POLITECNICO DI TORINO

Master Degree course in Data science and Engineering

Master Degree Thesis

Boundary Conditions for Human Gaze Estimation on A Social Robot: Evaluation of the State-of-the-Art Models and Implementation of Joint Attention Mechanism

Supervisors
Prof. Andrea Bottino
Prof. Koen Hindriks
Prof. Artem Belopolsky

Candidate
Nicola Scarano

Academic Year 2022-2023

Acknowledgements

A big thank you goes to Koen Hindriks that hosted me at the SocialAI lab and to Artem Beloponsky who supported my work with fundamental hints and guidance. Thanks to LinLin Cheng, we worked together daily, fighting frustration with excitement and resilience. Thanks to my family for the fundamental support to my studies.

C'era una volta un re, il cui nome nessuno poteva nominare, e neppure scrivere, e men che meno pensare. A dispetto di cio egli esisteva, egli viveva, egli faceva sentire la propria presenza, rendeva concreto ogni indizio del suo essere, fugava ogni dubbio sulla propria illusoria figura, imponeva nei confronti di sé stesso un crescente rispetto. Ma le lapidi non lo ricordavano, i libri non lo descrivevano, le poesie non lo cantavano, gli affreschi non lo ritraevano, le statue non lo raffiguravano, gli elenchi non lo catalogavano, i giudici non lo chiamavano a testimonio, i saggi non lo presentavano a modello, i miseri non lo imploravano, le dottrine non lo contemplavano, le profezie non lo incontravano, le teorie non lo asserivano. Tutti ne sapevano l'essenza, tutti non potevano negarne in alcun nodo l'essere e l'esistere, al di là di ogni altra congettura scettica o problematica. Ma non esisteva memoria - e per memoria si intende la funzione fisica, concreta, materiale del ricordare - di lui, della sua presenza, del suo agire, degli effetti da lui causati. Alcuni studiosi, invero, intuendone ingenuamente la presenza, come di fronte a qualsivoglia altro fenomeno, avevano cercato di arrivare ad una sua rappresentazione, o meglio si erano avventurati sulla strada dell'ipotesi scientifica - che cosa sia scienza è un altro mistero - di un "se" a cui però non era mai potuto seguire un "allora". Altri avevano anche coniato il binomio "genio e sregolatezza", ma nessuno capiva quale nesso ci fosse con il re, di cui nessuno parlava, di cui nessuno scriveva, di cui nessuno ritraeva l'immagine, di cui le lodi non erano contenute in nessun verso o in nessuna canzone. Le cose, che tutti conoscono come concrete e della cui esistenza parrebbe a chiunque cosa assurda dubitare, anch'esse esistevano in funzione di lui, ma non ne riuscivano a dimostrare coerentemente e soprattutto razionalmente l'esistenza. La razionalità - il dubbio sulla sua efficacia sempre più si accresceva - pareva portare alla conclusione che tutte le conclusioni si sarebbero confuse in un'unica conclusiva confusione. Egli, a dispetto di tutto, come a tutti era noto, esisteva. Il nome del re era Disordine. [1]

[1] Vittorio Marchis, "Dall'arte... allo zero, Piccolo dizionario filosofico dell'ingegneria", Mondadori Universitá, 2020.

Abstract

Humans are highly effective collaborators, able to quickly coordinate with each other, often without the need for detailed guidance or instructions. This is because they can rely on a variety of communication cues to guide interactions, both explicit, such as gestures or words, and implicit, namely gaze. Gaze is a powerful nonverbal communication cue in humans and can contain different types of information: interest, engagement, and attention in social interactions are just a few. Social robotics is a discipline that focuses on building robots with strong communication capabilities that are able to interact with each other, with humans, and with the surrounding environment. These particular types of robots can derive tremendous benefits from the ability to recognize the line of sight of their human partners. By knowing where humans are looking, social robots can share the attention of others and behave more humanely, increasing the trust of their partners. Recent advances in Deep-Learning-based gaze estimation improved the relevance of such cues in social robotics. A mix of good performance, scalability, and cost make these Deep Learning methods an effective alternative to older technologies (e.g., eye-tracking glasses). On the other hand, while there are several papers in the literature on datasets and algorithms, there are few studies on the application of appearance-based gaze estimation in Human-Robot Interaction (HRI) scenarios. There is a need for standardized, well-defined experiments to thoroughly evaluate the performance of these models in a social setting. In this thesis, we present an experiment to investigate in a social interaction scenario the performance of the most relevant gaze estimation methods currently available in the literature. During the experiment, images of people looking at different targets are captured by two cameras located in front of them. The experiment is conducted in a laboratory environment designed to resemble the human-robot interaction scene as closely as possible. The images with the ground truth annotations are then used to generate a small dataset for the evaluation of the algorithms. In this work, we test two deep learning models (L2CS and ETH) trained on the main datasets available today: Gaze360 [2019] and ETH-XGaze [2021]. The models are tested on our dataset and the results are analyzed through statistical and graphical tools allowing us to extract important insight on the performance of models and datasets in a social interaction scenario. In the last part of this work, we performed a fine-tuning of L2CS on our dataset, increasing its performance on our specific task. The model is then used to implement a joint attention behavior in the Pepper Softbank robot allowing it to perform real-time responses to human gaze.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Social AI lab

This thesis work has been conducted during an exchange period at the Social AI lab of the Vrije Universiteit Amsterdam. The months spent in the lab have had as output this thesis and a paper that will soon be published. The thesis is the beginning of a more complex and comprehensive work on gaze cues in Human-robot interaction scenarios. The Social AI lab focuses its work on applied AI in sectors all strictly related to humans and socially desirable applications of AI. The human-centered work performed in the group ranges from health to teaching applications eventually exploiting social robots such as Pepper and Nao from Softbank in social contexts. The lab successfully promotes interdisciplinary work between people coming from various backgrounds, from IT to natural sciences. This thesis fits into that part of the research in the lab that focuses on AI application in social interaction: exploiting social humanoid robots and modern AI tools to analyze visual or speech cues.

1.2 Research and contribution

Humans are highly effective collaborators, able to coordinate quickly with each other, often without the need for detailed guidance. This is because they can use a large number of communication cues to guide interactions, both explicit, such as gestures or words, and implicit, such as gaze. Gaze is a powerful cue of non-verbal communication [17] that indicates interest, engagement, and attention in social interactions [26]. For instance, in stores, when a customer is interested in one item, most of the time she or he will gaze at that object for a longer time than others. which is also a very important skill especially for sellers. In this way, they can know what's their customer's preference and then start to introduce that commodity to strengthen the customer's willingness to buy.

Also, social robots can get enormous benefits from the gaze direction detection of their human partners. By knowing where people are looking, social robots can share others' attention and increase their partners' trust in them. Which makes the interaction between humans and social robots more natural. This is quite vital for humanoid robots. Their human-like appearance gives us an expectation towards them: behave in a human-like way. It means humans will assume these robots have similar abilities to their own on visual perception. And the robots could respond to their gaze cues like following humans' gazes to anticipate their needs and intentions. However, this function has not been widely used.

The main reason why social robots with the ability of gaze tracking haven't been widely used might be the requirement of specific camera quality like extremely high resolution and narrow

field-of-view images. However, most social robots are equipped with wide field-of-view images to interact with people in a large environment. Similarly, for different reasons like real-time application and cost consideration. Some robots are also limited to low-resolution cameras. And this calculation is easily affected by light and shadow. There are two popular alternatives. The first one is using dedicated hardware like special glasses or helmets. This requires anyone who intends to interact with the robot to wear the special device. Which makes them difficult to use in open environments (e.g. shopping malls, airports, hospitals, etc.). The second one is choosing the so-called 'head gaze' instead of the eye gaze. The good point is that it's easier to get the head gaze direction. But the problem is that it's less accurate than eye gaze [35].

There is hence an urgent need for a method that uses an inexpensive off-the-shelf camera, doesn't require additional hardware, and is capable of detecting eye gaze direction. Appearance-based methods might be a good choice. This method directly uses a common camera to detect eye gaze direction. While there are rare studies applying this in HRI interaction. This is because appearance-based methods are mostly solved by deep learning models whose effects highly rely on the datasets they are trained. The early datasets have a limitation on the head pose and gaze variations and the max absolute yaw is about 40 degrees which is far from the real yaw range in HRI. This brings a big challenge to applying this method in HRI. Recently, this method has got a breakthrough, spurred by the release of two large datasets (Gaze360 dataset and ETH-XGaze dataset) [48], [22]. Unlike most datasets getting data only indoors, Gaze360 dataset was collected both indoors and outdoors. As for The XGaze dataset, while it was collected just in the laboratory, it used 18 custom high definition SLR cameras and included 16 adjustable illumination conditions. Both datasets are large-scale, and the maximum yaw is up to ± 70 degrees from directly facing the camera. It has led to the hope of using appearance-based methods in HRI. Some state-of-the-art models based on these two new datasets have been proposed lately. However, there have been no studies that use these models on social robots. Mostly since their qualities are measured using different protocols and metrics and it's unclear if the accuracies of the models are good enough in the HRI application. Therefore, there is a need for a more standardized, well-defined experiment to evaluate these models which will be applied to HRI.

In this thesis, we design an experiment called calibration experiment to explore the quality of two models based on the Gaze360 and ETH-XGaze dataset respectively. To get as close to the human-robot interaction scene as possible. It has nine different human positions including three rows (the distance from the robot is respectively 1meter,2 meter and 3 meter) and three columns (left, center, and right relative to the robot). Which covers the main area in the social zone where most interaction take place [30]. Moreover, we directly use the off-the-shelf camera inside Pepper, and choose the most common resolution 640*480. To explore the effects of different resolutions on these models. Additionally, we put a high-quality camera (4k camera) onto the robot head to take pictures at the same time. In the end, the dataset collected is used to fine-tune the best model on our specific task. The model is then deployed in the Pepper robot from Softbank and exploited to perform a joint attention mechanism in a real-time Human-robot experiment.

Chapter 2

Background

2.1 Robot appearance

Eye gaze research is performed using robots of really different between each other in both appearance and capacities ranging from cartoon-like robots to extremely human-like humanoids. The robots ranges also in the possibility of performing eye movements or just head ones as the Pepper robot we used in this thesis works. These type of robots use head orientation to indicate gaze direction, mechanical behaviour that has shown to be still communicative at a gross level in human-robot interactions (examples are the Pepper and the NAO robots from Softbank). Eye gaze research is also divided among the type of artificial agents, that can be virtual (a robot showed on a screen) or an embodied robot. Not all the research result agree on the same point but it seems to exist a confluence on results that shows that embodied robots helps people to be more engaged and to keep them more focused on the task, people seems to be trust more the agent and are rated more positively.

2.1.1 Pepper - Softbank Robotics

Pepper is a humanoid robot developed by Softbank Robotics. Pepper has been designed for human interaction, it has a mix of social capabilities that range from speech recognition to facial recognition including natural language processing and expressive body language. Pepper stands at approximately 1.2 meters tall and weighs around 28 kilograms. Its body is composed of a combination of plastic and metal components, providing durability and flexibility. The robot features a rounded head with expressive LED eyes and a tablet display on its chest for communication and information display.

The robot head is equipped with 4 cameras, a 2D camera in the mouth, a couple of 2D cameras behind the eyes that provide images in stereo vision, and a top 2D camera in the forehead Figure 2.2. The two 2D cameras can reach a resolution of 2560x1920 at 1 frame per second (fps) and down to 640x480 at 30 fps.

The robot's arms and hands are designed with multiple degrees of freedom, allowing it to perform various gestures and interact with objects. Pepper's hands are not designed for gripping or manipulation but are rather intended for expressive movements and touch interactions. Underneath its exterior, Pepper is powered by a combination of hardware components that facilitate its operation. It features a solid processing system, including CPU, GPU, and a set of sensors and actuators for locomotion and balance. To enable wireless connectivity and communication, Pepper is equipped with Wi-Fi and Bluetooth capabilities. It also has a range of speakers and microphones that enable it to listen to and speak with humans. These audio features, combined

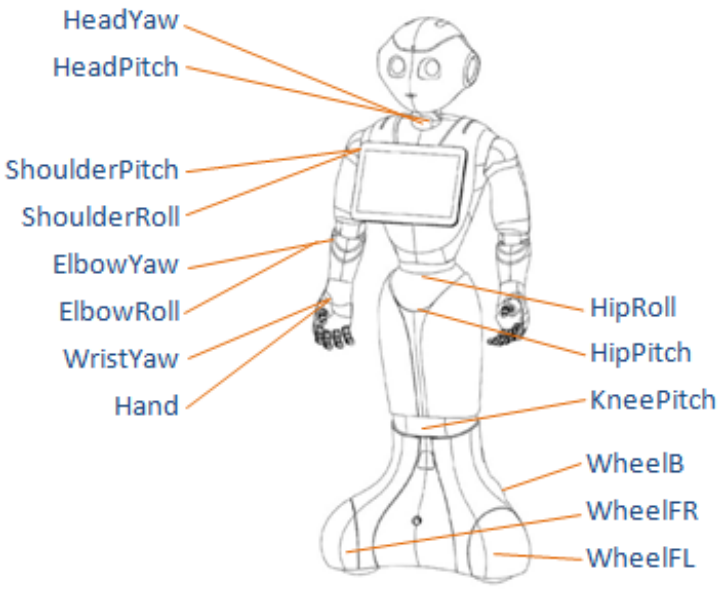


Figure 2.1: The picture shows all the joint connections available in the Pepper. [4].

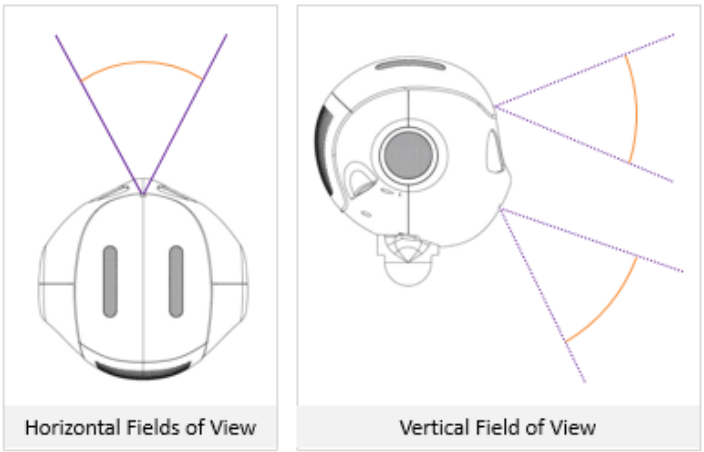


Figure 2.2: Horizontal and vertical Field of View of the Pepper. In particular, this picture is focused on the FoV of the two monocular cameras available: one on the mouth and one on the forehead of the robot [3].

with its natural language processing capabilities, allow Pepper to engage in conversation and respond to user queries and commands. Pepper has found applications in various domains, including retail, hospitality, education, and healthcare. It can serve as a receptionist, customer service assistant, educational companion, or even a companion for the elderly, providing social interaction and assistance.

2.2 Implicit interaction in HRI

For implicit communication we mean all those interaction where the transferring of information is not the main goal of the interaction but still some information is transferred citation—-. Implicit communication support the explicit communication adding more information to it [6], [2], make it more effective, reducing possible errors and making it more redundant and thus robust [25], [33]. For example imagine a top increase of the speech volume all of a sudden, this situation will help the listener to understand the speaker's emotional state (tired, angry, exited, etc), and the same happen for gaze cues shared between speakers and listeners. An clear example of the usefulness of implicit communication came from the military field, where this type of communication cues are constantly used to enrich the communications [44]. Frequently, implicit actions are directly linked to the communicator own goals of the interaction [28]. It has been shown that these goals can be transferred more easily through implicit cues rather than with direct communication. Of curse, its important to state that the perception capabilities of the other person play an important role in the effectiveness of these implicit cues [27]. In this work, we will focus our attention on probably the most information richer and complex implicit cue, the gaze. In particular, eye gaze is extremely important and various psychological studies have evidence that there is a special mechanism in the brain for just for interpreting eye gaze, suggesting the importance the human brain assigns to the gaze stimulus [12]. During the years many studies have been made on gaze from human-human interaction to humans interacting with computers, robot or interactive screens [43].

2.3 Social gaze

Social eye gaze is any gaze that is interpreted as communicative by an observer. Studies shows that human are able to shows gaze that are capable of this type of informal social communication. Interesting is the fact that evidence has shown that only humans have the capability of perceive others intentions from eye gaze. Examples of different types of eye gaze are:

- Mutual gaze: commonly defined as eye contact, it consist in an person that direct its gaze to another person's face or to its eyes and vice versa;
- Referential gaze: is a gaze directed to an object or a location. This type of gaze usually occurs together with verbal indication to an object;
- Joint attention: its a type if gaze that involves sharing attention focus on a common object. It usually involves several phases beginning with a mutual gaze to establish their attention and then a referential gaze to push attention to the object of interest, and then come back to a mutual gaze.
- Gaze aversions: are change of the gaze direction from a target. These behaviors are common when someone is listening to a speaker and shift its gaze from the speaker's face.

2.3.1 Gaze for conversation

In the early stage of human-human eye gaze research the focus was massively on the role of gaze in conversation and speech. In fact during conversation gazes may carry a lot of information:

regulating intimacy, conveying emotions and intentions, regulating the structure of the conversation. While listening, people focus their attention on the person who is speaking on average 88% of the time. Instead, the person that is speaking looks at the listener's face the 77% of the time [42]. During conversation is common to look at the floor, to the face of the interlocutor, or to look at other spots, and the duration and the sequence of these gesture can convey different information [45]. Moreover, in modern research the precise timing of gaze aversions, mutual gaze and other gestures have been recorded during conversations allowing us to precisely understand how the different behaviors are used and what they convey [5], [32]. Another important factor that affects eye gaze movements is the conversation topic, for example, people show less mutual gaze when the conversation is intimate [46], [21]. Moreover, gazes are also influenced by people's personalities, introverts tend to look less on their partner's face than extroverts. On the other hand, the duration of mutual gazes do not depend only on the single individual personality but also on the interpersonal dynamics shown between the partners.

2.3.2 Gaze for object reference and manipulation

When we talk about object reference and manipulation then eye gaze play a very important role. When people are interested in an object around them they look at the object before naming it [47], on top people are also really good at predicting what the partner is going to do (which object he is going to manipulate) just by looking at the partner's referential gaze [7]. Research has shown that people can respond more quickly to their partner when they can see his referential gaze on the object of interest. Instead, the response is slower when they cannot see their partner's face. Mutual gaze is also an important carrier of information in case of object manipulation between people. Partners usually signal their readiness to take the object establishing a mutual gaze. Caregivers in nursing homes demonstrate their availability to their patients by looking broadly, and people wait for the caregivers to make mutual eye contact before asking for help.

2.3.3 Designing robot behaviours

The appearances and behaviors of robots are developed by researchers that adopt a design-focused approach to achieve certain objectives, such as displaying involvement or engaging in shared attention. Here we present the positive and negative effects that manipulating robot gaze behavior can have on interactions between humans and robots. Robot gaze can be utilized to enhance human-robot interactions in a number of sectors, according to design-focused studies. Although the appropriate amount and direction of gaze depends on the topic of the conversation, robots can use a combination of mutual gaze and gaze aversions to control the pace and participation in conversations. Robot gaze can be employed for deictic references; when combined with vocal deictic references, this kind of information transfer is more efficient than using voice alone. Robots' ability to communicate their mental states through eye contact enhances collaboration and learning. Additionally, the gaze can convey emotion and personality, which can enhance user relations. The fact that socially and contextually contingent gazing is more efficient than gaze behaviors that are unrelated to the interaction is a common feature in this research. When a robot's attention is focused on what is being said or done, people react to them more favorably, recall discussion topics better, and finish tasks more rapidly. For instance, when a person's attention follows conversational speakers, they have a more favorable opinion of robots. Additionally, looking at human partners appears to improve information retention and the effectiveness with which those partners carry out cooperative tasks like handovers.

2.3.4 Joint attention

Joint attention behaviors refer to the coordinated attentional focus between two or more individuals towards a common object or event. It plays a crucial role in human social interactions, communication, and the development of social cognition. This phenomenon has garnered significant attention in psychology, neuroscience, and developmental research, as it provides insights into how humans establish shared understanding and engage in cooperative activities. During joint attention, individuals direct their attention to an external stimulus, such as an object or an event, while also being aware of others' attention towards the same stimulus. It involves three key components: attention coordination [9], mutual gaze [15], and shared intentionality [40]. Attention coordination refers to the ability to align attention with others towards a specific target. It involves the awareness of the attentional state of others and the ability to adjust one's own attention accordingly. This coordination can occur through the use of gestures, gaze cues, or verbal communication, allowing individuals to establish a common focus of attention. Mutual gaze, another critical aspect of joint attention, involves the shared visual engagement between individuals. It occurs when two or more individuals make eye contact and direct their gaze towards the same object or event. Mutual gaze serves as a powerful social cue, facilitating communication, and creating a sense of connectedness and shared experience. Shared intentionality refers to the understanding that the attentional focus is shared between individuals. It implies a mutual understanding that the attention directed towards an object or event is intentional and purposeful, leading to the formation of a common ground for communication and cooperation. The study of joint attention has been extensively explored in developmental psychology, particularly in infants and young children.

2.3.5 Technology behind eye gaze

Robots and virtual agents can communicate with one another through social gaze in a variety of ways. Modeling the underlying neurological or psychological processes of eye gaze is one method that is based on the study of human cognition. This biologically inspired strategy is based on the idea that creating gaze that seems natural can be achieved by imitating biological processes. The timings, frequencies, and locations of gaze aversions, for example, which are observed during observations of human interactions, are examples of empirical measurements of gaze attributes that are used to inform this method. While using slightly more in-depth observations than the biologically inspired approach, this data-driven approach still seeks to replicate observed human behavior. A third strategy is to create heuristic systems that are not based on biological or empirical findings but yet seem to produce expressive gaze (such with rules taken from animation principles). Although each of these strategies has advantages in terms of generating gaze that enhances interactions, they all have cons as well. Depending on how crucial it is for gaze to be anchored in realistic behaviors versus how crucial it is to have design control over the behaviors, one should choose a method for the technological deployment of eye gaze. Biological models frequently concentrate on the areas of the nervous system that psychologists are familiar with, like the visual attention system. Although cognitive architectures make an effort to produce more complex gaze behavior, it is impossible to precisely design the behavior because it arises from the structure of the system. Although data collection for empirical systems can take some time, the resulting gaze behaviors are comparable with or even superior to those of hand-tuned systems. A balance needs to be structured between the advantage of having gaze habits supported by empirical data and the expense of gathering and annotating this data. Designers can more precisely define gaze behaviors with heuristic systems, but it's possible that these specified behaviors won't match how gaze is really employed in social interactions.

2.4 Approaches in social robotics

Different approaches for gaze estimation have been proposed for human-robot interaction, e.g., [31], [29], [13]. It appears that one of the more popular methods is to use hardware [39] such as stationary or mobile (glasses). Hardware often uses an infrared light source to illuminate human eyes, and sometimes also uses another source to get a visible reflection from the eye. [37], [34]. These devices are capable of producing accurate gaze estimations, but require users to wear additional hardware and they often require calibration. This produces physical encumbrance and self-awareness of being monitored, which can affect social interactions [38]. Additionally, hardware solutions do not scale up when interaction involves multiple people. Overall, hardware solutions are not suitable for social interactions "in the wild". In order to avoid using additional hardware devices, many researchers have opted for using head orientation as a proxy for gaze estimation (also called "head gaze") [23], [11]. For instance, in Ivaldi et al [20] the iCub robot's camera was used to estimate human partner's attention from head direction. However, in social interactions, especially at close distances, people tend to make glances without moving their heads. Palinko et al [35] have systematically compared head and eye gaze during human-robot interaction and showed that eye gaze was more accurate and natural than head orientation for estimating human's attention. In more recent work, model-based approaches and regular cameras have been used for estimating gaze of users interacting with a social robot [35], [36]. These approaches use either 2D features (such as facial and eye landmarks) or a 3D geometrical model of the eye to estimate gaze [10]. In principle, they can estimate eye gaze without the need for an additional device, but in practice it has one major limitation: only a small range of head positions are allowed and often requires re-calibration. This can be time consuming and disruptive to the user experience, which makes it impractical "in the wild". To overcome some of the limitations highlighted above, it is interesting to evaluate whether appearance-based methods provide a suitable alternative that can be used for human-robot interaction. A key question is whether these methods perform well in combination with the cameras that are available on existing off-the-shelf and often used social robot platforms. Moreover, appearance methods work without explicit calibration e.g. [19], making them attractive for HRI. However, to our knowledge, no studies have systematically examined the feasibility of using the appearance-based models for HRI.

2.5 Gaze Estimation Datasets

Appearance-based methods are mostly solved by deep learning models which are highly subjected to the datasets they are trained. Though in earlier years many gaze datasets have been published, they are mostly limited to physically constrained applications, like desktops [19] or smartphones [24]. Mainly they are collected by a static recording or a camera inside smartphone. It is successful in easier control and greater performance but fails to be useful in more general applications. For example, the datasets captured on a smartphone are closer to users so the models based on this dataset may perform terribly in an open area. In order to capture human gazes in a natural scene, it is important not to unduly restrict the pose of the subject, allowing coverage of the full range of head and eye directions associated with the camera. Researchers have acknowledged this and many datasets with relatively small variations in head posture and gaze have been published [49] [14] [16]. However, these datasets suffer from limited ranges of gaze and head pose. The max absolute yaw is about 40 degrees [48] and it is too far from the natural range of gaze in HRI. Thanks to the release of two large-scale datasets (Gaze360 dataset [22] and ETH-XGaze dataset [48]), the appearance-based method has got a breakthrough lately. Overcoming the limitation of gaze range, the two new datasets can reach ± 120 yaw degrees from directly facing the camera. Gaze360, the largest publicly available dataset, consists of 238 subjects indoors and outdoors, it labels the 3D gaze of different head positions and distances. Unlike Gaze360, The

ETH-XGazeÂ dataset captures pictures only in the laboratory, but it adopted 18 custom high-definition SLR cameras and included 16 adjustable illumination conditions. These advantages make it much more possible to adopt appearance-based methods in HRI. Some state-of-the-art models based on the two datasets have been proposed afterward [1] [48]. However, there have been no studies that use the models in HRI. The aim of the present paper is to evaluate the the quality of the state-of-the-art models in the social zone where most human-robot interactions take place and analyze different factors for the models' performances.

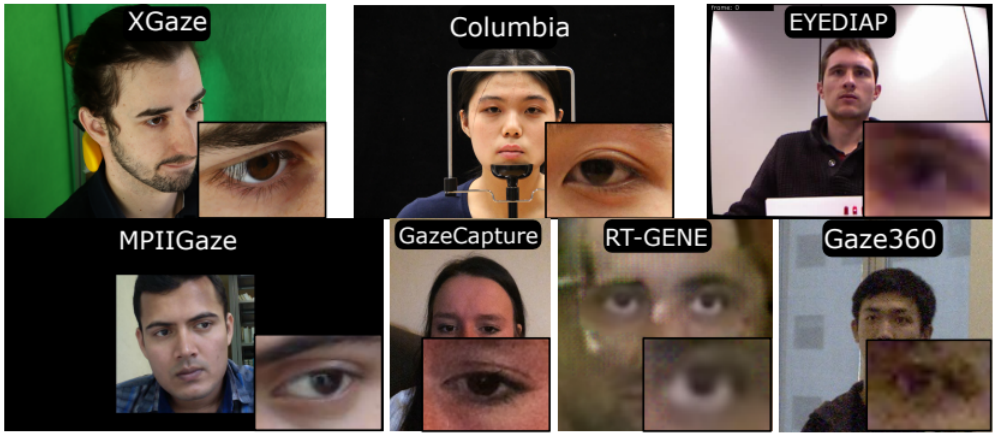


Figure 2.3: Samples from the main dataset for appearance-based gaze estimation currently available in the literature [48].

	MPIIGaze	EYEPIEAD	Gaze Capture	Gaze360	ETH-XGaze
# subjects	15	16	1474	238	110
# data	213 659	237 min	2 445 504	172 000	1 083 492
Max Head pose (Y, P)	$\pm 15°, 30°$	$\pm 15°, 30°$	$\pm 30°, 40°$	$\pm 90°$, unknown	$\pm 80°, 80°$
Max Gaze pose (Y, P)	$\pm 20°, \pm 20°$	$\pm 25°, 20°$	$\pm 20°, \pm 20°$	$\pm 140°, 50°$	$\pm 120°, \pm 70°$
Resolution	1280 x 720	HD & VGA	640 x 480	4096 x 3382	6000x 4000 (near to the camera)
Cons/ Uncons	U: laptop images	C:lab environment	U: phone/tablet images	U: Outdoor and Indoor	C: lab environment

Table 2.1: Quantitative comparison of the main datasets for gaze estimation

2.5.1 Gaze360

Gaze360 is a dataset published in the 2019 by researchers at MIT that is a real game-changer in the apperance based gaze estimation task. The reason for the usefulness of this dataset for the research is the ability of the authors to create a huge dataset that has together: high number of

subjects, high head and gaze ranges, pictures acquired in the wild, in a unconstrained environment. All these features together make Gaze360 stand out from the crowded group of datasets for gaze estimation. The characteristics presented above gave to Gaze360 some unique features incredibly useful for training deep learning models. One of the its most interesting capability is the one of generalizing well on other dataset. Its usefulness has been demonstrated through cross-dataset evaluation performed on other dataset [22]. Gaze estimation models have not yet reached such levels of performance mainly due to the lack of sufficiently large and diverse annotated training data for the task. Collecting precise and highly varied gaze data with ground truth, particularly outside of the lab, is a challenging task. Gaze360 is the first dataset that try to deal with highly varied gaze data, precisely collected outside the lab in unconstrained conditions. We also would like to highlight the big eye range in the dataset, in certain cases, correspond to gaze yaws of approximately ±140 (where the head pose is at 90 such that one eye remains visible, and that eye is a further 50 rotated).

Figure 2.4: Images samples from Gaze360 dataset with ground truth estimation [22].

2.5.2 ETH-XGaze

ETH-XGaze is a dataset published in 2020 from ETH Zurich [48]. The reason behind the publication of this dataset is mainly that existing gaze estimation datasets have limited head pose and gaze variations. ETH-XGaze consist over one million HD pictures of different gaze under extreme head poses. The participants of the experiment were 110 pictured with 18 cameras, adjustable illumination conditions and a system able to precisely record the ground truth of gaze direction. The dataset can significantly improve the robustness of gaze estimation methods across different head poses and gaze angles 2.5.

The images have been collected in a constrained lab environment trying to simulate extreme use cases: large variations in head poses, up to the limit of where both eyes are still visible (maximum ±70°) as well as gaze directions (maximum ±50° in the head coordinate system) that together with the head variation have a comprehensive variation higher that the ranges in Gaze360. Important to create a dataset able to train robust and general models are also the presence of viewpoint's variations combined with multiple lighting conditions, high input image resolutions, and the presence of occluders such as glasses. The majority of the dataset available in the literature to not consider this use-cases being limited to the frontal setting and thus lacking in ability to generalize.

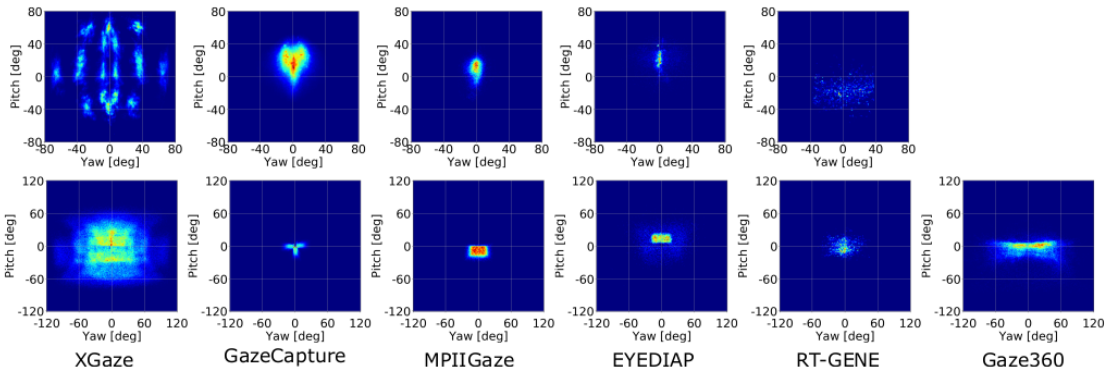


Figure 2.5: Head pose and gaze angles distribution in the main datasets available [48].

Figure 2.6: Samples from ETH-XGaze dataset [48].

2.6 Deep learning networks for gaze estimation

The most popular network for appearance-based gaze estimate are CNNs. Utilising image samples with actual gaze directions, the network is trained. A mapping function from unprocessed images to the human gaze is fundamentally learned in the gaze estimation problem. As a result, the deeper CNN architecture typically performs better, similar to computer vision tasks. Various CNN architectures have been proposed along the years for computer vision tasks, and some of them also show great success in gaze estimation task, e.g., LeNet, AlexNet, VGG, ResNet18 and ResNet50. The input of these models can be made of images or videos and can be single stream (face or eye pictures) or multiple stream (face and eye images). The most used loss functions in appearance based gaze estimation task are the mean square loss and the l2 loss although some experiment with pinball loss estimating gaze and error bounds together seems to improve performances especially in unconstrained environment [22]. In this work the focus is on the state

of the art models on Gaze360: L2CS [1]; and on ETH-XGaze [48]: the model is the one proposed in the paper, both of them use ResNet-50 as backbone.

2.6.1 L2CS-Net

In the paper proposed by A.A. Abdelrahman et al. [1] they present a set of new methods, from the losses to the network architecture, that reach the state of the art performance on Gaze360 dataset. First of all, the authors introduce a new way for computing the 3D gaze direction from images using a multiple losses approach. In particular, they propose to regress each gaze angle (yaw, pitch) separately, from the hypothesis that this technique could increase the accuracy of the models. Moreover, they used two identical loss for each gaze angle. Each loss consist in a combination of a cross entropy loss to predict binned gaze classification and mean squared error (l2 loss) that represent the regression component of the total loss. The gaze bin classification is performed thanks to a softmax layer together with the cross entropy loss allows the network to predict the neighborhood of the gaze angle in a more robustly. Based on the previous optimization scheme the authors presented the L2CS network. The network uses ResNet-50 as backbone to catch the spatial features of the face images and it predicts the gaze angles with a set of common convolutions layers and two separately Fully Connected layers. The predictions are used to compute the two losses and then both of them are back-propagated through the entire network In Fig. **??**

2.6.2 ETH-XGaze baseline

The model selected to test the performance of the ETH-XGaze dataset [48] is the network presented and trained by the author of the cited paper. The network uses ResNet-50 as backbone and it take as input full-face patch of 224 x 224 pixels and give as output the gaze angles: yaw and pitch. No better models trained on this dataset are today available in the literature.

2.7 Generalization problem in gaze estimation

Building accurate and general gaze estimation networks is not an easy task. A key challenge lies in making appearance-based models able to generalize among different people and environments different from the one on which the model has been trained. From its definition, this challenge can be considered a domain adaption problem, where the training set and test set represent the source and target domain. The test set (target domain) may contain unseen people, leading to a cross-person problem or unseen environments, resulting in a cross-dataset problem. Recently, fine-tuning has emerged as a prominent approach to address the domain adaptation problem in gaze estimation. Simple as effective it has shown good performance on both cross-person and cross-dataset evaluation.

2.7.1 Transfer learning

Deep neural networks are a powerful tool widely used nowadays for regression and classification tasks. However, there are drawbacks to using them, in particular, they require large datasets for training. Moreover, the models trying to achieve better performance are growing in size so necessitating even bigger datasets for training. These conditions give rise to some issues:

- The necessary amount of data for training a big deep neural network and achieve desirable performances are not always easy to get. For example, in very specific tasks, like object detection with special categories of objects or other computer vision tasks in niches or new problems, big datasets are usually not available.

- Training large deep neural networks on big datasets requires significant amount of time, energy, and resources.

To overcome these challenges, novel strategies have emerged that avoid training neural networks from scratch every time a new task arises. These approaches consist in reusing the information already learned by a model from a different dataset, leveraging the knowledge already acquired. This practice is known as transfer learning. A perfect example of transfer learning is our experiment, in which we employed two pre-trained models that were trained on datasets that present a distribution slightly different from the one on which they have been tested (our SocialAI dataset). These variations encompass different cameras, varying camera heights, and different distances from the subjects. Exploiting transfer learning, we were able to use pre-trained models exploiting their past knowledge on our task without the need of training them on our small dataset from scratch. We will later show that this strategy worked well in this situation, were obtaining a large labelled and task-specific dataset is an expensive task.

2.7.2 Domain-adaptation

Domain adaptation is a discipline close to machine learning and a subcategory of transfer learning, that deals with the problem of having a learning problem with a training set (source domain) and real deployment data (target dataset) that have instances sampled from two different but related distributions. The domain shift problem consists of a difference in the training dataset statistical distribution of a model and the data that the model will encounter during production. Nowadays it is common for machine learning algorithms to encounter this problem and several approaches have been experimented to deal with this problem. Let's define the domain shift in a thorough mathematical expression. Let X be the feature space and let Y be label space, and let f be a mapping function from the space X to the space Y trained from a set of samples (domain set) $D = (x, y) \in (XxY)_{i=0}^{m}$. The algorithm should be trained and tested on examples sampled from the same distribution D. In case of the domain shift instead the samples of the training set are sampled from a distribution D_s while the samples that the model will encounter in the future are sampled from another distribution D_t different but related to the other one. The are different types of domain adaptation strategies to deal with the domain shift problem:

- Supervised: In this approach, the target domain is labeled but typically it is limited in quantity. This approach assumes that there is sufficient similarity between the source and target domain;
- Unsupervised: Here, unlabeled data are available in the target domain, usually it is large. Various techniques are available such as adversarial domain adaptation or discrepancy-based methods;
- Semi-supervised: The target domain contains both unlabelled and labeled data, usually the dataset is large. The labeled data guide the model's adaptation process while the unlabeled data provides additional information used for domain alignment.

Each of these approaches tackles the domain shift problem in a different way, the choice usually resides in the type of data available for the task that we want to adapt.

2.7.3 Fine tuning

Fine-tuning is a transfer learning technique widely used in machine learning. With this technique, the weights of a pre-trained model are adapted to new data, allowing the neural network to learn a new task. During fine-tuning some of the layers of the model are kept "frozen" meaning that their weights are not updated during the backpropagation stage. This procedure allows us to

train just a part of the network, maintaining the information that we believe is useful also for the new task. In architecture like the convolutional neural networks (CNNs) the first layers capture lower-level features, like edges, basic structures, and textures. These low-level features are usually more general and common for different tasks making them transferrable for various domains and good candidates for being frozen during fine-tuning. On the other hand, the deeper layers of the networks capture the more specific and task-relevant features. For this reason, these layers are commonly trained during the fine-tuning phase making them able to adapt to the new target task. A common approach to increase the performance of pre-trained models on more general but robust datasets is to add new layers and train them from scratch on the new dataset keeping the other layers frozen as explained before. Another approach is to fine-tune the complete network. In this approach, any layer is kept frozen and all the pre-trained network is adapted to the new dataset, usually using a smaller learning rate. By fine-tuning the entire model all the layers adapt to the specific target task capturing its specific features. However, fine-tuning the entire model is more computationally expensive and can lead to overfitting. In this case, it is important to be aware of this problem and use techniques that can avoid over-specialization. In particular, it can be useful to use data augmentation, regularization, early stopping, or techniques like these which prevent overfitting and enhance the model generalization performances.

2.7.4 Regularization techniques: drop-out

The original paper introducing dropout was published in 2014 by N. Srivastava and other authors [41]. Dropout is a regularization technique used in deep learning to reduce overfitting and improve the generalization of models. It involves temporarily deactivating or "dropping out" a random subset of neurons in a neural network during training. Dropout provides a way to prevent the network from relying too heavily on specific neurons and encourages the learning of more robust and generalizable features. The key idea behind dropout is to introduce randomness and create an ensemble of multiple thinned networks within a single model. During each training iteration, a random subset of neurons is selected to be dropped out with a certain probability, typically set between 0.2 and 0.5. The dropped-out neurons are effectively ignored during that iteration, meaning their outputs are set to zero. As a result, the network becomes more resilient to relying on specific neurons or complex co-adaptations among neurons. By dropping out neurons, dropout forces the network to learn redundant representations and prevents overfitting. It also helps in preventing the network from memorizing noise or idiosyncrasies present in the training data. Furthermore, dropout acts as a form of regularization by implicitly adding noise to the network during training, which can improve its generalization ability. At test time, when making predictions, the entire network is used without dropout. However, the weights of the network are typically scaled down by the dropout probability to account for the increased number of active neurons during testing. This ensures that the predictions made at test time are comparable to the average predictions made during training. It's worth noting that while dropout is a powerful regularization technique, it may increase the training time since each training iteration works with a different thinned network. However, the regularization benefits often outweigh the additional computational cost, especially when dealing with complex models or limited training data.

Chapter 3

Methodology

3.1 Data collection

The present study endeavors to assess the efficacy of two models, trained on the most acclaimed datasets currently available in the scientific literature for appearance-based gaze estimation, namely Gaze360 and ETH-Xgaze dataset. To this end, we devised an experiment that entailed the collection of a limited dataset, resembling the customary scenarios encountered in human-robot interaction. The objective was to subject the models to scrutiny and evaluate their respective performances on this dataset. The experiment was carried out at the Vrije Universiteit located in Amsterdam, Netherlands. Prior to its commencement, the research proposal received ethical approval from the Research Ethics Review Committee of the Faculty of Science at Vrije Universiteit Amsterdam (BETHCIE), with the assigned approval code of 22-46. Moreover, all participants involved in the study provided informed consent after receiving comprehensive information about the study's purpose and procedures

3.1.1 Experimental Setup

The setup used for getting data is shown in Fig. 3.1. The experiment consists of a big screen, a Pepper robot with a built-in monocular camera, a 4K Pro webcam placed on its head, a keyboard, and nine fixed footprint placed on the ground to identify the position where the participant needs to stand. The screen used to show the dots is 150x88 cm (border included) and has been placed 155 cm from the ground (distance measured from the center of the screen). The screen is used to show the dots that the participant is required to gaze at. These dots appear in random order in 15 different fixed positions on the screen and they are organized in 3 rows and 5 columns. The vertical distance between the dots is 39cm while the horizontal one is 33.5cm. The Pepper robot is placed in front of the nine ground positions and just ahead of the screen. The monocular built-in camera in the head of Pepper captures pictures at a resolution of 640*480, the most used in real-time applications since it guarantees good speed performance. Additionally, a Logitech BRIO 4K Pro webcam with a resolution of 3840 * 2160 is put onto the robot head to take higher-quality pictures. A keyboard is provided to the participants to validate their gaze. The nine different human positions range from 1 meter to 3 meters of distance from the camera, covering the main area in the social zone [8] where most interaction takes place. Each of them is placed at a one-meter distance from the other: the first row is at 1 meter from the robot, the second at 2 meters, and the third at 3 meters. In every row, the right and the left position are 40cm from the center one (one small human step from the previous position). Finally, to keep the same illumination conditions (and to avoid reflection on the screen or the participant's glasses) the

curtains were closed during the whole experiment. The lab light was at the maximum level so that the faces and eyes were visible.

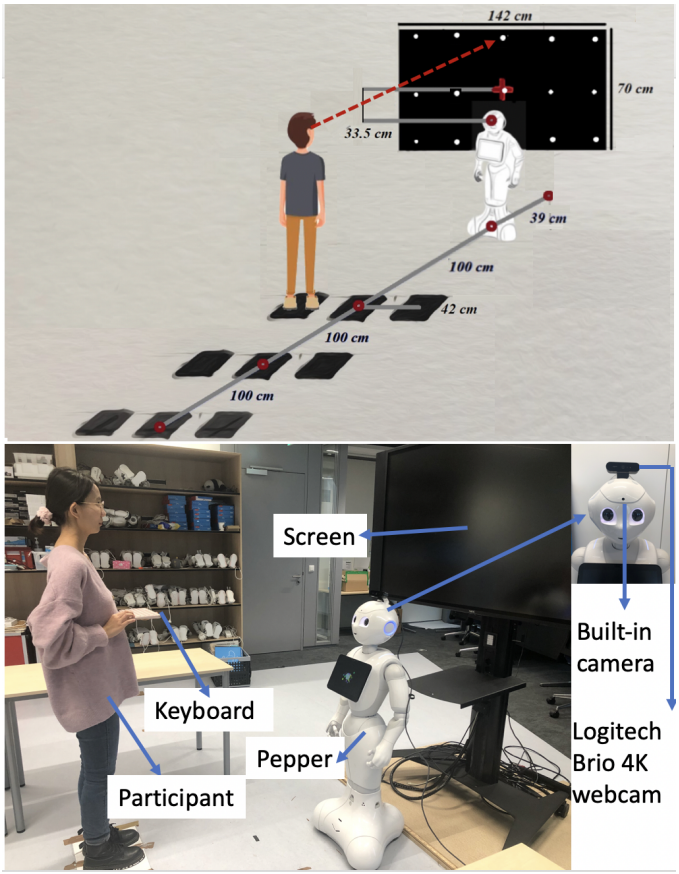


Figure 3.1: The picture shows a detailed visualization of the experiment setting. Above it is shown the digital representation of the experiment setup while in the bottom picture the physical experimental environment is presented

3.1.2 Participants

In total, twenty-one healthy adults (11 males and 10 females, mean height = $169.95 \pm 9.98cm$) aged between 21 and 35 years. 9 of them wore eyeglasses and 11 of them had normal vision without glasses or contact lenses during recording. The ethnicity of the participants includes Caucasian, Middle Eastern, and East Asian. Participants received a compensation of 10 euros for taking part in the experiment.

3.1.3 Experiment procedure

In the data collection experiment participants are required to sequentially gaze at various screen dots from the 9 different positions. For every position, 30 dots randomly appear on the screen in one of the 15 constrained positions allowed. The fixed ground footprint positions and the 15 constrained dot positions are designed to make the experiment repeatable and to make it feasible

to compute the ground truth of each participant's gaze. Each dot is followed by a validation procedure to check that the participant is correctly gazing at the dot shown on the screen. This procedure consists in showing after the dot an arrow directed on the left or the right and the participant is required to press the arrow key pointed in the same direction on the provided keyboard. The Pepper monocular camera and the Logitech 4k webcam, placed on the top of the robot standing in front of participants, capture their gaze during the experiment. Ten frames are recorded by each camera every time a dot appears on the screen. Ten among the twenty-one participants selected for the data collection have also been recorded by the high-quality camera while the others just with the built-in one. During the experiment, to keep all participants in the center of the camera view, the robot head changes pitch according to the participant's distance, at the end three different Pepper head pitches are used during the whole experiment (as shown in Fig.3.2). In order to make participants focus on the experiment, the robot reacts to participant's mistakes or correct validations by saying some words that either remind the participant when it needs to improve its performance (e.g. 'Oh, you miss one dot', 'You can do it better!') or congratulate him if he or she is concentrating on the screen (like 'good job!', 'well done!').

Figure 3.2: Three different views from the built-in monocular camera in the robot head according to different distances between humans and the Pepper robot

3.1.4 Social-AI dataset

The composition of our collected dataset is shown in Table 3.1. It shows the number of participants, image resolution and number of images in each group. The participants in Group 1 and 2 is same. 1350 is equal to the product of 9 (number of human positions), 15(number of fixed screen positions where dots are appear), and 10 (number of images per fixed screen position).

Camera	Participants	Resolution	Images	Valid Images
built-in	11	640*480	14850	14376
built-in	10	640*480	13500	13140
4K	10	3840*2160	13500	13045

Table 3.1: SocialAI dataset composition

3.2 Preprocessing

Before the analysis, a data cleaning step is necessary to remove unreliable data. In our experiment for unreliable data we mean all those images picturing a participant who is not looking at the dot displayed on the screen (because he or she moved the gaze direction from the target or for blinking). Blurred images are also dropped since they can be misleading for our analysis. The

first phase of the data cleaning procedure is to remove all those frames in which the keyboard validation procedure is missed by the participant. The frames associated with a wrong validation are removed since it is not sure if the participant is looking at the dot when they give the wrong reply on the keyboard. The second phase consist in checking the presence of outliers in the eth and l2cs prediction error using the Interquartile Range Method. We decided to look at the 3D error in the predictions, and in particular to the distribution of the gaze predictions of a given participant in a certain position looking at a dot. Once these predictions have been isolated we dropped just the frames whose predictions are outliers for both algorithms. All of this is done separately for the dataset subset containing low-quality images and the one with only 4k pictures since the performance of the models in the two subsets are considerably different.

Fig. 3.6 shows the difference in the L2CS predictions of participant 1 in the first three positions before and after the data cleaning. The bottom row shows the original prediction while the top one presents images cleaned after the outlier removal strategy. In the bottom-center and bottom-left plots the predictions considered outliers are shown. As we said before a frame is dropped if the prediction 3D error by both the models shows results that are far from the other predictions in the 10-frames set. In particular, in red are shown the predictions signed as outliers with both l2cs and ETH, predictions that make us drop those frames. In fact, looking at the top pictures those predictions have been deleted. In green instead is shown a prediction that for its distance from the ground truth is considered an outlier (with L2cs), but just with L2CS and not with eth, and thus it is not dropped.

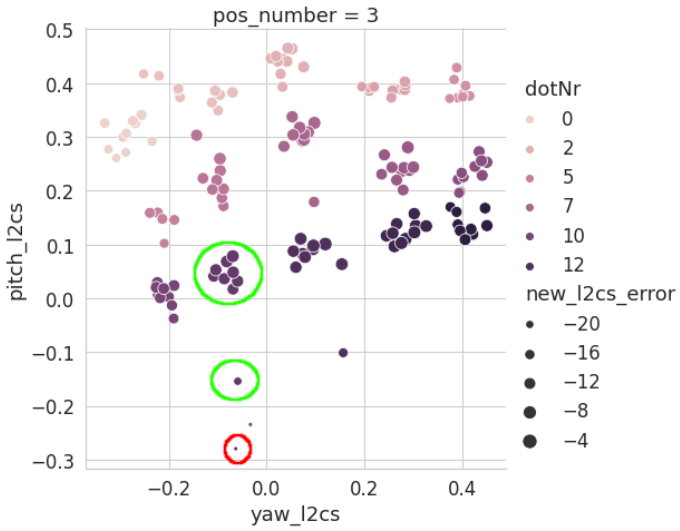
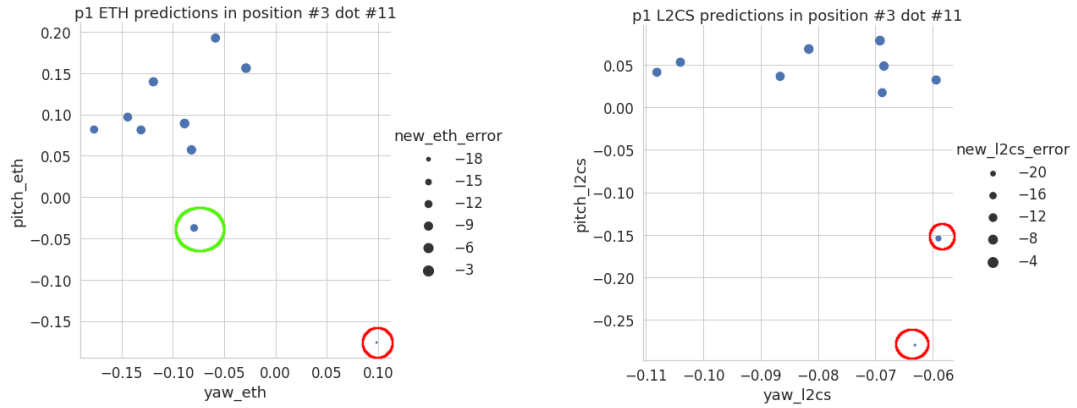


Figure 3.3: In the figure the predictions belonging to the gazes at dot 11 are highlighted, in red is shown the prediction that will be dropped

To better understand this approach in figure 3.4 is shown a macro of the predictions from ETH and L2CS of the dot 11 in the position 3 of the participant p1. In the left pictures we can clearly see that there is one prediction (red) that is an outlier with respect to the cluster. In the L2CS instead the predictions that represent an outliers are two. In this situation our method drop just the most the prediction that is red on both the plots (red circled dot in fig 3.3)

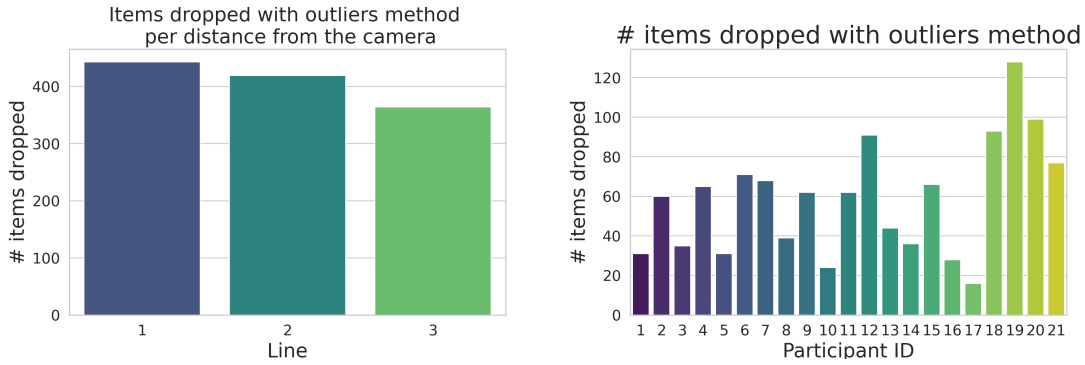
To understand the reasons for the outliers, we analyzed the data that was classified as outliers and discovered it is mostly caused by the three reasons including a very short glance at other places, blink, and blurry camera. Sometimes participants take a very short glance at other places but not the target or they are blinking or occasionally the high-quality camera is blurry as if it



(a) (b)

Figure 3.4: Predictions from ETH (left) and L2CS (right) of the dot 11 from participant p1 standing at position 3

is not properly focused on the subject, resulting in erroneous data. We notice that some models' predictions were missing and so was not possible to analyze the output of these frames. In this paper, we use the terms '3D error' and 2D error' to represent the accuracy of the estimated gazes. 3d error is defined as the angle between the vectors of the ground-truth gaze and the estimated gaze in three dimensions. Which only includes one value and the unit is degree.'2d error' is defined as the distance between the gaze point of the ground truth and that of the estimation on the 2D screen, including one value and the unit is a centimeter. Since the direct output of the ground truth and the estimation is only the 2D point and 3D vector respectively. Based on the locations of screen dots, camera, and human eyes, we use mathematical transformation to convert 2D points to 3D vectors on the ground-truth data and do the inverted conversion on the 3D estimated gaze directions.



(a) (b) cap

Figure 3.5: Count of the items dropped in the data cleaning procedure grouped per distance from the camera and per participant ID

The first picture shows the distribution of the l2cs predictions made by one participant on the first three positions of the experiment. The third and the second position show prediction

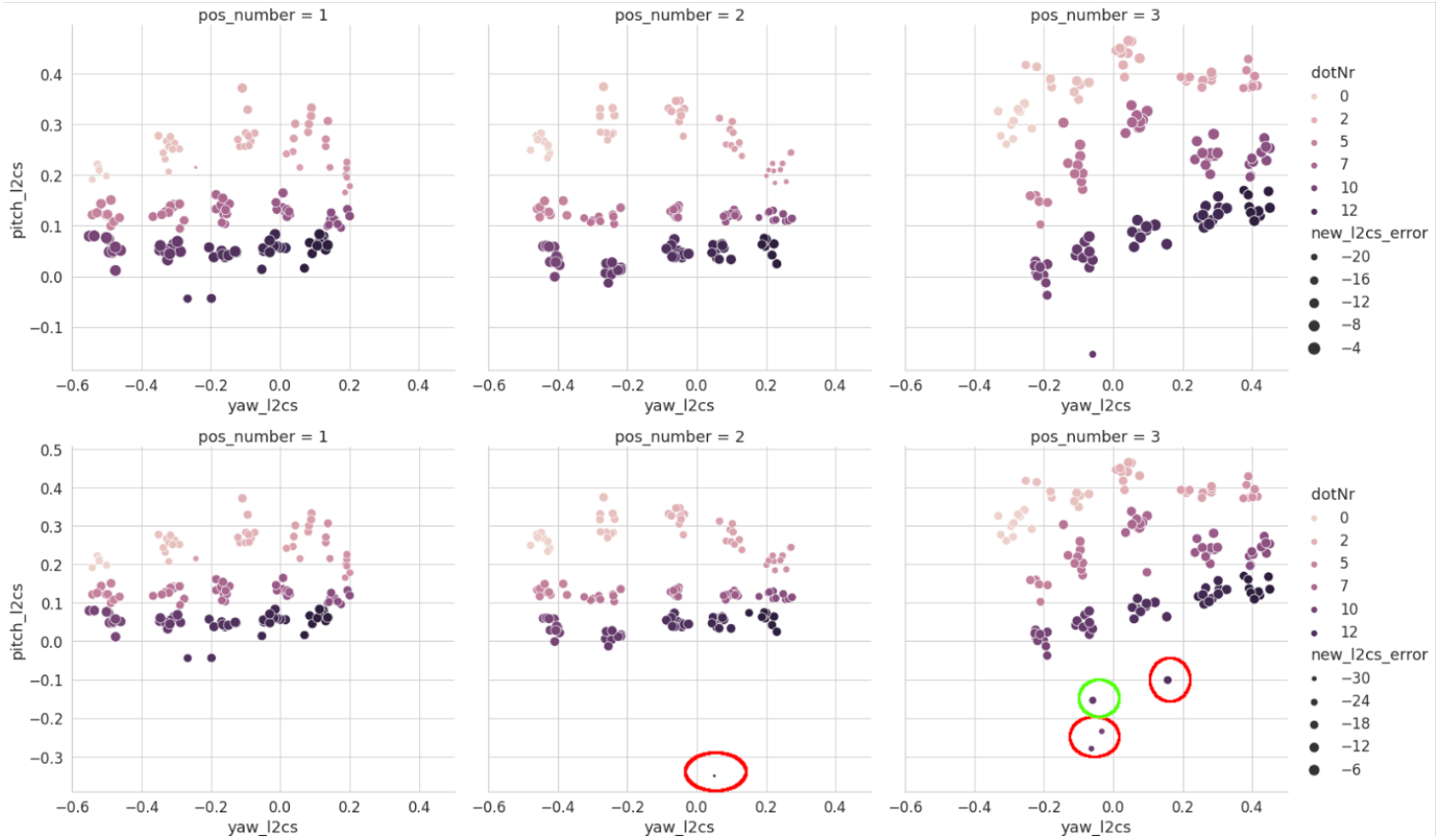


Figure 3.6: L2CS predictions of participant 1 in the first three positions before (lower line) and after (upper line) the data cleaning. The red dots are considered outliers while the green one is accepted

that are quite far from the other and they show an error considerably bigger than the one of the other predictions. These wrong predictions probably are the consequence of an error in the picture-capturing procedure. In the second series of plots the data after the outlier removing procedure are shown, and almost all the predictions with large errors and far from the cluster location have been deleted. Both the two series of pictures consider the prediction made by the L2CS model on the frames acquired by the monocular Pepper camera.

3.3 Ground truth computation

The Ground Truth of the images in the dataset where computed through manual measurement and annotation. During the data collection phase, the participant as we explained was required to gaze a screen dot from different standing positions, meanwhile two cameras were capturing all the action. We can define the following phases to extract from the experiment set up the ground truth of the gaze direction for each frame captured:

- Get the standing position of the participant, from 1 to 9;
- Read from the calibration file the dot number plotted when the picture that we want to annotate has been acquired;
- Measure the height of the participant;
- Use the screen coordinates of the dot, the vertical standing position and the height of the participant to establish correspondences between eye images and their respective calibration points;
- The gaze direction ground truth is computed in angular coordinates (yaw and pitch) using the participant's head coordinate system.

3.4 Finetuning

3.4.1 Motivation

As previously explained, the reason for the fine tuning in our case is related to the domain shift problem between the dataset on which the models have been trained and the data on which the models will be deployed that closely resembles the SocialAI dataset. With the introduction of Gaze360, the authors gave a huge contribution to the field allowing to create models with really good generalization capabilities. The dataset presented is way larger than the predecessor, presenting also various scenarios, and excellent annotation quality. In this thesis, our focus relies on a strictly constrained social setting consisting. The dataset is composed of frontal images captured from the Pepper robot at varying distances of one, two, and three meters. These images represent a realistic depiction of the perspective encountered in social interactions, making it ideal for analyzing and training gaze estimation models. In our specific situation, the fine-tuning procedure is conducted in order to generate models able to increase their performance in a social setting and be ready to be exploited in real-time experiments. We decided to proceed with the fine-tuning only on the L2CS model since it exhibited faster inference time compared to ETH-XGaze. We tested three different types of fine-tuning, which differ from each other for the layers updated during the procedure and the dataset used for the training.

3.4.2 Dataset composition

We trained the models using two different datasets: the first one contains only standard low-quality pictures while the second with both 4k images and low-quality pictures from the webcam. From this training set, a subset was extracted and employed as a validation set. For the test set instead, we decided to use only pictures from the Pepper monocular camera. 4K images are not used at test time since at deployment time (at least in our experiment) the model will predict gazes using its built-in monocular camera. Thus, using in the test set just images from the built-in camera we can have a better estimation of the real-time performance of the model. On the other hand, incorporating the 4k images in the training set can be beneficial to increase the dataset size. Furthermore, higher-quality images can potentially offer more informative cues than low-quality ones at training time allowing to generate more robust and accurate models.

The dataset for the training consist in three set: the training, the validation and the test set. The training set contains the data of the participant 1,2,4,5,7,8,9,14,15,18,19,20,21. the validation set instead constains the data of just two participant 12 and 13. The test instead contains the data of the participant 3,6,10,11,16,17. It's important to notice that the test set contains always only low quality images while training and validation set have both low and high. The idea behind is that the low-quality images are the one that the robot and the models will see at deploy time, since it will use the built in monocular camera of the robot.

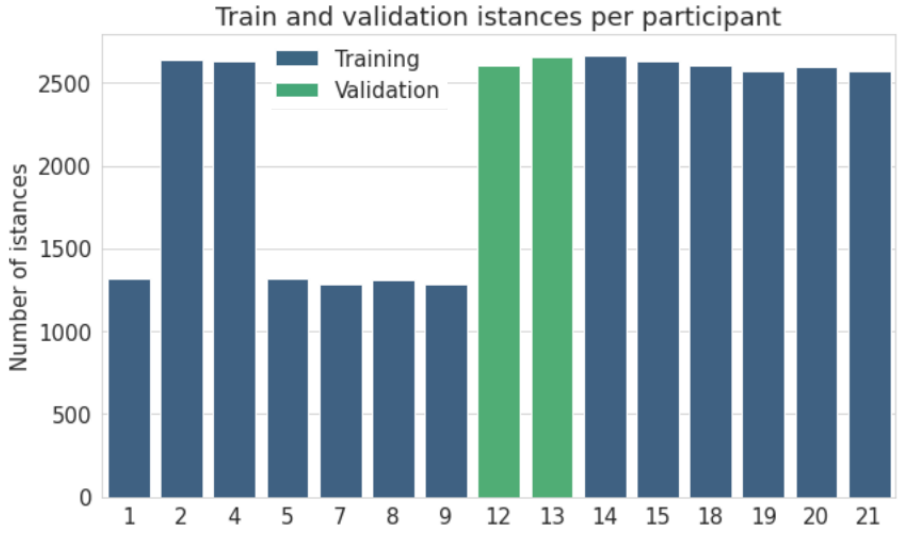


Figure 3.7: Barplot showing the number of instances used for training and for validation grouped per participant

3.4.3 Updates on the model

We tested two types of training for the L2CS model, these training procedures differ for the set of layers that are updated during the training:

- Freeze all the network and train only the last two fully connected layers;
- We freeze the first convolutional layer and we train the last two fully connected layers and four central convolutional layers;

Moreover, exploit drop-out layers as regularization techniques to preserve the generalization capability of the model. Particularly, we added four drop-out layers after each of the convolutional layer of the L2CS model as a way to avoid the network from overfitting.

Chapter 4

Results

In this section, we show the results of the data analysis using various data visualization tools. The chapter is divided into various sections where each of which is characterized by a set of questions that it tries to assess. Various aspects of the data are analyzed, always looking for insights that may help scientists to better understand the behavior of the gaze estimation deep learning models tested in our work. The social setting from which the data of our dataset are collected allows us to discuss how the models' performance is influenced by various features: for example the distance from the subject, the resolution of the camera, or the person-specific features. This is the core part of this work: here the effort made for the data collection is finally exploited for answering the key research questions stated at the beginning of this work. After the data analysis, the last part of this section is dedicated to the results obtained in the fine-tuning section. Now some small clarification about terms and practices that will be used in this section are needed. The words '4k' and 'no4k' are used to represent the built-in camera in the Pepper robot where a resolution of 640 x 480 is applied, and the Logitech BRIO 4K Pro webcam with a resolution of 3840 * 2160 respectively. While instead 'l2cs' and 'eth' are applied to express the L2cs model trained in the Gaze360 dataset and the baseline model trained in on the ETH-XGaze dataset.

4.1 Models person-specific resilience

In this first part of the analysis, we plotted the results achieved by the two models across different participants. The deep learning models used in gaze estimation usually are highly influenced by the facial features of the people (cross-person problem), in fact, it is common in real-time applications to perform person-specific calibrations. Our models are also subject to this problem and in the following pages we try to inspect how L2CS and ETH-XGaze deal with this problem.

Figure 4.1 shows the 3D error of the two models in the no4k dataset across all the twenty-one participants. The first aspect to notice is participant 21's performance with L2CS: it shows an L2CS 3D average error way higher than other participants' and also sensibly higher than the corresponding ETH-XGaze error on the same participant. This error represents a meaningless outlier (at least for the purpose of the analysis) that it's probably due to the quality of the images collected. After a careful inspection of the subject's pictures, we find out that the problem seems to be related to the thick glasses lenses worn by the participant thus making it difficult for the algorithms to precisely predict the gaze direction. Moreover, it is important to notice that the error is abnormal only with the L2CS predictions while ETH-XGaze performance is in a normal range. This result may have different interpretations but the one that seems more realistic for us is the one related to the robustness of the model, L2CS model is less robust in case of noisy data while ETH-XGaze is able to keep reliable performance. Each gaze estimation model relies on head

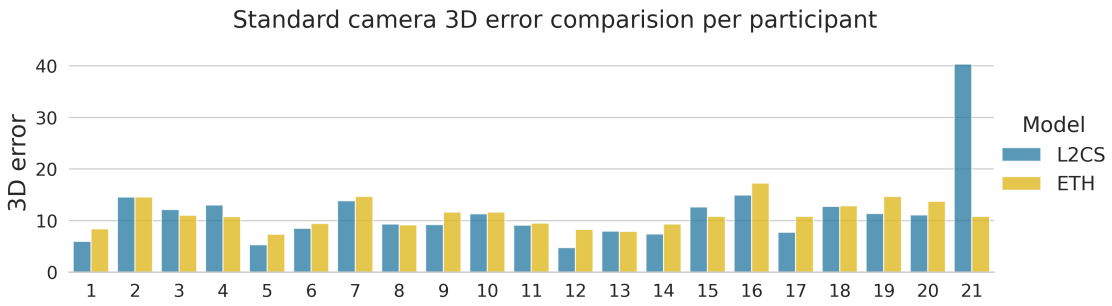
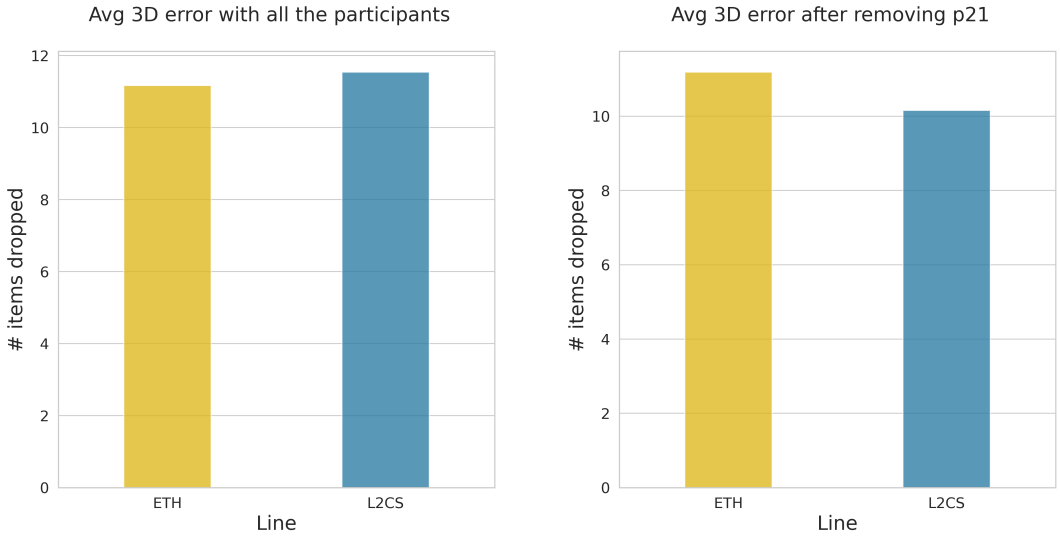


Figure 4.1: 3D error of L2CS and ETH-XGaze in the no4k dataset across all the twenty-one participants

orientation and eye features on different degrees and with a ratio that we do not know, a strong hypothesis is that probably L2CS is more reliant on eye features than ETH-XGaze which instead relies more on head orientation. This is just a hypothesis that could be interesting to validate with precise and targeted quantitative experiments in future works. We classify this result, on the 21st participant by L2CS as unreliable, and we drop participant 21 pictures in all the next analysis on L2CS.



(a) (b)

Figure 4.2: Average 3D error by L2CS and ETH-XGaze on the low-quality dataset before and after participant n21 cleaning

Figure 4.2 show the difference in 3D average error between the L2CS and ETH-XGaze on low-quality images with and without participant 21. L2CS is overall better than ETH-XGaze after the participant 21 pictures removal, highlighting how its exceptionally bad performance highly affects the overall results. In further analysis, the participant's 21 low-quality pictures are not considered and we will proceed to analyse the no4k dataset with just 20 participants.

The 4k camera pictures allow us to create a dataset that is smaller than the no4k one since

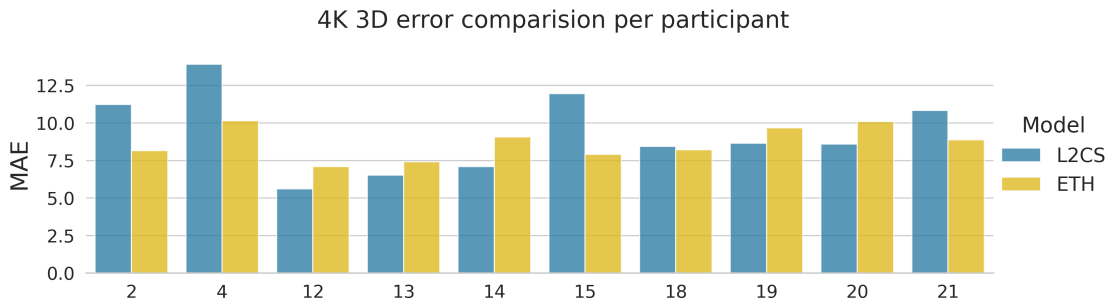


Figure 4.3: 3D error of L2CS and ETH-XGaze in the 4k dataset across all the twenty-one participants

it has been collected only for 10 participants. Also in this case we plotted the 3D average error across different participants as shown in figure 4.3. In this case, no abnormal values have been found and both models are quite stable across all the 10 participants. It is worth noticing that in this case participant 21 does not present abnormal values with L2CS and neither with ETH-XGaze meaning that the 4k camera is able to capture less noisy frames making L2CS predictions more precise. This is an important case in which we can understand how a higher quality camera could make less noisy images making the models' prediction more reliable.

4.2 Effect of images resolution on performance

In this paragraph, we explore the models' prediction accuracy on the two SocialAI datasets collected; one is composed of pictures captured with the Pepper monocular camera while the other one it's made of 4k pictures from the Logitech webcam. The low-resolution dataset includes 20 participants (after filtering out p21) and 26250 pictures, while in the high resolution, we took 4k images including 10 participants and 13045 pictures. In this section a couple of multiple bar plots are shown in order to answer some key questions: is the accuracy of L2CS and ETH-XGaze affected by the image resolution? How much does the accuracy of each of the Gaze estimation models change when we pass from a dataset with 640 x 480 pictures to one with 4k images? Which is the average variance of the single model's prediction errors on single participants? Better, how much the model is stable in the inner participant's predictions? Does this change increase the resolution of the images? In the figures below the 3D error standard deviation and average computed for each participant are used to draw model-specific bar plots. The data are aggregated by participants so each bar plot is printed using 20 and 10 values respectively for the low-quality dataset and the high-quality one.

In figures 4.4 and 4.5 we compare the performance of every single model on the two datasets, inspecting how much each of the two models is influenced by the camera resolution. In this case, the dataset used for the analysis contains only the 10 participants who were captured by both cameras, with the exception that p21 has been removed from the no4k dataset (11875 frames for low after the p21 removal, and 13045 frames for high resolution). Figure 4.5 shows 3D absolute error and standard deviation for L2CS with the two dataset partitions. In the no4k case the mean 3D error = 10.6° and mean std = 4.7° while for the 4k dataset Mean 3D error = 9.3° and Mean std = 5.7°. Figure 4.4 instead is related to eth and present in the no4k case Mean 3D error = 11.4°, Mean std = 5.8° while in the 4k dataset Mean 3D error = 8.7°, Mean std = 4.5°. Gaze estimation accuracy significantly increases for L2CS with the high-resolution camera ($t(9) = 3.1$, $p = .01$) but not precision that instead is higher with the low-quality pictures. For eth, both

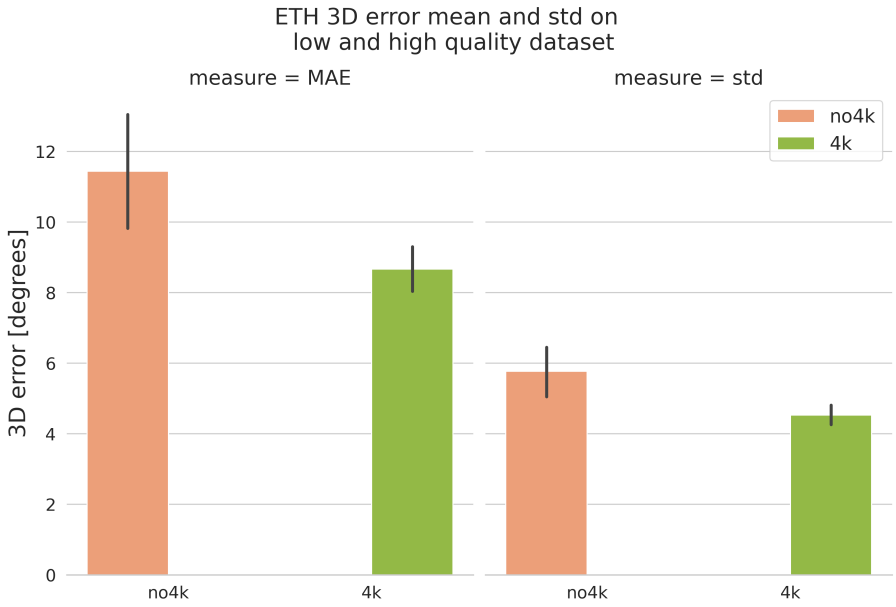


Figure 4.4: Average 3D error and standard deviation of L2CS predictions on the low quality and the 4K dataset

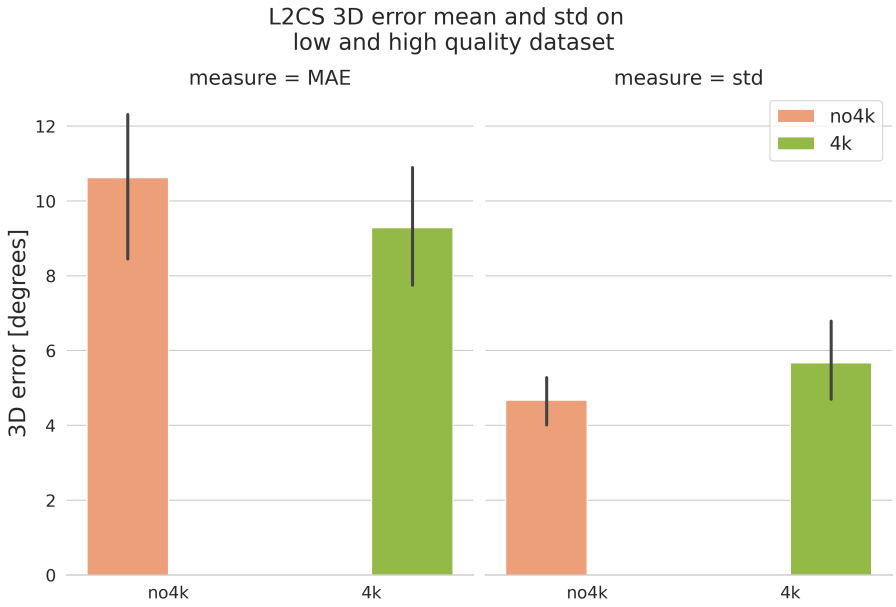


Figure 4.5: Average 3D error and standard deviation of L2CS predictions on the low quality and the 4K dataset

accuracy ($t(9) = 3.3$, $p = .01$) and precision ($t(9) = 3.7$, $p = .005$) are significantly better when we use high-quality pictures. Moreover, it's worth noticing that eth with the 4k images, not only perform better in terms of 3D average error with respect to the non4k but it also has a smaller

difference across the various participant as we can see in the green bar on the left side of 4.4. The variance of the prediction error on the different participants is smaller with the 4k pictures showing that higher-quality images make the models more robust. On the other hand, this does not happen with L2CS, that although has a lower average error with 4k images, it does not have a lower variance between participants (as we can see from the vertical lines on top of the bars in the left part of the 4.5).

4.3 Performance comparison: L2CS vs ETH-XGaze

In this section, we try to directly compare the two models on the two datasets generated. Which model has the lower error on the non-4k dataset and which one is more indicated for 4k applications? Also in this case we helped ourselves in answering these questions using multiple bar plots that perfectly fit with our visualization goals. The first graphs show the average 3D error of L2CS and ETH-XGaze and the average standard deviation on single participants one next to the other and separately for each dataset.

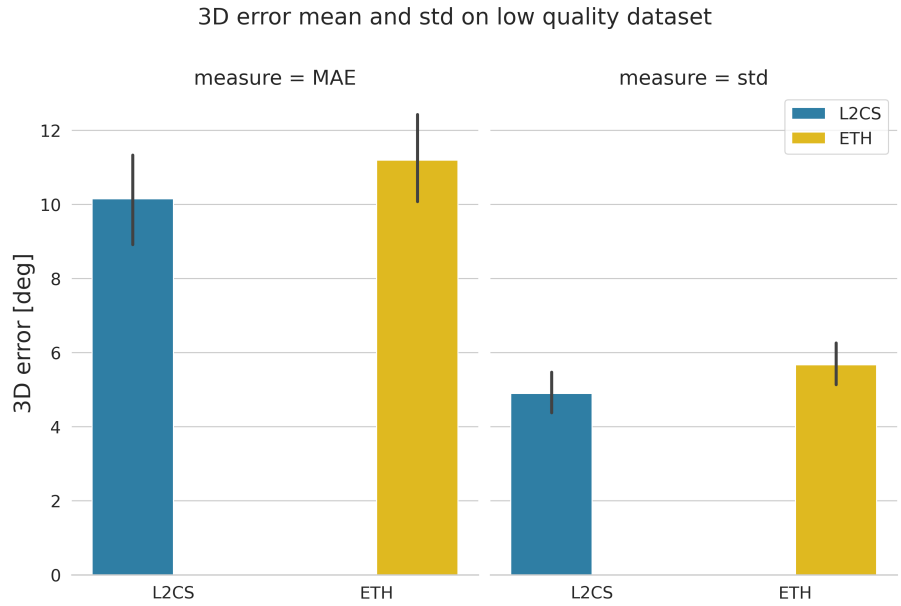


Figure 4.6: A comparison of the performance (in terms of 3D error and standard deviation) of the two models on the non4k dataset

Figure 4.6 shows that l2cs overall performances are better than the eth ones in the low-quality dataset, also presenting a better precision with a lower standard deviation. L2CS shows a mean 3D Error = 10.2° and a mean std = 4.9° while ETH presents mean 3D error = 11.2° and mean std = 5.7°. To check the significance of these results we perform a paired t-test on the values showing that L2CS performs significantly better than ETH for low-resolution images on both the metrics taken into consideration: accuracy with $t(19) = 2.9$ and p-value = 0.01 and precision with $t(19) = 4.03$ and p-value < 0.001. Symmetric to the previous picture figure 4.7 shows the 3D error and std distribution per participant with different models but the dataset is the high-resolution one. In this case, we found no significant performance differences between L2CS and ETH which may be due to the lower number (10) of participants we had for this condition: L2CS (Mean 3D error = 9.3°, Mean std = 5.7°) while ETH (Mean 3D error = 8.7°, Mean std = 4.5°). Another interesting

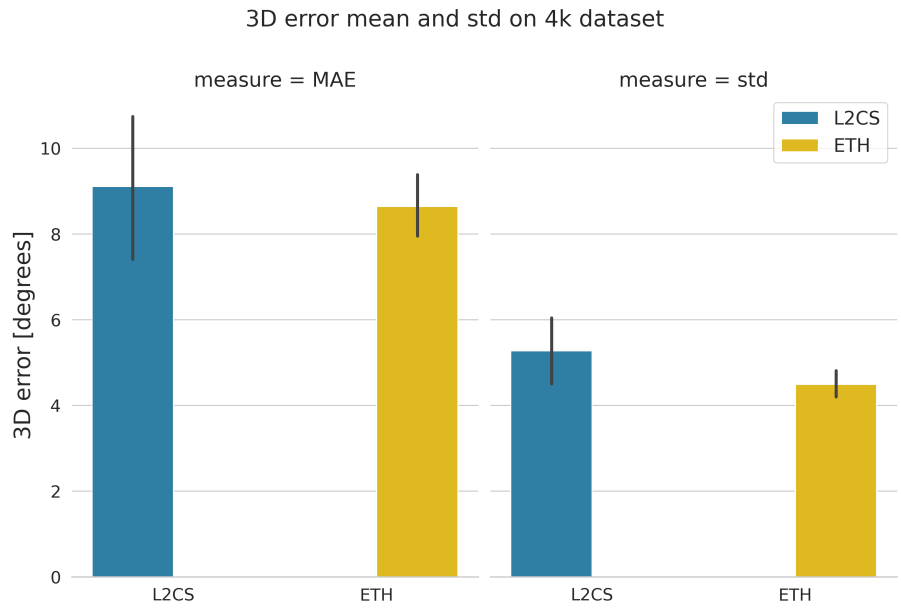


Figure 4.7: A comparison of the performance (in terms of 3D error and standard deviation) of the two models on the 4k dataset

result is visible from the 4k dataset plot. In particular, although the difference between 3d mean errors is not informative, we can see that the variance of the 3d errors across participant decrease from l2cs to ETH as it is shown in vertical bars on the left side of figure 4.7 (smaller on top of the eth bar).

4.4 Distance influence on performance

4.4.1 Horizontal distance

In our setup, the vertical distance between cameras and humans ranged from 1 to 3 meters, with a relatively small horizontal distance from the line perpendicular to the robot. This setting roughly matches the social zone where most interaction between humans and robots takes place [18]. The 9 positions we used were evenly distributed in this zone in terms of horizontal and vertical directions. In this paragraph, we analyze the effect of distance on the error performed by the models separately for each of the two datasets collected during the experiment. Before exploring how distance affects the performance of models, the Repeated measures ANOVA was employed to study the effect on the estimation performance of the three horizontal positions at the same distance (central, left side, or right side of the robots). Our results indicate that the standing positions at the same distance from the cameras have no significant effect on the estimation performance of both ETH-XGaze and L2CS (in both cases p-values > 0.5).

In figure 4.8 we can see a line plot showing the performance of ETH-XGaze and L2CS for different datasets in each of the nine standing positions. In this first analysis, we want to let the reader notice how for each line the 3D error change according to the position to avoid making comparisons across the models since this aspect will be discussed later. In particular, it's worth noticing how the 3D error is quite stable in the three consequent positions that are on the same line, at the same distance from the camera. The plot lets us understand visually the result of the

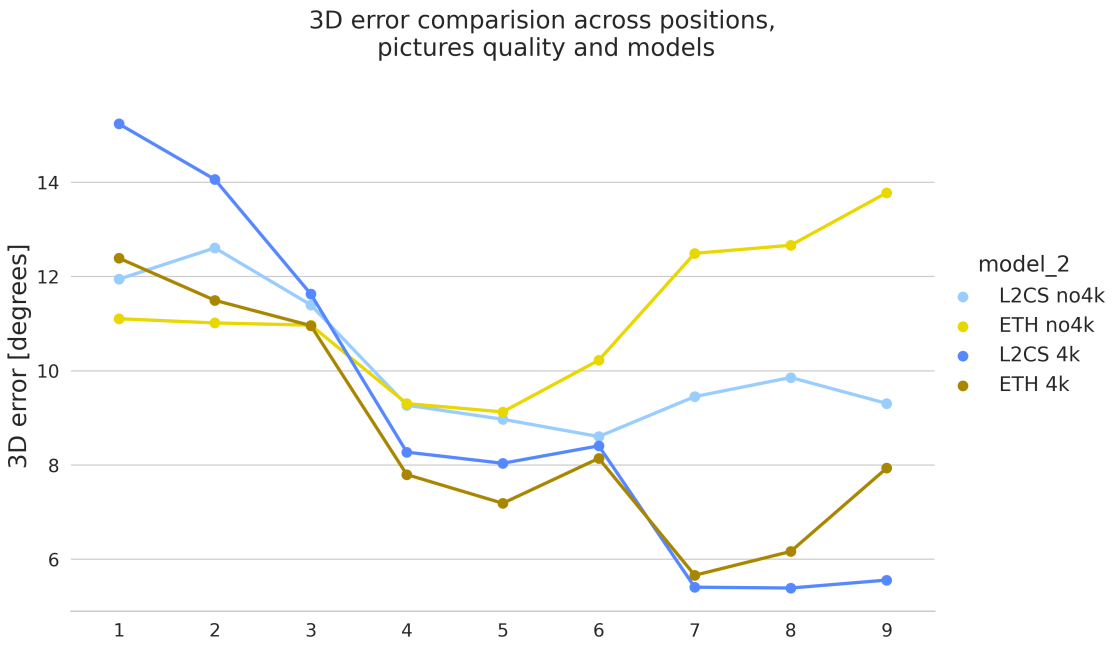


Figure 4.8: Line plot showing the performance of ETH-XGaze and L2CS for different datasets in each of the nine standing positions

ANOVA test described above that discards the hypothesis of the influence of different positions on the same line on the 3D error. The next analysis will consider this result and will merge the data, reducing the granularity to the three distances from the camera.

4.4.2 Three way ANOVA test: distance, model and resolution

To study the effect of the distance on the performance of the models we computed the average error and standard deviation of the L2CS and ETH model for the three distances. The results are shown in figure 4.9. We investigated the effect of three factors model (L2CS, ETH), resolution (4k and no4k), and distance (1m,2m,3m) - on the 3D error of gaze estimation. The data were analyzed using a three-way ANOVA. The result shows a significant effect of distance ($F(2, 108) = 18.1$, $p < .001$) and resolution ($F(1, 108) = 11.3$, $p < .001$) and significant interaction between resolution and distance ($F(2, 108) = 9.1$, $p < .001$). Further, in the left part of fig. 4.9, we can see that for the same resolution (4k and no4k) L2CS is better than ETH at a distance of 3 meters. The relationship is significant at 3 meters (under no4k, $p = .03$; under 4k, $p = .04$) but there is no significant difference at 2 meters (under no4k, $p = .05$; under 4k, $p > .1$). We also found that 4k has a significant positive effect on the accuracy of the L2CS model ($p = .01$) and ETH model ($p = .01$) at 2 and 3 meters, but has a significant negative effect on the accuracy of L2CS at 1 meter ($p = .03$).

Similarly, we use three-way ANOVA to study the effect of three factors: model (L2CS, ETH), resolution (4k and no4k), and distance (1m,2m,3m); on the precision (standard deviation) of gaze estimation 4.9 right side. We found significant effects of the model ($F(1, 108) = 16.3$, $p < .001$), and resolution ($F(1, 108)$, $p < .001$) but not of distance. We discovered that L2CS has a significantly higher precision compared to ETH under the same resolution at 2 meters (no4k, $p = .01$; 4k, $p = .02$) and 3 meters (no4k, $p = .03$; 4k, $p = .001$). 4k can significantly reduce the gaze

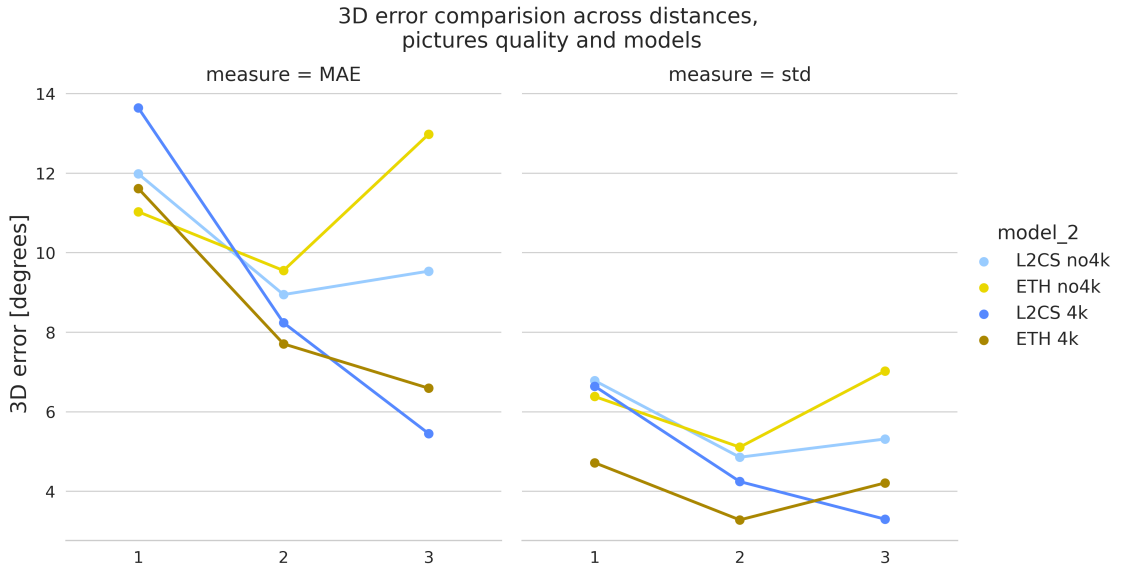


Figure 4.9: Effect of distance, picture quality and models on 3D error and standard deviation with emphasis on the variation in the three distances from the subject

dispersion for ETH at every distance (1m, $p < .001$; 2m, $p < .001$; 3m, $p < .005$). Moreover, it has a positive effect on the precision of L2CS at long distances (2m, $p = .02$; 3m, $p = .01$).

4.4.3 L2CS and ETH-XGaze distance-related performance

Figures 4.12, 4.10, 4.13, 4.11 show a more detailed visualization of the performance of the two models at different distances from the camera. The comparison is made separately for each dataset and we decided to show not only the results of the 3D cameras but also the 2D error on the screen. The 2D error is the error made in the prediction by the models computed after projecting it onto a plane and measured in centimeters. In particular, using the distance from the screen (where the dot appears), the height of the participant, and the coordinates of the ground truth, we were able to compute the projection of the predicted gaze directly onto the screen used for plotting the data and measure the distance from the ground truth. This metric is useful because it is easier to read than the 3D error that, although being the metric more used in the literature, show some problems when we try to make comparisons at different distances. In fact, in the standard camera graphs with the 3D error we can see how the mean error remains more or less stable in all the three positions. Instead, in the plot of the 3D error on the 4k camera the error keeps decreasing going further away from the camera. This behavior is explained by the fact that the yaw and pitch range that the models need to predict become more narrow going further away from the camera. This is due to the fact that the distance increases but the position of the dots on the screen is always the same. Furthermore, since the error used is an absolute error it decreases only because the angle that the models are asked to predict decreases too. This is the reason for a plot that uses the 2D error on the screen. Expressing the error in centimeter allow a better understanding of the real performance of the models when the distance change.

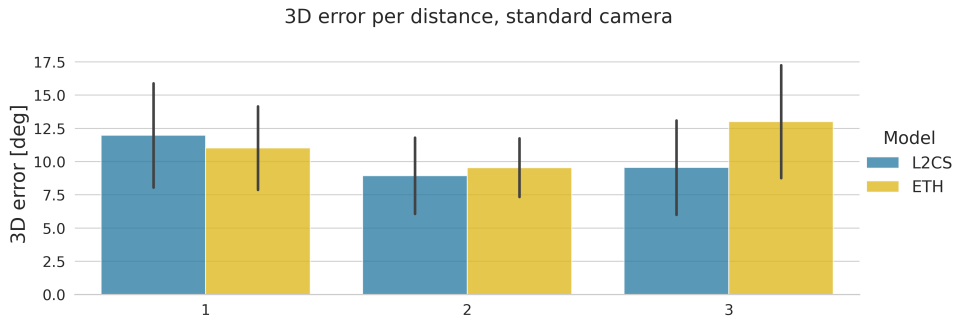


Figure 4.10: This first barplot shows a comparison of the 3D error of L2CS and ETH-XGaze made at a different distance from the camera (1,2 and 3 meters). This plot show the data of prediction made on the non4k dataset

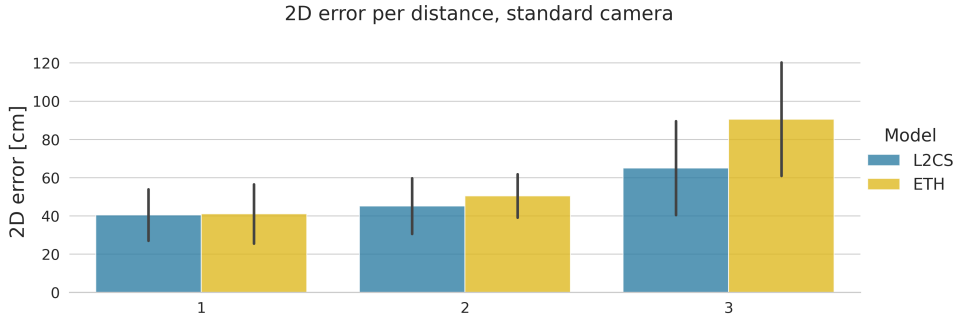


Figure 4.11: This barplot shows a comparison of the 2D error of L2CS and ETH-XGaze made at a different distance from the camera (1,2 and 3 meters). The plot shows the data of prediction made on the non4k dataset

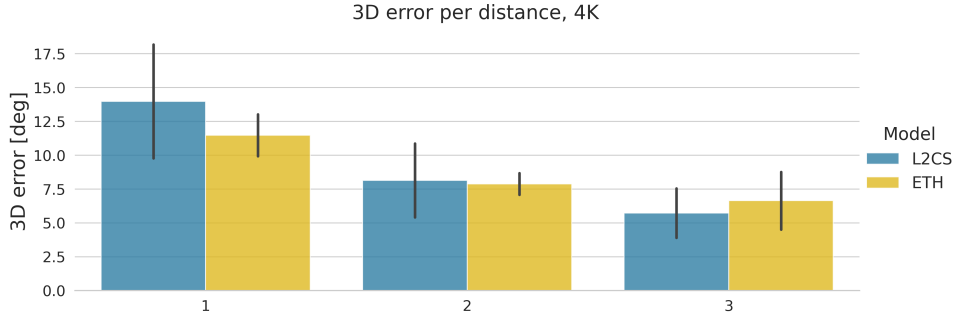


Figure 4.12: This barplot shows a comparison of the 2D error of L2CS and ETH-XGaze made at a different distance from the camera (1,2 and 3 meters). The plot shows the data of prediction made on the 4k dataset

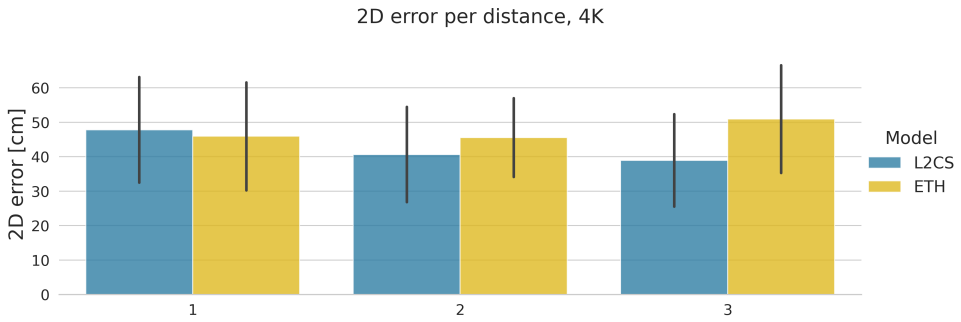


Figure 4.13: This barplot shows a comparison of the 2D error of L2CS and ETH-XGaze made at a different distance from the camera (1,2 and 3 meters). The plot shows the data of prediction made on the 4k dataset

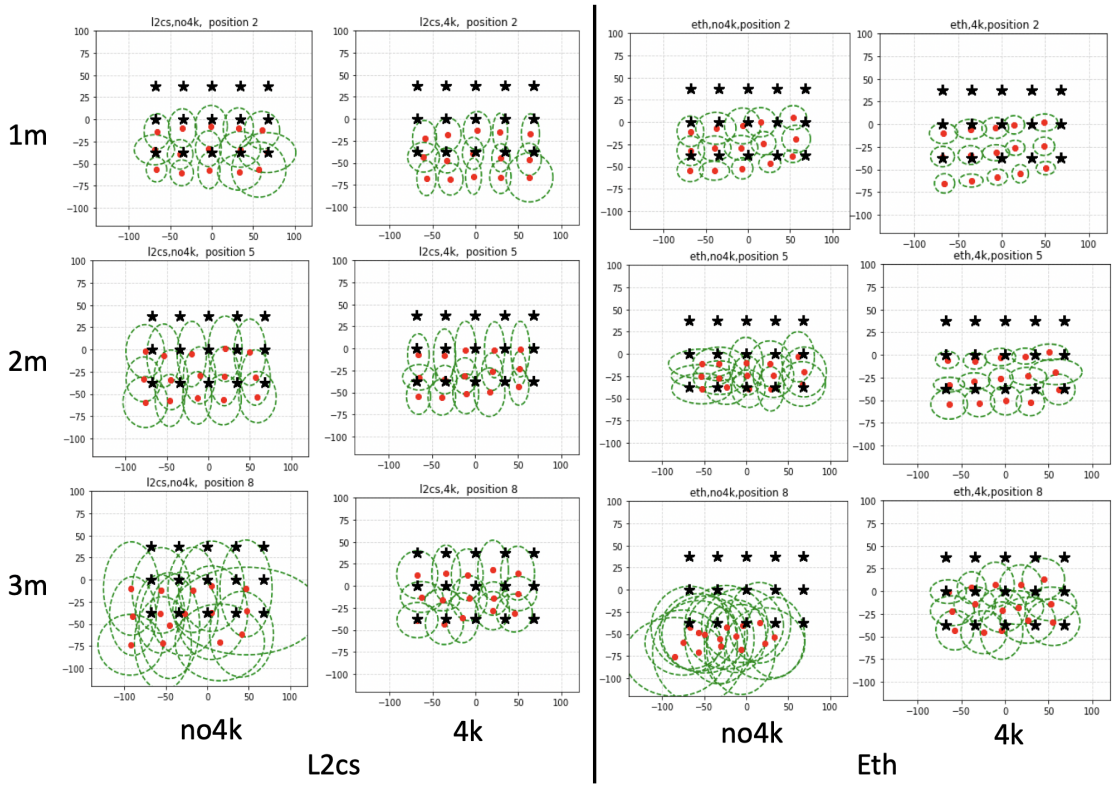


Figure 4.14: Gaze estimations from ETH and L2CS on a 2D plane. The black stars represent the ground truth gaze points, while the red dots stand for the average predictions. The green ellipses indicate the standard deviation of the estimate

To better visualize the models' performance and verify the above 3D result, we also plotted gaze estimations on a 2D plane (see Fig. 4.14 for the center positions at three different distances). The mean error and standard deviation are shown for each dot. The black stars represent the ground truth gaze points, while the red dots stand for the average values of the corresponding

gaze directions estimated by the models. The green ellipses indicate the standard deviation of the estimate. The shape is elliptical because the standard deviations are different in the vertical and horizontal directions. This figure shows that (i) L2CS is better than ETH model, especially at 2 m and 3 m. (ii) High camera resolution reduces the standard deviation of gaze estimation on the two models, especially for the ETH model. Also, it improves the accuracy of the models' estimation, especially at 3 meters. These are consistent with the 3D results (see 4.9). Additionally, we found that L2CS model has a higher precision along X-axis than the ETH at longer distances for both low- (2m, $p < .005$; 3m, $p < .005$) and high-resolution cameras (3m, $p = .02$). On the other hand, the ETH model has a higher precision along the y-axis at 1 meter distance (no4k, $p = .04$; 4k, $p = .03$).

4.5 Performance on horizontal and vertical directions

The dataset used for the training of gaze estimation models differs in the distribution of yaw and pitch angles. This feature led to the model's capability to accurately predict large gaze angles thus making the model more or less performative in wild and general environments. As presented in section 2.6 also the dataset used in this work, Gaze360 and ETH-XGaze, differs for gaze yaw and pitch distribution due to a different image collection method. In this section, we inspect through the use of multiple graphs the behavior of L2CS and ETH-XGaze on pitch and yaw variation. The goal is to increase the knowledge about the behaviors of the two models (and the effect of the relative datasets) on pitch and yaw variation present in our SocialAI dataset.

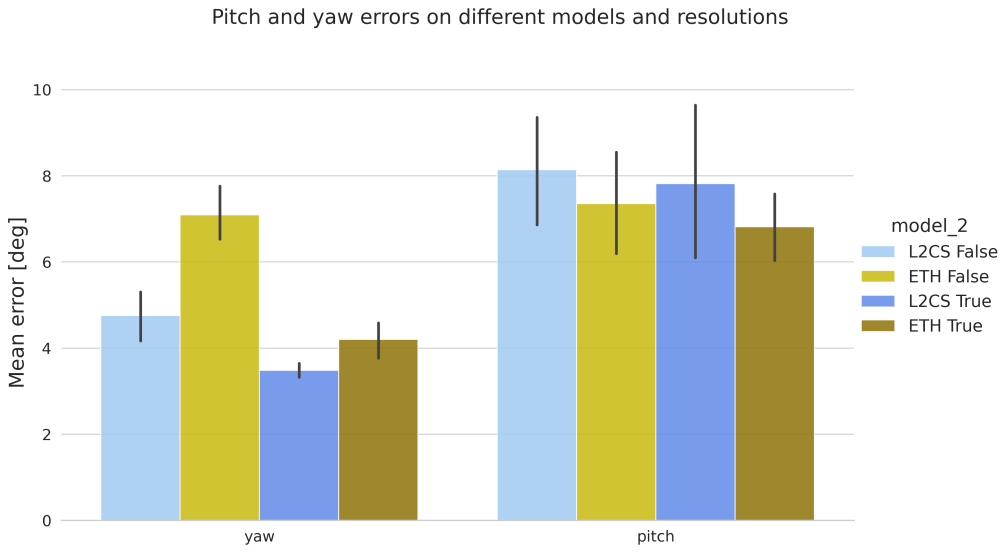


Figure 4.15: Bar plot showing respectively on the left and on the right side the comparison between yaw error and pitch error by L2CS and ETH-XGaze on 4k and non4k Social-AI dataset

4.5.1 3D error decomposition: yaw and pitch

In this subsection, we compare the performance of the models on pitch and yaw error. We try to better understand the strengths and weaknesses of the two models by inspecting their capability of prediction in the horizontal and vertical directions. We computed the pitch and yaw error for

each model on both the standard and the 4k Social-AI dataset. In fig 4.15 we plot the results of the previous computation through a bar plot showing respectively on the left and on the right side the comparison between yaw error and pitch error by L2CS and ETH-XGaze on 4k and non4k Social-AI dataset. From the plot, we can extract different information. First of all, we can clearly see how after separating the 3D error in its two components, the vertical (pitch) and the horizontal (yaw), the best model is different according to the metric we are looking at. In fact, ETH-XGaze has a lower pitch error on both 4k and non4k datasets while L2CS has a better yaw error. Another interesting finding is that ETH-XGaze has a far larger yaw error in the baseline Social-AI dataset than L2CS. The yaw error difference between the models gets smaller for the 4k dataset. Looking at the pitch instead although ETH-XGaze has a better performance the difference between the performance of the model from ETH and the one from MIT is small.

4.5.2 Effect of yaw and pitch angles on 3D error

In this section, we inspect the 3D prediction error according to changes in the horizontal and vertical range that the models will predict. The reason is that we want to check possible dependencies between the horizontal and vertical range of prediction and the prediction error. In the following pages are shown 4 scatter plot that shows the 3D error against pitch and yaw ranges for ETH-XGaze and L2CS. The yaw and pitch ranges on which the model are subjected are the one of our Social-AI dataset: yaw ranges from $-30°$ to $+30°$, while pitch ranges from $0°$ to $+35°$. Using a regression line we can see from the scatter plots the tendency that the error shows in all four cases. In particular, we can clearly see how the 3D error of the prediction increase when yaw and pitch angles increase. More interesting is the difference that the models show in these common behaviors: in fact, as we can see from the plots ETH-XGaze looks more robust to increase the angles to predict. The regression line of L2CS in the graphs of 3D error against pitch angle shows a higher slope with respect to the model from ETH. This result is repeated also in the graphs plotting 3D angle against yaw angle. Here instead the parable The first one, figure 4.16 shows the variation of the error based on the pitch. The picture shows a scatter plot with an estimated regression line that shows an increase in the error while the pitch to be predicted increases. The error goes from an average close to $9°$ degree at a pitch close to $0°$ to a maximum average of $15°$ at almost $35°$. The symmetric plot for L2CS is shown in figure 4.17. The ranges of pitch angles shown in the plot are exactly the same since the data used for deriving the plots are the same. Also in this case the scatter plot shows a increase in the error prediction when the input pitch increase. The error starts at around $6.5°$ at input pitch $= 0°$ and goes up to $18° \pm 2°$ at almost $35°$ of input pitch. The model behaves better than ETH-XGaze in the first part of the plot when the input pitch is low, while it tends to worsen its prediction when the input pitch increase. The regression line in the case of the ETH-XGaze error-pitch graph shows a trend that is linear against instead a regression line in the L2CS case that instead tends to grow in a faster way. Below we also show the scatter plot showing the 3D error of the models at different yaw variations. The first picture shows the 3D errors of ETH-XGaze variation according to the yaw angle of the gaze direction to predict. The error symmetrically goes from a value of approximately 12 degrees at the highest yaw range ± 30 down to a value of 10 degrees when instead the input yaw is close to 0 degrees. The regression line shows the average of the 3D error across all the yaw variations, it creates a parabola increasing at the highest ranges and reducing in the center of the graph when the range is lower. For L2CS the same scatter plot with the error against the yaw angles figure 4.19. For L2CS the plot shows the same behavior of ETH-XGaze with an average 3D error plotted in the red regression line that creates a parabola on the graph. Here the average error goes from 15 degrees at the greatest absolute yaw degrees (± 30) down to less than 10 degrees in the central part of the graph when the yaw angles are small. Also in this case the variation of the 3D error of eth along all the yaw variations is smaller than the one of l2cs.

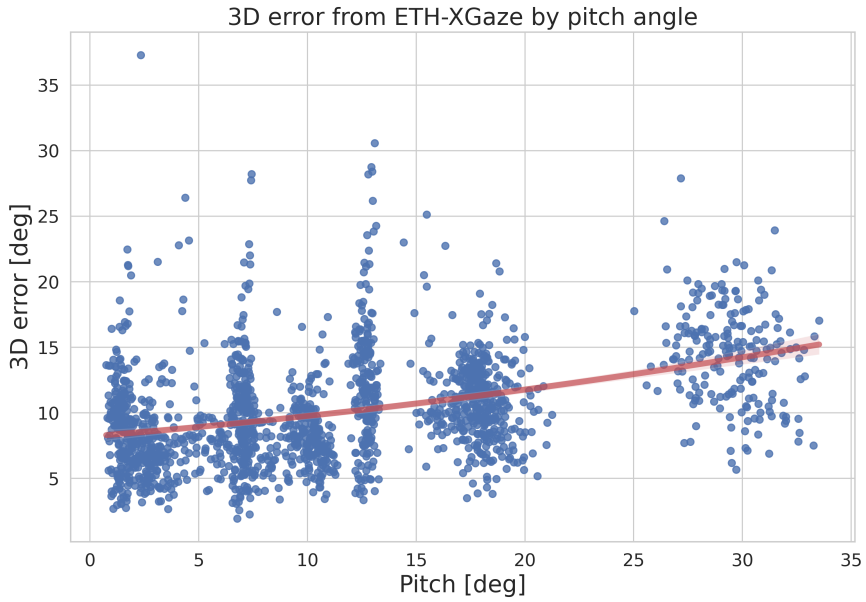


Figure 4.16: Scatter plot that shows the variation of the 3D error from ETH-XGaze based with respect to the pitch angle predicted and an estimated regression line.

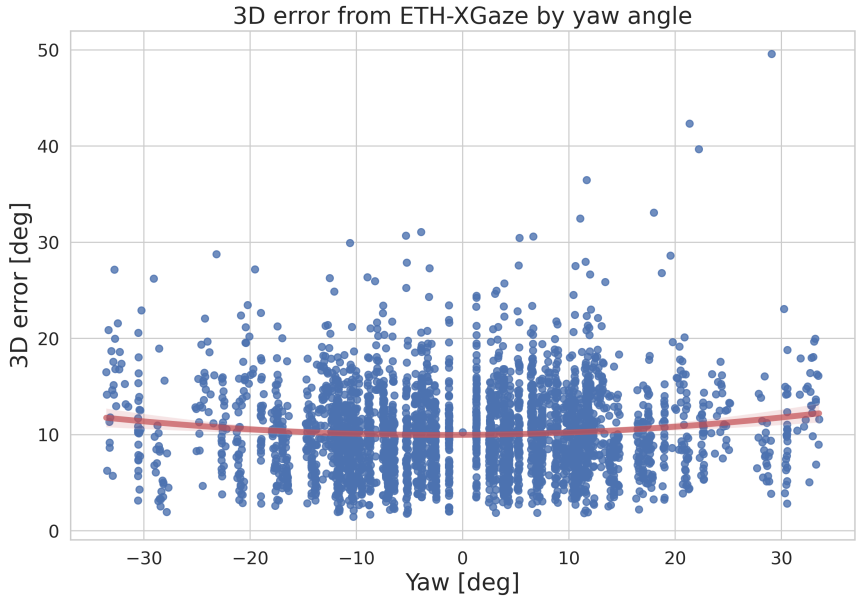


Figure 4.18: Scatter plot that shows the variation of the 3D error from ETH-XGaze based with respect to the yaw angle predicted and an estimated regression line.

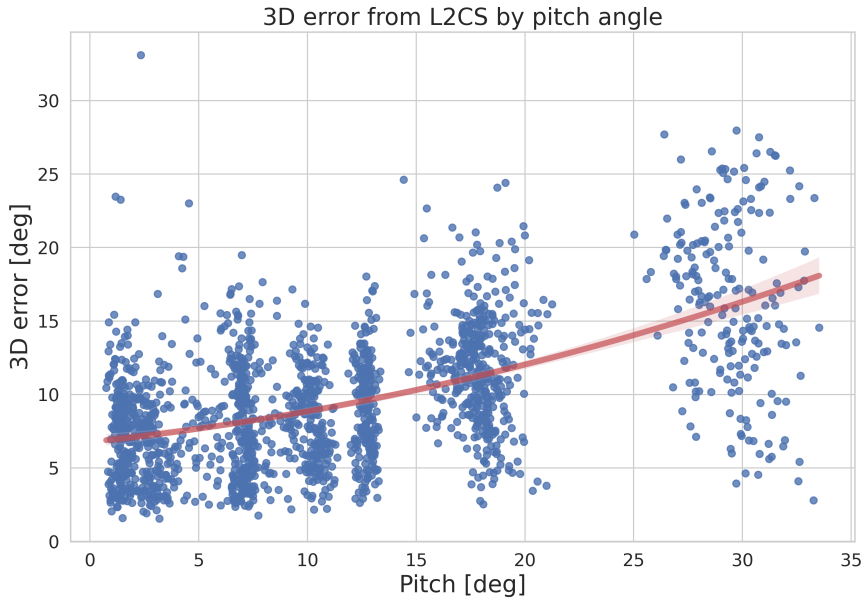


Figure 4.17: Scatter plot that shows the variation of the 3D error from L2CS based with respect to the pitch angle predicted and an estimated regression line.

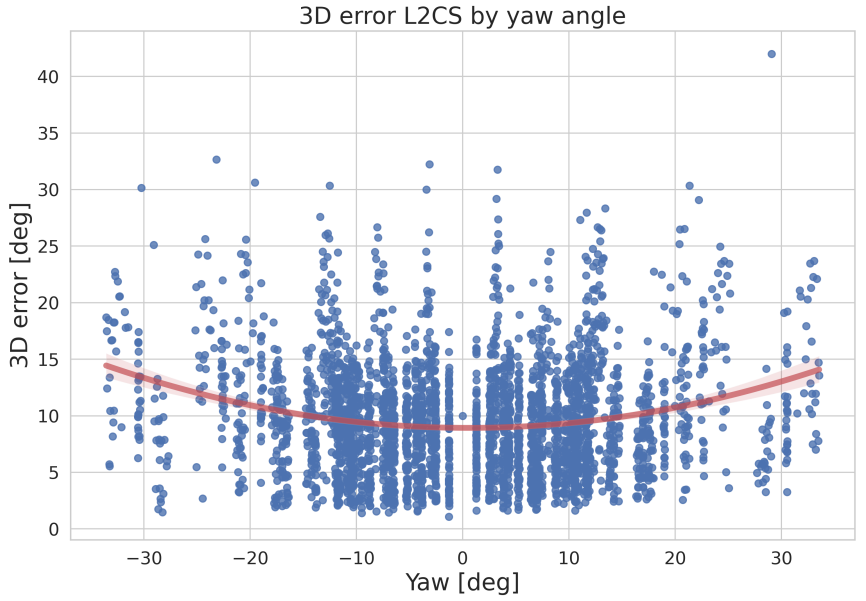


Figure 4.19: Scatter plot that shows the variation of the 3D error from L2CS based with respect to the yaw angle predicted and an estimated regression line.

Finally, we can state that with both models the yaw and the pith angles of the gaze direction affect the prediction accuracy. A bigger range in the yaw and in the pitch means an increase in the error on both models. More precisely, the two models are affected in a different way: ETH-XGaze has been shown to be more resilient to yaw and pitch variation than L2CS. In fact, the model from ETH shows a more stable error across the different yaw and pitch variations thus showing to be more robust. Bigger angles to predict in general has a worse effect on L2CS than on ETH-XGaze. On the other hand, L2CS shows better performance when the angles to predict are small.

4.6 Fine tuning

In this paragraph, we show the result of the different fine-tuning procedures performed on the L2CS model exploiting our Social-AI dataset. We performed three fine-tuning practices that differ between them for the section of the model updated and for the data used for the training.

- Training T1: In this type of fine-tuning only the last two Fully Connected layers of L2CS are trained. The data used for training and validation contains both low and high-quality images.
- Training T2: In this second type of fine-tuning, not only the Fully Connected layers but also the 4 central Convolutional layers are updated during training. The data used in the training and validation set contains both low and high-quality images.
- Training T3: In this third and last type of training the last two Fully Connected layers and the 4 central Convolutional Layers are updated during the fine-tuning. In this case, the dataset used for training and validation contains only low-quality images from the Pepper monocular camera.

The test set instead contains in all three types of training only pictures captured with the Pepper monocular camera for having a more precise understanding of the performance once the models will be deployed. Some training settings are shared by all three types of finetuning training: The training loop is set at 20 epochs on GPU; the optimizer used is Adam; the loss function is MSE; for the learning rate a polynomial scheduler is used with a starting value of $10^{-}5$.

4.6.1 Training T1

In this first training procedure, we fine-tuned only the last two Fully connected layers of the L2CS network on a training dataset containing both 4K and low-quality pictures of the participant. In the first picture Fig.4.20, we can see the value of the Mean Squared Loss on training set and validation set during all 20 epochs. The validation loss starts from approximately 130 and drops down constantly during the training for the first 10 epochs. Then it stays quite constant on what looks like an asymptote at approximately a value of 110. The training loss instead in the first 5 epochs reduces a lot its value showing a huge drop in the error. After the first 5/ 6 epochs the marginal reduction in the error at each epoch gets lower and lower. From the graph we can clearly see how using drop-out layers as regularization methods we avoided overfitting the validation set. Figure 4.21 instead shows separately the behavior of the validation pitch and the yaw loss during training. Freezing all the convolutional layers and updating the weights of just the final two Fully connected layers and based on the pictures in the training set, L2CS has a specific learning curve that can be characterized using the separate pitch and yaw loss values during the training. First of all, we can see how the yaw error for all the training keeps bigger than the pitch error. Moreover, the learning procedure affects mainly the yaw, which is actually the only loss that decreases during the training, the pitch loss instead does not show a significant decrease. The

increase in the performance of the L2CS model using the T1 training procedure is so explained mainly by an increasing capability of predicting the yaw components of the gaze direction.

4.6.2 Training T2

The difference with respect to the previous training procedure is characterized by the size of the section of the network that is updated during the training. In this case, the last two fully convolutional layers and the four central Convolutional layers are updated. The main threat in updating a bigger part of the network is overfitting. It is in fact for thinking about this possible problem that we decided to implement a set of drop-out layers in the architecture of the l2cs network used in the training. The learning curve for the training is characterized by a big improvement in performance in the first 3/4 epochs with a lower marginal improvement in the following epochs. The validation loss follows the training one on as an overall behavior, showing a big improvement in the very first epochs. The main difference between this training and the other is the asymptotes toward which the two curves tend. Particularly the training curve shows a lower value of the asymptotes that at the end of the 20 epochs go below 10 degrees. The validation set instead improves by 10 degrees the final value at the end of the training compared to the value of the previous finetuning. Figure 4.23 show the trend of the validation loss for pitch and yaw. Also in this second type of training the yaw loss is higher than the pitch loss on the validation set. A major difference with respect to the previous case is the fact that the major improvement in the validation performance is mainly explained thanks to the pitch loss reduction. In fact, the pitch error during all the training shows a way greater reduction than the yaw loss. From the beginning of the training to half of the training, the pitch loss goes from an MSE of 130 to 70 while instead, the yaw loss reduces in the first 2 epochs while staying quite stable for the rest of the training. At the end of the training, the pitch loss almost reach an MSE of 60 while the yaw stays near 130.

4.6.3 Training T3

Training T3 is the dual of training T2, they are exactly the same except for the only difference that in this case the training dataset and the validation dataset contain only low-quality images from the Pepper monocular camera. In this case, we dismiss the 4K images making the training dataset smaller and so the training faster. Moreover, using a dataset with a distribution closer to the one that the algorithm will see at deployment time we expect to increase its performance. In this case, we expect to have greater accuracy on the test set. The training's learning curves are characterized by a big improvement in performance in the first 5 epochs with a lower marginal improvement in the following epochs. The validation loss follows the training one as an overall behavior, showing a big improvement in the very first epochs. The two curves show a big difference in their value and the asymptotes toward which the two curves tend. Particularly the training curve shows a lower value of the asymptotes that at the end of the 20 epochs go below 10 degrees. The validation set instead reaches a final value lower than 100. Figure 4.23 show the trend of the validation loss for pitch and yaw. In this type of training the yaw loss is higher than the pitch loss on the validation set. Also in this case the major improvement in the validation performance is mainly explained thanks to the pitch loss reduction. In fact, the pitch error during all the training shows a way greater reduction than the yaw loss. From the beginning of the training to half of the training, the pitch loss goes from an MSE of 100 to 60 while instead, the yaw loss reduces in the first 2 epochs while staying quite stable for the rest of the training. At the end of the training, the pitch loss almost reach an MSE of 60 while the yaw stays near 130.

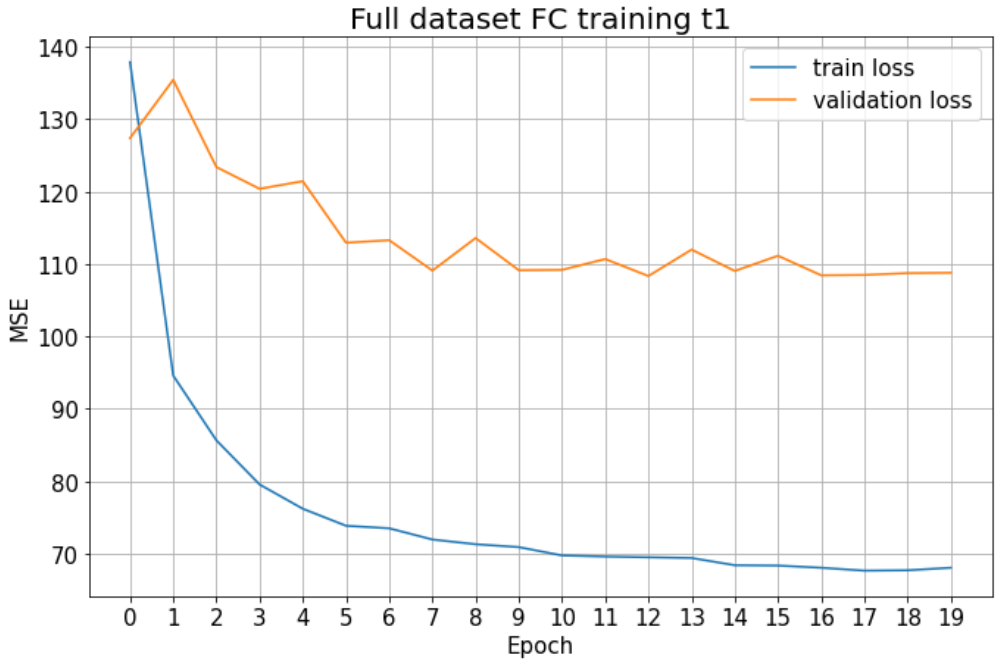


Figure 4.20: In this type of training we updated only the last two Fully Connected layers of L2CS with both low and high-quality images. The picture shows the value of the training and validation Mean Squared Error Loss during all 20 epochs of training

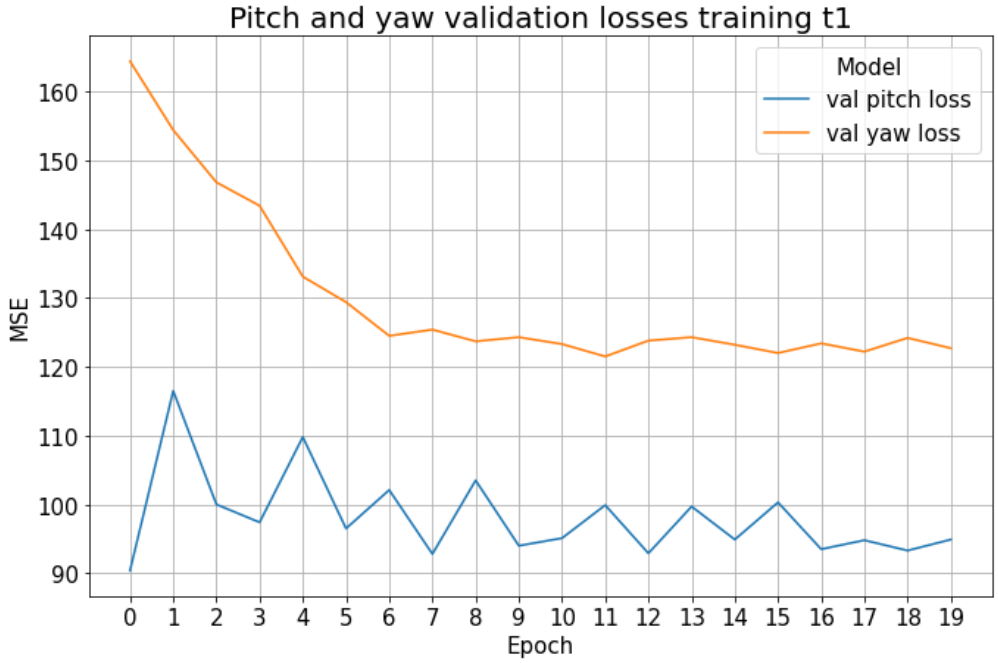


Figure 4.21: Behavior of the validation pitch and yaw loss during training T1

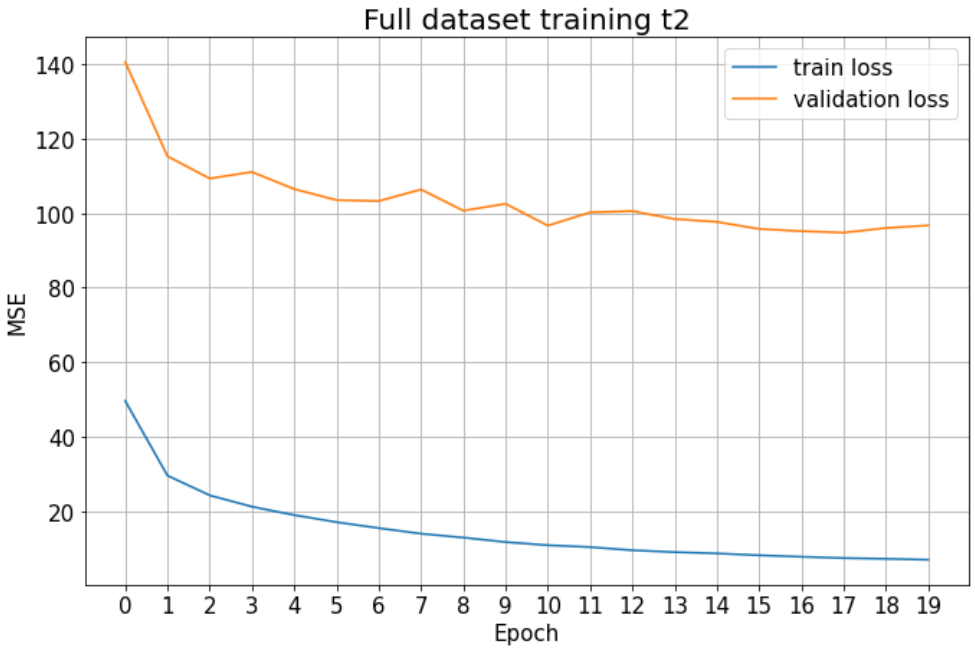


Figure 4.22: In this type of training we updated the last two Fully Connected layers and the 4 central ConvLayer of L2CS with both low and high-quality images. The picture shows the value of the training and validation Mean Squared Error Loss during all 20 epochs of training

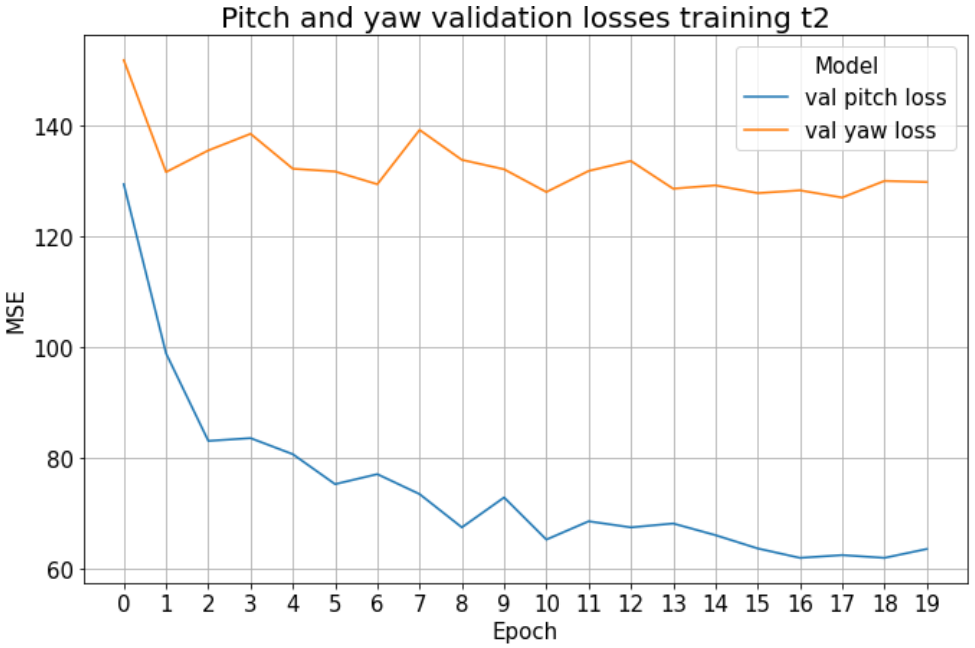


Figure 4.23: Behavior of the validation pitch and yaw loss during training T2

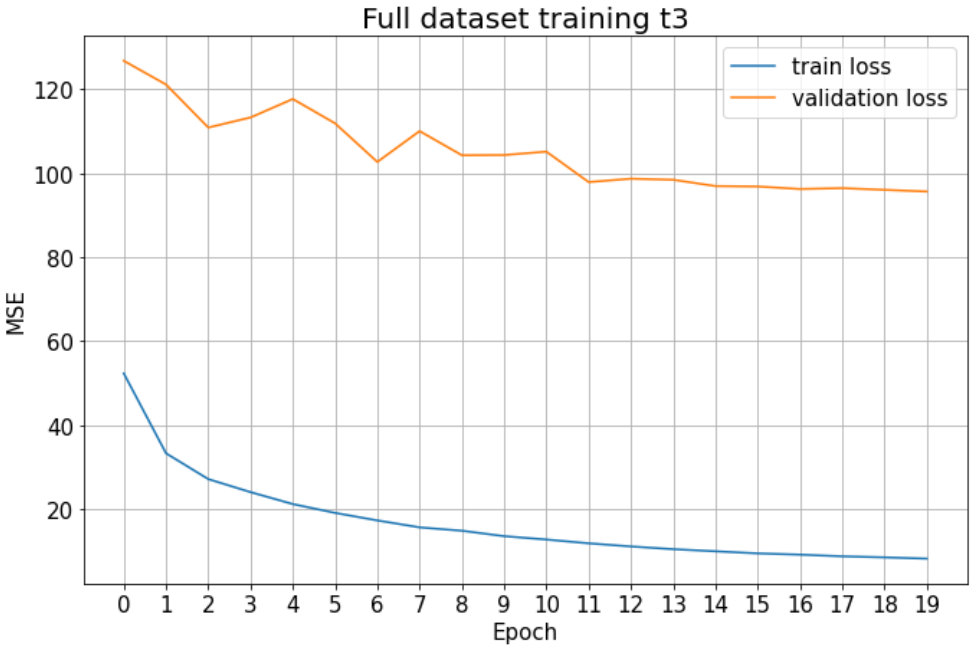


Figure 4.24: In this type of training we updated the last two Fully Connected layers and the 4 central ConvLayer of L2CS with only low-quality images. The picture shows the value of the training and validation Mean Squared Error Loss during all 20 epochs of training

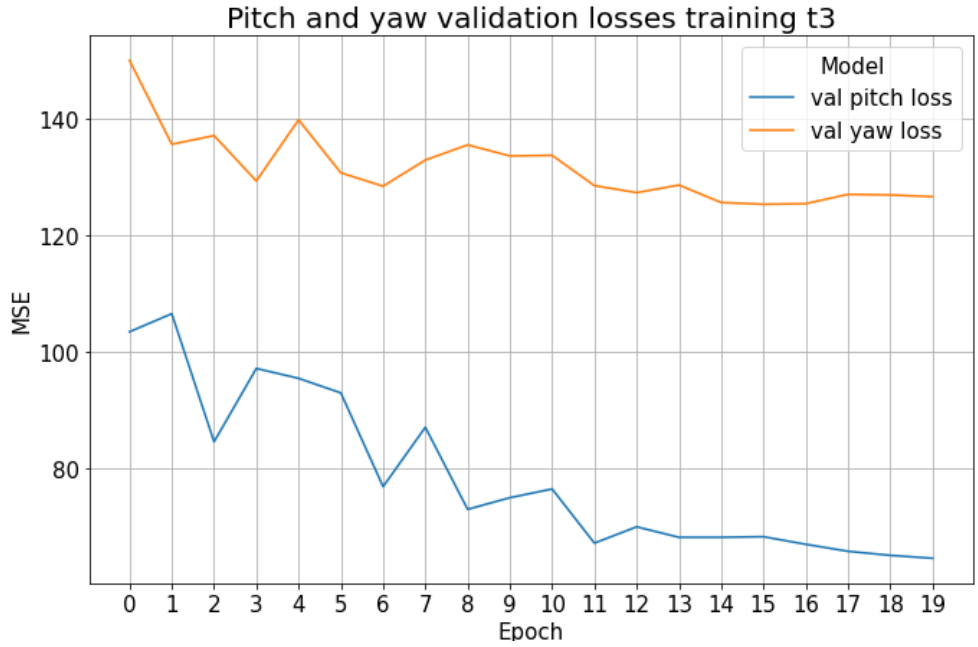


Figure 4.25: Behavior of the validation pitch and yaw loss during training T3

4.6.4 Test Results

The training results are shown in 4.1. The values reported in the table are the average 3D error computed by the best model from each of the training procedures on the test dataset. The test contains the data of participants 3,6,10,11,16,17. All the images have been captured from the low-quality Pepper built-in camera.

Training Type	Training Dataset	# data	Model	Test 3D Error
T1	4k + non 4k	32704	L2CS only FC	7.8629
T2	4k + non 4k	32704	L2CS FC +4Conv	5.4096
T3	non 4k	19659	L2CS FC +4Conv	5.5877

Table 4.1: Final results on the test set of the models after finetuning

Chapter 5

Discussion

5.1 Comment on the results

The L2CS model outperforms the ETH model when applied to low-resolution data, with accuracy and precision increases of 9% and 14%, respectively. Looking instead at the dataset with high-quality pictures ETH-XGaze performs better than L2CS although the data are less and the confidence of this result is lower than the previous case. In particular, eth improves the L2CS performance by 6% in accuracy and by 11% the precision. The use of a high-resolution camera increases the quality of prediction for both models, but we highlight the biggest improvement in quality is shown by the ETH model. Specifically, a higher resolution significantly improves the accuracy of the L2CS model (12.2%), while it improves with with a bigger range both the accuracy (23.7%) and precision (22.4%) of the ETH model. The resolution of the dataset used for training the models might influence their sensitivity to camera resolution. The ETH-XGaze dataset used to train the ETH model has a higher resolution (6000*4000) than the Gaze360 dataset used for the L2CS model (4096*3382). This suggests that if the resolution of the images contained in the dataset for training a model is higher, improving the camera's resolution will be more effective in improving the gaze estimation quality of the model. The performance comparison for distance follows the overall results stated before, showing that the L2CS model outperforms the ETH model in terms of accuracy and precision at 2 and 3 meters, regardless of resolution. In particular, at 2 meters the models have similar accuracy, instead at 3 meters l2cs is slightly better than eth on the 4k dataset while it has a relevant better accuracy on the standard dataset (27% better than eth). Looking instead at the performance at 1 meter, with the standard camera the models have close errors in prediction while with the 4k camera eth shows a significantly better accuracy (20%). We speculate that this may be due to differences in the range of distances covered by the datasets used to train the models. The distance between humans and cameras covered in the ETH-XGaze dataset is just one meter, while it ranges from 1 to 3 meters in the Gaze360 dataset. Unlike the Gaze360 dataset, where the camera height equals the subject height, the ETH-XGaze dataset provides multiple perspectives in the vertical direction. More generally, this suggests that further improvements can still be obtained by creating more extensive datasets or by fine-tuning the models. Furthermore, it is useful to note that 3D errors of the models decrease as the distance increases, although this is not the case for low-resolution data at larger distances. This is expected as in general at a longer distance the variation in yaw range will be smaller. Other work has also found that the 3D error decreases as the absolute values of yaw are closer to 0 [19]. It also appears that the performance of models drops somewhat when higher-resolution images are used at a close distance (1 m). We speculate that at a close distance, the eyes occupy a larger portion of the image, and the finer details captured by high resolution can introduce

more noise and artifacts, leading to a decrease in accuracy. In the result section we analyzed the performance of the two models in the vertical and the horizontal directions, looking at the yaw and the pitch error. The yaw and pitch performance of the models on the two Social AI datasets are different; from these results, we can extract useful information. In particular, with both models, the prediction on 4k images decreases the error on both directions. With both l2cs and eth the greater increase in performance using the 4k camera is visible in the yaw loss. We can see a decrease of the yaw loss of nearly 40% on eth and almost 30% on l2cs while on the pitch the error decreases by less then 10% with both the models using 4k pictures. This means that in the horizontal direction, the models are much more sensible to the quality of the camera than in the vertical position. Moreover, the pitch error is also greater than the yaw one for all the models and datasets although the range of yaw gaze directions (-30;+30 degrees) is higher than the pitch direction range (0; +30). This difference is explainable always for the data used for the training, where the different variations of pitch are less represented than the yaw variation Fine-tuning has shown to be a good way to increase the performance of the two models on our specific task. First of all, it is important to notice that the best practice is to one in training data and also the pictures from our 4k camera. Watching the results instead, we can see how the training using also the 4 central layers led to a really big improvement in the training data while staying away from overfitting thanks to the strategy used to avoid it for example dropout. The decision on which model to use depends on the use cases: if the robot will be deployed in situations that are close to the experiment setting thus might be useful to implement the models trained on a bigger section of the network, while instead if we want to keep some generalization capabilities it might be successful to implement the model fine-tuned on just the last layers. In the real-time experiment at the end, we decided to use the model fine-tuned only on the last layer, since it can guarantee a higher degree of generalization even in different contexts. Our findings suggest that state-of-the-art models for gaze estimation can be usefully applied to off-the-shelf robots such as Pepper and similar robots. In particular, the L2CS model already seems to perform well in the social zone even with a low-resolution camera that is available on most social robots that can be bought today. In a short range of up to 2m, the resolution has little effect on the accuracy of L2CS. Our work thus shows that existing models are promising for application in HRI scenarios, although clearly more work is needed to validate this for the "in the wild" scenarios.

5.2 Real-time joint attention

As a first step, we informally present a small example application to demonstrate the feasibility of using state-of-the-art models for gaze estimation in an HRI scenario. In our example, we simulate a joint attention task where the robot and user need to look at the same object. Such a task can be useful for a scenario where focusing on the same object communicates interest, e.g., in a store.

We applied the L2CS model after fine-tuning T1, which for us is the best one also considering its generalization performance. We used the built-in camera of the Pepper platform with a resolution set to 640*480. In our example scenario, objects were placed on a table, with a human user and the Pepper robot on either side of the table. Figures 5.1 and 5.2 illustrate the setup by means of images from the robot's camera (the robot perspective) and from an upper view. We varied the distance between the user and robot and positioned both either 0.5m or 1m away from the table on opposite sides. As a result, the distance between the user and the robot varied from either 1m or 2m. We asked two participants to look at the objects on the table, asking the robot with identifying to which object the gaze was directed. At the start, each time, the robot looks straight ahead towards the participant and based upon an analysis of their gaze moves its head to look in the same (from an opposite perspective) direction. The robot looks straight ahead again after 2s when the eyes (gaze) of the participant can no longer be detected by means of the camera. We collected a small sample of 60 data points, which we classified either as correct or

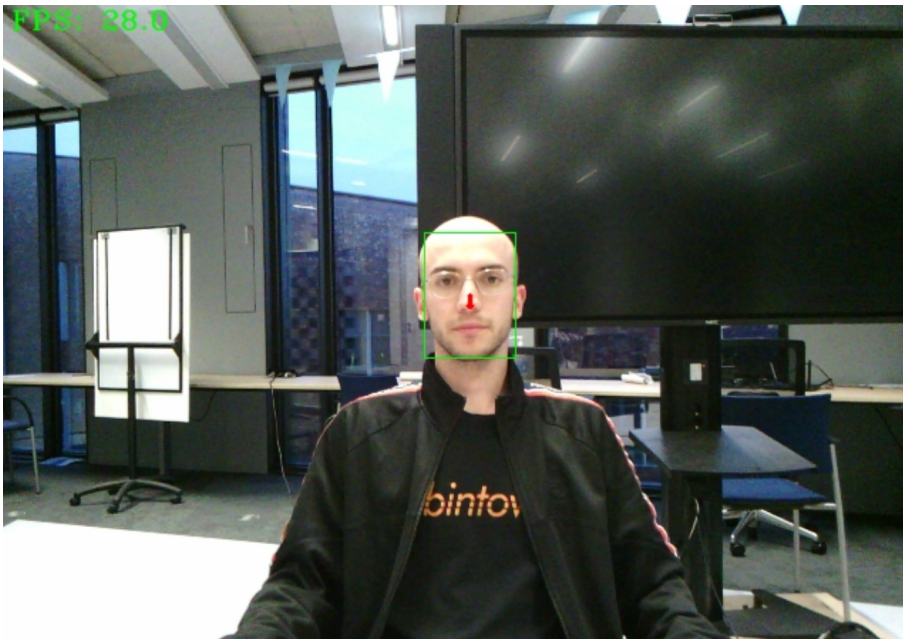


Figure 5.1: Robot view during the joint attention experiment. The image has been acquired from the upper monocular camera of the robot

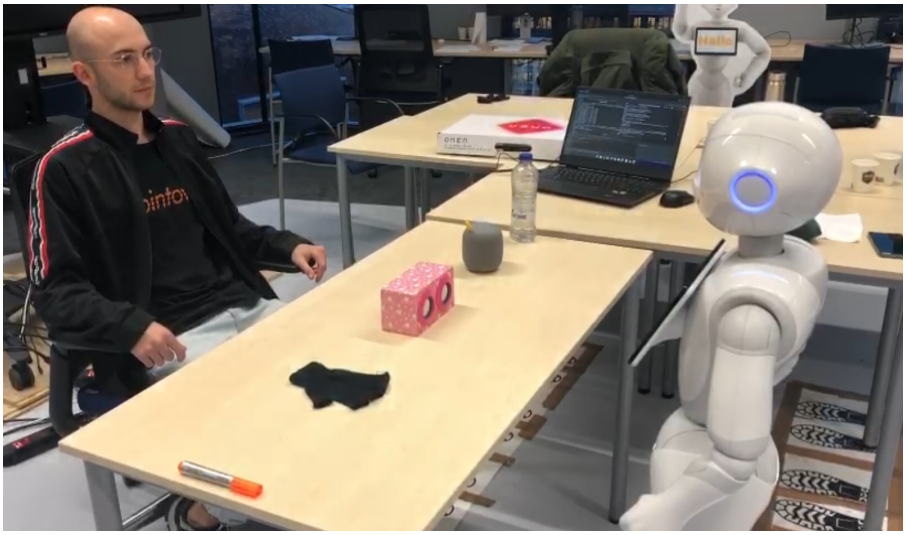


Figure 5.2: The picture shows the set-up of the joint attention experiment from a top view. The two actors look at each other and between them, there is the table with on top the objects that will be subject to the human gaze direction

incorrect based on the 2D error of eye gaze on the virtual plane where the objects are located. It is regarded as correct if the error is smaller than 12.5cm (half of the shortest distance between objects). The accuracy for our sample is 85%. Gaze estimation and joint attention were achieved successfully for all objects except for the gray speaker. Moreover, and relevant to report here,

the time it took to move the robot's head to follow human gaze is less than 1s where about 0.5s is needed to estimate the user's gaze. Although still a bit slow, which likely can be improved by using better hardware in future setups, this shows the potential of the appearance-based method for responding to human gaze cues by a social robot.

Chapter 6

Conclusion

In this thesis, we investigated the use of two state-of-the-art models for gaze estimation in a controlled laboratory setting. We restricted our study to the social zone, where estimating gaze is most useful to facilitate social interaction between social robots and users. We evaluated the prediction quality of the two models, L2CS and ETH, at different distances from the subject and with different camera resolutions.

In order to perform the analysis, we collected a small dataset of people standing at different distances from the camera and gazing at target dots appearing in front of them. The dataset contains 28350 images captured at 3 distances from the subjects (1 to 3 meters) and with two resolutions: 640x480 and 3840x2160. After the data collection, we tested L2CS and ETH on the SocialAI dataset. The analysis of the results suggested that L2CS is more performative with low-resolution cameras at every distance. This model demonstrated promising performance with an average gaze estimation error of 10°. ETH instead performs better with 4k images but only at a distance of 1 meter. On the other hand, ETH gains more in performance when we use 4k images instead of standard ones. As a general rule so, ETH can be preferred to L2CS when the prediction involves pictures with high resolutions. L2SC instead outperforms ETH when we need to predict more noisy images.

In the last part of this work, we presented the results of three simple fine-tuning practices applied to our dataset. Fine-tuning showed to be a good way to improve L2CS performance in our social setting scenario. In the end, we showed how after the fine-tuning L2CS is ready to be deployed with sufficiently good performance in the Pepper robot. With the implementation of a simple joint attention task, we illustrated that the built-in monocular camera of Pepper can be used to effectively estimate human gaze using the L2CS model after fine-tuning. L2CS reached 85% success rate in the example application and there is still much room for improvement. This thesis opens up future works in different directions: from the collection of images in more complex environments to the inspection of more advanced domain adaptation techniques. Moreover, after the implementation of L2CS in the Softbank Pepper, it can be worth studying the implementation of different other natural behaviors of the robot in human-robot-interaction scenarios focusing the attention on the human response. On one hand, the challenge is to increase the prediction capability of the appearance-based gaze estimation methods, on the other hand instead we need to test and experiment with deployed models in real-world tasks making the robots able to exploit these powerful tools for meaningful collaboration with humans.

Bibliography

[1] A.A. Abdelrahman, T. Hempel, A. Khalifa, et al. L2cs-net: Fine-grained gaze estimation in unconstrained environments. In *Proceedings of the International Conference on Image Processing (ICIP)*. IEEE, 2022.

[2] J. Adams, P. Rani, and N. Sarkar. Mixed initiative interaction and robotic systems. *AAAI Workshop on Supervisory Control of Learning and Adaptative Systems*, 2004.

[3] Aldebaran, United Robotics Group. Softbank robotics pepper - cameras. [http://doc.aldebaran.com/2-5/family/pepper_technical/video_2D_pep.html#d-camera-pepper].

[4] Aldebaran, United Robotics Group. Softbank robotics pepper - motors. [http://doc.aldebaran.com/2-5/family/pepper_technical/motors_pep.html].

[5] Sean Andrist, Bilge Mutlu, and Michael Gleicher. Conversational gaze aversion for virtual agents. In Ruth Aylett, Brigitte Krenn, Catherine Pelachaud, and Hiroshi Shimodaira, editors, *Intelligent Virtual Agents*, pages 249–262, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] E. L. Blickensderfer, R. Reynolds, E. Salas, and Cannon bowers J. A. Shared expectations and implicit coordination in tennis doubles teams. *J. Appl. Sport Psychol.*, pages vol. 22, no. April 2013, pp. 486–499, 2010.

[7] Jean-David Boucher, Ugo Pattacini, AmÃ©lie Lelong, GÃ©rard Bailly, FrÃ©dÃ©ric Elisei, Sascha Fagel, Peter Dominey, and Jocelyne Ventre-Dominey. I reach faster when i see you look: Gaze effects in human-human and human-robot face-to-face cooperation. *Frontiers in neurorobotics*, 6:3, 05 2012.

[8] N Brown. Edward t. hall: Proxemic theory, 1966,. In *Center for Spatially Integrated Social Science. University of California, Santa Barbara*, page vol 18: 2007, 2001.

[9] Malinda Carpenter, Katherine Nagell, and Michael Tomasello. Social cognition, joint attention, and communicative competence from 9 to 15 months of age. *Monographs of the Society for Research in Child Development*, 63 4:i–vi, 1–143, 1998.

[10] Yihua Cheng, Haofei Wang, Yiwei Bao, and Feng Lu. Appearance-based gaze estimation with deep learning: A review and benchmark, 2021.

[11] M.W. Doniec, G. Sun, and B. Scassellati. Active learning of joint attention. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 34–39. IEEE, 2006.

[12] N.J. Emery. The eyes have it: the neuroethology, function and evolution of social gaze. *Neuroscience Biobehavioral Reviews*, 24(6):581–604, 2000.

[13] T. Farroni, G. Csibra, F. Simion, et al. Eye contact detection in humans from birth. *Proceedings of the National academy of sciences*, pages 99(14): 9602–9605, 2002.

[14] T. Fischer, H.J. Chang, and Y. Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *Proceedings of the European conference on computer vision (ECCV)*, pages 334–352, 2018.

[15] Alexandra Frischen, Andrew P. Bayliss, and Steven P. Tipper. Gaze cueing of attention: visual attention, social cognition and individual differences. *Psychological Bulletin*, 133(4):694–724, 2007.

[16] K.A. Funes Mora, F. Monay, and J.M. Odobez. Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras. In *Proceedings of the symposium on eye tracking research and applications*, pages 255–258, 2014.

[17] Conty L George N. Facing the gaze of others. *Neurophysiologie Clinique/Clinical Neurophysiology*, pages 38(3):197–207, 2008.

[18] T. E. Hall. *The hidden dimension.* Doubleday, Garden City, N.Y., 1st edition, 1966.

[19] Q. Huang, A. Veeraraghavan, and A. Sabharwal. Tabletgaze: dataset and analysis for unconstrained appearance-based gaze estimation in mobile tablets. *Machine Vision and Applications*, pages 28(5): 445–461, 2017.

[20] S. Ivaldi, S.M. Anzalone, W. Rousseau, et al. Robot initiative in a team learning task increases the rhythm of interaction but not the perceived engagement. *Frontiers in neurorobotics*, page 8: 5, 2014.

[21] Sin-Hwa Kang, Albert (Skip) Rizzo, and Jonathan Gratch. Understanding the nonverbal behavior of socially anxious people during intimate self-disclosure. In Yukiko Nakano, Michael Neff, Ana Paiva, and Marilyn Walker, editors, *Intelligent Virtual Agents*, pages 212–217, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[22] P. Kellnhofer, A. Recasens, S. Stent, et al. Gaze360: Physically unconstrained gaze estimation in the wild. In *IEEE/CVF international conference on computer vision*, pages 6912–6921. IEEE, 2019.

[23] H. Kim, H. Jasso, G. DeÃ¡k, et al. A robotic model of the development of gaze following. In *7th IEEE International Conference on Development and Learning*, pages 238–243. IEEE, 2008.

[24] K. Krafka, A. Khosla, P. Kellnhofer, et al. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184. IEEE, 2016.

[25] S. Lackey, D. Barber, L. Reinerman, N. I. Badler, and I. Hudson. Defining nextgeneration multi-modal communication in human robot interaction. In *Proc. Hum. Factors Ergon. Soc. Annu. Meet*, pages vol. 55, no. 1, pp. 461–464, 2011.

[26] Argyle M. Facing the gaze of others. *Royal Institute of Philosophy Supplements*, pages Volume 10: 63 – 78, 1976.

[27] M. F. Martins and Y. Demiris. Impact of human communication in a multi-teacher, multi-robot learning by demonstration system. *Proceedings of the Workshop on Agents Learning Interactively from Human Teachers*, 2010.

[28] M. J. Mataric. Cooperative behaviors in multi-robot systems through implicit communication. *Rob. Auton. Syst.*, pages vol. 16, no. 2–4, pp. 321–331, 1995.

[29] L.P. Morency, C.M. Christoudias, and T. Darrell. Recognizing gaze aversion gestures in embodied conversational discourse. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 287–294. IEEE, 2006.

[30] M.M.E. Neggers, R.H. Cuijpers, P.A.M. Ruijten, et al. Determining shape and size of personal space of a human when passed by a robot. *International Journal of Social Robotics*, pages 14(2): 561–572, 2022.

[31] F. NegKaplan and V.V. Hafner. The challenges of joint attention. *Interaction Studies*, pages 7(2): 135–169, 2006.

[32] Catharine Oertel, Marcin Wlodarczak, Jens Edlund, Petra Wagner, and Joakim Gustafson. Gaze patterns in turn-taking. In *Proc. Interspeech 2012*, pages 2246–2249, 2012.

[33] E. Pagello, A. D'Angelo, F. Montesello, F. Garelli, and C. Ferrari. Cooperative behaviors in multi-robot systems through implicit communication. In *Rob. Auton. Syst*, pages vol. 29, no. 1, pp. 65–77. IEEE, 1999.

[34] L. Paletta, A. Dini, C. Murko, et al. Towards real-time probabilistic evaluation of situation awareness from human gaze in human-robot interaction. In *Proceedings of the Companion of the 2017 International Conference on Human-Robot Interaction*, pages 247–248. IEEE, 2017.

[35] O. Palinko, F. Rea, G. Sandini, et al. Robot reading human gaze: Why eye tracking is better than head tracking for human-robot collaboration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5048–5054. IEEE, 2016.

[36] Oskar Palinko, Francesco Rea, Giulio Sandini, and Alessandra Sciutti. Eye gaze tracking for a humanoid robot. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 318–324, 2015.

[37] G. Perugia, M. Paetzel-Prüsmann, M. Alanenpää, et al. I can see it in your eyes: Gaze as an implicit cue of uncanniness and task performance in repeated interactions with robots. *Frontiers in Robotics and AI*, page 8: 645956, 2021.

[38] Evan Risko, Kaitlin Laidlaw, Megan Freeth, Tom Foulsham, and Alan Kingstone. Social attention with real vs. reel stimuli: Toward an empirical approach to concerns about ecological validity. *Frontiers in human neuroscience*, 6:143, 05 2012.

[39] L. Scalera, S. Seriani, P. Gallina, et al. Human-robot interaction through eye tracking for artistic drawing. *Robotics*, page 10(2): 54, 2021.

[40] Atsushi Senju and Gergely Csibra. Gaze following in human infants depends on communicative signals. *Current Biology*, 18(9):668–671, 2008.

[41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[42] R. Vertegaal, R. Slagter, G. van der Veer, and A. Nijholt. Eye gaze patterns in conversations: There is more to conversational agents than meets the eyes. In *In Proceedings of the ACM Conference on Human Factors in Computing Systems (SIGCHI)*, pages (Vol. 3, p. 301–308), 2001.

[43] D. Vogel and R. Balakrishnan. Interactive public ambient displays: Transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. 17th Annu. ACM Symp. User interface Softw. Technol*, volume 6, pages 137–146. UIST, 2004.

[44] P. N. Wilson. Cooperative behaviors in multi-robot systems through implicit communication. *Mar. Corps Gaz.*, pages vol. 91, no. 4, pp. 29–32, 2007.

[45] Tian (Linger) Xu, Hui Zhang, and Chen Yu. See you see me: The role of eye contact in multimodal human-robot interaction. *ACM Trans. Interact. Intell. Syst.*, 6(1), may 2016.

[46] Akiko Yamazaki, Keiichi Yamazaki, Matthew Burdelski, Yoshinori Kuno, and Mihoko Fukushima. Coordination of verbal and non-verbal actions in human-robot interaction at museums and exhibitions. *Journal of Pragmatics*, 42(9):2398–2414, 2010. How people talk to Robots and Computers.

[47] Chen Yu, Paul Schermerhorn, and Matthias Scheutz. Adaptive eye gaze patterns in interactions with human and artificial agents. *TiiS*, 1:13, 01 2012.

[48] X. Zhang, S. Park, T. Beeler, et al. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision*, pages 365–381. Springer, Cham, 2020.

[49] X. Zhang, Y. Sugano, M. Fritz, et al. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, pages 41(1): 162–175, 2017.