



EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPT. OF ARTIFICIAL INTELLIGENCE

Composite AI for Request Detection during Construction Task in ADOS-2 Module 3

Supervisor:

Dos Santos Melício Bruno Carlos

PhD student

Author:

Xiang Linyun

Computer Science MSc

Budapest, 2023

Abstract

Autism Spectrum Disorder (ASD) is a developmental condition that affects individuals' social interaction and communication skills. Detecting and analyzing verbal and non-verbal cues is one of the fundamental approaches to having a better understanding of autism behaviors. However, the complexity and variability of such cues cannot be robustly tackled by a single Deep Learning (DL) method. In this paper, we propose a Composite AI (CAI) framework, that combines DL with traditional AI methods, to detect verbal and non-verbal requests during the Construction Task of the Autism Diagnostic Observation Schedule-Second Edition (ADOS-2) Module 3. Our framework comprises feature extraction, episode detection and request detection. The extracted features allow for an understanding of the environment, humans and context. They are then combined using rule-based systems in order to restrict the request detection only to key moments called "episodes", resulting in zero false positives. Precisely, the hands and puzzle pieces features are crucial indicators in detecting episodes and requests. Our main contributions include hand pose estimation, puzzle piece segmentation and the overall composite AI framework.

Contents

1	Introduction	3
2	Background	5
2.1	The Autism Diagnostic Observation Schedule-Second Edition (ADOS-2)	5
2.2	Composite AI	6
3	Related work	8
3.1	Hand Pose Estimation	8
3.2	Puzzle Pieces Segmentation	9
3.3	Construction Task in ADOS-2	10
4	Composite AI Framework for Request Detection	12
4.1	Feature Extraction	12
4.1.1	Person and Object Tracking	13
4.1.2	Body Pose Estimation	13
4.1.3	Hand Pose Estimation and Tracking	14
4.1.4	Puzzle Pieces Segmentation	19
4.1.5	Speech Recognition	21
4.1.6	Combined Features	22
4.2	Episode Detection	23
4.3	Request Detection	24
5	Experiments	27
5.1	Dataset	27
5.1.1	Data Pre-processing	27
5.2	Hand Detection and Tracking	28
5.3	Puzzle Pieces Segmentation	30
5.3.1	MiVOS semi-automated annotation	31

5.3.2 Mask2Former fine-tuning	35
6 Results and Discussion	38
7 Conclusion	44
Acknowledgements	45
Bibliography	46
List of Figures	50
List of Tables	52

Chapter 1

Introduction

According to Centers for Disease Control and Prevention (2020)[1], the estimate of Autism Spectrum Disorder (ASD) prevalence in the United States children is 1 in 36 (or 2.78%). This represents a significant increase from the 2016 data, which showed a prevalence of 1 in 54 (or 1.9%). The rise in ASD prevalence among American children is a cause for concern, action is needed to address this issue and improve the quality of life for those with autism. Finding effective interventions is the most pressing challenge that needs to be overcome.

Autism is a neurodevelopmental disorder that affects an individual's communication, social interaction, and behavior. Children with ASD (cwASD) often have difficulties in understanding and using spoken language, and engaging in social interactions, and may display repetitive behaviors. Therefore, observing, analyzing, and understanding subtle expressions of autism is key for the diagnosis.

AI can be used as a valuable tool for diagnosing ASD by providing more accurate and objective assessments, identifying potential signs of ASD at an earlier age, and reducing the time and cost associated with traditional diagnostic methods.

Thus, we propose a Composite AI framework that combines rule-based systems and various Deep Learning methods, such as body and hand pose estimation, object detection and tracking, and speech recognition to analyze the behavior of patients during a standardized Autism diagnostic test, called ADOS-2. Specifically, we analyze whether patients make requests during a puzzle completion task. With our proposed framework, we are able to detect verbal and gestural requests, which are significant to the ASD analysis.

For the remainder of this thesis, we provide all the necessary knowledge to fully

understand the structure and functioning of the proposed method. In section 2, ADOS-2 and Composite AI are specifically introduced. In section 3, recent works related to our research topic are reviewed. Section 4 describes the method proposed in this paper, including the tools being used, the different components, and their purposes. Furthermore, the experiments and results are shown in sections 5 and 6. Finally, in section 7, the potential applications of the framework are highlighted, and final remarks are provided.

Chapter 2

Background

2.1 The Autism Diagnostic Observation Schedule-Second Edition (ADOS-2)

The Autism Diagnostic Observation Schedule-Second Edition (ADOS-2) is a standardized assessment tool that helps clinicians diagnose autism spectrum disorder (ASD) in individuals of all ages. It is widely used by clinicians, researchers, and educators to assess social interaction, communication, and repetitive behavior patterns in individuals who exhibit potential autism signs.

The ADOS-2 consists of a series of structured tasks and activities that are designed to elicit behaviors that are characteristic of autism. The tasks include activities such as playing with toys, conversation, and social interactions, and are designed to assess the individual's ability to initiate and respond to social interaction, use language in a social context, and engage in imaginative play.

ADOS-2 has the following different modules targeting different age groups, languages, and developmental levels:

- Module T: For toddlers between 12 and 30 months of age.
- Module 1: For individuals who are 31. months and older, with pre-verbal or single words
- Module 2: For individuals with phrase speech.
- Module 3: For children/adolescents with fluent speech.
- Module 4: For adolescents/adults who have functional communication skills.

In this paper, the focus is on Module 3. Within Module 3, the activity of our interest is the Construction Task. In this task, the clinician presents the patient with a shape drawn on a piece of paper and a set of eight puzzle pieces. However, the clinician does not provide all eight pieces at once. Rather, they offer a subset of two to three pieces and wait for the patient to request more. This approach enables the clinician to observe and assess the patient’s behavior, as the manner in which they make the request (if at all) provides valuable insights. The request may take the form of a verbal or gestural cue, or the patient may attempt to acquire additional pieces without making any overt request. The clinician evaluates these various forms of requests differently during the ADOS-2 evaluation. The Construction Task is deemed completed once the patient has successfully placed all eight puzzle pieces within the shape on the paper. However, this can only be achieved by requesting more pieces as required. The process of receiving the initial set of puzzle pieces, placing them, and requesting more provides numerous cues that can be analyzed to evaluate the patient’s level of autism.

2.2 Composite AI

Composite AI refers to the integration of multiple AI models with the aim of enhancing the system’s robustness, accuracy, and adaptability, thereby enabling the creation of more powerful AI systems that can operate effectively with limited data. It combines different AI components, such as machine learning, natural language processing, computer vision, and robotics, to solve complex problems that require a more comprehensive understanding of the data.

The idea behind composite AI is that different AI technologies can complement each other and work together to achieve better results than any single AI technology could achieve alone.

Composite AI offers several advantages, including its ability to adapt to changing environments and situations. By combining multiple AI components, the system can be designed to be more flexible and resilient, enabling it to learn from new data and adjust its behavior accordingly. Moreover, it is an effective solution for addressing challenges such as data scarcity, model uncertainties, and explainability.

Composite AI has recently emerged as a powerful tool for solving complex problems in a wide range of industries, including healthcare, finance, manufacturing, and

transportation.

In our case, Composite AI is essential due to the limited data available, and the precise estimations required in medical applications. By combining Deep Learning methods with expert knowledge, the system can be tailored to meet the specific needs of different applications. This approach allows researchers to select the most appropriate AI technologies for each part of the system, increasing its flexibility and adaptability to changing requirements.

Chapter 3

Related work

The conventional screening and diagnosis of Autism Spectrum Disorder (ASD) typically involve a manual evaluation conducted by clinicians, which can be a time-consuming process. However, the emergence of AI technologies provides an opportunity for clinicians to integrate these technologies into their evaluations, resulting in a more efficient and accurate diagnostic process. Nonetheless, achieving accurate diagnoses of ASD with AI requires the combined expertise of both human understanding and human-machine cooperation.

This work proposes a Composite AI framework that includes a combination of different AI methods. Our main contributions include a new hand pose estimation method, a puzzle piece segmentation method and the overall system for aiding clinicians during the Construction Task of ADOS-2. Therefore, in this section, we only review the areas of development which are related to our contributions.

3.1 Hand Pose Estimation

Hand pose estimation is currently a widely researched area, and numerous methods have been developed. Tang et. al. (2021) [2] proposed a Real-time 3D Hand-Mesh Reconstruction method that includes joint and mesh detection, and fine-tuning with an offset mesh for mesh-image alignment. Nevertheless, this method is limited to predicting a single-hand pose and faces difficulties with occlusions. In cases where the hand is partially occluded, the method attempts to estimate the full hand since it lacks a concept of visibility. To overcome this limitation, we leverage MediaPipe’s [3] visibility scores.

A method named TOCH for refining incorrect 3D hand-object interaction sequences using a data prior is proposed by Zhou et. al. (2022) [4]. The key component of the method is a point-wise object-centric representation that encodes the hand position relative to the object. While this method performs adequately for slow hand-object interactions, it encounters difficulties in situations where patients are constantly manipulating puzzle pieces within a confined space. The continuous hand movements result in a significant amount of motion blur which can lead to imprecise pose estimation. To address this issue, we have incorporated tracking software [5] within our method.

Most recently, Anchor-to-Joint Transformer Network for 3D Interacting Hand Pose Estimation from a Single RGB Image named A2J-Transformer was proposed by Jiang et. al. (2023) [6]. The method is quite robust to blurriness, but it struggles with occlusion, especially when the object is big enough to cover at least half of the hand. Our combination of methods which includes tracking helps in dealing with this problem.

Unfortunately, all the aforementioned methods are not strong enough to detect hands precisely, especially in our case where the hands are interacting with objects. The primary factors behind those limitations are occlusions and the complex and subtle hand movements, which are difficult to detect with a single method. Thus, our composite AI method for hand detection is introduced in this paper.

3.2 Puzzle Pieces Segmentation

Image segmentation is a well-advanced field that has benefited from advances in deep learning models. Usually, these models are trained with datasets containing a finite set of day-to-day objects. However, when accessing children with autism, specifically in ADOS-2, there are special objects which are used. There are a few studies that use segmentation and detection techniques for ADOS-2 objects. Presecan et. al. [7] presented a Deep Learning-based approach in object classification for ADOS-2. The study was centered on the "free play" task, which involves the use of various objects such as plates, dolls, dump trucks, and other similar items. However, none report the puzzle piece object, which is of interest in the Construction Task discussed in this paper.

In the Construction Task, patients have to complete a puzzle. Thus, it is needed to segment these pieces in order to observe how the patient interacts with them. The most relevant work we could find regarding this subject matter is proposed by Sárkány et. al. [8]. Optical flow was utilized in this study to ascertain the disparity between two consecutive frames, enabling the detection of changes in puzzle pieces' positioning. However, the methodology adopted in this research is unsuitable for our purposes, as we require knowledge of the number of puzzle pieces in order to determine if there was a request, as well as the completion status of the task by the patient.

Henceforth, we resolved to train our own instance segmentation model. To achieve this, we conducted extensive research of relevant literature to identify the most advanced segmentation models presently available. From the time of this writing, our search yielded models such as Mask R-CNN [9], Swin-L [10], and Mask2Former [11]. After experimenting with these models, we determined that Mask2Former demonstrated the highest average precision on COCO val2017, which comprised 80 categories, surpassing the performance of the other methods. Consequently, we opted to employ Mask2Former as our segmentation method, by training it to segment ADOS-2 puzzle pieces.

3.3 Construction Task in ADOS-2

Several studies [12, 13, 14, 15, 16] on autism with AI assistance have been conducted. Most of them focus on using AI with different types of features including speech, vision (facial expressions, head pose, gaze), questionnaires, among others.

For instance, the previously mentioned method proposed by Presecan et. al. [7] uses vision features to recognize the two main actions including putting down and picking up an object. They note that grabbing or releasing an object involves a temporary stop of the hand's movement. By combining these ideas, the process is to propose several stationary hand candidates initially and screening by observing whether there is a difference between the two frames showing the (dis)appearing object. In our case, grasping and releasing an object is only important in scenarios when a request does not happen. Instead, it is of higher relevance to detect communicative signs, such as verbal (spoken language) and non-verbal (gestures) in order

to detect a puzzle piece request by the patient. Detecting such signs increases the complexity of the problem, from which we tackle with a combination of methods.

A 3D Human Sensing Action and Emotion Recognition of Autism children was proposed by Marinoiu et. al. [17]. They recognize the actions of the patients by fine-tuning a pose estimation model which was named DMHSPV, using a shape-based volumetric refinement based on the SMPL body representation. They classify gestures such as pointing, grabbing and interacting with objects, which are relevant for our purpose. However, our work requires a more extensive set of gestures related to requesting puzzle pieces. For instance, gestures with the finger expressing the number of puzzle pieces needed.

Typically, research in the field of Autism Spectrum Disorder (ASD) aims to achieve precise diagnostic outcomes, often with the objective of either enhancing or replacing the decisions made by clinicians. For instance, Cai et. al. presented an advanced deep learning framework for the video-based diagnosis of ASD, which sought to achieve greater accuracy in diagnosing the condition [18]. However, our research takes a different approach, seeking to provide additional cues for clinicians to aid their decision-making process, specifically by detecting and classifying patient requests. This methodology has broader applications beyond the specific context of ADOS-2 and can be applied more generally in scenarios where social interaction occurs.

Chapter 4

Composite AI Framework for Request Detection

In this section, the overall Composite AI framework is presented (see Figure 4.1). We describe in detail each individual component of the framework. Various features are extracted and combined to identify key moments on the input video, referred to as episodes. Furthermore, the integration of deep learning methodologies with rule-based systems facilitates the detection of requests in a robust and precise manner. The primary contributions, which include the proposed hand pose estimator, trained segmentation model, and the overall system, are described in detail.



Figure 4.1: Overall Composite AI framework with three main modules: feature extraction, episode detection, and request detection.

4.1 Feature Extraction

The feature extraction module is responsible for extracting relevant features from the input. These features are the building blocks of the system, as they are utilized by the other components. However, it is crucial to note that a single feature may not possess sufficient information to effectively solve the task at hand. It is through the combination of all the features that a complex system capable of robustly solving the problem is established. Figure 4.2 shows all the necessary features in order to

understand the environment, the human behavior and the context. Each feature is described in the next subsections.

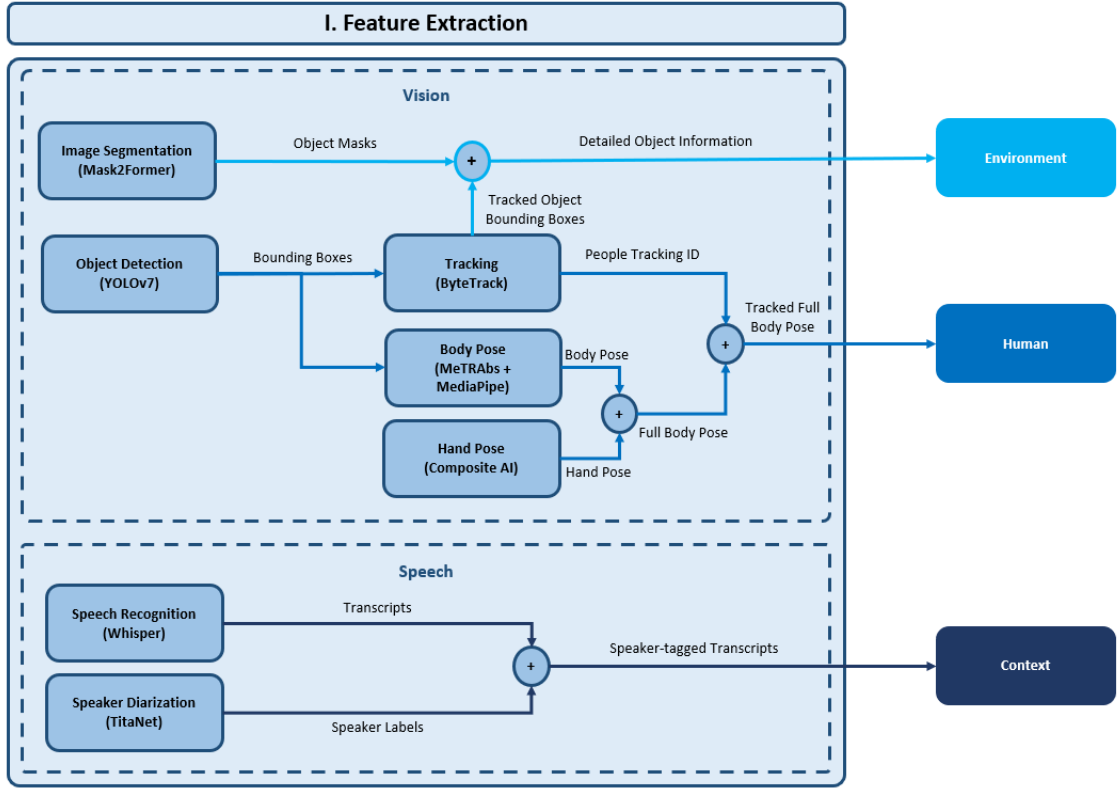


Figure 4.2: Feature Extraction module comprises a vision and speech component which uses multiple Deep Learning methods to extract features representing the environment, human and context.

4.1.1 Person and Object Tracking

The initial phase involves detecting and tracking both people and objects. YOLOv7 [19] was used for the detection. However, as YOLOv7 does not generate bounding box IDs, it fails to differentiate among distinct individuals and objects. For that reason, ByteTrack [20] was employed to track each individual and object by associating their detection with a unique ID.

4.1.2 Body Pose Estimation

In order to understand human behavior, we extract further features, specifically, the 3D body pose. The YOLOv7 and ByteTrack provide tracked bounding boxes, which enable us to estimate multi-person poses. To obtain the 3D body joint positions, we utilized MeTRAbs v2 [21] and MediaPipe [22]. Although MeTRAbs has

a more accurate pose estimation than MediaPipe, it faces difficulties with occlusion due to the fact that during the sessions patients are mainly sitting. To address such limitation, the visibility scores from MediaPipe were integrated. This results in an accurate and occlusion-aware body pose estimation.

4.1.3 Hand Pose Estimation and Tracking

Hands play an indispensable role in daily life, they can be the way of communication and interaction, and also the outward manifestation of mental state. Thus, hand movement is one of the essential indicators of ADOS-2. By detecting and tracking hands, we could obtain more imperceptible cues which might be ignored by human eyes. A modified version of Hartyányi method [23] is proposed. It consists of the following parts:

- Hand segmentation
- Body keypoints
- Tracking
- Hand pose

Hand segmentation

Mask R-CNN [9] is the first and core step in the method, we fine-tuned it to segment hands.

The method receives an input image and proposes several regions potentially containing hands. We only keep masks if the confidence score is at least 95% (refer to Figure 4.3). Despite this strict threshold, the model still produces some false detections. For example, in some cases, it might segment the face, because the dataset it was trained on contains samples with hands touching the face. Moreover, under certain circumstances, MaskR-CNN fails to detect both hands, for instance, when the hands move swiftly and result in a blur or when they are partially obscured. Consequently, additional measures must be implemented to address such inaccurate detections.



Figure 4.3: An example output of the MaskR-CNN model, with both hands detected

Body keypoints

To address some of the limitations of the Mask R-CNN, body pose estimation is included. In this step, Mediapipe is employed to estimate the body pose, which provides 33 keypoints representing the entire body (refer to Figure 4.4). However, in our scenario, only eight of these keypoints, specifically the wrist, thumb, index, and pinky of each side, are deemed necessary to solve the false hand detections.

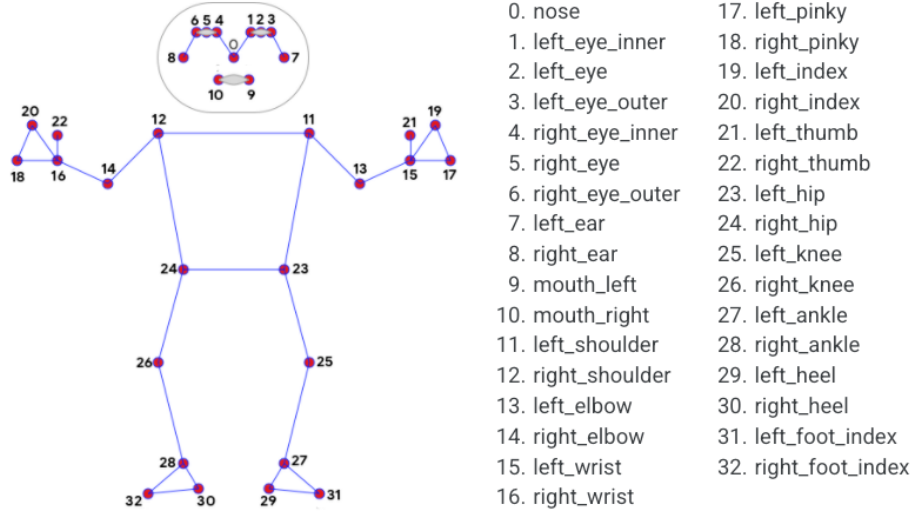


Figure 4.4: An body pose estimation example from Mediapipe, including 33 returned keypoints with names listed. The image is from the Mediapipe site [22].

As previously stated, certain cases may contain erroneously detected hand masks. Upon processing Mediapipe, the hand keypoints are obtained. Subsequently, the center points of the detected masks and hand keypoints are computed separately.

The L2 distance between the two center points is then calculated with a specific threshold of 40. Thus, only those masks which are within the ideal distance to the hand keypoints are kept. The example is shown in Figure 4.5.

Besides, though the clinician and the patient are both included in the video, our focus is only on the patient. Therefore, only one body pose is required. In the episode detection subchapter, we show how to identify the patient.



Figure 4.5: An example of misdetection from MaskR-CNN (middle) discarded by calculating the L2 distance (right) with the assistant of hand keypoints (left).

With this approach, we solve the false detections from Mask-RCNN. However, MediaPipe can still miss the hand pose if it is occluded. For that reason, tracking is used.

Tracking

To address the issue of missing detections, due to blurriness and occlusions, Modular interactive Video Object Segmentation (MiVOS) [5] is introduced for tracking as the next submethod, which aims to produce high-quality segmentation of a target object instance across an input video sequence.

For effective tracking, the essential step is to separate the left and right hands. Here the output from the body pose estimation is used to differentiate between the left and right hands, then the tracking is processed with the masks of both hands separately, which means that the entire process needs to be run twice for each frame, thereby inevitably increasing the time cost two-fold.

The MiVOS tracking ideally solves missing mask problems arising from either the original MaskR-CNN or due to the improper estimation of hand poses. It is noteworthy, the model is capable of tracking through time (back and forth). In the event of a missing hand detection in a specific frame while being present in adjacent frames, MiVOS can infer the position of the hand in the missing frame. An output of tracking for the mask missing frame and adjacent frames is demonstrated in Figure 4.6.

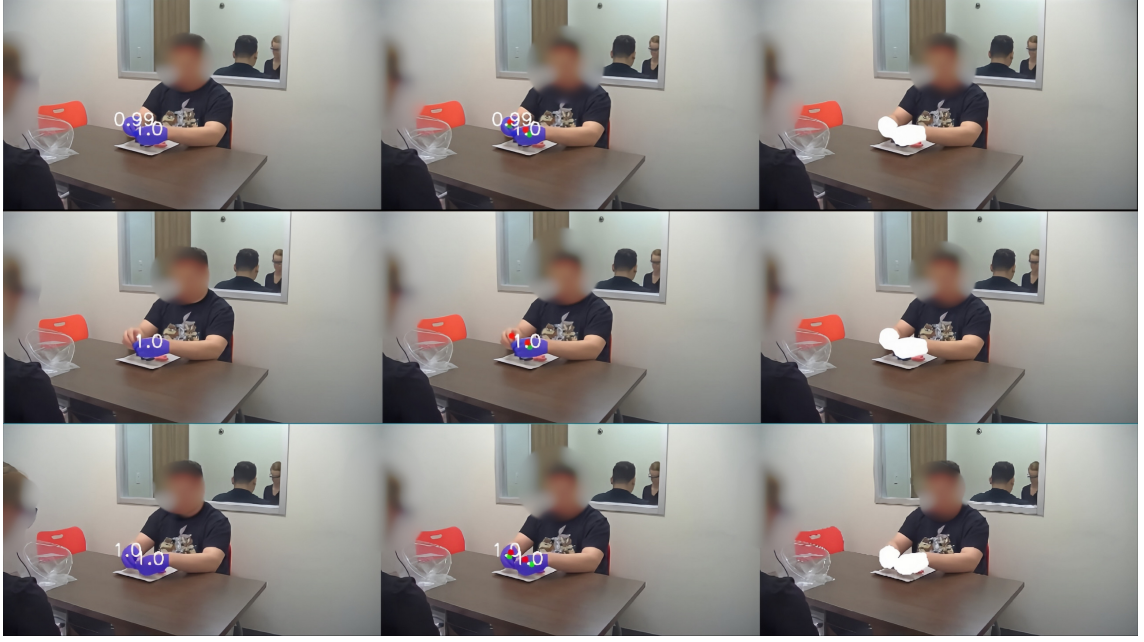


Figure 4.6: A comparison of outputs from the MaskR-CNN and tracking. The middle row shows the mask missing frame, the output from MaskR-CNN only has one hand detected because of partially hands overlapping (left). The MiVOS tracking generates an output with both hands detected correctly (right). The first and third rows are the adjacent frames with both hand masks.

Hand keypoints

Limiting the analysis to hand movements alone is insufficient, as fingers constitute essential components of the evaluation. In the context of autism, even the slightest of movements is crucial, serving as a key reference for clinicians.

To get more specific hand information, MediaPipe Hands [3] is applied to detect hand key points. The model returns 21 hand keypoints, the illustration can be seen in Figure 4.7.

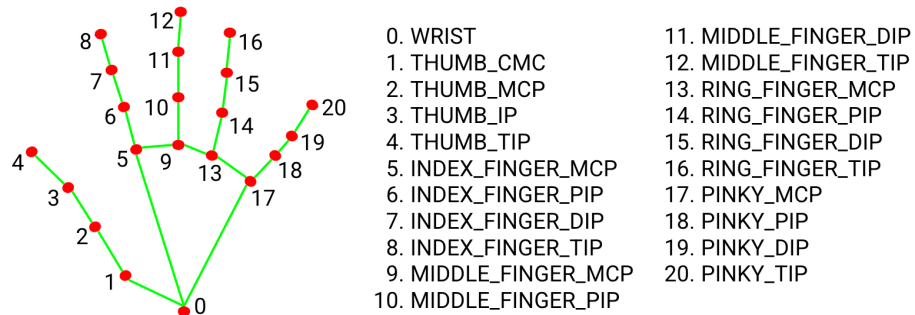


Figure 4.7: 21 hand landmarks from Mediapipe Hands site [3].

In order to obtain more precise results, the cropped image with the shape of masks is taken as the input. In this case, the Mediapipe Hands receives the image

solely with the hands part, which immensely improves the accuracy of detection. The demonstration can be seen in Figure 5.3.



Figure 4.8: An example of the hand keypoints output

Full hand detection pipeline

Putting everything together, the overall architecture of the hand detection system is demonstrated in figure 4.9. This figure provides a comprehensive view of the system and its components, including the inputs, processing steps, and outputs. Additionally, it allows for a better understanding of the relationship between the different stages and how they contribute to achieving the final result.

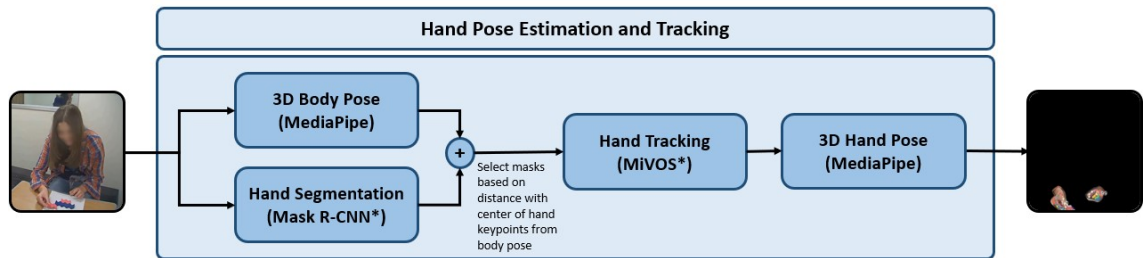


Figure 4.9: The architecture of full hand detection pipeline

An example concatenated output with all steps is illustrated in Figure 4.10. In the order from left to right, the images are respectively: Body pose estimation with Mediapipe, hand masks from MaskR-CNN, selected hand masks with the assistance

of body estimation, tracking with MiVOS and filling the frames that contain missing detections, and the hand keypoint detection based on the mask outputs from previous steps.

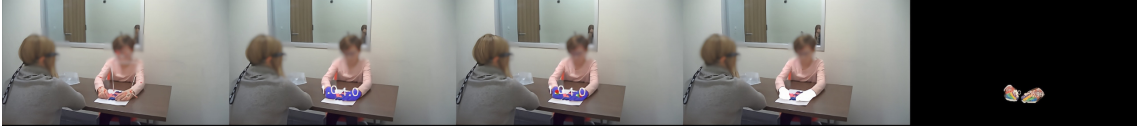


Figure 4.10: The final output of the full hand detection pipeline, which summarizes the different stages of the process.

4.1.4 Puzzle Pieces Segmentation

As mentioned in chapter 2, this paper mainly focuses on the Construction Task which involves filling a shape with puzzle pieces. Therefore, how to segment puzzle pieces precisely and efficiently is of immense significance in our case.

In the Construction Task, the eight pieces provided by the clinician are in the same shape with two different colors which are red and blue. Since the puzzle pieces have a unique shape, training a model to fulfill the segmentation task is required. This is achieved through a two-step process:

- Semi-Automated Annotation with Modular Interactive Video Object Segmentation (MiVOS) [5]
- Segmentation using Masked-attention Mask Transformer (Mask2Former) [11]

Semi-Automated Annotation with MiVOS

In order to get the annotation efficiently, a modified version of MiVOS [24] was used. It is a semi-automated annotation tool implemented in the following three steps: Initially, the user selects and interactively annotates one frame (e.g., using scribbles or clicks) to produce a mask. Then, the method generates segmentation for every frame in the video sequence. After that, the user examines the output quality, and if needed, starts a new “round” by correcting an erroneous frame with further interactions [24]. The process of annotation is shown in Figure 4.11.

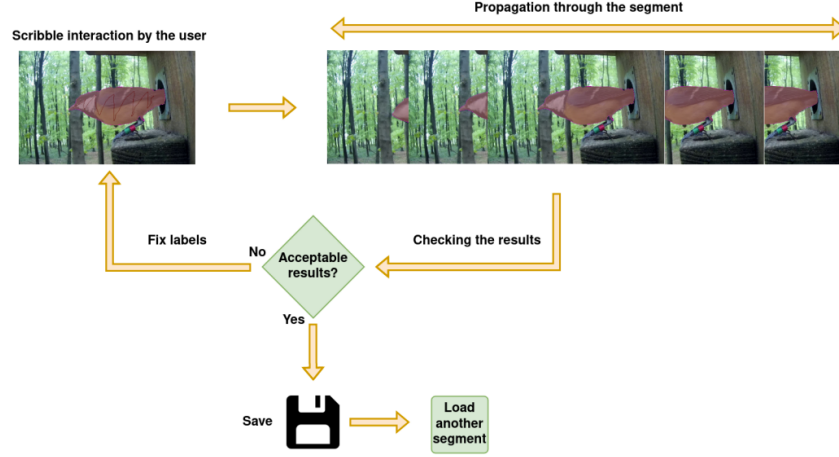


Figure 4.11: The example process of annotation from Szász’s Bachelor thesis [24].

Mask2Former

Mask2Former is introduced in this section. It is a new architecture that can be used for any image segmentation task (panoptic, instance, or semantic) and outperforms the current state-of-the-art methods by using masked attention to extract localized features. In addition, it is easy to train on specific tasks. The model is constitutive of a backbone, a pixel decoder, and a Transformer decoder. The architecture can be seen in Figure 4.12.

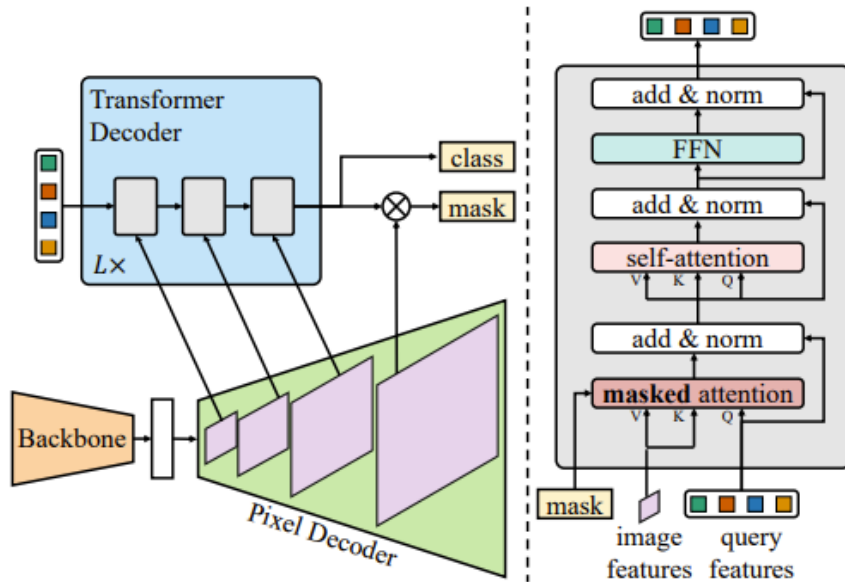


Figure 4.12: Mask2Former architecture [11]

Since the small puzzle pieces used in our case require precise segmentation, we

have retrained the model with annotated videos to improve its ability to detect and segment the puzzle pieces. The retrained model is then applied to all the videos in order to obtain puzzle information accurately and consistently. This allows us to analyze the behavior and movements of the patient in relation to the puzzle pieces in a more detailed and precise manner. The example is shown in Figure 4.13.



Figure 4.13: The upper image is the original frame, and the lower one is the output of Mask2Former with puzzle pieces segmented.

4.1.5 Speech Recognition

Apart from the previously mentioned methods, our approach also incorporates speech recognition. To recognize verbal requests made by the patient, the speech must be transcribed. The Whisper [25] model from OpenAI is employed to achieve this.

Whisper provides precise transcripts but it does not differentiate between the speakers. For that reason, a speaker diarization model, named TitaNet [26], was used to obtain speaker labels.

4.1.6 Combined Features

As the set up of the Construction Task, there are two persons in the video, to effectively focus on the patient, it is necessary to distinguish between the patient and clinician. To achieve this, we combine the different features with rules.

Clinician and Patient Identification

From the features, there are bounding boxes for each detected person and an ID associated with each. However, the IDs are just numbers and do not tell who is the clinician or the patient. Therefore, a rule-based system based on known knowledge from the task is used:

- Clinician: The clinician can be identified as the person who initiates the task by placing the puzzle pieces on the table and is also the last person to interact with the pieces, as they are responsible for packing them up at the end. Following this approach, the rules below were created:

If the table is empty at the start of the video and puzzle pieces appear afterward once the task starts, then whoever introduces the puzzle pieces, meaning whose hands put the pieces on the table first, is the clinician. However, if this first case fails, we can wait until the end of the video and whose hand was the last to touch the puzzle pieces will be the clinician. Then it can backtrack and associate each box with the clinician's hands.

- Patient: The patient is the person manipulating the puzzle pieces on the paper and spending the most time interacting with the pieces. Following this approach, the rules below were created:

If there are hands moving together with puzzle pieces within the bounding box of the paper, then those hands belong to the patient. In addition, for robustness, the patient can be identified as the person whose hands interact with the puzzle pieces the for the longest number of frames.

After identifying the patient and clinician, the next step is to detect the episodes. By analyzing their actions, we can determine the specific parts of the interaction.

4.2 Episode Detection

The episode detector has the function of identifying key moments in videos where a specific event happened. For that, the method should identify a start and end point of the event. For example, the method should be able to find when the task started and ended. There are 4 episodes:

- Start - denotes the start of the task
- Interaction - denotes the moment when the patient starts interacting with the puzzle pieces
- Request - denotes moments when we predict that a request might happen
- End - denotes the end of the task

Episode detection rules

The rule for detecting the four episodes are as follows:

- Episode Start: Monitor the puzzle pieces and save the frame number when more than one piece is first placed on the table. Mark this frame as the task's starting point.
- Episode Interaction: Once the task begins, the patient starts interacting with the puzzle pieces to fit them into the shape on the paper. Track the patient's hands and the pieces; when at least one hand and one piece move together within the paper's bounding box, record the frame number and mark it as the beginning of the interaction.
- Episode Request: The most critical episode is the request episode, where we estimate when a request may occur. To achieve this, we identify an interval [start point, end point] during which a request is likely to happen. The rules for this episode are as follows:
 - If the interaction with the puzzle pieces has already started, then count the number of puzzle pieces within a radius of the paper. Then, save the maximum number of pieces. Next, track the number of pieces throughout the task.

- If the number of puzzle pieces increases, record the frame number and call it "End". This means that a request has happened. Then, go back in the video, and track hands moving with puzzle pieces.
 - If hands stop moving together with pieces, before the "End" frame, then record the current frame number and call it "Start", which denotes that the patient is about to make a request.
 - Return the interval between "Start", which is when the patient stopped interacting with the puzzle pieces, and "End", which is when the puzzle pieces increased. [Start, End]
- Episode End: The task concludes when there are eight puzzle pieces on the paper and no hands are moving together with any piece.

4.3 Request Detection

Verbal requests detection

For detecting verbal requests, speech features are analyzed which would express the need for more puzzle pieces by the patient. The following features are used:

- Transcript - The whisper method was used to transcribe the videos, generating text from what was being said on the videos.
- Speaker Diarization [26] - a method provided by ARGUS Cognitive extracted speaker labels in order to be able to identify which speaker said what on the videos.
- Common Request Words - a dictionary containing the most common request words was used to determine whether what was said by the patient was a request or not: ["I", "can", "have", "mind", "if", "piece", "pieces", "need", "more", "some", "please", "one", "two", "three", "four", "want", "ask", "few", "puzzle"].
- First, the video is transcribed with Whisper method which produces text transcripts.
- Then, from the transcripts, what was said by the clinician was filtered out based on the speaker diarization time stamps and labels (Assume that the

clinician is always the first person to talk). Therefore, the text from the patient was kept only.

- Sentences from the patient that contain a certain number of occurrences of words present in the common request words vocabulary was selected only.
- Finally, only those sentences that happened in the request episode were kept.

These features were combined as follows (Figure 4.14):

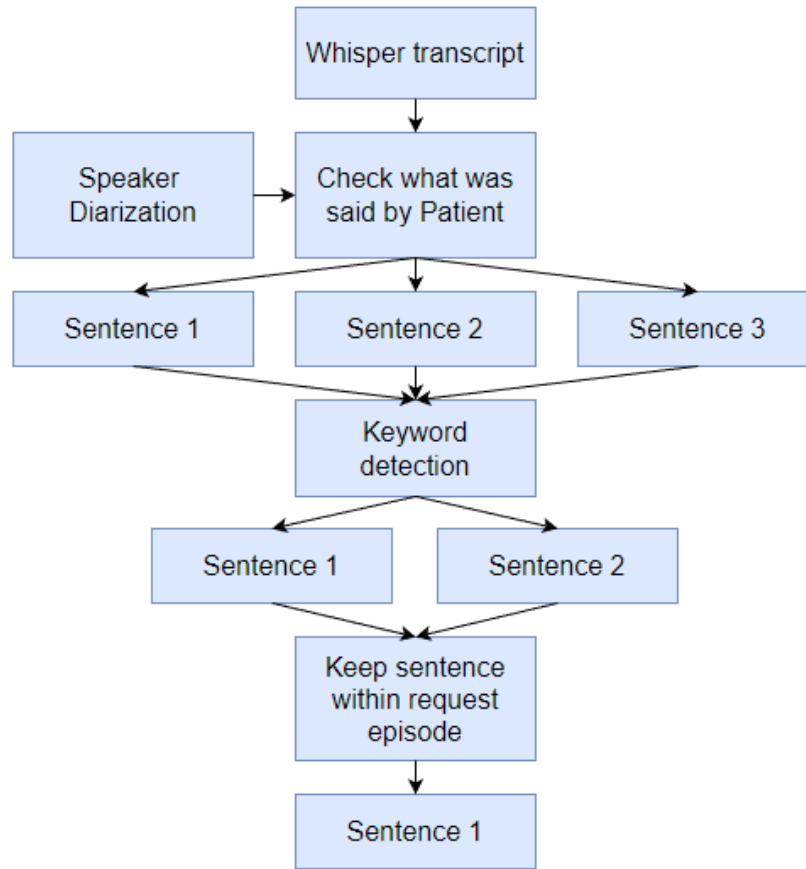


Figure 4.14: Pipeline for Verbal Request Detection

Gesture requests detection

To detect gesture requests, we use MediaPipe’s gesture recognition features. Within the detected request episode, typically 8 seconds, we employ a 2-second window. The following rules are created:

- If MediaPipe gesture recognition classifies at least one frame in the window as having a gesture, we consider the whole 2-second window as a single gesture.
- If gestures are the same across different windows, we treat them as one gesture.

- If the gestures differ, we use context from speech to decide the most appropriate gesture. For instance, a patient may verbally request for two pieces while showing the number two with their fingers. In those cases, we would select the "victory" label from MediaPipe's gestures. In the absence of context, we choose the gesture with the most occurrences or the highest confidence score.

Chapter 5

Experiments

5.1 Dataset

The original datasets provided by ARGUS and RUSH Medical School include two synchronous videos recorded by the wall camera and the testing camera separately. However, only the videos that come from the wall camera were used. 29 videos were chosen from the full video sessions of Module 3 ADOS-2, from which a smaller set containing only the Construction Task was created. For each of the videos of this smaller set, the following annotations were created:

- request time: [hh:mm:ss - hh:mm:ss], for example, a request that happened in the first minute of the video and lasted for 2 seconds was annotated as [00:01:00 - 00:01:02]
- request type: which could be verbal; gestural; grabbing (without request); clinician (no request); verbal + gesture; verbal + grabbing

This resulted in a small annotated video set, called the Construction Task Dataset. Each video ranges from one to three minutes long. On most occasions, the unprocessed dataset can not be used straightforwardly. That is why data pre-processing was introduced as the first step.

5.1.1 Data Pre-processing

As the execution of MiVOS is one of the integral parts, our case faces a significant challenge in that MiVOS requires a considerable amount of GPU memory, which cannot be accommodated. A solution is proposed to partition the whole video into

short segments and process these sequences respectively. Hence, the separation of the videos was taken as the initial step. After several attempts with status monitoring on a 24GB GPU, it was figured out that the 10 seconds video reached the limitation of the GPU memory. Therefore, all the videos were divided into small segments with a length of 10 seconds. Those 10 seconds segments were taken as the input of the Hand Detection and Tracking method.

All the other methods except MiVOS were compatible with our dataset, thus, the raw dataset was fed into other methods.

5.2 Hand Detection and Tracking

The hand detection and tracking method was operated with the segmented data. The process took a considerable time as expected. It was measured that a 10 seconds video needed approximately 7 minutes to complete with $2\times$ TITAN RTX GPUs.

The obtained result is consistent with the time cost. The method outputted fairly good results under most circumstances. It detected all the visible hands accurately with exact hand keypoints as Figure 5.1 shows.

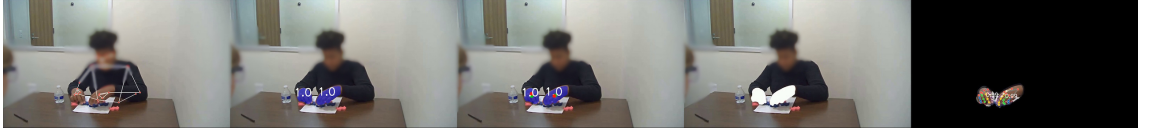


Figure 5.1: A perfect output example

However, the model did not always generate perfect results. In some cases, it fails to detect the hand, and the hand of the clinician might be detected instead of the patient.

As explained in the previous method, the distance between the hand mask center and body pose hand center point was taken into consideration as the hands' screen criterion. The threshold was chosen discreetly. After several different trials with thresholds of 20, 30 and 50, it was eventually set to 40, which worked out in the majority of cases and optimize the output.

The reason for detecting a hand other than the patient's is that the input dataset has a setup with at least two persons included (a clinician and a patient), it is doubtless that the MaskR-CNN model would detect more than two candidates of the hand when the clinician's hands are also visible. The following Figure 5.2 shows

the case when the hand from the clinician was detected at the beginning and it was eliminated because it was out of the threshold.



Figure 5.2: A proper example of removing the clinician's hand

The specific task given to the patient is to manipulate the puzzles which means that the patient's hands keep moving most of the time. This caused detection missing with a series of difficulties.

The most frequent one is that the variability in hand position and posture resulted in missing detection. As Figure 5.3 shows, both hands of the patient were detected correctly in the first row, but then the model lost detection in the next two frames (the second image in row 2 and row 3). The missing gaps of the hand were filled after hand tracking (the last column) eventually.



Figure 5.3: An example of filling a vacancy in missing hand detection

Secondly, continuous movements generated inevitability cumbrous blur without saying. To our great delight, the method has a satisfactory handle on the case of pixel blur. A typical example is presented in Figure 5.4. It demonstrates three frames with obvious hand moving, in the first and third frames, both hands were clear and detected directly. In the second frame, there was a blur around one of the hands, it was not able to be recognized by the model. With the help of tracking, the blurred hand was identified successfully, as seen in the last image in the second row.



Figure 5.4: An example of recognizing blurred hand

Thirdly, in some cases, the hands were tricky to detect validly because of partial covering. However, with the whole process of the method, the generated result is quite decent. For instance, there is a patient in the dataset who was wearing a pair of gloves when she was manipulating the puzzles. The example is shown in Figure 5.5, the first image is the body pose estimation, and the second image shows the hands proposals which are not perfect because one hand from the patient is missing and the clinician’s arm and hands are included. After combining with the body pose, the unwanted parts are eliminated, as shown in the third image. Furthermore, after executing the MiVOS tracking, the inclusion of both hands can be seen in the fourth image. Thus, both hands of the patient are detected in the final output which is presented in the last image.



Figure 5.5: A proper example of hand detection, where the hands are partially covered by gloves

5.3 Puzzle Pieces Segmentation

As described before, the puzzles provided in Construction Task are in a unique shape with two colors. The task contains 8 pieces of the puzzle, 4 in red and 4 in blue. The puzzle pieces have approximate dimensions of 5cm x 3cm x 0.5cm. Therefore, we retrained Mask2Former to segment puzzle pieces.

5.3.1 MiVOS semi-automated annotation

The most basic and essential step for puzzle pieces segmentation is to create the train, test and validation dataset for model retraining.

In order to make the process efficient, the MiVOS was included as the assistant. The interface of the software is shown in Figure 5.6. After choosing the specific video, the categories should be defined. In our case, they are named:

- red_puzzle#1
- red_puzzle#2
- red_puzzle#3
- red_puzzle#4
- blue_puzzle#1
- blue_puzzle#2
- blue_puzzle#3
- blue_puzzle#4

It is necessary to separate the same color puzzles with different category names as in most situations those pieces with the same color are connected. Otherwise, if they are not differentiated from each other, the model can not learn the exact shape of the puzzles.

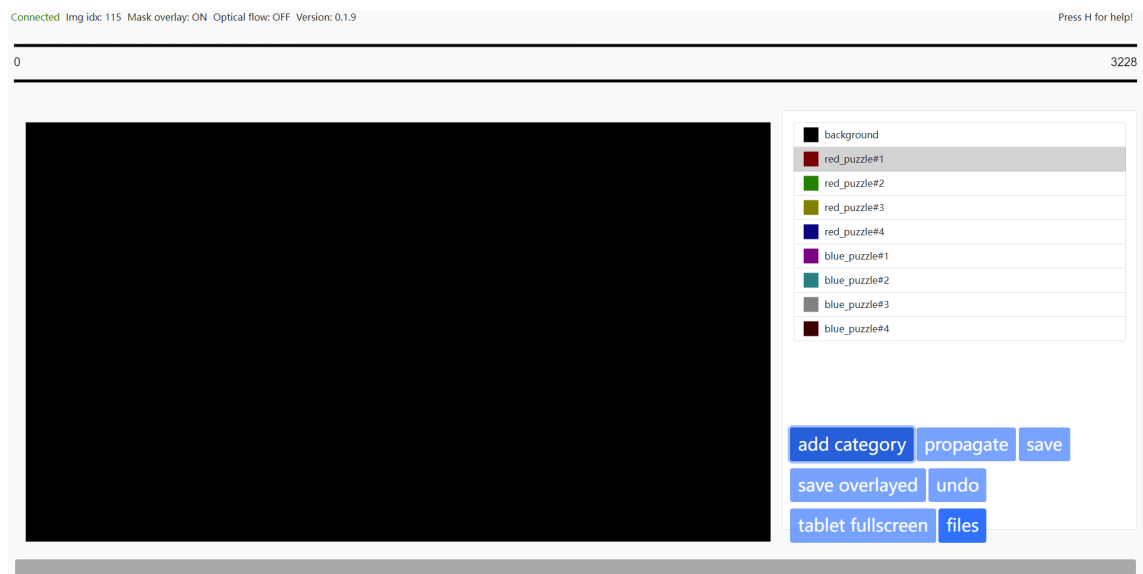


Figure 5.6: MiVOS main interface

The MiVOS has a propagation settings option (Figure 5.7) in which the max propagation size and segment size can be adjusted. The segment size should be feasible according to the GPU memory composition. Through multiple trials and comparative observations, it was determined that more satisfying precision could be achieved through the utilization of a reduced propagation size. In order to strike a balance between time consumption and result accuracy, the size was set to 10 as the optimal solution.

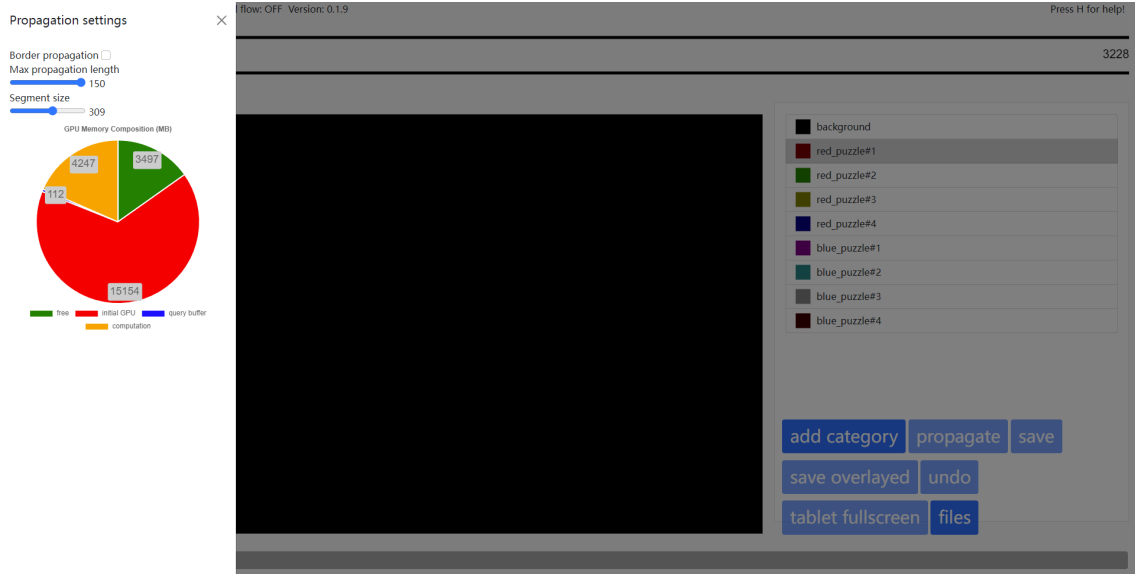


Figure 5.7: MiVOS propagation settings

After loading the video, adding categories and choosing the proper size, the manual segment can start.

The MiVOS can automatically recognize the entire puzzle with a scribble on it when the object color has a significant difference from the surroundings. Figure 5.8 shows the frame which is manually segmented. The adjoin puzzles were well separated and the one in the patient's hand was also fully recognized.



Figure 5.8: Manually segmented frame

The propagation process can be executed with one manually marked frame. All the frames within the propagation size can be segmented in a short time period even if the position of puzzles is changed frame by frame.

However, not all frames can be perfectly handled. It is normal that the model finds a similar object and generates a mask on it. The example is presented in Figure 5.9. In this frame, all the puzzles were properly segmented even though they were partially covered by the patient's hands, but it is non-negligible that the pen in the clinician's hand was also recognized as a puzzle.



Figure 5.9: Falsely segmented frame

The aforementioned case has an obviously incorrect mask. Nonetheless, in some frames, tiny mistaken objects exist, the example can be seen in Figure 5.10. After some frames, the small block of the patient's shirt was deemed as a puzzle that might be ignored easily.



Figure 5.10: Propagated frame with tiny mistake

Those defective frames can affect the accuracy of the model, so they can not be neglected. Thus, every frame needs to be double check to make sure there are no

mistakes. It increased the segmenting time without any doubt.

Eventually, 13 featured videos were segmented which are severed as the retrain dataset.

5.3.2 Mask2Former fine-tuning

13 videos with annotation JSON files are possessed. However, the JSON format is not compatible with the Mask2Former input file format. In our case, the Mask2former for video instance segmentation model was used. It requires an input with the youtube_vis format [27].

Therefore, the indispensable part is data format reorganizing. Within this step, the data was divided into the train, test, and validation sets with certain proportions.

After disposing of the data, the training process was conducted. There are several hyperparameters that can be adjusted according to the circumstances. In our case, only the batch size, learning rate, and maximum iteration have significance. Due to insufficient memory of the GPU, the batch size was set to 2. Following several experiments, the maximum iteration was determined to be 5999 and the learning rate was established at 0.000025. Several rounds of retraining were carried out with different settings of dataset distribution under those hyperparameters.

At first, the percentage was set to 50%, 25%, and 25% for train, test, and validation respectively. However, with this setting, the model was not properly trained due to insufficient data. The predictions were not satisfying, one example of a predicted frame can be seen in Figure 5.11. It is apparent that the result is unacceptable with only 2 predictions upon 8 puzzles.



Figure 5.11: First retrain model output with 50% training data

To improve accuracy, another model with 90% of training data, 5% of testing and validation data was trained. The outcome exceeded our expectations as shown in Figure 5.12 with the same frame.



Figure 5.12: Second retrain model output with 90% training data

The loss function is a combination of classification loss and mask loss as shown in Equation 5.1. The λ_{cls} is set to 2.0 for the predictions which are matched with

the ground truth. Otherwise, $\lambda_{cls} = 0.1$ is applied.

$$\mathcal{L}_{mask} + \lambda_{cls}\mathcal{L}_{cls} \tag{5.1}$$

The mask loss is calculated by adding binary cross-entropy loss and dice loss as Equation 5.2 with both λ_{ce} and λ_{dice} set to 5.0 [11].

$$\mathcal{L}_{mask} = \lambda_{ce}\mathcal{L}_{ce} + \lambda_{dice}\mathcal{L}_{dice} \tag{5.2}$$

Comparing the total loss function of the two above-mentioned models, the one with 50% data served as a train set exhibited a loss of 55.8424. However, the total loss of the other retrained model with a larger training dataset is 32.1705. This represents a reduction in loss of 42.39% and translates into a significant improvement in prediction accuracy.

Chapter 6

Results and Discussion

With the proposed Composite AI framework, we are able to detect two types of requests, which are verbal and gesture. The set dataset includes 29 videos. However, due to the individual diversity and differences in interaction skills of those patients, the type and the number of requests are different in the videos. Moreover, not all videos contain requests. In this section, we visualize the result with graphs and present the employed model evaluation metrics:

- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative
- Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$
- Precision: $PPV = \frac{TP}{TP+FP}$
- Recall: $TPR = \frac{TP}{TP+FN}$
- F-1 score: $F - 1 = \frac{2TP}{2TP+FP+FN}$

The following confusion matrix (Table 6.1) presents verbal detection compared with the manually annotated ground truth. We have zero false positives along with 33 true positives, 10 false negatives, and 2 true negatives.

		Detected	
		Request	No Request
Ground Truth	Request	TP 33	FN 10
	No Request	FP 0	TN 2

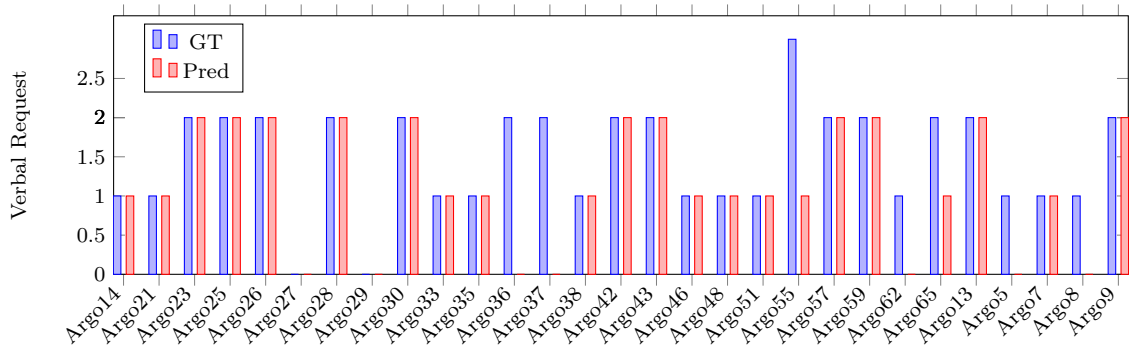
Table 6.1: Confusion Matrix of verbal request

The evaluation metrics for verbal request are as follows:

- TP: 33
- TN: 2
- FP: 0
- FN: 10
- Accuracy: 87%
- Precision: 100%
- Recall: 77%
- F-1 score: 87%

The histogram shows the comparison of the verbal request numbers between ground truth and detection of 29 videos separately (refer to Figure 6.1).

Figure 6.1: Verbal requests comparison between ground truth and detection



The following confusion matrix (Table 6.2) presents gesture detection compared with the manually annotated ground truth.

		Detected	
		Request	No Request
Ground Truth	Request	TP 4	FN 3
	No Request	FP 0	TN 23

Table 6.2: Confusion Matrix of gesture request

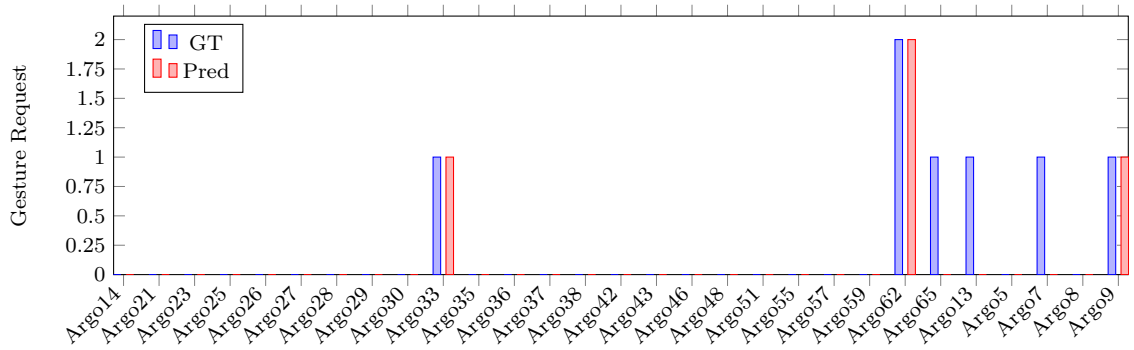
We have the evaluation metrics for gesture request detection as follows:

- TP: 4
- TN: 23
- FP: 0
- FN: 3
- Accuracy: 89%

- Precision: 100%
- Recall: 57%
- F-1 score: 73%

The histogram shows the comparison of the gesture numbers between ground truth and detection of 29 videos separately. As demonstrated in the Figure 6.2, only 6 videos contain gestures.

Figure 6.2: Gesture requests comparison between ground truth and detection



Due to the size of the project, my focus was mainly on hand pose estimation and tracking and puzzle pieces segmentation, the other methods were run by others with default values. There are several typical gestures being recognized within the request episode. For instance, the patient rises the hand (Figures 6.3) and uses fingers to represent the number of puzzle pieces he/she wants, usually with one (Figure 6.4) or two fingers (Figure 6.5) showing.



Figure 6.3: Patient makes a request with one hand up



Figure 6.4: Patient asks for one more piece by showing one finger

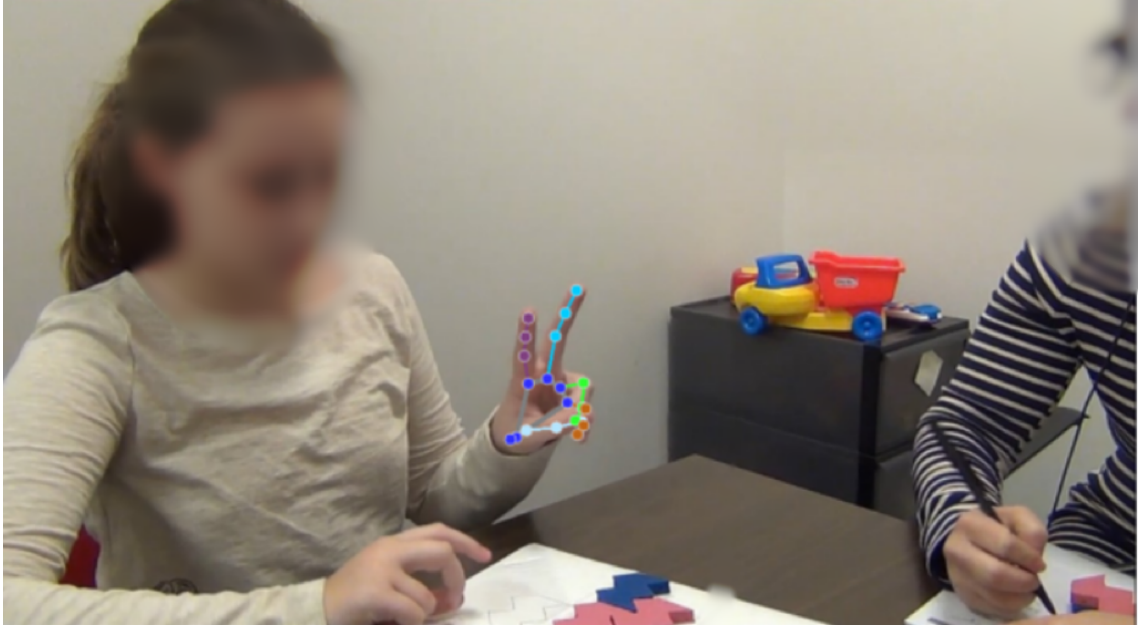


Figure 6.5: Patient asks for two more pieces by showing two fingers

According to the result, our method has the capability to detect the gesture and verbal requests respectively with quite good performance in accuracy, precision and F-1 score. Besides, our model can also detect grabbing motions, however, annotators' opinions were different when annotating grabbing motions, specifically, in cases where there are instances of verbal requests and grabbing actions that occur simultaneously or in close proximity to each other. Eventually, we decided not to include the full evaluation of grabbing motion due to its inherent ambiguity in children with ASD.

Chapter 7

Conclusion

Interaction with objects (which in our scenario are puzzle pieces) is considered one of the essential indicators of autism diagnosis. However, subtle and imperceptible movements are easily neglected by human eyes. The rise of Composite AI, a combination of several deep learning and traditional methods, provides the opportunity and convenience for AI assistant autism request detection including verbal and non-verbal communication cues. It not only enhances the clinicians' human power but also improves detection accuracy. Our proposed Composite AI framework is able to detect both verbal and non-verbal requests accurately and robustly with 100% precision. This is only possible due to the framework's ability to constrain the analysis to only key moments. The well-detected result serves as a potent clue for clinicians' diagnosis. Furthermore, the method can have more general application scenarios which include social interaction.

There are some limitations to our proposed framework. Currently, we only evaluate the patients and their requests, which means the information we get is confined. However, in the future, we plan to evaluate clinicians as well, in this way, we might add valuable information about the interaction and what caused or limited certain requests.

Acknowledgements

This project is part of the research conducted by Neural Information Processing Group (NIPG). My contributions are related to the hand pose estimation and puzzle pieces segmentation parts. Thanks to Szlatinicsán Adrián for the initial work on puzzle pieces segmentation and András Sárkány for the contribution to the speech recognition.

Bibliography

- [1] Maenner MJ, Warren Z, and et al. Williams AR. “Prevalence and Characteristics of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2020.” In: (2023). DOI: <http://dx.doi.org/10.15585/mmwr.ss7202>.
- [2] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. “Towards Accurate Alignment in Real-time 3D Hand-Mesh Reconstruction”. In: *arXiv e-prints*, arXiv:2109.01723 (Sept. 2021), arXiv:2109.01723. DOI: 10.48550/arXiv.2109.01723. arXiv: 2109.01723 [cs.CV].
- [3] Google. *Mediapipe Hands*. URL: <https://google.github.io/mediapipe/solutions/hands.html>.
- [4] Keyang Zhou et al. “TOCH: Spatio-Temporal Object-to-Hand Correspondence for Motion Refinement”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2022.
- [5] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. “Modular Interactive Video Object Segmentation: Interaction-to-Mask, Propagation and Difference-Aware Fusion”. In: *CVPR*. 2021.
- [6] Changlong Jiang et al. “A2J-Transformer: Anchor-to-Joint Transformer Network for 3D Interacting Hand Pose Estimation from a Single RGB Image”. In: *arXiv e-prints*, arXiv:2304.03635 (Apr. 2023), arXiv:2304.03635. DOI: 10.48550/arXiv.2304.03635. arXiv: 2304.03635 [cs.CV].
- [7] Mihael Presecan, Frano Petric, and Zdenko Kovacic. “Object Classification for Child Behavior Observation in the Context of Autism Diagnostics Using a Deep Learning-based Approach”. In: *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2018, pp. 1–6. DOI: 10.23919/SOFTCOM.2018.8555752.

- [8] András Sárkány, Máté Csákvári, and Mike Olasz. “Combining Common Sense Rules and Machine Learning to Understand Object Manipulation”. In: 24 (2019), pp. 157–172. DOI: 10.14232/actacyb.24.1.2019.11. URL: <https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/4013>.
- [9] Kaiming He et al. “Mask R-CNN”. In: *arXiv e-prints*, arXiv:1703.06870 (Mar. 2017), arXiv:1703.06870. DOI: 10.48550/arXiv.1703.06870. arXiv: 1703.06870 [cs.CV].
- [10] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *arXiv e-prints*, arXiv:2103.14030 (Mar. 2021), arXiv:2103.14030. DOI: 10.48550/arXiv.2103.14030. arXiv: 2103.14030 [cs.CV].
- [11] Bowen Cheng et al. “Masked-attention Mask Transformer for Universal Image Segmentation”. In: *arXiv e-prints*, arXiv:2112.01527 (Dec. 2021), arXiv:2112.01527. DOI: 10.48550/arXiv.2112.01527. arXiv: 2112.01527 [cs.CV].
- [12] Olesia Makhnytkina, Olga Frolova, and Elena Lyakso. “Morphological and Emotional Features of the Speech in Children with Typical Development, Autism Spectrum Disorders and Down Syndrome”. In: *Artificial Intelligence and Natural Language*. Ed. by Valentin Malykh and Andrey Filchenkov. Cham: Springer Nature Switzerland, 2022, pp. 49–59. ISBN: 978-3-031-23372-2.
- [13] Giuseppe Palestra, Berardina De Carolis, and Floriana Esposito. “Artificial Intelligence for Robot-Assisted Treatment of Autism”. In: *WAIAH@AI*IA*. 2017.
- [14] ali rashid and Shaimaa shaker. “Autism spectrum Disorder detection Using Face Features based on Deep Neural network”. In: *Wasit Journal of Computer and Mathematics Sciences* 2.1 (2023), 112–124. DOI: 10.31185/wjcm.100. URL: <https://wjcm.uowasit.edu.iq/index.php/wjcm/article/view/100>.
- [15] Diego Fabiano et al. “Gaze-based classification of autism spectrum disorder”. In: *Pattern Recognition Letters* 135 (2020), pp. 204–212. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2020.04.028>. URL: <https://www.sciencedirect.com/science/article/pii/S0167865520301501>.

- [16] Sid Ahmed Hadri and Abdelkrim Bouramoul. “Towards a Novel Machine Learning and Hybrid Questionnaire Based Approach for Early Autism Detection”. In: *Key Digital Trends in Artificial Intelligence and Robotics*. Ed. by Luigi Troiano et al. Cham: Springer International Publishing, 2023, pp. 53–61. ISBN: 978-3-031-30396-8.
- [17] Elisabeta Marinoiu et al. “3D Human Sensing, Action and Emotion Recognition in Robot Assisted Therapy of Children with Autism”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2158–2167. DOI: 10.1109/CVPR.2018.00230.
- [18] Miaomiao Cai et al. “An Advanced Deep Learning Framework for Video-Based Diagnosis of ASD”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*. Ed. by Linwei Wang et al. Cham: Springer Nature Switzerland, 2022, pp. 434–444. ISBN: 978-3-031-16440-8.
- [19] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: *arXiv e-prints*, arXiv:2207.02696 (July 2022), arXiv:2207.02696. DOI: 10.48550/arXiv.2207.02696. arXiv: 2207.02696 [cs.CV].
- [20] Yifu Zhang et al. “ByteTrack: Multi-Object Tracking by Associating Every Detection Box”. In: *arXiv e-prints*, arXiv:2110.06864 (Oct. 2021), arXiv:2110.06864. DOI: 10.48550/arXiv.2110.06864. arXiv: 2110.06864 [cs.CV].
- [21] István Sárádi et al. “MeTRAbs: Metric-Scale Truncation-Robust Heatmaps for Absolute 3D Human Pose Estimation”. In: *arXiv e-prints*, arXiv:2007.07227 (July 2020), arXiv:2007.07227. DOI: 10.48550/arXiv.2007.07227. arXiv: 2007.07227 [cs.CV].
- [22] Google. *Mediapipe*. URL: <https://google.github.io/mediapipe/>.
- [23] Hartyányi K Attila. “Detecting and Determining Hand Movements and Object Configurations in Videos”. Master’s thesis. 2022.
- [24] Milán Szász. “Optimizing interactive pixel-wise labeling of video databases with the tools of machine learning”. Bachelor’s thesis. 2022.
- [25] Alec Radford et al. “Robust Speech Recognition via Large-Scale Weak Supervision”. In: *ArXiv abs/2212.04356* (2022).

- [26] Nithin Rao Koluguri, Taejin Park, and Boris Ginsburg. “TitaNet: Neural Model for speaker representation with 1D Depth-wise separable convolutions and global context”. In: *arXiv e-prints*, arXiv:2110.04410 (Oct. 2021), arXiv:2110.04410. DOI: 10.48550/arXiv.2110.04410. arXiv: 2110.04410 [eess.AS].
- [27] Linjie Yang, Yuchen Fan, and Ning Xu. “Video Instance Segmentation”. In: *CoRR* abs/1905.04804 (2019). arXiv: 1905.04804. URL: <http://arxiv.org/abs/1905.04804>.

List of Figures

4.1	Overall Composite AI framework with three main modules: feature extraction, episode detection, and request detection.	12
4.2	Feature Extraction module comprises a vision and speech component which uses multiple Deep Learning methods to extract features representing the environment, human and context.	13
4.3	An example output of the MaskR-CNN model, with both hands detected	15
4.4	An body pose estimation example from Mediapipe, including 33 returned keypoints with names listed. The image is from the Mediapipe site [22].	15
4.5	An example of misdetection from MaskR-CNN (middle) discarded by calculating the L2 distance (right) with the assistant of hand keypoints (left).	16
4.6	A comparison of outputs from the MaskR-CNN and tracking. The middle row shows the mask missing frame, the output from MaskR-CNN only has one hand detected because of partially hands overlapping (left). The MiVOS tracking generates an output with both hands detected correctly (right). The first and third rows are the adjacent frames with both hand masks.	17
4.7	21 hand landmarks from Mediapipe Hands site [3].	17
4.8	An example of the hand keypoints output	18
4.9	The architecture of full hand detection pipeline	18
4.10	The final output of the full hand detection pipeline, which summarizes the different stages of the process.	19
4.11	The example process of annotation from Szász’s Bachelor thesis [24].	20
4.12	Mask2Former architecture [11]	20

4.13	The upper image is the original frame, and the lower one is the output of Mask2Former with puzzle pieces segmented.	21
4.14	Pipeline for Verbal Request Detection	25
5.1	A perfect output example	28
5.2	A proper example of removing the clinician's hand	29
5.3	An example of filling a vacancy in missing hand detection	29
5.4	An example of recognizing blurred hand	30
5.5	A proper example of hand detection, where the hands are partially covered by gloves	30
5.6	MiVOS main interface	31
5.7	MiVOS propagation settings	32
5.8	Manually segmented frame	33
5.9	Falsely segmented frame	34
5.10	Propagated frame with tiny mistake	34
5.11	First retrain model output with 50% training data	36
5.12	Second retrain model output with 90% training data	36
6.1	Verbal requests comparison between ground truth and detection . . .	40
6.2	Gesture requests comparison between ground truth and detection . .	41
6.3	Patient makes a request with one hand up	42
6.4	Patient asks for one more piece by showing one finger	42
6.5	Patient asks for two more pieces by showing two fingers	43

List of Tables

6.1	Confusion Matrix of verbal request	39
6.2	Confusion Matrix of gesture request	40