

Duy Vu

MACHINE UNDERSTANDING OF MOTHER-INFANT INTERACTION

Using motion capture and eye tracking technology

Bachelor of Science Thesis
Faculty of Information Technology and Communication Sciences
Examiner: University Lecturer Sari Peltonen
April 2022

ABSTRACT

Duy Vu: Machine understanding of mother-infant interaction
Bachelor of Science Thesis
Tampere University
International Bachelor degree of Science and Engineering
April 2022

Parents' interaction with their infant is crucial in the children's early cognitive development, but capturing and measuring these interactions is not a trivial task. Currently, a professional psychologist is required to watch the recording of parent and infant attentively to detect and label all interactions. While human is the one setting the baseline and standard for the encoding system, and hence the most reliable source of measurement, it is no doubt that relying on human is highly inefficient, and we shall aim at automating the pipeline using state-of-the-art technology. Therefore, in this thesis, an automatic workflow of capturing and analysing mother-infant interaction with the help of Machine Learning is demonstrated. To be specific, two focus modes of interaction in this thesis are the amount of mother's gaze on baby and the head-to-head distance between the pair.

Keywords: mother-infant, interaction, computer vision, eye tracking, gaze capture, motion capture, instance segmentation

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

PREFACE

I would like to sincerely convey my gratitude to my supervisor, University Lecturer Sari Peltonen, for her practical guidance and tremendous support throughout the process of my research and writing. Moreover, I also want to thank those who gave me the opportunity to work in this interesting project as a trainee last spring: Pasi Pertilä, the principal investigator of the technical side of the project at that time, and Professor Atanas Gotchev, director of CIVIT. Last but not least, I would like to thank Jani Käpylä with his constant and considerable assistance in the technical side, from devices and data to necessary knowledge in order to perform the experiment.

Tampere, 20th April 2022

Duy Vu

CONTENTS

1. Introduction	1
2. Background	2
2.1 Gaze estimation	2
2.2 Instance segmentation	3
2.3 Motion capture	5
3. Methods	6
3.1 Experimental design	6
3.2 Data extraction and processing	6
3.2.1 Gaze capturing	6
3.2.2 Motion capture	6
3.3 Software implementation	7
3.3.1 Checking mother's gaze at infant	7
3.3.2 Distance between mother and infant	10
4. Results	11
4.1 Comparing gaze and segmentation results	11
4.2 Analysing distance between mother and infant	14
5. Discussion	17
6. Conclusions	19
References	20
Appendix A: Extracting gaze data using software Pupil Player	24
Appendix B: Processing gaze data using Python	26

LIST OF FIGURES

2.1	A person wears Invisible glasses which look just similar to normal glasses [10].	2
2.2	Example of instance segmentation result generated by Detectron2.	4
2.3	The Mask R-CNN framework for instance segmentation [23].	5
3.1	Software flowchart.	7
4.1	IoU formula.	11
4.2	Comparison between models based on inference time and AP.	12
4.3	Comparison of software result on part 1 of the video.	13
4.4	Comparison of software result on part 2 of the video.	14
4.5	Comparison of 2 histograms of distance between mother and infant in the first part of the experiment.	15
4.6	Comparison of 2 histograms of distance between mother and infant in the second part of the experiment.	16
A.1	Pupil Player screen.	24
A.2	Plugin manager of Pupil Player.	25
A.3	Added plugins appear at the bottom of listed plugins.	25

LIST OF TABLES

3.1	How result is visualized (gaze color) and written (last 2 columns).	9
-----	---	---

LISTINGS

B.1	Converting gazes in normalized coordinate to video coordinate	26
-----	---	----

LIST OF SYMBOLS AND ABBREVIATIONS

AP	Average Precision
COCO	Common Objects in Context
CPU	Central Processing Unit
CSV	Comma-Separated Values
FAIR	Facebook AI Research
FPN	Feature Pyramid Network
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
IR	Infrared
LED	Light Emitting Diode
R-CNN	Region-based Convolutional Neural Network
RegNet	Regulated Residual Networks
ResNet	Residual Neural Network
RoI	Region of Interest
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine

1. INTRODUCTION

It has been shown that early parent-infant interaction can largely affect infant development [1, 2, 3]. These interactions can be movement [4], touch [5], gaze [6], vocal communication like singing [7], or even a combination of features which is also called multimodal interactions [8, 9]. Normally, to capture and analyse these interactions, a professional psychologist is needed to watch the recording of parent and infant and analyse everything manually. Not only this is costly, but also capturing the multimodal interactions is nearly impossible. Hence, a software pipeline with the capability to capture and analyse the needed features can significantly help researchers to analyse parent-infant interaction. Among all interactions psychologist needs to characterize, in this thesis, only 2 main aspects are focused on: the gaze quantity of mother on infant and the distance between mother and infant. For the first objective, we want to check whether mother's gaze is at her baby by retrieving the gaze direction from the view of the mother from an eye tracking device called Pupil Invisible, and segmenting the baby using a computer vision machine learning algorithm. Secondly, a motion capture system called Optitrack is used to find spatial locations of the mother-infant pair, and, based on that information, a time series data of distance between the pair is produced.

The thesis is structured as follows. Firstly, Chapter 2 will describe in detail the algorithm, mechanism of the technology used in this thesis. Then, in Chapter 3, the step-by-step instructions will be demonstrated in order to conduct the experiment and obtain the results. After that, Chapter 4 will display these results. Finally, Chapter 5 will discuss some problems as well as future improvements and Chapter 6 will summarize everything.

2. BACKGROUND

Before going to the implementation and experiment, we need to understand some core concepts, mechanisms and algorithms used in this thesis: gaze capturing, instance segmentation to find gaze of mother on infant, and motion capture for calculating distance between mother-infant pair.

2.1 Gaze estimation

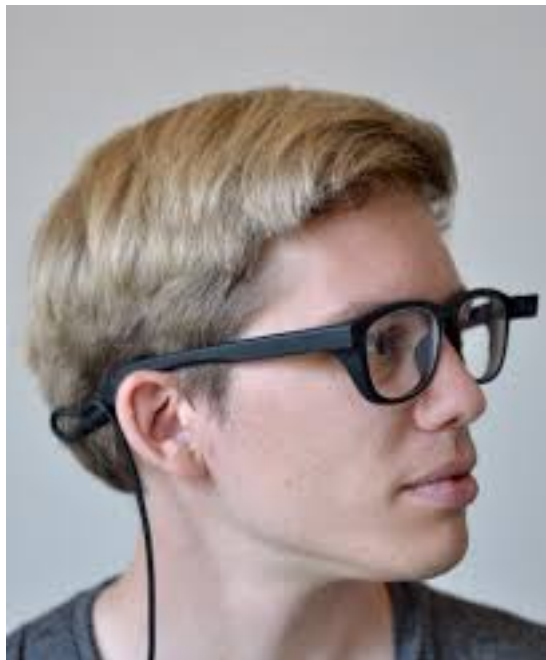


Figure 2.1. A person wears Invisible glasses which look just similar to normal glasses [10].

The eye tracking glasses we use to learn where the mother is looking at is Pupil Invisible [10, 11] which is preceded by the first product released by Pupil Labs, Pupil Core [12]. The hardware of the glasses consists of 2 eye cameras recording at the frame rate of 200 frames per second at the resolution of 192 by 192 pixels. These cameras are attached in the interior of the glasses frame. Next to each of these cameras is an Infrared (IR) LED, acting as a light source to the eyes in dark conditions. On the outside of the glasses frames, there is a detachable scene camera recording at the smaller frame rate of 30 frames per second, but with higher resolution of 1 088 by 1 080 pixels, along with the field of view of 82 by 82 degrees. One great advancement of this product compared to its

predecessor is its simplicity (neither calibrations nor adjustments are needed), accuracy (robust performance in diverse environments) and elegance (look-alike normal glasses) it brings to the wearer [10], and this is also why Pupil Invisible is used in the experiment. An example of the glasses is shown in Figure 2.1. This experiment is not just about interaction between any humans, but interaction between a mother and her small child, and hence we do not want the infant to feel foreign, and be distracted by a strange-looking device on the mother face. Moreover, even though the wearer, the mother in this experiment, may not need to move drastically and suddenly, a small glasses slippage is still unpredictable and unavoidable. However, Pupil Invisible can compensate for such slippage [10].

To understand how the glasses can be so versatile, we need to understand the way the glasses estimate the gaze direction. There are 2 stages in the estimation process: calculating gaze direction in a device-independent coordinate system, and then, based on the camera of a specific device, mapping it to a more accurate position [10]. The reason behind these 2 separate phases is the variance in both intrinsic and extrinsic device parameters when manufacturing the glasses [10]. Hence, firstly, some preliminary gaze prediction can be acquired with the assumption of ideal scene camera [10]. After that, with the knowledge of known specific device parameters, the final gaze prediction can be obtained for each pair of glasses [10]. To get the initial gaze prediction, Pupil Labs trained a convolutional neural network with their own dataset in which the input and output data are eye images and 3D gaze respectively [10]. With the 3D gaze predictions attained in the previous step and the scene camera's specific parameters, these gazes will be transformed into 2D gaze points on the scene camera view [10]. All of these steps are done by Pupil Labs.

2.2 Instance segmentation

With the mother's gaze data on hand, now we only need to locate the baby in the video so that we can check whether the mother is looking at the baby. In each video frame, a Boolean value True is assigned for each pixel in the baby segment, False otherwise. This locating method is often known as instance segmentation. Since letting human annotate every frame is time-consuming, an accurate and fast machine learning algorithm should be adopted instead. To be more precise about the definition, an instance segmentation algorithm is a combination of object detection algorithm (locating object using bounding box) and semantic segmentation algorithm (assigning every image pixel to a predetermined list of categories but not discriminating objects belonging to the same class) [13]. An example of instance segmentation can be seen from Figure 2.2. In this thesis, we choose to use Detectron2, one of the most widely adopted open-source project developed by Facebook AI Research (FAIR) [14]. One of the most important advantages of Detectron2 to this research is the wide range of tasks the library supports [14]. Even though currently

the method we choose to find the baby is through segmentation, other techniques with their own merits can also be considered in the future, such as bounding box (object detection), or localizing keypoints of the baby like mouth, eyes, hands (keypoint detection). Therefore, this makes the tool handy for future work as well. Moreover, regarding instance segmentation alone, there is still a huge list of pre-trained models with various architectures to choose from the library [15, 16, 17].

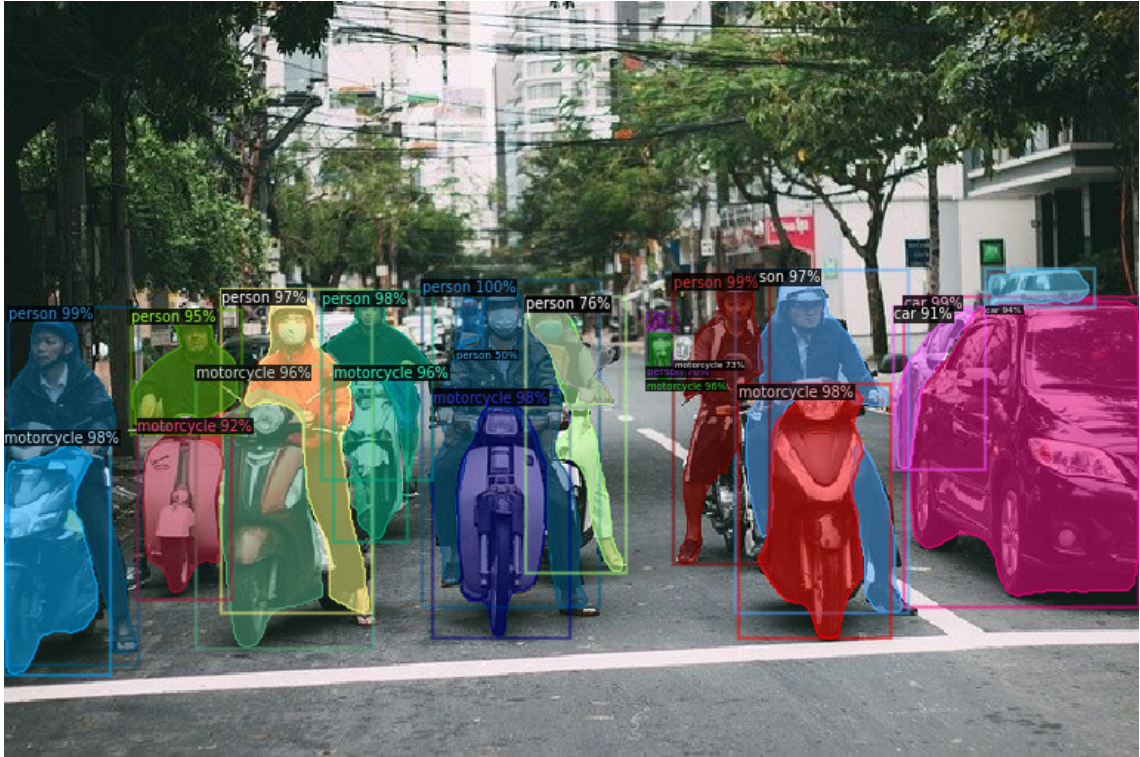


Figure 2.2. Example of instance segmentation result generated by Detectron2.

In this thesis, we use 3 Mask Region-based Convolutional Neural Network (R-CNN) models from original baselines [15] in which Residual Neural Network (ResNet) [18] and Feature Pyramid Network (FPN) [19] are the backbone network with standard Convolution layer and Fully connected heads for mask and box prediction [20]. Moreover, 2 models trained in longer training schedule and large-scale jitter [16] in which one has slightly different backbone architecture (Regulated Residual Networks (RegNet) [21] as bottom-up pathway) are also used in this thesis. The criteria for this choice of models are fast inference time together with high accuracy, and we base the choice on the available result on the website [15, 16]. Detailed information about these 2 criteria are discussed in Chapter 4.

In short, Mask R-CNN is an extension of Faster R-CNN [22], with a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression [23]. Besides, instead of using RoIPool like R-CNN, RoIAlign is used since the mechanism of using coarse spatial quantization from RoIPool causes misalignments between the extracted features and the original RoI, and thus cannot perform well on predicting masks at pixel level [23]. Instead of

applying quantization twice, RoIAlign just simply divides the feature maps into equal bins (not rounding the size of bins), takes four middle sampled points, computes bilinear interpolation from the closest neighbouring cells. Afterwards, aggregate function is used normally. Visualization of the architecture can be seen from Figure 2.3.

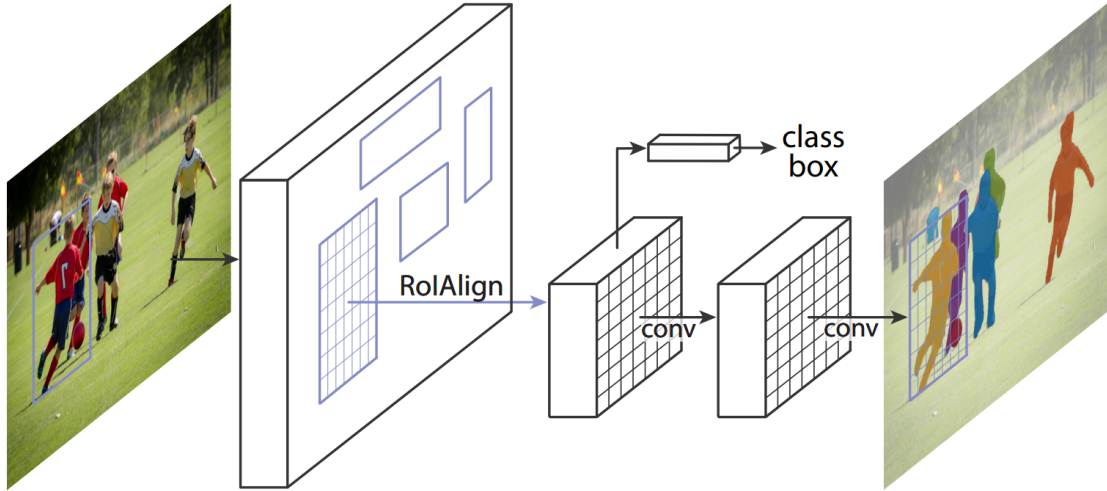


Figure 2.3. The Mask R-CNN framework for instance segmentation [23].

Models with ResNet backbone as depth of 50 and 101 layers, were trained with around 37 Common Objects in Context (COCO) epochs [20]. For models trained using Copy-Paste method [24], they are trained with 400 epochs where each epoch consists of training on 118 000 COCO images [16]. The detailed configuration of all models can be found from Detectron2's repository [25, 26, 27, 28, 29].

2.3 Motion capture

To track the motion of mother and infant, we use Optitrack system. Its main components are cameras which, in this case, are Prime 17W [30], Prime 41 [31], markers, and the software called Motive. In order for the camera to capture the positions of participants, we need to put some markers on them. In this thesis, we focus on the markers placed on the head only since head is body part that is naturally most unlikely blocked from all cameras. Moreover, Motive Rigid Body Tracking is used to track solid objects like heads in this case. Four markers are needed to accomplish this since 3 points can define a surface in 3D space, but 4 points (not on the same surface) can define a 3D volume. Moreover, more markers provide more information about the rigid body location in 3D space, and hence make the workflow smoother. Furthermore, even in the case of marker occlusions, the system can still deduct position of markers from other visible markers, and thus compute position and orientation of the rigid body. The way the system works is that multiple cameras placed around the laboratory will capture 2D images of the markers on participants, calculate the positions of these markers, and these overlapping data are used to finally calculate the 3D positions via triangulation.

3. METHODS

3.1 Experimental design

A recording session has two parts of 10 minutes each. In the first part, the mother and infant freely interact with each other. Then, the mother is instructed to do 5 interaction tasks. During the experiment, the mother will wear the eye-tracking glasses to extract the gaze data. For motion capture data, markers are put on both mother and infant body: 4 markers on each person head, 1 marker on each person arm, and 1 marker on each person leg. Two cameras are set to capture facial expressions of the mother which are used for later manual analysis, and there are as well several other cameras in the Optitrack system. They are synced with Optitrack system with a trigger signal.

3.2 Data extraction and processing

3.2.1 Gaze capturing

Gaze position can be retrieved from Pupil Player software by dragging the recording folder to the application and pressing "download" icon on the left side. Downloaded data is in the sub-folder with 3-digit number as folder name. If data from one recording is exported multiple times, the latest data can be found in the folder with the name as the largest number in that directory "exports". An example of a downloading can be found in Figure A.1 of Appendix A.

In addition, we can see the gaze visualization made by Pupil Player by using "Vis Circle" and "Vis Polyline" from "Plugin Manager" of the application. Information and figures about this can be found in Figure A.2 and Figure A.3 of Appendix A.

3.2.2 Motion capture

As said in Chapter 2, the system includes cameras Prime 17W, Prime 41, and passive markers (also called retro-reflective markers) which reflect incoming light back to its source instead of active markers which can emit IR LED light by itself. Anyway, IR light emitted from the camera is reflected by passive markers on participant's body and detected by

the camera's sensor. Since the system obtains the data through detecting emitted or reflected light, the experiment lab also minimize the ambient lighting, such as sunlight and extraneous illumination or reflection sources.

3.3 Software implementation

Two different pieces of software are created to process the corresponding data and produce the necessary results. The first section will check the mother's gaze at her baby, followed by the second one about analysis of distance between mother and infant.

3.3.1 Checking mother's gaze at infant

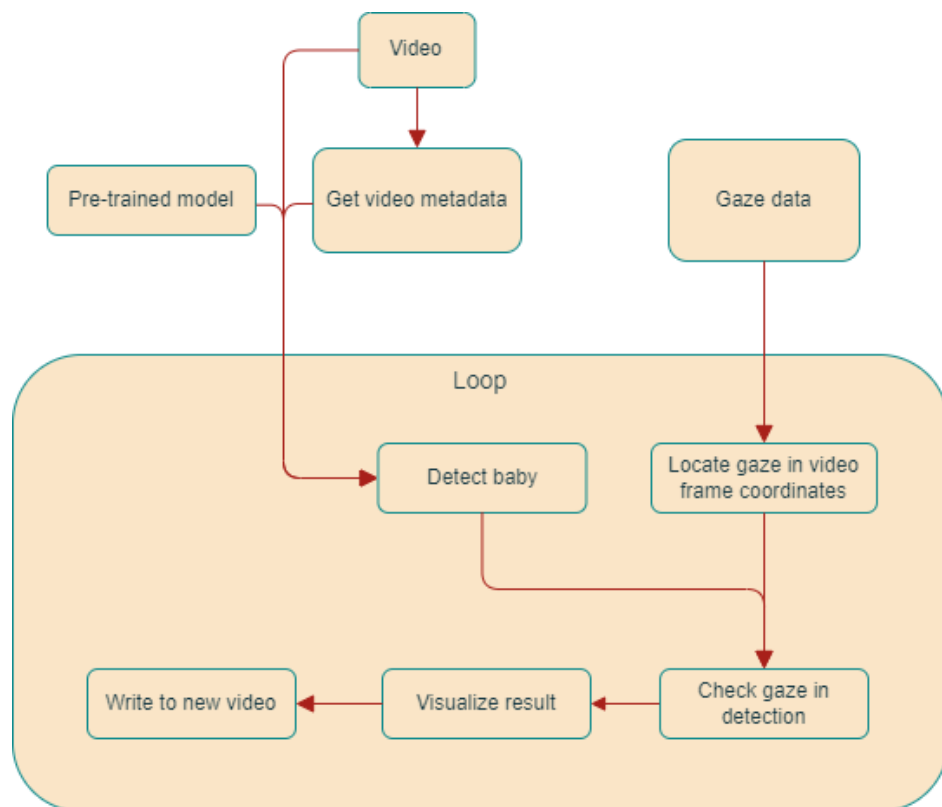


Figure 3.1. Software flowchart.

The raw video from world camera (without using visualizing plugins from Pupil Player software) is available immediately after each recording session, and we can then use OpenCV-Python, one of the most ubiquitous Python libraries for image and video processing, to create the video with visualization of whether mother is looking at the baby. Firstly, we need metadata of the video: frame rate, number of frames, width and height of the video frame. Then, we can read each frame of video, detect baby, visualize the baby instance, mother's gazes on it, and write it to the new video. These steps of visualizing and processing frame are done in the loop of reading video frame. The mechanism of the program can be seen from Figure 3.1.

Finding baby

For each video frame, given a pre-trained instance segmentation model, we define before the loop, we find the baby by feeding the video frame as an input image to the model and ask the model to return only the "person" instance segmentation result (generally, the model returns results of all classes it knows). The dataset that Detectron2 is trained on (COCO dataset) does not have specific class for "baby" but since from the mother's point of view, there is only one baby, and "baby" is also a "person", this approach is feasible. Sometimes, the model can make false positive prediction, but the prediction score (a float number in the range from 0 to 1) for such case is usually significantly lower than normal true positive prediction by at least 0.2. Hence, a threshold to filter lower confident prediction should be set properly. The choice of this parameter is more or less from observation of the model performance, but a good starting point is 0.9. Moreover, if there are more than one predicted "person" instance, only the one with the highest classification probability is retrieved. A trickier scenario we might encounter is when the mother's hands cover the baby and since the model only has the knowledge of human (as well as body parts), it will conclude the mother's hand as part of the same human instance, or in other words, baby. This is, however, acceptable since the final goal is to understand the mother's eye interaction with her infant, and if the mother's hands interact with the baby by touching, holding, we can consider those hands as parts of the same "baby" instance.

Retrieving gaze

Besides the baby segmentation, the other required element is the gaze location relative to the video frame. These data are stored in the `gaze_positions.csv` in the sub-folder exported in the previous section, but since the gaze data from Pupil Player is in the normalized coordinates (the origin is at bottom left), we need to convert it to the video frame coordinate (the origin is at the top left) by multiplying normalized x and y coordinates with the frame width and height [32] [33, see cell 10]. However, sometimes, the normalized gaze coordinates may be larger than 1 or smaller than 0. This might seem ridiculous, but completely normal as the mother's eye pupils might move all the way to the edge. The way we handle these edge cases is by clipping the denormalized coordinates in the range from 0 to maximum width and height correspondingly. Moreover, details of this processing gaze data step can be found in Program B.1 of Appendix B. However, not all gazes are detected by Pupil Invisible. For example, mother may rub her eyes, scratch her nose, and thus accidentally block the eye cameras. Hence, in the same file where gaze data is stored, there is a confidence score for each gaze which has only two possible confidence values: 1.0 and 0.0 corresponding to the algorithm estimation of whether the person is wearing the glasses or not at each gaze. To filter out 0.0 confident gaze, we need to specify a minimum gaze confidence score that we want to get gaze data which can be any number between

0.0 or 1.0. Another problem we might encounter is missing gaze data which sometimes happens in the first few frames of the recording. This can happen due to the fact that gaze estimation process sometimes starts later than the scene video recording. Anyway, the number of these abnormal gazes only takes up to at most 5 % of the total gazes, and we can simply ignore them.

Visualization

With the gaze and segmentation results on hand, the only steps needed to be taken are visualizing and writing the result which is used for later analysis. There are 3 cases we need to handle: baby is found in the frame and gazes are at the baby, baby is found in the frame but gazes are not at the baby, and baby is not found from the video frame. In order to visualize effectively, different colors for the gaze are used for each scenario. Besides the visualization, the result of baby segmentation as well as checking gazes at baby are written in a CSV file. The goal of the visualization is that user of this software can easily see all of these things at the same time: baby segmentation result and the confidence score (more useful for technical user who wants to work on the segmentation algorithm), movement of the gaze, and know whether the mother is looking at the baby or not (the latter two are more useful for psychologist analysing mother-infant interactions). Since baby, as a big object compared to the gaze points, can easily be seen on the video by human eyes, we make the segmentation 10 % more transparent than the default level which allows user to see the gaze more easily as well as the details of the baby like facial expression. Moreover, the confidence score is visible next to segmentation result for developer analysis. Regarding logic of the gaze color and writing result of gaze and segmentation, it is described in the Table 3.1.

Table 3.1. How result is visualized (gaze color) and written (last 2 columns).

baby	gaze available	gaze in segmentation	gaze color	is_baby	in_segmentation
True	True	True	Green	True	True
True	True	False	Red	True	False
False	True	False	Yellow	False	Nothing
False	False	False	Nothing	False	Nothing

There are 4 circumstances. Firstly, if the baby is detected, and the gaze can be seen inside the detected segmentation, then we draw the green dot on the segmentation. Secondly, if the baby is still detected, but the gaze is not inside the segmentation, then we use red color for the gaze. Thirdly, if the segmentation is now unavailable, in other words the model cannot find the baby, but the gaze is, we use yellow color for the gaze. Finally, if both segmentation and gaze data are unavailable in the frames, then we do not draw anything on the video frame. In short, the color choice can be understood as: green - looking at

visible baby, red - not looking at visible baby, yellow - waiting for the baby to be visible.

In addition to the visualization, the detection gaze in segmentation is written to a CSV file so that we can analyse them in the future. Two important columns needed to add to the file are "is_baby" which is filled with binary values about whether the baby is detected and "is_segmentation" which, besides the binary values about whether the gaze is in the segmentation, sometimes contains no values since not all frames have baby detected in the first place.

3.3.2 Distance between mother and infant

In this section, we focus mostly on the distance between mother's and infant's head. Markers on the head do not usually have missing data as much as other body parts like legs. Furthermore, we are studying the interaction of the mother's eyes which locate on her head. In the data file, the data of head position are labeled as "RigidBody BabyHead (or MotherHead) Position X (Y, or Z)". They are the 3D coordinates of the heads, and the unit of these data is meter. The origin where these coordinates are calculated is a point on the ground of the capture scene. Even though this origin may vary between recording sessions, this is not important as we only focus on the distance between the mother and infant. Hence, the positions of the participants relative to each other matter the most. This is explained later with the distance formula at the end of the next paragraph.

After retrieving data, it is very important to check for missing data and interpolate the data appropriately, for example, by cubic or linear interpolation. If there are too many missing data (more than half), we might want to consider use data from other experiment instead since it is likely that there are some issues with the experiment. However, in this thesis, there are no missing data for the heads' markers. After any necessary pre-processing data step, we can calculate the distance between mother and infant over time using Euclidean distance:


$$d = \sqrt{(x_{mother} - x_{baby})^2 + (y_{mother} - y_{baby})^2 + (z_{mother} - z_{baby})^2} \quad (3.1)$$

where d is the distance between mother and infant, x_{mother} , y_{mother} , z_{mother} , are 3D coordinates (x, y, z coordinates in the Cartesian coordinate system) of mother's head and x_{baby} , y_{baby} , z_{baby} are 3D coordinates baby's head. From Equation 3.1, it is worth noting that the distance between 3D coordinates of mother and infant is what matter the most. Mathematically speaking, the distance between 2 points is always the same regardless of the choice of origin. Therefore, this allows us to do comparison between different mother-infant pairs without being bothered with the accuracy of mother and infant locations relative to the origin.

4. RESULTS

4.1 Comparing gaze and segmentation results

Generally, there is no problem in the gaze data we have in terms of confidence. In most cases, all gazes are detected by the glasses, with 1.0 confidence score, and in the worst case scenario, we have maximally 9 % of all gazes undetected which means the glasses are not worn at that time. However, there are a couple of things to discuss about instance segmentation model. Two of the most important criteria we should look out for a good model is short inference time and high accuracy. To be precise about the metric, inference time is the time it takes for a model to process a single video frame, and accuracy is mask Average Precision (AP). To understand what is this mask AP, we need to understand how a predicted mask can be considered a correct mask. Given the predicted and ground truth mask, if the ratio of overlapping area and total area of these 2 masks is greater than a predefined threshold (usually 0.5), then it is a correct mask. This evaluation metric is often known as Intersection over Union (IoU). Visualization of the formula is demonstrated in Figure 4.1.



$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Figure 4.1. IoU formula.

With this metric, we can calculate the precision and recall of a class given a threshold for IoU. The area under the precision-recall curve is the AP for a class given a threshold. The final mask AP is the average of the APs calculated over all 80 classes in the dataset and all 10 threshold values ranging from 0.50 to 0.95 with 0.05 spacing [34].

Fortunately, models provided by Detectron2 have already been evaluated with at least these 2 metrics (inference time and AP), so we summarize all of them in the graph below. All the numbers can be found from the original sources [15, 16] and visualized in Figure 4.2.

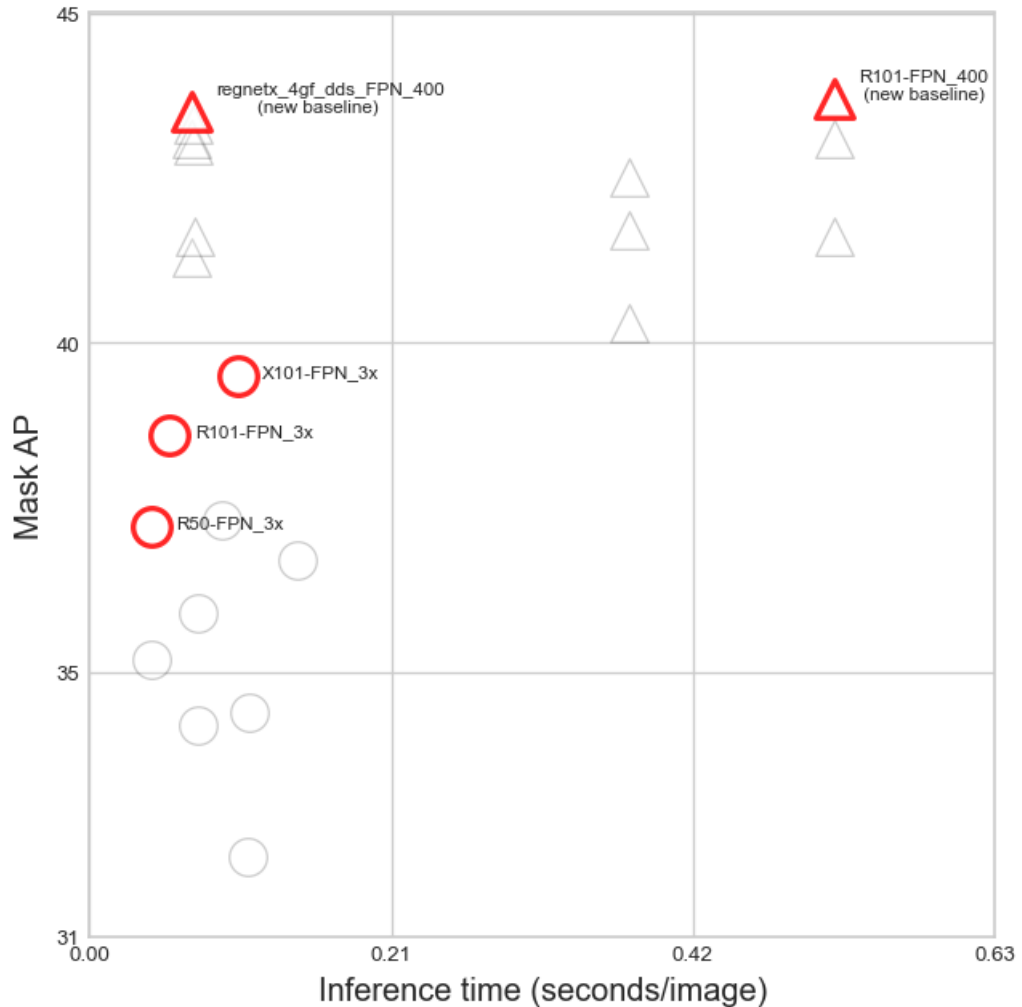


Figure 4.2. Comparison between models based on inference time and AP.

Figure 4.2 shows 10 models created as the original baseline, denoted by circle, and 12 new baseline models, denoted by triangle. We pick models with the following characteristics: the highest and second highest AP (and fastest if there are at least 2 models with the same AP) from original and new baselines, and the fastest of all. Picked model are colored in red, and we denote them with the following names: "R50-FPN_3x", "R101-FPN_3x", "X101-FPN_3x", "regnetx_4gf_dds_FPN_400", and "R101-FPN_400".

Without any labeled data and manual labeling is not considered in the thesis, the results of gaze and segmentation are analyzed through observation only. In order to facilitate this process, we also try to choose videos in which mother gazes on the baby the most which is checked via observation as well. Subsequently, the gaze and segmentation results made by the software are summarized on a graph.

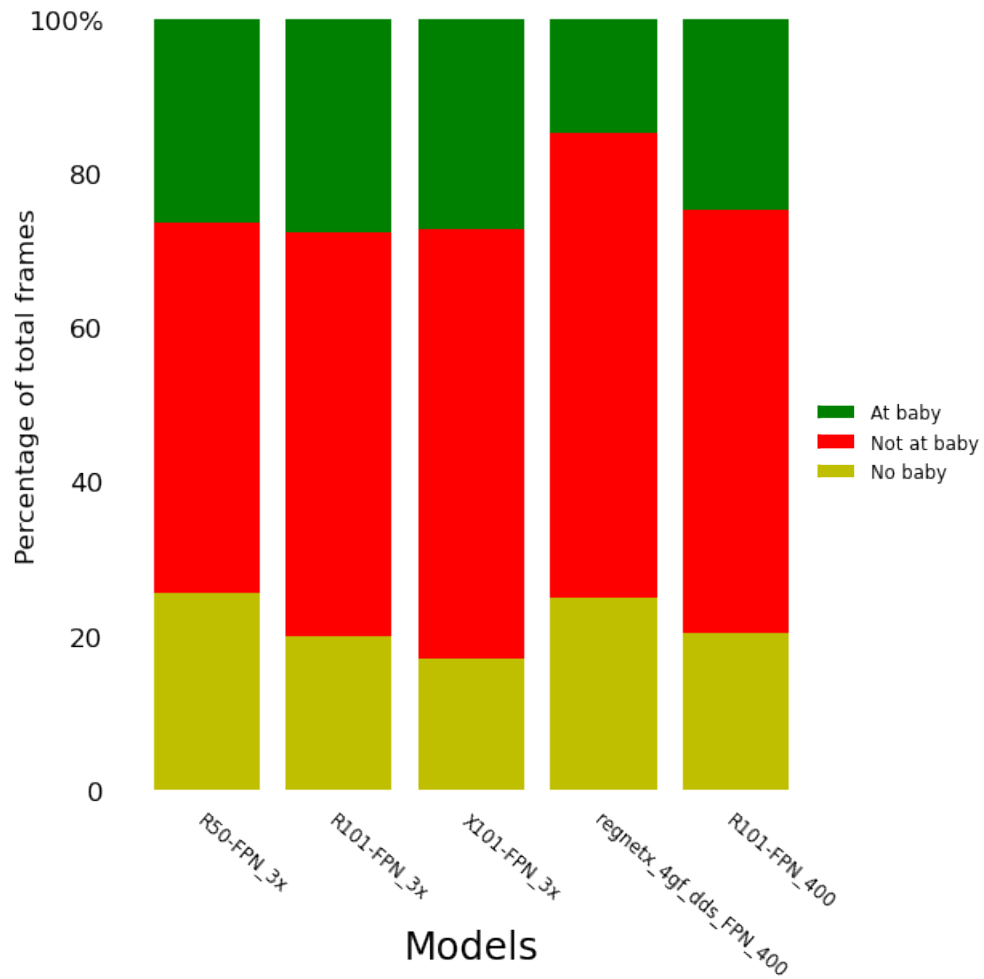


Figure 4.3. Comparison of software result on part 1 of the video.

As discussed earlier, the mother-infant experiment has 2 parts. For the first part of the video and Figure 4.3, we can make one key observation: Better (higher AP) models generally perform better, with an exception of the "regnetx_4gf_dds_FPN_400". In many frames, baby segmentation cannot be seen using worse models (lower AP), but it is possible with high AP models. This can be because better models give true baby instance segmentation higher score than worse models give, and the score is also higher than the minimum threshold for confidence score. In other cases, even with the existence of both infant and mother hand in the frame, worse model puts more confidence on mother body parts which are clearly not interacting with the infant (mother hand is reaching to the toy) while better model can find the baby. Put "regnetx_4gf_dds_FPN_400" aside, the first model "R50-FPN_3x" having the lowest AP out of all reports the least amount of frames with baby, and going to the right with higher AP models, the amount of frames baby detected increases, compared to the leftmost case. However, there is a fairly equal number of frames the mother is looking at the baby. This can mean that these frames have highly confident segmentation where all models have the same prediction and the mother is

looking directly in the middle of the baby segmentation. Regarding the circumstances of the model "regnetx_4gf_dds_FPN_400", the low number of frames in which the mother is looking at the baby may imply that the model outputs false positive prediction in many frames. To put it simply, many segmentation results are actually on the mother and the mother is looking at the baby. Also, the high number of frames without segmentation can mean that the confidence score of segmentation is so low that it cannot be counted in the end.

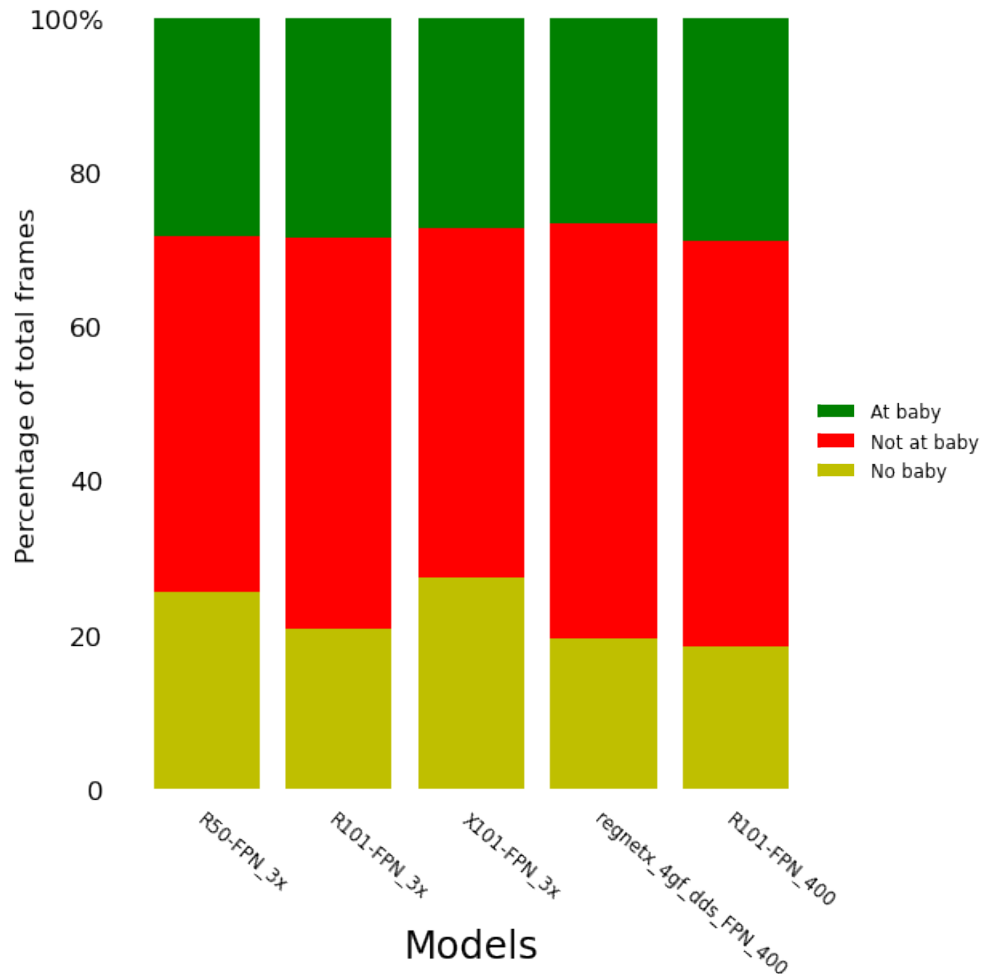


Figure 4.4. Comparison of software result on part 2 of the video.

From Figure 4.4, we also see similar pattern in the second part of the video, but now with an exception of the model "X101-FPN_3x" which has high number of frames not detecting the baby. We believe that it has the same problems as in model "regnetx_4gf_dds_FPN_400".

4.2 Analysing distance between mother and infant

Depending on the goal, we might need different kinds of motion capture data, but for this task we only need data from mother and infant heads' markers, and fortunately for us,

there are no missing data, and we can easily do the calculation. The time series distance data between mother and infant can be used for many purposes. Nevertheless, in this thesis, we demonstrate a comparison between the distance of 2 mother-infant pairs in 2 scenarios: when free to do anything and when given specific tasks to follow.

Without adequate knowledge or expertise in this psychology field, the detailed analysis of this data should be done later by professional psychologists. However, we can still draw some statistical analysis from these time series data. As said in Chapter 3, each experiment has 2 parts. Each part is 10-minute long, and we have 2 mother-infant pairs. The recording rate of the motion capture system is 100 frames per second. The statistical result drawn from this data is that when given some tasks, the mother tends to get closer to the baby. The average distance in Figure 4.5 is 0.6, whereas the number is only 0.4 in Figure 4.6.

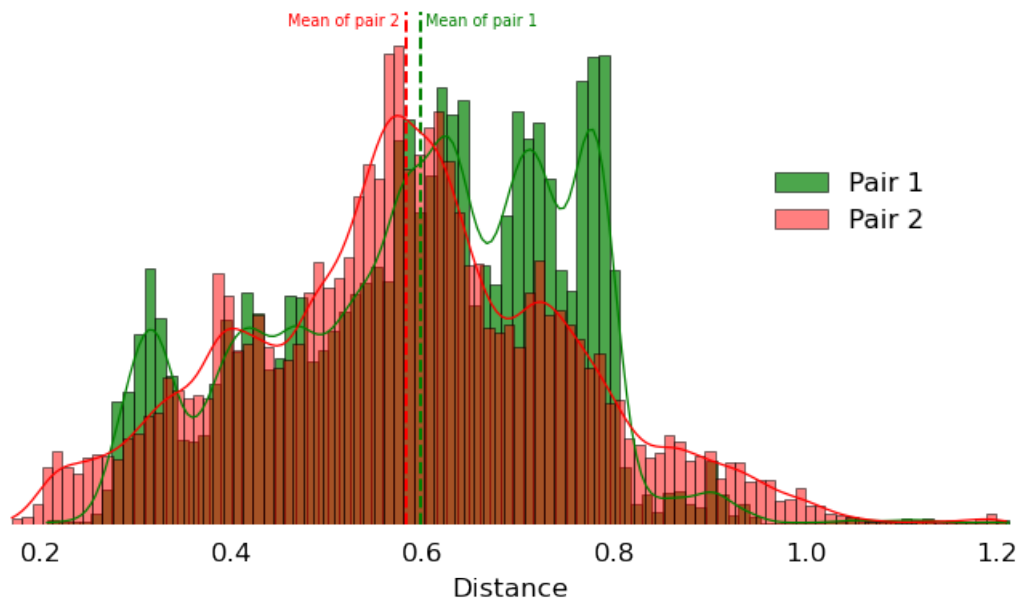


Figure 4.5. Comparison of 2 histograms of distance between mother and infant in the first part of the experiment.

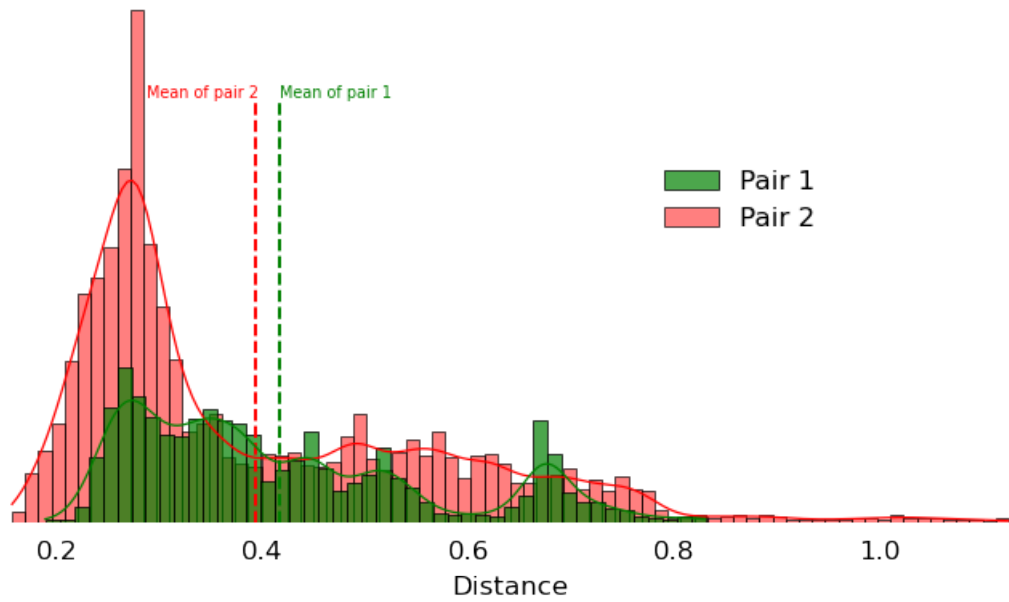


Figure 4.6. Comparison of 2 histograms of distance between mother and infant in the second part of the experiment.

5. DISCUSSION

In this project, since there are several assumptions and hypotheses we must make for this project, errors and problems are inevitable, and we have mentioned them all in the previous chapter.

Regarding baby detection, false detection result can be considered as the most notorious problem. It is easier to make, and thus create misleading result for data analysis. For example, mother's gaze is at the baby can be seen by human eye but mother's body part which happens to exist in the same frame is detected by model instead. Hence, a misleading result that "mother is not looking at the baby" is created. This number should be decreased as much as possible since in most of the videos, mother is looking at the baby for a majority of the time.

The problem with the current detection algorithm is that it can only classify "person" instance from other non-person instances, not a whole human with human body part. Even though we have forced the model to only pick the person with the highest score, the problem does not go away completely. One possible cause of this issue lies in the training dataset that the models are trained on. Possibly, it contains mainly normal adult and teenager instances, with a few or no infant instances, and thus adult's body parts may be recognized more easily by the model, compared to even a whole infant in the frame.

This problem cannot easily be solved by telling the parent not to look to their body which is unrealistic. One solution we can take is to train a better model where the algorithm can differentiate infant from adult. This will require a new dataset of baby with some introduction of adult's body part to "trick" the model. We already have this data collection, but it still needs to be sorted out and labelled carefully. With a good dataset, we can take the pre-train models we have already now to continue training to serve our purpose. Another possible solution is tracking, use other algorithm to draw bounding box first then segmentation. Besides the idea of using another Machine Learning model which relies on weights and biases the model learns through the training process, some traditional Computer Vision approaches like Histogram of Oriented Gradients (HOG) together with Support Vector Machine (SVM) to detect human in video [35], or Scale Invariant Feature Transform (SIFT) features which are proved to be "invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in

illumination" to extract the baby keypoints and localize the baby in the video [36]. Optical flow estimate can be considered as well since the majority of the video is the baby. One of the most prevalent optical flow methods is the Lucas–Kanade method [37] or the advanced version using Shi-Tomasi feature [38].

Regarding segmentation algorithm, vision Transformer model is on the trend nowadays. For example, recent Swin Transformer models have outperformed the traditional Mask R-CNN models on the COCO dataset [39, 40].

Moving to the eye gazing part, we can even consider fixation instead of gaze points which are quite sensitive, and noisy due to the constant movement of eye and high capturing rate. Fixation can be an option as it is only detected under some minimum and maximum constraints on gaze dispersion, duration, and even confidence.

Last but not least, since the final product is the software, fast software is also vital. Currently, the software has to read one video frame at a time and process everything on that frame before moving to the next one. However, we can already start reading the next frames and process them upon doing image processing on the previous frame. This process of letting multiple Central Processing Unit (CPU) threads execute concurrently is called multi-threading.

6. CONCLUSIONS

The goal of this thesis was to illustrate an automatic workflow including capturing and analysing interactions between a pair of mother and infant. Specifically, we aimed at estimating the amount of mother's gaze at infant and analysing the interspace between the pair. These goals were achieved via a system of hardware and software. Regarding the hardware, a pair of eye tracking glasses and a motion capture system were utilized to track the mother gazes at the infant and find the location of participants respectively. The data retrieved from the hardware were then processed depending on the goals of the thesis.

For quantifying mother's gaze at infant, a machine learning model was used to find the segmentation of baby from the glasses' scene camera. Afterwards, the gaze data from glasses were checked with the segmentation results obtained in the previous step to get the estimation correctly. Several different machine learning models were also tested to see how different models infer the input video. However, through observation, the baby segmentation result was not perfect, regardless of the models, and some alternative solutions were devised in Chapter 5.

Regarding the location data of mother and infant, a time series distance data was acquired by calculating Euclidean distance between mother and infant heads. Even though markers were attached in other body parts of participants, head was chosen as it usually did not have any missing data. Anyway, for this thesis, a small data analysis on the distance of the pair in different scenarios was the best result we can reach, and it was better to let those with suitable expertise conduct a full analysis.

In conclusion, this thesis has shown the capability of automating a human-intensive tasks via a set of advanced technology. Even though there are a lot of ongoing research in this field of automatic interaction analysis, this is still a very active field of research and requires more development with the help of the state-of-the-art technology.

REFERENCES

- [1] Niedźwiecka, A., Ramotowska, S. and Tomalski, P. Mutual Gaze During Early Mother–Infant Interactions Promotes Attention Control Development. *Child Development* 89.6 (2018), pp. 2230–2244. DOI: <https://doi.org/10.1111/cdev.12830>. URL: <https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/cdev.12830>.
- [2] Murray, L., Cooper, P., Creswell, C., Schofield, E. and Sack, C. The effects of maternal social phobia on mother–infant interactions and infant social responsiveness. *Journal of Child Psychology and Psychiatry* 48.1 (2007), pp. 45–52. DOI: <https://doi.org/10.1111/j.1469-7610.2006.01657.x>. URL: <https://acamh.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-7610.2006.01657.x>.
- [3] Stein, A., Gath, D. H., Bucher, J., Bond, A., Day, A. and Cooper, P. J. The Relationship between Post-natal Depression and Mother–Child Interaction. *British Journal of Psychiatry* 158.1 (1991), pp. 46–52. DOI: 10.1192/bjp.158.1.46. URL: <https://doi.org/10.1192/bjp.158.1.46>.
- [4] Lev-Enacab, O., Sher-Censor, E., Einspieler, C., Daube-Fishman, G. and Beni-Shrem, S. The Quality of Spontaneous Movements of Preterm Infants: Associations with the Quality of Mother–Infant Interaction. *Infancy* 20.6 (2015), pp. 634–660. DOI: 10.1111/j.1469-7610.2009.02066.x. URL: <https://doi.org/10.1111/j.1469-7610.2009.02066.x>.
- [5] Hertenstein M, J. Touch: Its Communicative Functions in Infancy. *Human Development* 45 (2002), pp. 70–94. DOI: <https://doi.org/10.1159/000048154>.
- [6] Beebe, B., Messinger, D., Bahrnick, L., Margolis, A., Buck, K. and Chen, H. A systems view of mother-infant face-to-face communication. *Developmental psychology* 52.4 (2016), pp. 556–571. DOI: 10.1037/a0040085. URL: <https://doi.org/10.1037/a0040085>.
- [7] Trehub, S. E., Ghazban, N. and Corbeil, M. Musical affect regulation in infancy. *Annals of the New York Academy of Sciences* 1337.1 (2015), pp. 186–192. DOI: 10.1111/nyas.12622. URL: <https://doi.org/10.1111/nyas.12622>.
- [8] Abu-Zhaya, R., Seidl, A. and Cristia, A. Multimodal infant-directed communication: how caregivers combine tactile and linguistic cues. *Journal of Child Language* 44.5 (2017), pp. 1088–1116. DOI: 10.1017/S0305000916000416. URL: <https://doi.org/10.1017/s0305000916000416>.

- [9] Bahrick, L. E. and Lickliter, R. Intersensory redundancy guides early perceptual and cognitive development. *Advances in child development and behavior* 30 (2002), pp. 153–87. DOI: 10.1016/s0065-2407(02)80041-6. URL: [https://doi.org/10.1016/s0065-2407\(02\)80041-6](https://doi.org/10.1016/s0065-2407(02)80041-6).
- [10] Tonsen, M., Baumann, C. and Dierkes, K. A High-Level Description and Performance Evaluation of Pupil Invisible. *ArXiv abs/2009.00508* (2020).
- [11] *Pupil Labs: Pupil Invisible*. URL: <https://pupil-labs.com/products/invisible/> (visited on 04/17/2022).
- [12] *Pupil Labs: Pupil Core*. URL: <https://pupil-labs.com/products/core/> (visited on 04/17/2022).
- [13] Hafiz, A. M. and Bhat, G. M. A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval* 9.3 (2020), pp. 171–189. DOI: 10.1007/s13735-020-00195-x. URL: <https://doi.org/10.1007/s13735-020-00195-x>.
- [14] Wu, Y., Kirillov, A., Massa, F., Lo, W. Y. and Girshick, R. *Detectron2: A PyTorch-based modular object detection library*. Oct. 10, 2019. URL: <https://ai.facebook.com/blog/-detectron2-a-pytorch-based-modular-object-detection-library-/> (visited on 04/17/2022).
- [15] *Detectron2: COCO Instance Segmentation Baselines with Mask R-CNN*. July 21, 2021. URL: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#coco-instance-segmentation-baselines-with-mask-r-cnn (visited on 04/17/2022).
- [16] *Detectron2: New baselines using Large-Scale Jitter and Longer Training Schedule*. July 21, 2021. URL: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#new-baselines-using-large-scale-jitter-and-longer-training-schedule (visited on 04/17/2022).
- [17] *Detectron2: LVIS Instance Segmentation Baselines with Mask R-CNN*. July 21, 2021. URL: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#lvis-instance-segmentation-baselines-with-mask-r-cnn (visited on 04/17/2022).
- [18] He, K., Zhang, X., Ren, S. and Sun, J. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: <https://arxiv.org/abs/1512.03385>.
- [19] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. *Feature Pyramid Networks for Object Detection*. 2016. DOI: 10.48550/ARXIV.1612.03144. URL: <https://arxiv.org/abs/1612.03144>.
- [20] *Detectron2: Common settings for COCO Models*. July 21, 2021. URL: https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md#common-settings-for-coco-models (visited on 04/17/2022).

- [21] Xu, J., Pan, Y., Pan, X., Hoi, S., Yi, Z. and Xu, Z. *RegNet: Self-Regulated Network for Image Classification*. 2021. DOI: 10.48550/ARXIV.2101.00590. URL: <https://arxiv.org/abs/2101.00590>.
- [22] Ren, S., He, K., Girshick, R. B. and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015).
- [23] He, K., Gkioxari, G., Dollár, P. and Girshick, R. *Mask R-CNN*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.
- [24] Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T. Y., Cubuk, E. D., Le, Q. V. and Zoph, B. *Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation*. 2020. DOI: 10.48550/ARXIV.2012.07177. URL: <https://arxiv.org/abs/2012.07177>.
- [25] *Detectron2: Configuration of model mask_rcnn_R_50_FPN*. Oct. 10, 2019. URL: https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml (visited on 04/17/2022).
- [26] *Detectron2: Configuration of model mask_rcnn_R_101_FPN*. Oct. 10, 2019. URL: https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_101_FPN_3x.yaml (visited on 04/17/2022).
- [27] *Detectron2: Configuration of model mask_rcnn_X_101_32x8d_FPN*. Oct. 10, 2019. URL: https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_X_101_32x8d_FPN_3x.yaml (visited on 04/17/2022).
- [28] *Detectron2: Configuration of model mask_rcnn_R_101_FPN_LSJ*. June 9, 2019. URL: https://github.com/facebookresearch/detectron2/blob/main/configs/new_baselines/mask_rcnn_R_101_FPN_400ep_LSJ.py (visited on 04/17/2022).
- [29] *Detectron2: Configuration of model mask_rcnn_regnetx_4gf_dds_FPN_LSJ*. June 9, 2019. URL: https://github.com/facebookresearch/detectron2/blob/main/configs/new_baselines/mask_rcnn_regnetx_4gf_dds_FPN_400ep_LSJ.py (visited on 04/17/2022).
- [30] *Optitrack: Prime 17W*. URL: <https://optitrack.com/cameras/prime-17w/> (visited on 04/17/2022).
- [31] *Optitrack: Prime 41*. URL: <https://optitrack.com/cameras/prime-41/> (visited on 04/17/2022).
- [32] *Gaze Datum Format*. Mar. 22, 2022. URL: <https://docs.pupil-labs.com/developer/core/overview/#gaze-datum-format> (visited on 03/23/2022).
- [33] *Pupil tutorials: Frame Extraction and Gaze Visualization*. Jan. 26, 2021. URL: https://github.com/pupil-labs/pupil-tutorials/blob/master/07_frame_extraction.ipynb (visited on 04/17/2022).

- [34] *COCO dataset: Detection Evaluation*. Aug. 25, 2021. URL: <https://cocodataset.org/#detection-eval> (visited on 04/17/2022).
- [35] Surasak, T., Takahiro, I., Cheng, C.-h., Wang, C.-e. and Sheng, P.-y. Histogram of oriented gradients for human detection in video. *2018 5th International Conference on Business and Industrial Research (ICBIR)*. 2018, pp. 172–176. DOI: 10.1109/ICBIR.2018.8391187.
- [36] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60.2 (2004), pp. 91–110. URL: <https://doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [37] Lucas, B. D. and Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. Vol. 81. *IJCAI'81*. Morgan Kaufmann Publishers Inc., Apr. 1981, pp. 674–679.
- [38] Shi, J. and Tomasi. Good features to track. *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. DOI: 10.1109/CVPR.1994.323794. URL: <https://doi.org/10.1109/cvpr.1994.323794>.
- [39] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. DOI: 10.48550/ARXIV.2103.14030. URL: <https://arxiv.org/abs/2103.14030>.
- [40] Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F. and Guo, B. *Swin Transformer V2: Scaling Up Capacity and Resolution*. 2021. DOI: 10.48550/ARXIV.2111.09883. URL: <https://arxiv.org/abs/2111.09883>.

APPENDIX A: EXTRACTING GAZE DATA USING SOFTWARE PUPIL PLAYER

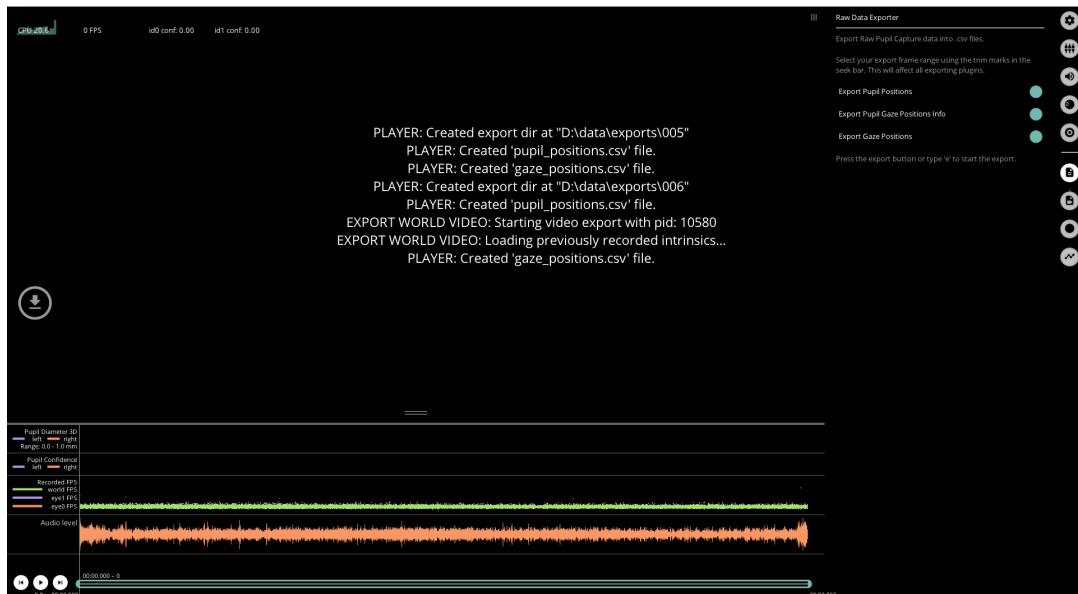


Figure A.1. Pupil Player screen.

This is what the screen of the application Pupil Player exporting gaze data looks like. Making sure in "Raw Data Exporter", "Export Gaze Positions" is enabled. Two options above it are unnecessary, but at least the middle option is nice to have since it explains the column names of exported data file. "Download" icon on the left side of the screen or button "e" on the keyboard can then be used to export the data. The exported directory name will be the new largest 3-digit number in the "exports" folder.

Regarding gaze visualization made by Pupil Player, in "Plugin Manager" from the right sidebar, one can add for example "Vis Circle", "Vis Cross", and "Vis polyline" by pressing "Add" button next to them. These newly added plugins will appear below the horizontal separator.

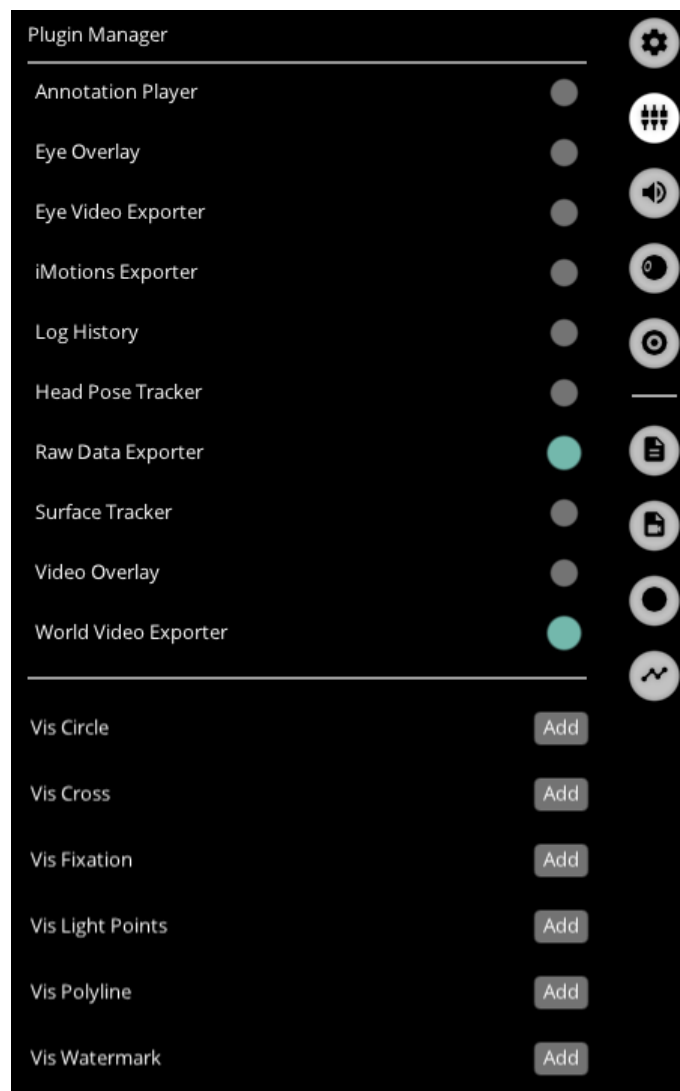


Figure A.2. Plugin manager of Pupil Player.

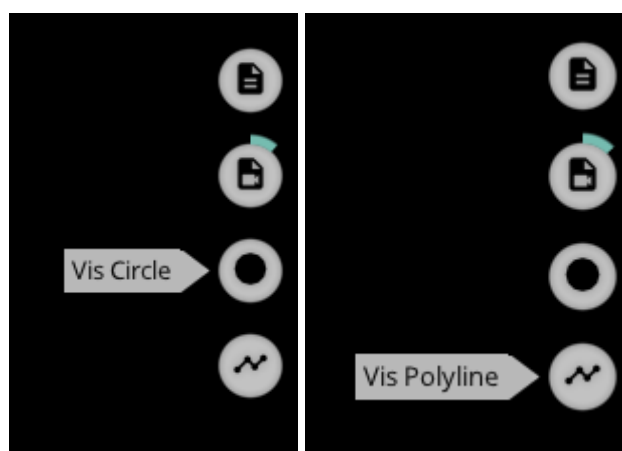


Figure A.3. Added plugins appear at the bottom of listed plugins.

APPENDIX B: PROCESSING GAZE DATA USING PYTHON

```

1  import numpy as np
2  import pandas as pd
3
4
5  def clipping_gaze(denormalized_gazes: pd.Series, frame_dimension:
    int):
6      """Returns gazes in denormalized coordinate (float numbers) to
        the indices on the video frame.
7
8      Args:
9          denormalized_gazes (Series): Gazes in denormalized
            coordinate.
10         frame_dimension (int): Dimension of the frame to clip the
            coordinates to avoid index out of range.
11
12     Return:
13         list: List of gaze coordinates that can be used as indices
            on the video frame.
14     """
15
16     return (
17         np.clip(
18             a=np rint((denormalized_gazes - 1).to_numpy()),
19             a_min=0,
20             a_max=frame_dimension - 1
21         )
22         .astype(int)
23         .tolist()
24     )
25
26 def cal_gaze_in_frame(gaze_data: pd.DataFrame, frame_size: tuple):
27     """Converts gazes in normalized coordinate to video coordinate.
28
29     Args:
30         gaze_data (Dataframe): Gaze dataframe.
31         frame_size (tuple of int): Width and height of the video
            frame.
32

```

```

33     Return:
34         np.ndarray: Array of tuple of gaze coordinates.
35     """
36
37     width, height = frame_size
38     x_denormalized = gaze_data.loc[:, "norm_pos_x"] * width
39     y_denormalized = (1 - gaze_data.loc[:, "norm_pos_y"]) * height
40     x_world_coor = clipping_gaze(x_denormalized, width)
41     y_world_coor = clipping_gaze(y_denormalized, height)
42
43     return np.array(zip(x_world_coor, y_world_coor))

```

Program B.1. *Converting gazes in normalized coordinate to video coordinate*

The main function handling the conversion is `cal_gaze_in_frame` which takes the normalized gaze data and frame size as inputs. The normalized x and y coordinates are multiplied with the width and height of the frame, and then they are clipped by the function `clipping_gaze`. However, the sole purpose of this function is to make sure the previous obtained coordinates are within the video frame. Hence, only gazes with coordinates that fall outside the frame range of either width or height are clipped, otherwise gazes remain unchanged after the function.