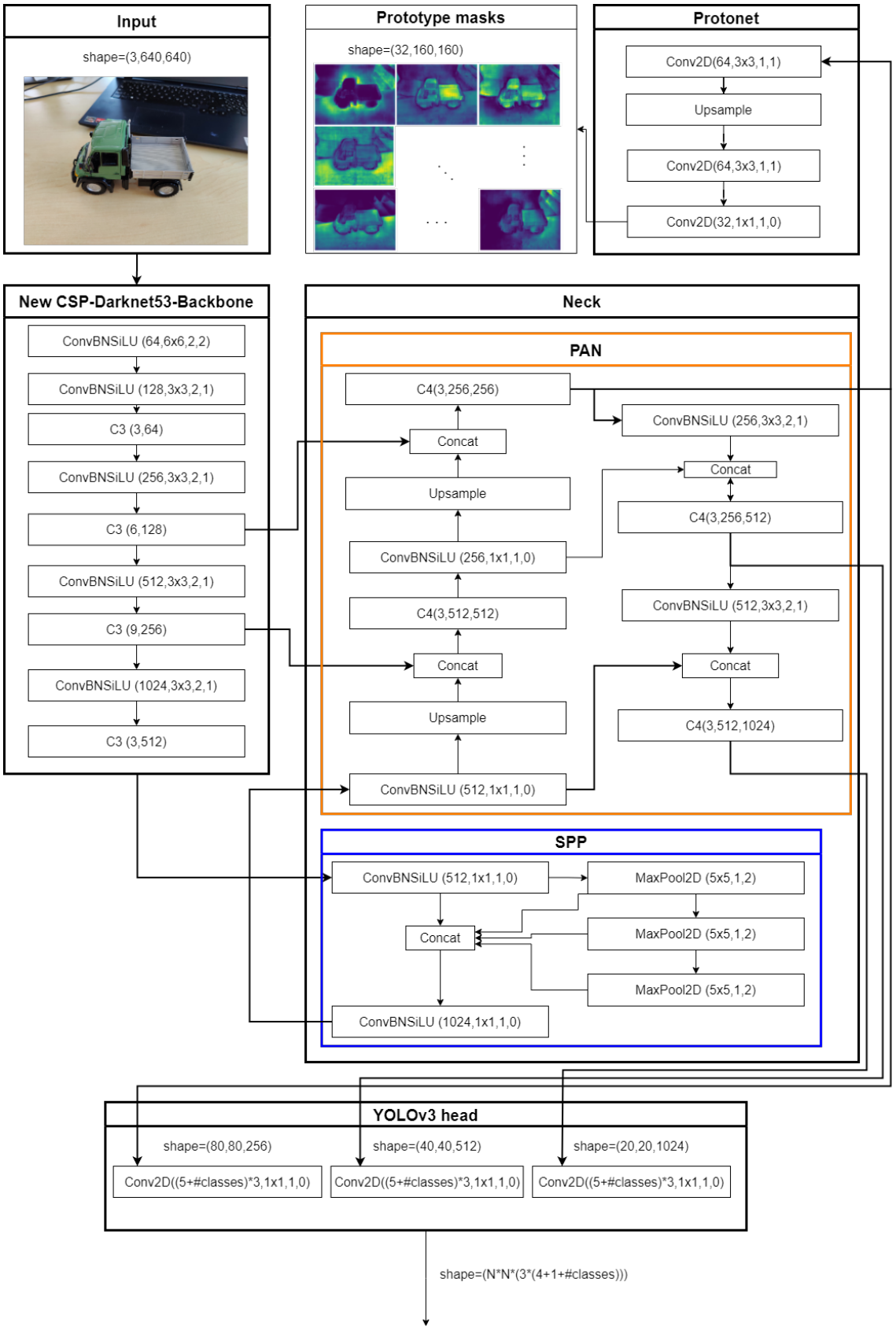


Figure 3.17.: High level overview of used YOLO implementation

Figure 3.17 gives clarity where the protonet (see Figure 2.8) is mounted. The backbone feature layer for the protonet is the one in the highest stage of the top-bottom path of the PAN. The rationale behind this are as follows: as the backbone feature layer is in the very top of the feature pyramid the prototypes will be of higher resolution and therefore of better quality and better in detecting smaller objects (as described by Bolya et al., 2019).

This implementation of PAN uses concatenation like YOLOv4 by Bochkovskiy et al., 2020 instead of addition by original paper by Liu et al., 2018.

The three heads in the end are representing different object scales. Each is weighted differently:

$$L_{obj} = 4.0 * L_{obj}^{small} + 1.0 * L_{obj}^{medium} + 0.4 * L_{obj}^{large} \tag{3.8}$$

As described in Chapter 3.1 I am bound to an UWP application on server side. Therefore I had to reimplement parts of the segmentation mask generation in C# as follows:

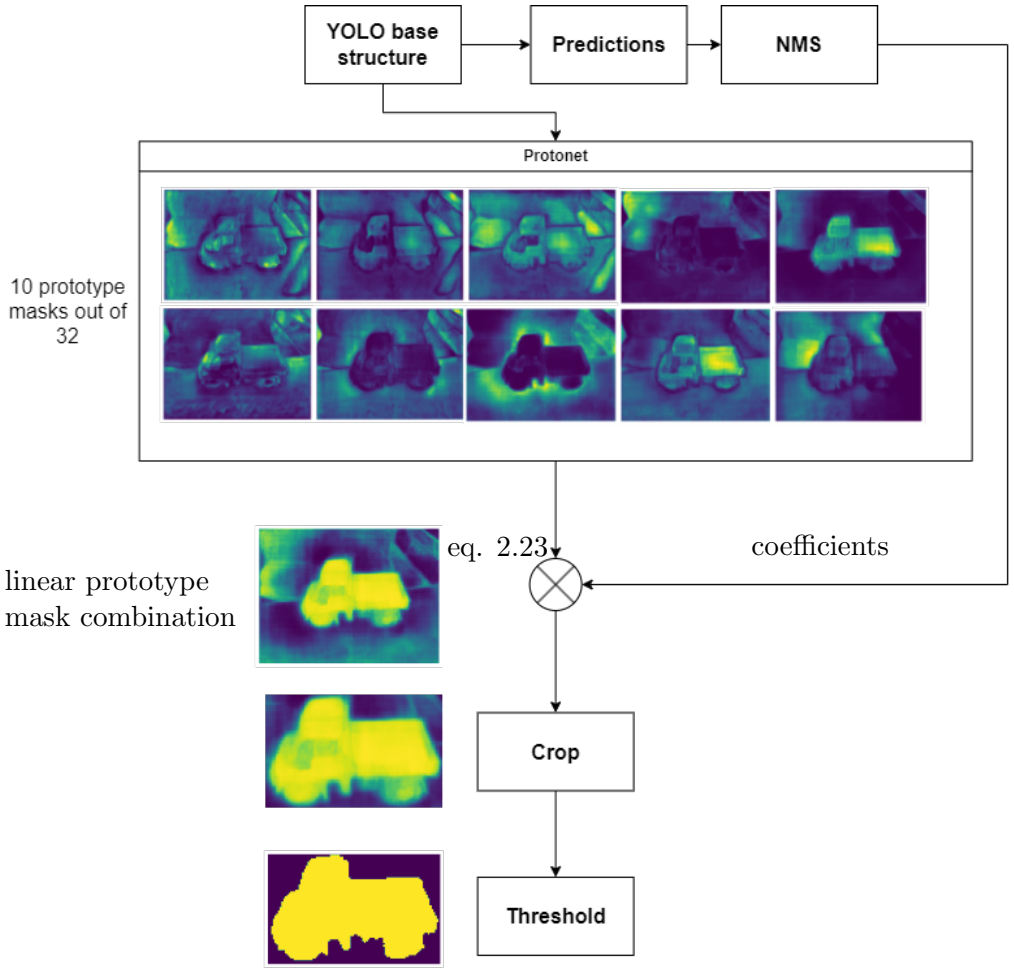


Figure 3.18.: High level overview of reimplemented YOLACT approach

The steps in Figure 3.18 are as described in Section 2.3.4.2. I have marked the place where eq. 2.23 takes place. One can see interesting properties of the prototype masks as

described by Bolya et al., 2019 after they were generated by the protonet: some prototype masks emphasize certain parts of the truck, other masks the surrounding and yet others concentrate on certain partitions of the image.

I use model weights that were pre-trained on the MS-COCO data set. Furthermore a rich set of **hyperparameters** were used during training. These hyperparameters were not validated via exhaustive grid search but with a genetic algorithm[11]. As optimizer I utilize SGD with initial LR=0.01, momentum=0.937 and a weight decay of 0.0005. I started with initial 2000 epochs but enabled early stopping. Due to GPU memory and DRAM restrictions I used a mini-batch size of 16. For the IOU threshold I chose 0.2. Regarding the data augmentation I use the common (as described in Bochkovskiy et al., 2020) augmentation steps like the HSV related: hue with a fraction of 0.015, the saturation with a fraction of 0.015 and value with a fraction of 0.4 augmentation. Besides color editing I also make use of positional changes like image translation of factor 0.1, image scaling of factor 0.5 and image flipping with a probability of 0.5

3.3.5. Evaluation

After collecting the data and training the models an evaluation step is needed for verifying the expressiveness of the models.

3.3.5.1. User Intention Prediction Module

For visualizing the results of the user intention model the previously defined technique of a confusion matrix 2.3.1 is used.

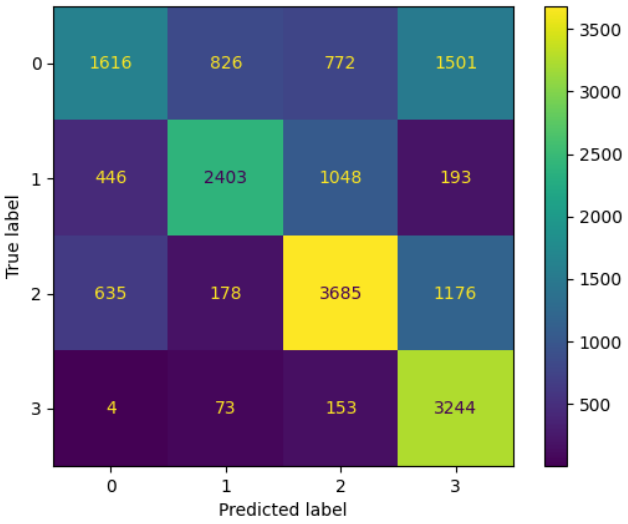


Figure 3.19.: Confusion matrix of user intention NN

One can see that the amount of correct predicted labels (see the diagonal of the matrix) is significantly higher than the miss classified labels. But the worst classified class is the

[11] Ultralytics. (2023). Hyperparameter evolution. https://docs.ultralytics.com/yolov5/tutorials/hyperparameter_evolution/.

class that represents no intention at all. This implies that the number of false positives is quite high. On this point I want to anticipate one user feedback from my later conducted study. The user exactly criticized this behaviour (see in the Appendix Section D.2 for full list). He stated that the amount of detected intentions is too high and could be one reason for it.

An other method for visualizing the results is the OvR-ROC curve plot as in Figure 3.20. One can clearly see that all curves are above the chance level $AUC = 0.5$. This means that the system performs better than random guessing and gives a first hint for the quality of the system. In a next step I investigated all curves representing an individual intention (bright blue, dark blue, green, orange). The worst performing of them is the bright blue curve representing the classification performance on samples belonging to no intention. This confirms the already made statements about the confusion matrix in Figure 3.19 about the problem of false positives.

Regarding the performance over all classes I now investigate the micro and macro averages as defined in equations 3.2 and 3.3. I have chosen to evaluate the micro average in addition to the standard macro average because I suspected a class imbalance. The reason for that is that in the data set creation process the amount of available time series samples correlates to data recording length which is different for every intention respectively. But in practice, the macro and micro average (dotted blue, dotted pink curves) don't differ much. Both have a decent AUC of 0.85 and 0.83.

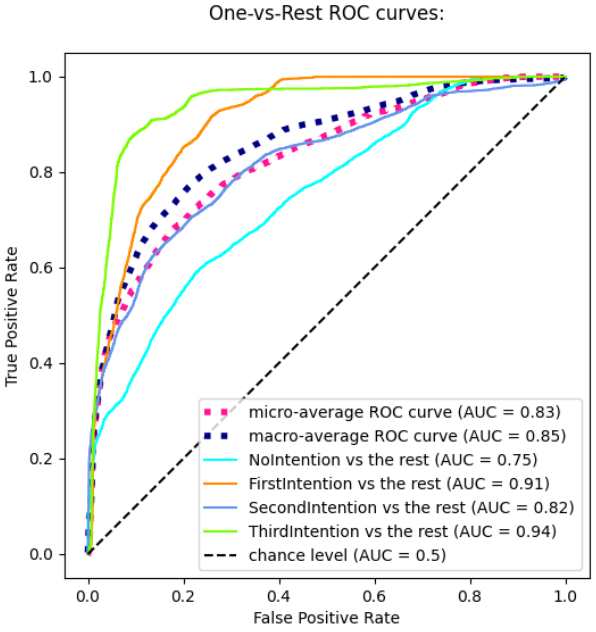


Figure 3.20.: Evaluation of user intention NN in OvR approach using ROC curves

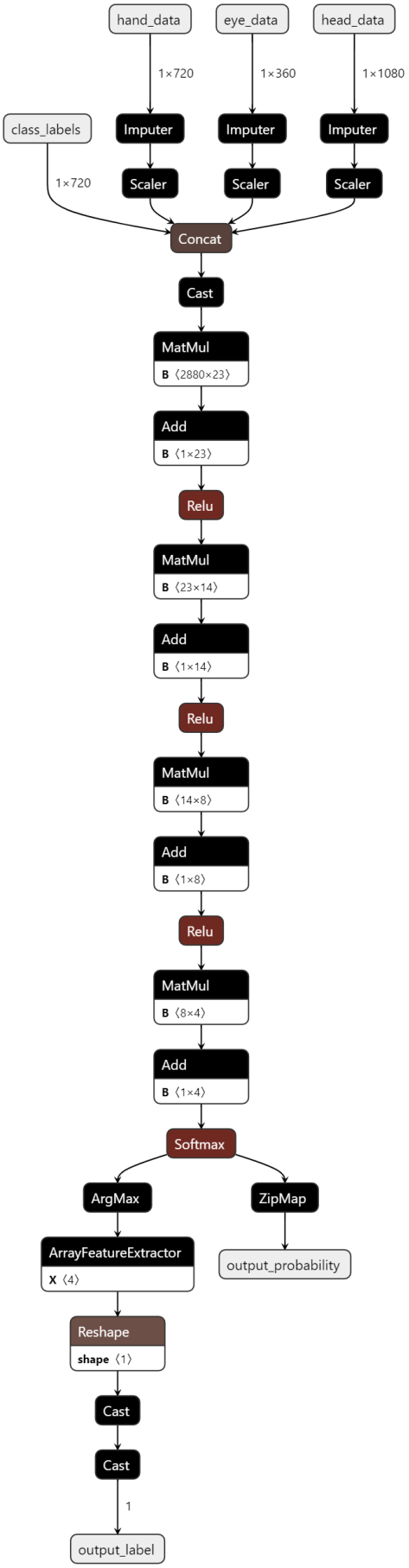


Figure 3.21.: Resulting MLP structure for user guidance system

Next I want to give with Figure 3.21 a more precise overview of the resulting MLP for intention detection after I outlined the high-level overview already in Figure 3.15: Firstly I want to explain the hyperparameters with their corresponding final values. The hyperparameters were validated by Exhaustive Grid Search to burst size = 180 samples ($\approx 5.4sec$) and hidden layer sizes = (23,14,8). They were chosen because they simply produced the model with the highest quality metrics. Next I go into NN inputs. For all numerical data (head, hand, eye data) an imputer and scaler were pre-staged. The Imputer ensured that no incorrect data (e.g. NaN) were supplied to the MLP and whenever necessary replaced with the statistical mean whereas the Scaler guarantees that all values are in the range [0,1]. The class labels were already given in ordinal scale from the preprocessing step in Section 3.3.3. Therefore no pre-stages are given for this input.

All hidden layer neurons are preceeded with ReLU activation units. The used softmax and neg-log-loss were previously defined in equation 3.5 and 3.6 respectively.

3.3.5.2. Object Detection/Segmentation Module

For the sake of brevity and clarity I packed the results of the object detection part into the Appendix Section B as instance segmentation has become the driving force during my research. The YOLO instance segmentation part produced the following results:

First of all I plotted the precision against the F1-score (as defined in eq. 2.12). Since it is a instance segmentation scenario both the box precision and the mask precision are plotted against their F1-score. Both depicted in Figure 3.22 and 3.23.

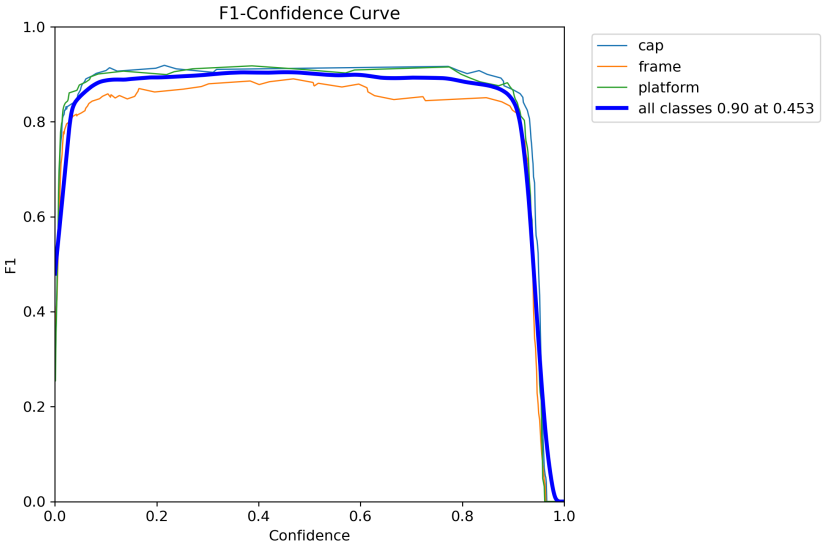


Figure 3.22.: Box F1 curve for instance segmentation

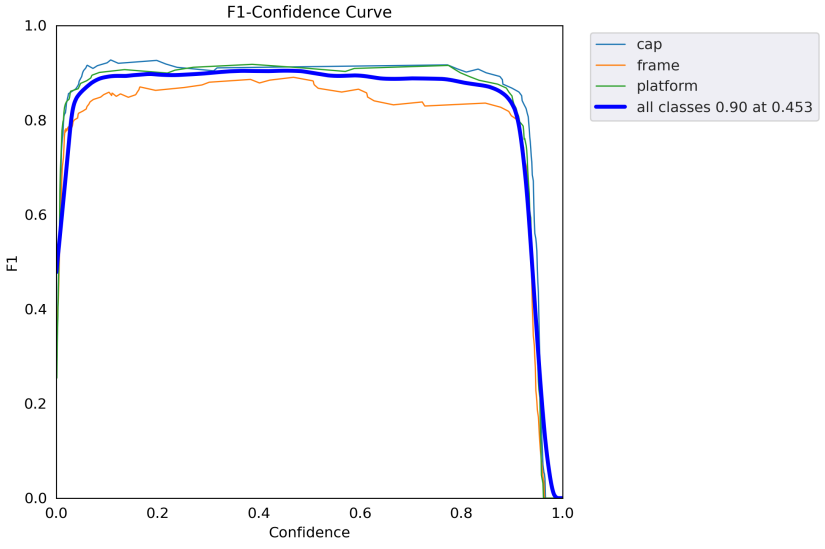


Figure 3.23.: Mask F1 curve for instance segmentation

The F1-confidence curves in Figure 3.22 and 3.23 have a maximum at confidence ≈ 0.45. One can read this value from the legend of the plot. Over all classes (represented by blue

curve) the maximum F1-score is 0.90 at a confidence of 0.453. Therefore I have chosen ≈ 0.45 to be my confidence level for accepting/refusing new samples. One can also say that a confidence value of ≈ 0.45 is the best trade-off between recall and precision.

I have put the complete training results for the instance segmentation model into Figure C.5 in the Appendix Section due to its sheer size. I will give some remarks to the notation: mAP@0.5:0.95 notation means that the model has been tested by firstly evaluate it with different IOU values between 0.5 and 0.95 in steps of 0.05 and secondly average all these results of first step into concluding result. mAP@0.5 simply denotes a fix IOU value of 0.5.

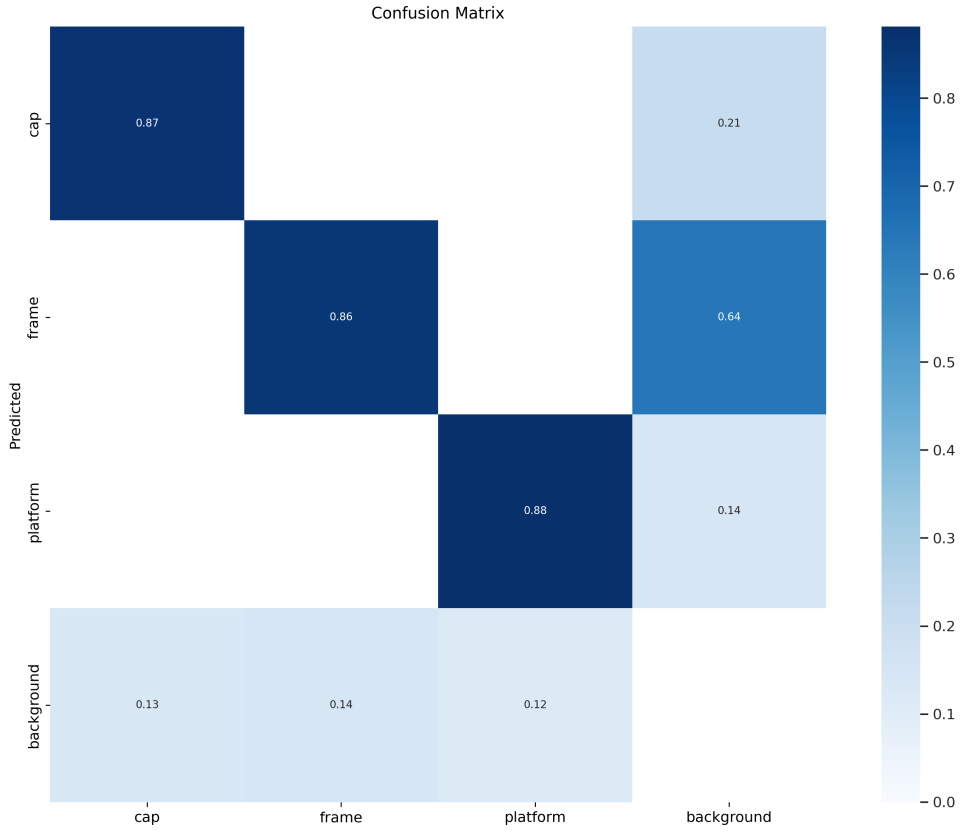


Figure 3.24.: Confusion matrix for instance segmentation task

Figure 3.24 shows us the confusion matrix for the instance segmentation model. The diagonal shows for each segmentation class (platform, frame, cap) a good classification probability of 0.87 for the cap, 0.86 for the frame and 0.88 for the platform. Whereas the lowermost row as well as the all right column points out that the FP rate is quite high. Especially for the frame class there are many samples belonging to the background but are classified as frame. This FP rate should be lowered in follow up works.

I had to weigh different network sizes against each other in regards to their mAP, number of parameters, FLOPS and overall application FPS. The results of this comparison are given in Table 3.4. The best trade-off between all measures is given by the medium network size. Hence a medium YOLO network was deployed on the server.

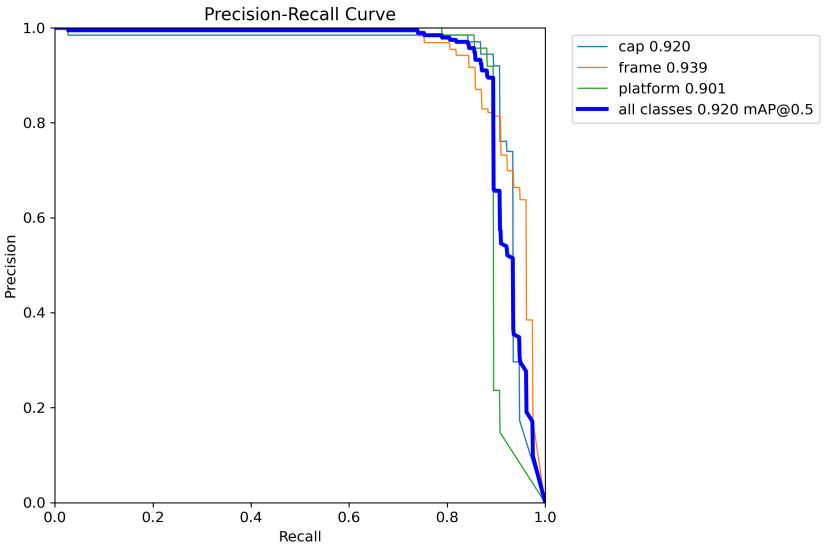


Figure 3.25.: PR curve for instance segmentation

Model	mAP box 50-95	mAP mask 50-95	params [M]	FLOPS [B]	FPS
Nano	0.75546	0.76663	2.0	7.1	23
Small	0.7644	0.76103	7.6	26.4	20
Medium	0.78109	0.77455	22.0	70.8	19
Large	0.75546	0.76663	47.9	147.7	21
X-large	0.79007	0.7769	88.8	265.7	14

Table 3.4.: Comparison of different instance segmentation NN sizes

The PR curve in Figure 3.25 reveals further insights into the quality of the trained model. During the training of the NN I wanted to push the curve as much as possible to the upper right. I have achieved an mAP score (defined in eq. 2.14) of 0.92 when IOU=0.5 threshold is chosen respectively. Based on these results and the findings about the PR curve in Figure 3.25 I can draw following conclusion: the instance segmentation module has a good quality in predicting object classes alongside their bounding boxes, masks and objectness scores. The server/client architecture allows me to run larger models on high enough frame rates (as examined in Table 3.4). This allows me to use it for my user guidance system.

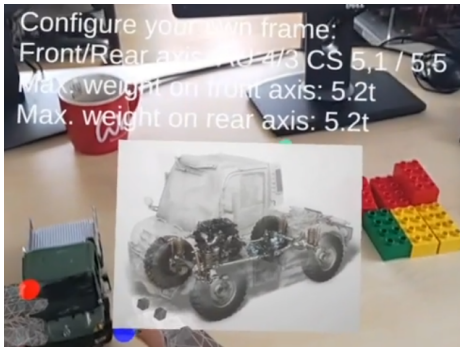
3.3.6. Deployment

All NN were trained in python and converted to Open Neural Network Exchange (onnx)[12] for further usage in an UWP server.

[12]onnx. (2023). Open neural network exchange. https://onnx.ai/.

3.4. User guidance

After the last step of Section 3.3 an user intention NN was deployed on the server as described in the application overview in Figure 3.1. Thus, the application is able to automatically detect user intentions. This is achieved by gathering user data and recording the users' FOV as video stream and collectively send them both from the HoloLens2 to the server. The video stream serves as input to the deployed YOLO network. The results of the YOLO network in turn serve together with the user data subsequently as input to the user intention NN. With this input, the user intention detection NN does an inference and displays MR content accordingly. This user adaptive MR content is portrayed in Figure 3.26. Furthermore, the menu opens to the right of the users' current eye hit position. This part of the application answers the in Section 1 secondly formulated RQ which is about building user-adaptive guidance based on detected user intentions.



(a) User adaptive menu for first intention

(b) User adaptive menu for second intention

(c) User adaptive menu for third intention

Figure 3.26.: User adaptive content for MR

The images for the intentions were taken from the official Daimler Unimog webpage as this work also uses a Daimler Unimog truck model. As the first intention is about discovering the bottom parts of the model the corresponding menu displays a proper image with information about the front and rear axis. The second intention, on the other hand, pops up because the user focuses the cap with its interior. The menu for that hints that the user is able to configure the cap interior ergonomically and the standard engine for this model is called OM 934. Last but not least, the menu for the third intention gives information about the loading capacity of the truck as the user focuses and grabs the platform.

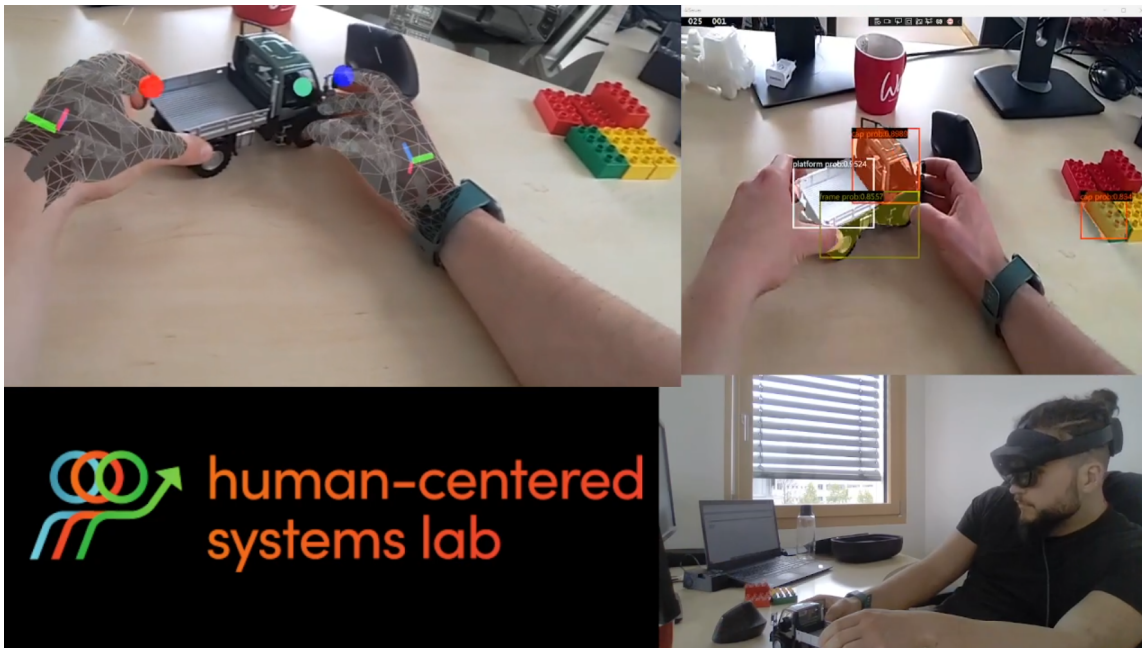


Figure 3.27.: Situation before menu as in Figure 3.26b for second intention pops up

Figure 3.27 shows the situation right before the menu opens. The upper left shows the MR scene as seen by the user. This part reveals the eye hit position (green) and both index finger tip positions (red, blue). Whereas the upper right shows the YOLO instance segmentation output. Every individual part of the truck is detected and segmented. The menu for the cap interior opens as the user focuses on the cap.

3.4.1. User Evaluation

Before this Section all necessary foundations for user-adaptive guidance have been laid out. The user intentions are detected and an informative, adaptive menu pops up when such an intention is detected. Whether these menus provide user guidance must now be tested in a survey. Therefore, a group of 15 participants tested the application and afterwards filled out a Short User Experience Questionnaire (S-UEQ) as described by Schrepp et al., 2017 (see Section 3.4.1.1). Additionally a short interview was conducted (see Section 3.4.1.2). The test of the application has been as follows:

The participant was handed over the HoloLens2. In a first step all mandatory configurations were done which included eye tracking calibration and manual adjustments for different head sizes. After that the user was told to investigate the truck as a customer who wants to know more about the truck and its individual parts. The participant was able to end the test as soon as he finished examining the truck by tacking off the HoloLens2.

3.4.1.1. Short User Experience Questionnaire (S-UEQ)

The S-UEQ yields a quantitative measure for the quality of the user adaptive guidance provided by the developed application. For the S-UEQ Schrepp et al., 2017 uses a differentiation between pragmatic and hedonic quality. Pragmatic quality measures (perspicuity, efficiency and dependability) are goal-directed and describe the so called utility and usability of the app whereas the hedonic quality measures (stimulation and novelty) refer more to joy of use. This fundamental distinction of each question can be seen in Table 3.5.

Item	Mean	Variance	Std. Dev.	Negative	Positive	Scale
1	1.2	0.9	0.9	obstructive	supportive	Pragmatic Quality
2	1.2	1.9	1.4	complicated	easy	Pragmatic Quality
3	1.4	0.8	0.9	inefficient	efficient	Pragmatic Quality
4	1.5	1.7	1.3	confusing	clear	Pragmatic Quality
5	2.5	0.6	0.7	boring	exciting	Hedonic Quality
6	2.6	0.8	0.9	not interesting	interesting	Hedonic Quality
7	2.0	1.1	1.1	conventional	inventive	Hedonic Quality
8	2.5	0.3	0.5	usual	leading edge	Hedonic Quality

Table 3.5.: Detailed quantitative results of sueq

More details on the exact structure of the S-UEQ can be seen in the Appendix Section D.1. In the following an interpretation of the results in Table 3.5 is given. Based on the mean column of Table 3.5 you can draw conclusions about pragmatic and hedonic quality. Values between -0.8 and 0.8 represent a neutral evaluation of the corresponding scale. For the evaluation in Figure 3.5 no value falls into this range. All values are above 0,8 and thus, represent a positive evaluation. The range of the scales is between -3 (horribly bad) and +3 (extremely good). So for every entry in Table 3.5, the application is evaluated to the positive.

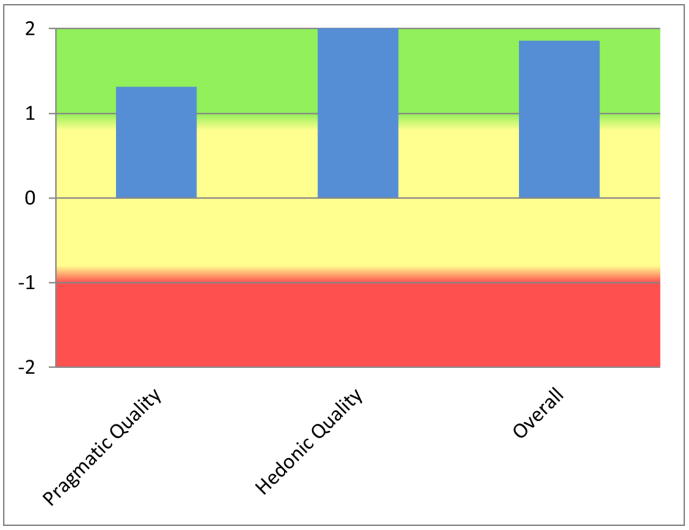


Figure 3.28.: S-UEQ quality measures

Figure 3.28, however, gives a quantitative measure for the overall (pragmatic/hedonic) quality of the system. Every unique and the overall quality is evaluated to the positive. Nevertheless, the pragmatic quality stays behind the hedonic quality and could be further improved. The results of the short interview can hint to the reasons behind it.

3.4.1.2. Short Interview

The short interview yields both, quantitative and qualitative measures for the user adaptive guidance provided by the developed application. All details on the short interview can be found in the Appendix Section D.2. Apparently, every participant has liked the provided adaptive guidance. Moreover, all participants agreed on the fact that the application should be further developed.

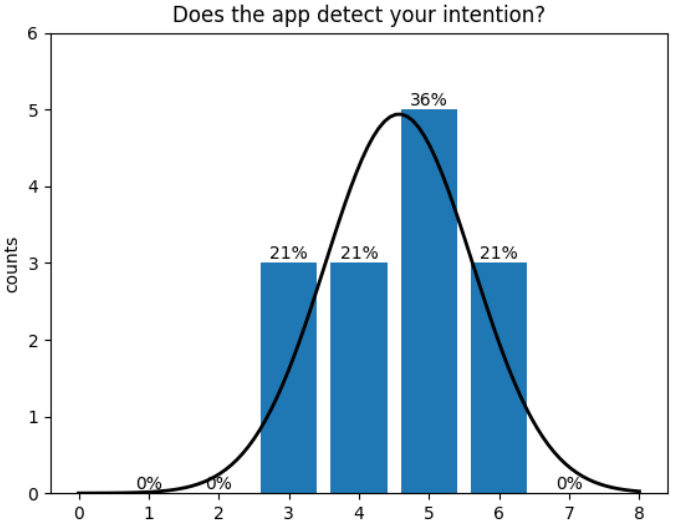


Figure 3.29.: User survey question on how good the application has detected their intentions (from 1 (not at all) to 7 (very good)); results overlaid with fitted Gaussian

In Figure 3.29 quantitative results of the interview are presented. The user was asked about how good the application has detected their intentions on a scale of 1 (not at all) to 7 (very good). The above displayed results are pretty decent as no participant has felt their intention not to be detected at all. Also the majority of participants have found the intentions good detected (scale 5,6). Around 40% (scale 3,4) are OK with the detection which means there is room for improvement in future work (see Section 3.5).

Regarding the qualitative measures, the survey provided some general proposed questions. The participants were asked about their biggest challenges while using the application. For users who were newly introduced to MR, the MR buttons turned out to be difficult to push. One user complained about the frequency of discovered intentions. At this point, one can make a link to the quantitative metrics of the ML models in Section 3.3.5.1. The ROC curve namely showed that the system has problems in distinguishing no intention from some intention at all. Exactly this fact is described by this participant. An other

participant would have liked the user-adaptive pop up menu to follow the gaze for having better guidance. Some other participant, on the other hand, has complained about the adaptive user feedback to be not expressive enough. This user would have wanted the menu to pop up directly on the truck. The critique point of an other user, in contrast, was about the direct opposite. This user has regarded the pop up menu as too close to the truck.

In reference to the HoloLens2, the users have been really enthusiastic. The majority of users perceived the HoloLens2 as very futuristic and sustainable which confirms the results from the S-UEQ. Though, some critique points about the HoloLens2 stick out. The limited FOV, lack of image quality and the reduced visibility during large sun exposure have been the three most named critique points.

The next question in the survey has concerned suggestions for other fields of application where the user guidance system could be useful. The participants not only see the user guidance system at work (f.e. in a warehouseman, manufacturing or real estate setting) but also in private use cases such as reading an instruction manual or receiving general information about the personal FOV. At this point, the educational purposes of this application need to be stressed out. Many participants can see themselves taught by such a system.

3.5. Limitations & Future Work

This Section reveals hereinafter not only limitations of the system but also discusses promising starting points for future work.

3.5.1. Posing TSC with CNN methods

In the Related Work Section 2 I decided to pose the user intention detection as a TSC problem using MLP. The SLR however exhibits that not only MLPs but also CNNs for TSC are very promising. Therefore it is up to future work to investigate their effectiveness.

3.5.2. Hand tracking

Multiple participants complained during the survey about the hand tracking capabilities of the HoloLens2 because it sometimes was not stable enough. The eye tracking however has been very stable and satisfying.

Regarding the hand tracking the application is limited by only taking index finger tip positions into account. This assumption has been made for limiting the scope of this work. The survey showed though that many participants were using e.g. their thumb for investigating the truck too. For future work it therefore would be interesting to additionally take the users' thumbs (and/or other parts og the hand) into consideration.

3.5.3. YUV to RGB conversion problems

The HoloLens2 streams its video in YUV color space. But the YOLO network consumes as input only RGB images. This circumstance has the consequence that a conversion step between both color spaces is interposed. But this conversion can have the following occurring artifacts:

Figure 3.30.: YUV to RGB conversion artifacts

Figure 3.30 shows one exemplary output of the conversion step (already resized to network quadratic input size). Especially on edges one can see green, red, yellow dots and/or edges. This could mitigate the quality of the instance segmentation NN.

3.5.4. Latency vs. Sampling Rate

Regarding the HoloLens2 the average sampling rate is at $\approx 30\frac{samples}{s}$ whereas the average measured latency is about 3 times higher resulting in $\approx 10\frac{samples}{s}$. The latency of the system affects every part of the application where an intermediate network transmission step is involved. This mainly biases the storage of NN results from the instance segmentation part into the CSV file. NN results (bounding box coordinates, class probabilities, masks, objectness score) are thus stored only every third sample together with user data leading to data holes. Finding a good policy in tackling these data holes is needed and not fully investigated yet.

3.5.5. The (dis)similarity of intentions

In Section 3.2 the three most observed intentions were discovered in a first exploratory study. These intentions hold discriminative features as further investigations have shown. But in the survey I discovered inter alia situations where different intentions were mixed.

Some participants f.e. have dragged the truck while raising the platform. In this situations unique decisions are hardly made.

4. Conclusion

Mixed Reality continues to grow in popularity and relevance. Moreover, MR has a broad field of application. These MR applications again put the interaction of the users with the augmented scene in the center of their functionality. The conducted SLR shows that MR has a great use and potential in guiding users in the augmented world which also includes teaching persons such as students or customers. The SLR, however, shows a research gap when the users' intentions should be taken into account for those interactions. Hence, this work firstly deals with detecting user intentions in MR applications through user data (eye, hand and head) tracking features. And secondly proposes a design of user-adaptive guidance for MR applications based on detected user intentions.

In a first step, the user guidance system provides a new way of user intention detection. The system is new in its way of training jointly on eye, hand and head data for making a classification step. Furthermore, it is not only new in the way it takes user data into account but also in how the system uses detected objects with their information (bounding box, class probabilities, segmentation masks) for predicting intentions. For achieving this, state of the art object detection and instance segmentation techniques were deployed and their usefulness for achieving user-adaptive guidance was investigated. This step is based on the fact that this work identified not only user data but also data about objects in the users' FOV as crucial for understanding users' interaction with the augmented scene. Based on detected user intentions, the system subsequently augments the users' surrounding accordingly with appropriate content. The augmentation consists of additional images and texts that are displayed in an intention adaptive manner. The quality of this new way of intention detection is not only supported by well-established quantitative methods for evaluating and optimizing ML models but also with qualitative metrics within a survey. This survey has validated the resulting user-adaptive guidance system for a specific use case. This use case has been about a participant who wants to learn more about a truck and thus investigated the truck in MR. Based on their actions (related to their eyes, hands or head) and the detected objects of the truck the user thereby received adaptive content about parts of the truck. Every participant has liked the user guidance provided by the system. The majority of the users ($\geq 50\%$) perceived their intentions well or very well detected. No user strongly disliked the provided guidance. Also every user in the survey have stated that the system should be further developed in future. The previously mentioned ML metrics and the survey, both reinforce the effectiveness of the new user guidance.

I want to conclude with the following remarks on the generalization possibilities of the system. For feasibility reasons this work has been limited to three intentions in regards to one special use case (truck investigation). But the participants of the survey and I, we both can think of many more domains a similar application could be utilized and be beneficial. One could use it f.e. for real estate, assembly lines or as warehouse operator with specific intentions like retrieving information of objects.

Bibliography

Abdrabou, Y., Schütte, J., Shams, A., Pfeuffer, K., Buschek, D., Khamis, M., & Alt, F. (2022). "Your Eyes Tell You Have Used This Password Before": Identifying Password Reuse from Gaze and Keystroke Dynamics. *CHI Conference on Human Factors in Computing Systems*, 1–16. https://doi.org/10.1145/3491102.3517531

Bahri, H., Krcmarik, D., & Koci, J. (2019). Accurate Object Detection System on HoloLens Using YOLO Algorithm. *2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, 219–224. https://doi.org/10.1109/ICCAIRO47923.2019.00042

Berkovsky, S., Taib, R., Koprinska, I., Wang, E., Zeng, Y., Li, J., & Kleitman, S. (2019). Detecting Personality Traits Using Eye-Tracking Data. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300451

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. (2019). YOLACT: Real-time Instance Segmentation.

Bottani, E., & Vignali, G. (2019). Augmented reality technology in the manufacturing industry: A review of the last decade. *IISE Transactions*, *51*(3), 284–310. https://doi.org/10.1080/24725854.2018.1493244

Brereton, P., Kitchenham, B. A., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583. https://doi.org/10.1016/j.jss.2006.07.009

Choi, M., Sakamoto, D., & Ono, T. (2022). Kuiper Belt: Utilizing the "Out-of-natural Angle" Region in the Eye-gaze Interaction for Virtual Reality. *CHI Conference on Human Factors in Computing Systems*, 1–17. https://doi.org/10.1145/3491102.3517725

David-John, B., Peacock, C., Zhang, T., Murdison, T. S., Benko, H., & Jonker, T. R. (2021). Towards gaze-based prediction of the intent to interact in virtual reality. *ACM Symposium on Eye Tracking Research and Applications*, 1–7. https://doi.org/10.1145/3448018.3458008

Duchowski, A. T. (2017). *Eye Tracking Methodology*. Springer International Publishing. https://doi.org/10.1007/978-3-319-57883-5

Fang, W., & Zhang, K. (2020). Real-time Object Detection of Retail Products for Eye Tracking. *2020 8th International Conference on Orange Technology (ICOT)*, 1–4. https://doi.org/10.1109/ICOT51877.2020.9468806

Gasques Rodrigues, D., Jain, A., Rick, S. R., Shangley, L., Suresh, P., & Weibel, N. (2017). Exploring Mixed Reality in Specialized Surgical Environments. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2591–2598. https://doi.org/10.1145/3027063.3053273

Gebhardt, C., Hecox, B., van Opheusden, B., Wigdor, D., Hillis, J., Hilliges, O., & Benko, H. (2019). Learning Cooperative Personalized Policies from Gaze Data. *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 197–208. https://doi.org/10.1145/3332165.3347933

Georges, V., Courtemanche, F., Senecal, S., Baccino, T., Fredette, M., & Leger, P.-M. (2016). UX Heatmaps. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 4850–4860. https://doi.org/10.1145/2858036.2858271

Guillaume, A., Vrain, C., & Wael, E. (2021). Random Dilated Shapelet Transform: A New Approach for Time Series Shapelets. https://doi.org/10.1007/978-3-031-09037-0{_}53

Guo, J., Chen, P., Jiang, Y., Yokoi, H., & Togo, S. (2021). Real-time Object Detection with Deep Learning for Robot Vision on Mixed Reality Device. *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, 82–83. https://doi.org/10.1109/LifeTech52111.2021.9391811

Hahn, J., Ludwig, B., & Wolff, C. (2018). Mixed Reality-Based Process Control of Automatic Printed Circuit Board Assembly Lines. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–6. https://doi.org/10.1145/3170427.3188652

He, H., She, Y., Xiahou, J., Yao, J., Li, J., Hong, Q., & Ji, Y. (2018). Real-Time Eye-Gaze Based Interaction for Human Intention Prediction and Emotion Analysis. *Proceedings of Computer Graphics International 2018 on - CGI 2018*, 185–194. https://doi.org/10.1145/3208159.3208180

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. https://doi.org/10.1007/978-3-319-10578-9{_}23

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition.

Holmqvist, K., & Andersson, R. (2017). *Eye-tracking: A comprehensive guide to methods, paradigms and measures.*

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, *33*(4), 917–963. https://doi.org/10.1007/s10618-019-00619-1

Johnson, J. G., Gasques, D., Sharkey, T., Schmitz, E., & Weibel, N. (2021). Do You Really Need to Know Where "That" Is? Enhancing Support for Referencing in Collabo-

rative Mixed Reality Environments. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3411764.3445246

Karolus, J., Wozniak, P. W., Chuang, L. L., & Schmidt, A. (2017). Robust Gaze Features for Enabling Language Proficiency Awareness. *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2998–3010. https://doi.org/10.1145/3025453.3025601

Koochaki, F., & Najafizadeh, L. (2018). Predicting Intention Through Eye Gaze Patterns. *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 1–4. https://doi.org/10.1109/BIOCAS.2018.8584665

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature Pyramid Networks for Object Detection.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2014). Microsoft COCO: Common Objects in Context.

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation.

Mariscal, G., Marbán, Ó., & Fernández, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review*, *25*(2), 137–166. https://doi.org/10.1017/S0269888910000032

Moro, C., Birt, J., Stromberga, Z., Phelps, C., Clark, J., Glasziou, P., & Scott, A. M. (2021). Virtual and Augmented Reality Enhancements to Medical and Science Student Physiology and Anatomy Test Performance: A Systematic Review and Meta-Analysis. *Anatomical Sciences Education*, *14*(3), 368–376. https://doi.org/10.1002/ase.2049

Moro, C., Phelps, C., Redmond, P., & Stromberga, Z. (2021). HoloLens and mobile augmented reality in medical and health science education: A randomised controlled trial. *British Journal of Educational Technology*, *52*(2), 680–694. https://doi.org/10.1111/bjet.13049

Nikolenko, S. I. (2019). Synthetic Data for Deep Learning.

Pakdamanian, E., Sheng, S., Baee, S., Heo, S., Kraus, S., & Feng, L. (2021). DeepTake: Prediction of Driver Takeover Behavior using Multimodal Data. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–14. https://doi.org/10.1145/3411764.3445563

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection.

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, Faster, Stronger.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large

Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(4), 40. https://doi.org/10.9781/ijimai.2017.445

Schwarz, J., Marais, C. C., Leyvand, T., Hudson, S. E., & Mankoff, J. (2014). Combining body pose, gaze, and gesture to determine intention to interact in vision-based interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3443–3452. https://doi.org/10.1145/2556288.2556989

Shearer C. (2000). *The CRISP-DM model: the new blueprint for data mining.*

Speicher, M., Hall, B. D., & Nebeling, M. (2019). What is Mixed Reality? *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3290605.3300767

Taha, A. A., & Hanbury, A. (2015). Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, *15*(1), 29. https://doi.org/10.1186/s12880-015-0068-x

Ungureanu, D., Bogo, F., Galliani, S., Sama, P., Duan, X., Meekhof, C., Stühmer, J., Cashman, T. J., Tekin, B., Schönberger, J. L., Olszta, P., & Pollefeys, M. (2020). HoloLens 2 Research Mode as a Tool for Computer Vision Research.

Vazquez, C. D., Nyati, A. A., Luh, A., Fu, M., Aikawa, T., & Maes, P. (2017). Serendipitous Language Learning in Mixed Reality. *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2172–2179. https://doi.org/10.1145/3027063.3053098

Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN.

Wang, X., Ley, A., Koch, S., Lindlbauer, D., Hays, J., Holmqvist, K., & Alexa, M. (2019). The Mental Image Revealed by Gaze Tracking. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. https://doi.org/10.1145/3290605.3300839

Weibel, N., Gasques, D., Johnson, J., Sharkey, T., Xu, Z. R., Zhang, X., Zavala, E., Yip, M., & Davis, K. (2020). ARTEMIS: Mixed-Reality Environment for Immersive Surgical Telementoring. *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–4. https://doi.org/10.1145/3334480.3383169

Yin, Y., Juan, C., Chakraborty, J., & McGuire, M. P. (2018). Classification of Eye Tracking Data Using a Convolutional Neural Network. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 530–535. https://doi.org/10.1109/ICMLA.2018.00085

Zeiler, M. D., & Fergus, R. (2013). Visualizing and Understanding Convolutional Networks.

Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2019). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression.

Appendix

A. SLR

	Mixed Reality		Tracking				Machine learning											User guidance		
							Traditional ML								Deep Learning			ML		
Authors	Augmented Reality	Virtual reality	HoloLens 2	Eye tracking	Head tracking	Hand tracking	Person tracking	Support Vector Machines	ADABoost	Logistic regression	Decision tree	Naive Bayes	k-nearest neighbors	Reinforcement Learning	Convolutional Neural Network	Multi Layer Perceptron	Object detection	Visual feedback	Audio feedback
Hahn et al., 2018	x	x				x											x	x	
Vazquez et al., 2017	x	x	x			x											x	x	x
Gasques Rodrigues et al., 2017	x				x	x												x	
Weibel et al., 2020	x	x	x		x	x	x												
Johnson et al., 2021	x		x															x	
Choi et al., 2022	x			x															
Schwarz et al., 2014	x	x		x	x	x	x		x								x		
David-John et al., 2021		x		x						x									
Gebhardt et al., 2019		x		x				x						x				x	
H. He et al., 2018				x	x			x			x	x							
Berkovsky et al., 2019				x					x	x	x	x	x						
Abdrabou et al., 2022				x															
Pakdamanian et al., 2021				x												x		x	
Karolus et al., 2017				x				x			x	x				x		x	
Georges et al., 2016				x															
X. Wang et al., 2019				x											x				
Koochaki and Najafizadeh, 2018				x				x									x		

Table A.1.: Coding table for the SLR

B. Object detection

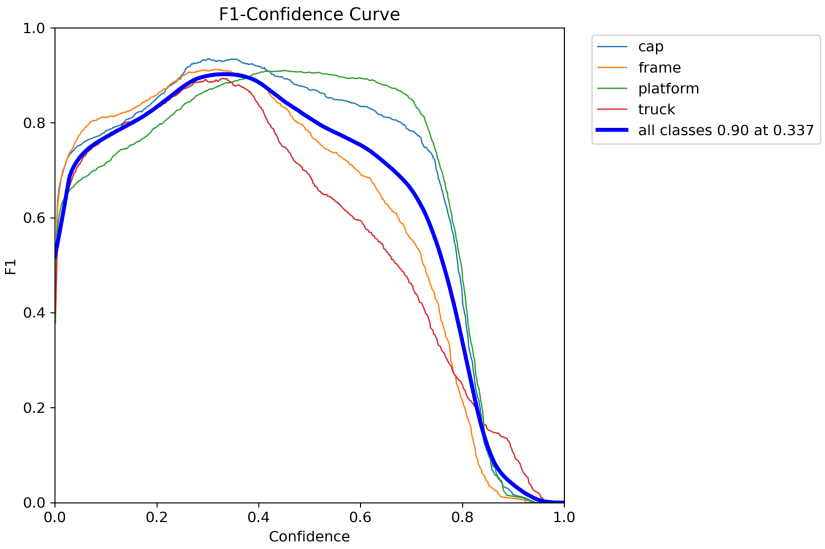


Figure B.1.: F1 curve for object detection

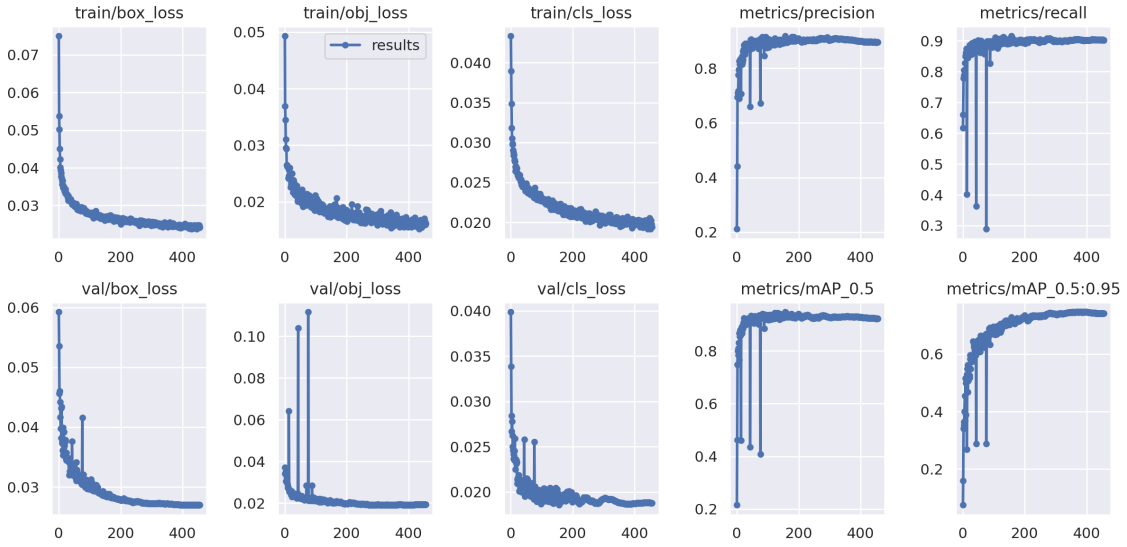


Figure B.2.: Training results for object detection

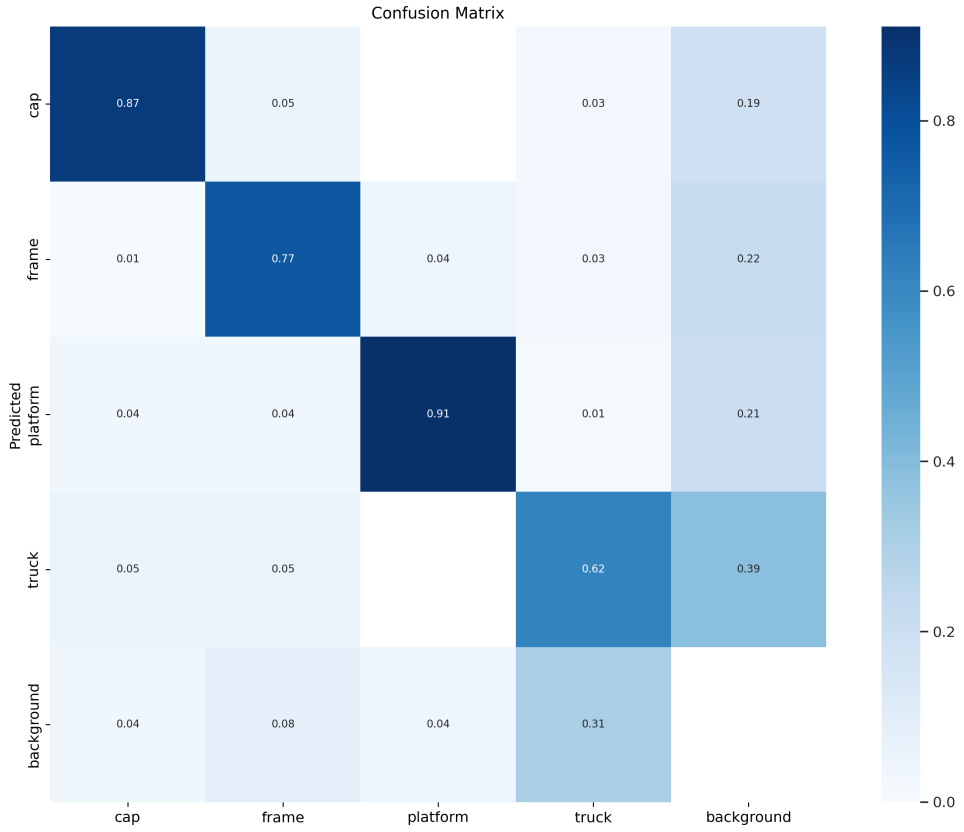


Figure B.3.: Confusion matrix for object detection

C. Instance segmentation

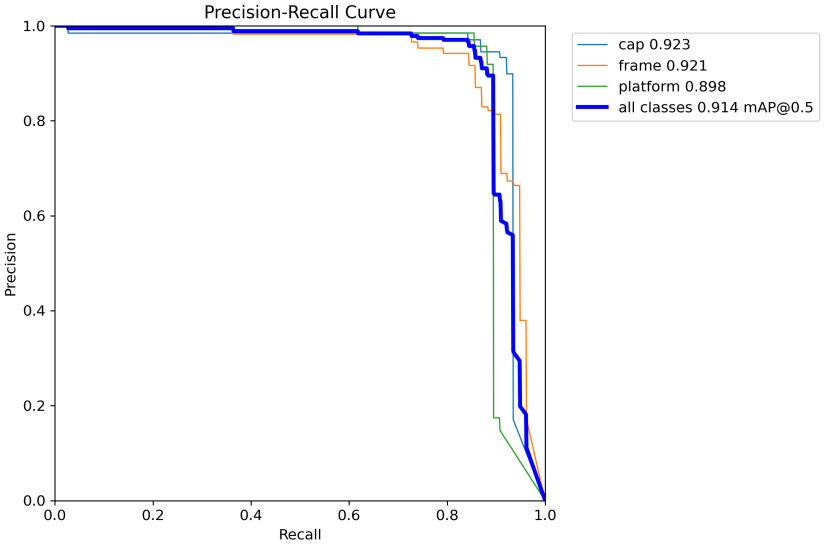


Figure C.4.: Mask PR curve for instance segmentation

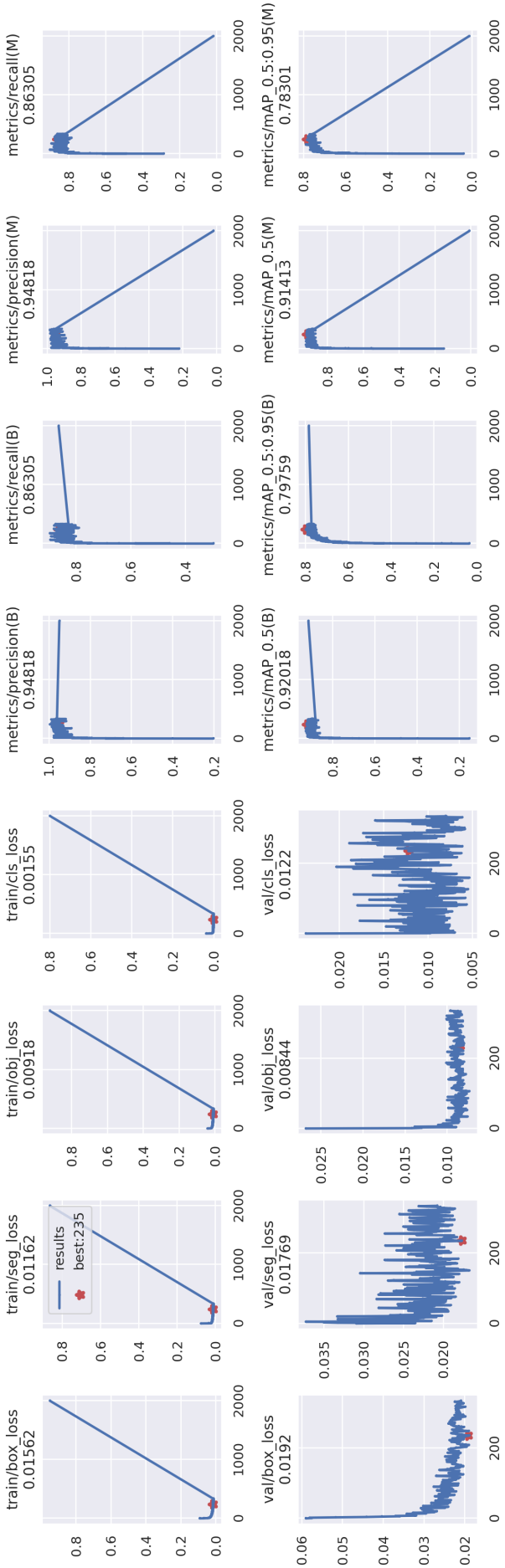


Figure C.5.: Training results of the instance segmentation model; mAP@0.5:0.95 notation means that model has been evaluated by first evaluating the model with different IOU values between 0.5 and 0.95(steps of 0.05) and concluding averaging. mAP@0.5 simply denotes fix IOU value of 0.5; As we have enabled early stopping the model does not run for the planned 2000 epochs but rather stops at $\approx$ 300 steps; All values after step $\approx$ 300 can therefore be omitted for evaluation

D. Survey

D.1. S-UEQ

obstructive	o o o o o o o	supportive
complicated	o o o o o o o	easy
inefficient	o o o o o o o	efficient
confusing	o o o o o o o	clear
boring	o o o o o o o	exciting
not interesting	o o o o o o o	interesting
conventional	o o o o o o o	inventive
usual	o o o o o o o	leading edge

Table D.2.: Short version of the User Experience Questionnaire (S-UEQ)

For the questions that are displayed in Table D.2 an introductory text were given as described by Schrepp et al., 2017:

Please make your evaluation now

For the assessment of the product, please fill out the following questionnaire. The questionnaire consists of pairs of contrasting attributes that may apply to the product. The circles between the attributes represent gradations between the opposites. You can express your agreement with the attributes by ticking the circle that most closely reflects your impression.
Please decide spontaneously. Don't think too long about your decision to make sure that you convey your original impression.
Sometimes you may not be completely sure about your agreement with a particular attribute or you may find that the attribute does not apply completely to the particular product. Nevertheless, please tick a circle in every line.
It is your personal opinion that counts. Please remember: there is no wrong or right answer!

Figure D.6.: S-UEQ means per item

Item	Mean	Std. Dev.	Confidence	Confidence Interval
1.Question	1.2	0.941	0.476	0.724 - 1.676
2.Question	1.2	1.373	0.695	0.505 - 1.895
3.Question	1.4	0.910	0.461	0.939 - 1.861
4.Question	1.467	1.302	0.659	0.808 - 2.126
5.Question	2.467	0.743	0.376	2.091 - 2.843
6.Question	2.6	0.910	0.461	2.139 - 3.061
7.Question	2.0	1.069	0.541	1.459 - 2.541
8.Question	2.533	0.516	0.261	2.272 - 2.795
Pragmatic Quality	1.317	0.782	0.396	0.921 - 1.712
Hedonic Quality	2.400	0.480	0.243	2.157 - 2.643
Overall Quality	1.858	0.502	0.254	1.604 - 2.112

Table D.3.: Confidence intervals ($\alpha = 0.05$)

D.2. Short interview

What follows are all questions from the short interview with the corresponding answers.:

me ***participants***

1. Did you like the user guidance?
   - 100% *answered yes (15 answers)*
2. Did the application detect your intention?
   scale from 1(not at all) - 7(perfectly)
   - *1: 0 votes 2: 0 votes 3: 3 votes*
     4: 3 votes 5: 5 votes 6: 3 votes
3. Should the application be extended and more possibilities be explored?
   - 100% *answered yes (15 answers)*
4. Where else would you use the system?
   - *Informationsquelle, wenn man in einem Lager ist*
   - *evtl. zur Unterstützung von Bedienungsanleitungen von Maschinen mit Videos oder eingeblendeten Erklärungen bzgl. der Bedienung*
   - *Can be a good idea for Real Estate, Aerospace and Yatch companies*
   - *Bei neuen Arbeitsplätzen oder zur Maschinenbedienung Generell in verschiedenen Anwendungsdomänen der Produktkonfiguration (bspw. Maschinen- und Anlagenplanung Gebäudeplanung)*
   - *Unterstützung bei Entstörprozessen*
   - *Informationen über Objekte, die in meinem Sichtfeld sind, erhalten (z.B. Baumarten, Alter eines Gebäudes).*
   - *Wartung, Gebrauchsanleitung, Bauprojekte*
   - *Zum Informationen über Objekte im eigenen Sichtfeld erhalten. Zum Analysieren von Kaufverhalten*
5. What were the biggest challenges you met while using the app?
   - *Die Knöpfe richtig drücken.*
   - *Drücken von Knöpfen zum Beginn.*
   - *Die Häufigkeit der ekannten Handlungen*
   - *Das noch ungewohnte Tool*

- *Zeitversetzte Erkennung der Hände, Gegenlicht des Fensters, Angezeigte Infos werden mit Bewegung des Kopfes abgeschnitten*
- *Persönliches empfinden beeinträchtigt durch Darstellung virtueller Inhalte im Sichtfeld (Bei VR auch immer schnell schlecht)*
- *Erkennung der Kabine*
- *Das System zu starten*
- *Die Fläche des Sichtfelds, auf der Inhalte gerendert werden können ist zu klein, dadurch findet man gelegentlich Inhalte nicht*
- *Teilweise wurde mein Fokus auf Teile des Objektes nicht korrekt erkannt, weshalb ein falsches oder gar kein Infofenster aufpoppte.*
- *Ich glaub die AR Brille hat mir nicht ganz gepasst, da ich den Rand nicht gut sehen konnte. evtl hat es wirklich wieder was mit brillenträgern zu tun gehabt, aber bin mir nicht sicher. nur ähnliche Probleme habe ich auch mit meiner VR Brille. Leider ist es nicht für Brillenträger entworfen (so mein blickwinkel)*
- *Das Fenster hat teilweise die Sicht auf das Fahrzeug behindert bzw. abgelenkt, Der grüne Punkt war etwas irritierend, Der Sinn der Anwendung hat sich mir nicht ganz erschlossen*

6. What else would you like to have?

- *Die Brile könnte kleiner sein.*
- *Die Brille könnte kleiner sein.*
- *deutlicheres feedback bei erfolgreicher erkennen von x oder y. zb ein kurzer pop up um das eingeblendete fenster besser in verbindung zu dem objekt bringen zu können*
- *UI / Anzeigesettings*
- *Eine leichtere Bedienung bis zum tatsächlichen Programmstart*
- *größeres Sichtfeld, mehr Informationen zu dem beobachteten Objekt*
- *evtl interaktion direkt am fahrzeug. d.h. wenn man die tür öffnet, dass dann ein fenster in der nähe sich öffnet.*
- *Etwas mehr Erklärung zu Beginn, was das Ziel der Anwendung ist, hilfreichere Texte, die Angezeigt werden, unterschiedliche Texte, je nachdem was ist betrachte*

7. Regarding the application in total how much did you like it?

    - *Sehr gut, wirkt sehr futuristisch und zukunftsträchtig.*
    - *Sehr gut und sehr futuristisch.*
    - *Sehr gut, ist nicht nur praktisch sondern auch überraschend innovative. Hinterlässt einen guten Eindruck welchen man mit dem Produkt/der Firma in Verbindung bringt.*
    - *Gut und interessant*
    - *Guter Ansatz, verbesserungsfähig bei Flüssigkeit und Erkennung der Objekte (Ladefläche wurde nicht erkannt)*
    - *sehr gut, spannend und interessant.*
    - *sehr toll*
    - *Es war sehr interessant und hat gut gefallen*
    - *gut*
    - *Sehr spannender Anwendungsfall, der im Kontext der Produktkonfiguration als Begeisterungsfaktor/USP fungieren kann.*
    - *sehr gut*
    - *Okay, kann bestimmt spannend sein, wenn unterschiedliche Informationen gegeben werden. (oder ich habe nur nicht gesehen, dass es unterschiedliche Texte sind und dachte nur, dass es immer der selbe ist)*

8. What's your opinion to the HoloLens2?

    - *Finde ich sehr interessant und es gibt viele potentielle Einsatzgebiete.*
    - *Sehr gut mit vielen potentiellen Anwendungsfeldern.*
    - *Eine sehr interessante Technik, welche in seiner Anwendungsvielfalt, aufgrund seines Preises, auf die Industrie/Marketing begrenzt ist.*
    - *Hat viel Potential für spannende AnwenduNgen*
    - *Gewöhnungsbedürftige Nutzung aber gute Unterstützung*
    - *leider grafisch noch sehr hinter her was zu meinem schumrigen gefühl beigetragen haben könnte.*
    - *hilfreich, aber noch nicht der Höhepunkt von AR*
    - *Mangelhaft so wie alle anderen VR-Brillen. Schlechte Bildqualität und schnelle Motion-Sickness*

- *Interessante Technik mit etwas zu sehr eingeschränktem Sichtfeld*
- *die selbe antwort wie bei "Was waren deine größten herausforderungen bei der Benutzung?"*
- *Find ich cool, Anzeige bei hellem Hintergrund muss allerdings noch verbessert werden.*

Prototype Video Publication Agreement

I hereby agree that the prototype video submitted by me may be published on the Internet.

Karlsruhe, 22.05.2023 _____

Jonas Heinle