POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Automating Upper Limb Activity Labeling

Supervisors

Prof. Danilo DEMARCHI

Prof. Paolo BONATO

Dr. Giulia CORNIANI

Candidate

Marta DE IASI

JULY 2024

POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Automating Upper Limb Activity Labeling

Design and Assessment of a Machine Learning Pipeline for Video Annotation in Post-Stroke Patient Analysis

Supervisors

Prof. Danilo DEMARCHI

Prof. Paolo BONATO

Dr. Giulia CORNIANI

Candidate

Marta DE IASI

JULY 2024

'To my sister,
my greatest strength.'

Acknowledgements

I would like to extend my heartfelt gratitude to Prof. Demarchi and Prof. Bonato for giving me the incredible opportunity to carry out my Master's thesis at the Spaulding Rehabilitation Hospital in Boston. The experience was truly enriching, not only professionally but also personally, as I met friends and amazing colleagues.

I owe the greatest thanks to my parents for their steadfast support throughout my journey, enabling me to be here in Boston. I also want to express my gratitude to my sister Chiara for her continuous encouragement to always strive for excellence.

I am deeply thankful to Gianluca, my boyfriend, for his unwavering support throughout our nine months apart. His presence and encouragement have been a constant source of strength, ensuring that I never felt alone during this time.

Throughout my stay in Boston, I had the pleasure of meeting many wonderful people with whom I share amazing and unforgettable memories. All the laughter, travels, and shared experiences made this journey truly special.

A heartfelt thanks to my Italian friends who have always believed in me throughout my master's journey and inspired me to become who I am today.

Summary

The advent of cutting-edge medical technologies and telehealth services has resulted in an explosion of health-related data, highlighting the urgent need for efficient data annotation in healthcare research. Manually labeling video footage to identify specific actions or features in medical imaging is both time-consuming and requires specialized expertise, causing significant delays in research progress. This thesis addresses this challenge by focusing on the annotation of upper limb movements in egocentric video data. It introduces an innovative minimally-supervised deep learning system designed to streamline this process. The proposed framework analyzes video recordings from head-mounted cameras capturing individuals performing everyday tasks. Central to the system are two key components: the Hand Object Detector (HOD) and the Snorkel model. The HOD, based on Faster R-CNN and CNN architectures, excels in identifying hands and their interactions with objects. Complementarily, Snorkel generates probabilistic labels for unlabeled data by applying custom labeling functions tailored to the observed actions. The pipeline enhances these models with customized modules and crucially integrates a Large Language Model (LLM) to support the labeling functions in Snorkel, thereby improving the accuracy of the functions by refining the results based on the output of HOD. This combination significantly reduces the need for manual annotation, automating much of the video labeling process. To validate the approach, the framework was applied to a carefully curated dataset. The results demonstrate its capability to accurately detect hand-object interactions and classify various hand activities, proving particularly beneficial for monitoring upper limb function in stroke survivors. This advancement marks a significant breakthrough in medical data annotation. By automating the identification and categorization of hand movements, the method not only reduces the manual workload but also enhances the precision of healthcare-focused machine learning models. Moreover, it offers a scalable solution to manage the ever-increasing volume of medical data. This approach demonstrates the potential of minimally-supervised deep learning and LLM in medical video annotation, promising to advance medical technology development and improve patient care in the evolving healthcare landscape.

Table of Contents

List of Tables

List of Figures

Acronyms

AFO
Ankle Foot Orthoses

AI
Artificial Intelligence

AVG
Active Video Games

DL
Deep Learning

FMA
Fugl-Meyer Assessment

GIT
Generative Image-to-Text

HAR
Human Activity Recognition

HOD
Hand Object Detector

HMM
Hidden Markov Model

HSMM
Hidden Semi-Markov Model

LF

Labeling Function

LLM

Large Language Model

MA

Massachusetts

MAS

Modified Ashworth Scale

MIL

Multi-Instance Learning

ML

Machine Learning

MMSE

Mini-Mental State Examination

SAR

Socially Assistive Robot

SRH

Spaulding Rehabilitation Hospital

UCSD

Brief Assessment of Capacity for Consent

Chapter 1

The Motion Analysis Lab

During my research tenure, I had the distinguished opportunity to work at the Motion Analysis Lab at Harvard Medical School, located within Spaulding Rehabilitation Hospital in Boston, Massachusetts. This state-of-the-art facility focuses on the extensive study of human movement biomechanics, leveraging the latest innovations in robotics and wearable technology. The lab's primary mission is to enhance understanding and develop novel therapeutic approaches for conditions such as stroke, Parkinson's disease, and cerebral palsy,.

Over the course of my time at the motion analysis laboratory, I was fortunate to engage in numerous projects. This invaluable opportunity enabled me to delve into various aspects of research, enhancing my comprehension of how technology and biomechanics work together to address mobility issues. This period was crucial for my growth and learning, highlighting the significance of a multidisciplinary approach in research to improve the lives of individuals with movement disorders.

1.1 REHAB-PAL

The project aims to develop and evaluate a home-based rehabilitation system for children with cerebral palsy. Using a socially assistive robot (SAR) along with active video games (AVGs), the system will provide personalized exercises, detect compensatory movements, and provide interactive feedback during therapy sessions. The comparative clinical study will test the effectiveness of the SAR-based REHAB-PAL system compared to a traditional AVG system, with the goal of improving motor skills and quality of life for children with cerebral palsy.

1.2 DEPHY

The project aims to evaluate the effectiveness of the new ExoBoot device developed by Dephy Inc. in simulating the mechanical characteristics of different AFOs (Ankle-Foot Orthoses). The objectives include verifying the capability of the Dephy platform to simulate AFOs on a test bench, assessing its feasibility during ambulation in stroke survivors to simulate the characteristics and performance of their habitual AFOs, and exploring the feasibility of using the Dephy platform to simulate the characteristics and performance of different AFOs and select the most suitable AFO characteristics on a subject-by-subject basis.

1.3 SYNPHNE

The purpose of the project is to introduce the SynPhNe platform to the market, which includes a connected wearable solution designed to provide an innovative approach to neuro-motor rehabilitation. This platform simultaneously trains the brain and muscles using EEG and EMG signals during activities to create a self-correcting learning loop. The main goal is to assist stroke survivors and individuals with disabilities in their rehabilitation journey, enabling them to achieve greater independence and improved performance in daily activities. The device offers various care programs tailored to address issues related to the rehabilitation and recovery of patients with neurological conditions.

Chapter 2

Introduction

The swift advancement of machine learning techniques has brought about substantial progress in numerous fields. Machine learning is increasingly becoming a fundamental part of everyday life, improving the ease, efficiency, and customization of people's interactions and experiences.
In the domain of rehabilitative treatment, ML grants the capability to individualize therapies, improve the outcome of the clinical path, deliver more streamlined and impactful care [1].
Examples include:

- **Physical Therapy Support**[2]: Machine learning systems assess patients' movement patterns during physiotherapy sessions, delivering immediate feedback to both patients and therapists. This ensures exercises are performed accurately and tracks improvement.

- **Gait Evaluation**[3]: Machine learning techniques are employed to examine the walking patterns of individuals convalescing from harm or surgical interventions. Sensors and cameras gather data from movement, which is then examined to detect walking pattern alterations and provide guidance for rehabilitation approaches.

- **Fall Prevention**[4]: AI models assess mobility data to forecast the risk of falling in senior individuals, enabling healthcare professionals to adopt preventive strategies.

- **Recovery Monitoring** [5]: Body-worn sensors and devices utilizing intelligent systems consistently supervise patients' motion and essential health indicators, aiding medical providers in following improvement and fine-tuning rehabilitative protocols as required.

The success of such techniques largely depends on having extensive and well-annotated datasets, which form the basis for AI applications for recovery programs. These datasets enable frameworks to train, forecast, and direct rehabilitation procedures with a high degree of accuracy and customization. This type of accuracy and customization can significantly enhance treatment results and the general standard of healthcare [6]. The proposed customized method allows rehabilitation programs to be specifically designed for each individual's needs, significantly increasing the probability of successful results.

However, it's important to acknowledge that the labeling process greatly influences the deployment time for these models. The labor-intensive nature of data annotation impacts the overall efficiency of the process. Enhancements made to speed up the labeling process can lead to a ripple effect, hastening the implementation of machine learning models.

This script introduces a detailed method aimed at significantly speeding up the The video annotation procedure. By integrating DL techniques with weakly supervised machine learning, the system is capable of accurately identify hand movements in videos and autonomously tag individual frames. These developed annotations will then be utilized to annotate data collected from worn sensors, particularly those on the wrist and fingers [7], thus expediting the implementation of algorithmic systems for processing data coming from the sensors.

In the ensuing portion of this chapter, related studies will be examined to provide a unabridged overview of current approaches in the field of automatic data labeling from sensor elements. Next, the *RingSensor* study, which supplies the data to develop and test the framework proposed in this thesis, will be described.

Chapter three will provide a detailed account of the procedures and resources utilized, encompassing Deep neural networks and the weakly guided architecture, as well as the criteria applied to evaluate the findings.

Subsequently, the outcomes will be reviewed in chapters four and addressed in chapters five, ending with an overview of the procedure, an analysis of constraints, and suggestions for future research.

2.1 Previous studies

In prior investigations into human actions and motions identification via body-mounted devices, hand-operated approaches such as film recordings and direct monitoring broadly applied to collect captions.

Plotnik et al.[8] developed a body-worn support for Parkinson's disease individuals exhibiting freezing of gait. During the study, the patient's gait is recorded using cameras and wearable sensors. One annotator labels the videos, while a second

annotator labels the acceleration transmitted from the wearable device to a computer. The team also includes a physiotherapist who identifies the endpoints of freezing events in the gait analysis video.

Following a similar approach, *Anguita et al.* [9] incorporated a Samsung Galaxy S II smartphone into their data collection. The aim was to segment the various activities performed by patients, based on their movement, integrating environmental information into the recordings. Simple activities such as standing still, seated, reclining, strolling, going down stairs, and climbing steps were repeated twice by each patient, with a 5-second break between repetitions. Afterwards the labeling process of the data was performed by manual means.

In the study by *Banos et al.* [10], two IMUs are placed on 10 volunteers, specifically on the right wrist and left ankle. ECG data from two leads are integrated using a sensor placed on the chest. The collected data, pertaining to approximately 15 outdoor movements, include acceleration, geomagnetic data, angular velocity, ECG signals, and video recordings. Manual labeling of the data is also conducted in this study.

On the internet, there are numerous other publicly available datasets accessible to everyone for human activity recognition (HAR) [11] based on wearable sensors and portable devices. These datasets provide data on acceleration, angular velocity, and geomagnetic field. However, achieving high-precision labeling still relies primarily on manual annotation, which demands significant effort and lengthy timelines.

The greatest hurdle today is obtaining a fully labeled dataset for extended monitoring of activities of daily living. These datasets are crucial for training algorithms capable of automating the annotation process, thereby reducing the workload of human labeling.

One of the new proposals involves the application of weakly supervised methodologies [12], specifically MIL with experiential sampling. Instead of labeling individual occurrences, sets of instances known as *bags* are annotated. This approach increases the generalization of labeling, resulting in a significant reduction in annotation burden. A positive *bag* is classified as so if it contains at the minimum one positive occurrence, and negative if the instances in one group are non-positive.

The study presented in [13] represents the first application of MIL to time series of activities performed by subjects. The model effectively segments daily activities, thereby reducing the workload for annotators. Building upon this work, Guan et al. [14] develop an integrated MIL approach with an auto-regressive Hidden Markov Model, operating offline on multivariate time series data, annotating individual instances as well as *bags*.

Unsupervised learning methods are used to analyze human activity data to identify and extract covert patterns within these data, without requiring predetermined

labels or categories. Mentioning, Wyatt et al. [15] handled activities as sequences of actions or behaviors described in natural language, where object usage is considered integral to these linguistic sequences. They applied generic models derived from everyday activities found on the internet, which represent a form of common knowledge universally recognized in human behavior. This facilitates the segmentation of activities of daily living through a context or reference base on how people typically interact with objects during their daily activities.

A further instance is the unsupervised pipeline proposed by *Bottcher et al.* [16], which uses clustering methods to identify transitions between different activities. This approach completely eliminates the need for predefined labels, provided that the order and number of steps are known in advance.

Comparably, *Van Kuppevelt et al.*[17] utilize non-supervised machine learning methods to examine accelerometer readings from everyday activities. They implement a Hidden Semi-Markov Model (HSMM) to divide the data into segments of five seconds each, identifying up to ten different activity states based on average acceleration data. This strategy uncovers patterns of movement and inactivity without the need for manual data labeling.

The aforementioned methods address the issue of labeling mainly through human annotation and machine learning models. However, when the context shifts from a regulated environment, such as a laboratory, to actual settings, the precision and accuracy of the data and annotations decrease drastically.

It is evident that the core concept of MIL and unsupervised methods is the integration of pre-existing knowledge about performed activities with a limited dataset of labeled data. However, there remains a need to manually label some initial instances. This initial labeling effort is essential to create a solid foundation upon which these advanced techniques can develop, thereby improving the effectiveness and precision in activity detection.

Consequently, this thesis concentrates on an innovative method for labeling manual tasks using video footage gathered in a controlled laboratory environment. The videos originate from the *RingSensor study*, which aims to track upper limb movements in stroke survivors using sensors placed on the finger and wrist. To expedite the annotation pipeline, the suggested networks employs DL to detect hands and objects within frames and weakly supervised learning and LLM to produce annotations.

2.2 The RingSensors Project

2.2.1 Objectives

The clear targets of this research are:

- **Objective 1**: A set of data is collected through the use of sensors applied to the fingers of a group of participants - 20 people - who have suffered a stroke. The participants are monitored while performing various activities of daily living, in order to collect data, which will form the foundation of an automatic machine learning algorithm (Objective 2).
- **Objective 2**: Validate the adequacy of sensors in accurately measuring upper limb performance in daily living activities among participants - up to 60 subjects - through the implementation of machine learning-based algorithms. It is presumed that the wearable sensors will be able to accurately simulate measures of upper limb performance in habitual activities.
- **Objective 3**: Analyze the feedback obtained from the study patients (users) and the healthcare providers (prescribers) involved, regarding the Performance and user-friendliness of the planned solution.

Currently, Objective 1 has been effectively accomplished, allowing data and camera recordings integration into the proposed process; Objective 2 is currently ongoing.

2.2.2 Background and significance

Upper limb paralysis represents the main consequence after a stroke, affecting up to 75% of subjects who have suffered such an event [18]. This condition severely compromises the individual's ability to perform a wide range of essential daily activities. Despite rehabilitation, almost half (49%) of stroke survivors continue to experience difficulties in using the impaired arm, even five years post-event. Consequently, ensuring the best clinical results requires a more individualized and organized approach to planning rehabilitation pathways. Substantial research findings demonstrates the effectiveness of rehabilitative therapies in improving movement abilities [19], mainly derived from motor skill development.

The increasing use of body-worn sensors presents a hopeful method to impartially monitor mobility ability in everyday contexts. At present, sensors worn on the wrist hold a prominent position in wearable technology, primarily focusing on measuring arm use, such as the span and strength of daily arms motions [20]. However, this presents a weakness: these wrist-worn sensors primarily record large-scale limb motion, such as natural arm swing while walking. This frequently results in an overly positive evaluation of motor abilities.

On the other hand, emerging research sheds a promising light on finger-worn sensors, highlighting their potential to more accurately monitor arm movements [21]. Initial findings obtained from control subjects, as illustrated in Section 2.1, show a robust correlation between the acceleration data captured by sensors worn on the fingers and the actual upper limb activities in real-world settings, both in

laboratory and outside settings. Based on the literature, RingSensor's research seeks to investigate a new method to improve the resolution of upper limb data in chronic post-stroke individuals during typical daily activities. This approach involves sensors positioned on both the subjects' fingers and wrists. Furthermore, the study intends to investigate sensor data can be used to provide valuable feedback to both wearers and healthcare providers.

Figure 2.1: Suggested ring-shaped sensors.

2.2.3 Study design and methodologies

In the intended research, 60 subjects were enrolled for Aim 2, out of which 20 were invited to be involved in Phase 1 and 3. 10 clinicians are required to participate in Phase 3 of the study.

It's important to note that this study does not involve any interventions. Preliminary screening is carried out over the telephone by the investigators, followed by a conclusive evaluation at the Motion Analysis Laboratory (MAL) of Spaulding Rehabilitation Hospital. Face-to-face preliminary evaluation is available on demand.

Eligibility requirements for stroke survivors

Inclusion Criteria

- Individuals who have experienced a stroke (ischemic or hemorrhagic), more than 6 months post-stroke at the time of consent.
- Mild to moderate remaining impairments in upper limb function, scoring > 35 on the Fugl-Meyer Assessment (FMA) without severe limitations in range of motion.
- Age between 18 and 80 years.

Exclusion Criteria

- Incapable of raising upper limb against gravity (more than 30 degrees of flexion and abduction).
- Intense upper-limb spasticity hindering passive finger movement (MAS > 3).
- Incapable to independently wear/remove sensors or needing assistance from a caregiver.
- Cognitive deficits impacting understanding and following instructions (score < 23 in the MMSE).
- Possession of implantable medical devices that do not conform to ISO 14117:2012 or ANSI/AAMI PC69 Bluetooth compatibility standards. Subjects will submit their medical device record card for confirmation.

Criteria for clinician eligibility for Phase 3

Inclusion Criteria

- Practitioners with at least one year of experience in stroke recovery.
- Minimum age of 21.

Research protocol for Phase 1

Out of the designated cohort of 60 subjects, around 20 individuals are chosen to engage in an initial in-person session before the procedures specified in Phase 2. The first investigative meeting takes place at the Spaulding Rehabilitation Hospital (SRH) in Charlestown, MA, and has an estimated maximum duration of three hours, during which participants sign the consent form and undergo an initial screening process. Subsequently, they wear sensors on their hands, upper limbs, and torso; thus equipped, the patients perform the activities listed in Table 1, under the supervision of the research team. The entire visit is recorded using GoPro cameras, which allow for subsequent synchronization with the acceleration data from the sensors. After the session, the research team annotates the recorded video data, contributing to the completion of Aim 1.

Research protocol for Phase 2

The investigation carries on at the participant's domicile, as agreed upon during the consultation with a clinician, providing sensors and an encrypted device for the entire duration of the study. The entire research is structured with a duration limit of one week, during which the patient regularly wears the sensors for eight hours a day and records the activities performed every one and a half hours. The annotations provide a crucial tool for associating the collected acceleration data with the type of movement performed by the participant.

Figure 2.2: Study protocol pathway

First Visit

Subjects who, following their involvement in Aim 1, are willing to proceed, must consent to the subsequent aims of the study. After passing a second initial screening, the subjects consult with a researcher to comprehend and conclude the consent procedure. Their cognitive function and comprehension of instructions are tested using the Mini-Mental State Examination (MMSE). Consequently, potential participants who do not meet this criterion are excluded.

The potential patients' judgment capacity is assessed through the UCSD questionnaire. It is crucial that the subjects fully understand the research nature of the study, which does not provide any medical therapy, and are conscious of the possible risks and benefits. Not understanding these aspects leads to the exclusion. Upon successfully completing these assessments, participants either endorse the consent form or give verbal consent if participating remotely.

To assess upper limb functionality, a clinician performs a FMA [22] assessment, providing a rating from 0 to 66 to evaluate upper limb motor dysfunction; a score under 35 results in exclusion from the study. Subsequently, the participants' muscle stiffness is examined using the MAS [23] and by reviewing patient records. Arm function is tested by completing tasks aligned with the Wolf Motor Function Test [24]. The patient is required to self-assess using the Motor Activity Log [25], a clinical approach to evaluate the use and effectiveness of movement during simple activities. The assessment period varies between one and two hours, depending on the subject's motor abilities.

Participants have the option to consolidate Visit 1 and Visit 2 into a single day.

Second Visit

After completing the clinical assessment, the research team places a sensor on the chest of the patient, and on the hands, fingers, and wrists of both limbs. The sensors are housed in silicone cases for the rings, while wrist and chest sensors are mounted on Velcro straps, both off-the-shelf. The predicted setup time is roughly 10 minutes 10 minutes.

Once the sensors are set up, the subjects simulate activities of daily living in a simulated home environment located at the Spaulding Rehabilitation Hospital or in their own home, in case of virtual sessions. All tasks are recorded through GoPro cameras or via a webcam in virtual sessions, allowing for subsequent offline analysis. The cameras, affixed to the participant's torso and head, grant researchers a participant's perspective, which aids in aligning movement with measurements from body-worn sensors. Likewise, the portable recording devices supply supplementary perspectives that may fall outside the GoPro's point of view (Figure 2.3).

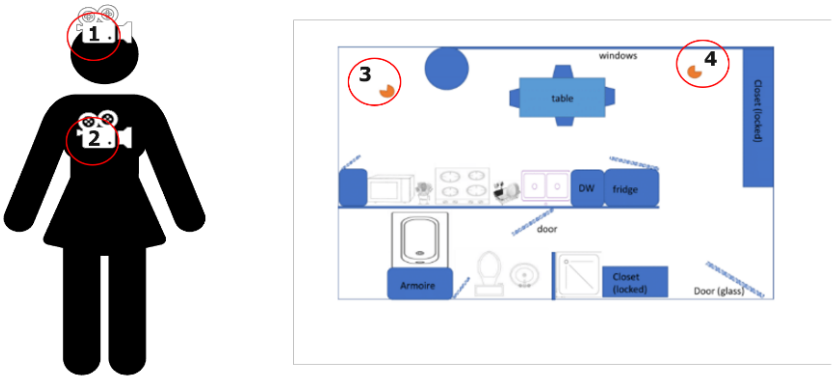

Figure 2.3: GoPros set up

Only IRB-authorized personnel are permitted to capture and view video footage, and participants' permission for recording is obtained beforehand. The patient has to perform 14 tasks, each repeated 3 times, allowing for the observation of individual variability in movement patterns. The execution of tasks is divided into two sessions to avoid fatigue for the patiens. Detailed descriptions of the activities are provided in Table 2.1.

The approximate duration of the first laboratory session is 1.5 hours. To complete the second session, the patient will return after 7 days.

Home-based surveillance

The provided upper limb sensors must be worn by the patient for 7 days, 8 hours each day. They are instructed to remove the sensors each night to charge them using the provided charger. Since the sensors are not waterproof, participants need to be instructed to remove them before engaging in activities such as showering or swimming.

Activity Tasks	
Sit to Stand transitions	Using the phone
Pour and drink a glass of water	Brush teeth
Find a recipe	Prepare a sandwich
Make and fold the laundry	Take the coat off
Mop the floor	wipe the countertop
Set the table for eating	Eat the meal
Unlock and lock the doorknob	Open and close the door

Table 2.1: Experimental tasks.

Participants are provided with a prepaid smartphone, which comes preinstalled with:

- **Google Timeline**, for recording movement patterns and types. It is crucial to obtain this information as it allows for the identification of passive actions detected by sensors and their respective filtering. Although it is recommended for patients to have their phone with them to facilitate the annotation process, since the application tracks their location everywhere, patients are not obligated to carry it with them.

- **Application for monitoring sensor conditions** manages communication between sensors and the phone, and consequently between researchers and participants. It allows patients to monitor the charge status of sensors and alerts any issues in data collection, ensuring the research team stays updated on patient behavior during the at-home period.

- **Application for annotating daily activities**. The patient will receive a notification every 90 minutes within the eight-hour window established during the first session, reminding them to enter a brief description of the upper limb activities performed during that interval.

The MGB IT department has conducted a security assessment on these specially tailored apps.

Third Visit

At the end of the seven-day monitoring period, participants return to the MAL to return the equipment and undergo a second laboratory assessment, similar to the previous visit.

Research protocol for Phase 3

Interview - Patients)

The 20 participants of Phase 2 undergo an interview, either in person or remotely (via platforms such as UMass, UMD, or MGB Enterprise Zoom), where they are shown the collected data, aligning sensor accelerations with patient annotations. This allows researchers to understand any issues or challenges participants may encounter.

The interviews are recorded for both participation modalities.

Interview - Healthcare providers

Through interviews conducted either in person or via the MGB Enterprise Zoom platform, the opinions of 10 clinicians are sought to assess the acceptance of using ring sensors for monitoring patients' activities of daily living at home. Prior to starting the recording, the clinician participants are briefed on the protocol and asked to provide verbal consent.

2.2.4 Recording markup

To label the lab-based assessment videos with clinical relevance, healthcare providers have defined a grasping ontology, to better understand what types of grasps were more frequently used by post-stroke subjects. This way, it is possible to provide information to guide interventions, encouraging subjects to use their fingers more during grasping activities. The proposed labels are shown below.

Movement

- Arm
- Hands

No Movement

- *Ambulatory* - arm swing motion.
- *Non-Ambulatory* - duty-focused.

- *Bilateral*
  - Wide arm
  - Fine Hand grasping
    - Full hand
    - Finger
    - Lateral pinch
    - Uncertainty
- *Bilateral* - dual control & individual item
  - Wide arm
  - Fine hand grasping
    - Full hand
    - Finger
    - Lateral pinch
    - Uncertainty
- *Bilateral* - separate handling & isolated item
  - Gross Arm
  - Fine Hand
    - Full hand grasp
    - Finger grasp
    - Lateral pinch
    - Flag for uncertainty

This thesis aims to develop a method to quicken the process of annotating grasping activities by focusing only on a subset of frames, specifically those where hand-object contact is present, rather than the entire video dataset. This approach leverages the ELAN [26] annotation platform, which provides predefined labels for hand-object contact. By using this method, the work of the clinical team is greatly simplified, allowing them to concentrate solely on the intervals where contact occurs, thereby improving the efficiency and accuracy of the annotation process.

Chapter 3

Materials and Methods

The proposed pipeline is composed of main blocks, as depicted in Figure 3.1, that enable automatic annotation of first-person videos, with hands as the primary subject.

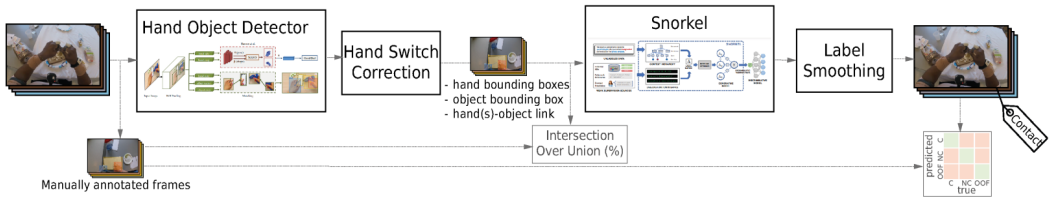

Figure 3.1: Two-stage pipeline.

The videos, which have been cropped and blurred to ensure the privacy of the subject, are processed by a deep learning model - first stage. This model is capable of detecting interactions between hands and objects, generating rectangular bounding boxes around the manipulating hands and the manipulated object. A significant advantage of this approach is that it does not require ground truth labels for training. However, to evaluate its performance, a subset of the data has been manually annotated. The results from this initial hand object detection (hod) are then passed to an improvement block that enhances the hod results before moving to the second stage.

Subsequently, all the data was processed in Snorkel - second stage - using labeling functions to assign the subsequent annotations:

- **Movement label**: occurrences where the hand is grasping an object.
- **No Movement label**: occurrences where the hand is not grasping of the object.

- **Uncertain label**: occurrences where hands are out of the GoPros field of view.

In the end, the labeled dataset is displayed on ELAN to observe the produced annotations.

3.1 RingSensor Study - Video Data

Understanding the dataset is crucial, before delving into the specifics of each phase. As defined earlier in section 2.2.3, during the activities planned for aim 1, conducted in a simulated kitchen, the actions of the subjects are recorded from 4 different perspectives:

- **Head point of view**: The subjects wear a GoPro mounted on their heads with a headband.
- **Chest point of view**: The subjects wear a GoPro mounted on their chest with a chestband.
- **Room point of view**: GoPro camera placed on a tripod to capture a panoramic view of the simulated kitchen.
- **Room Door point of view**: GoPro camera placed on a tripod to capture a panoramic door-view of the simulated kitchen.

A remote device is employed to synchronize the start and stop of recordings across the 4 GoPro cameras.

Initially, it was chosen to use the videos from the GoPros positioned on the subjects' heads. As mentioned earlier, these videos were pre-processed before being fed into the pipeline to remove frames related to the sensor placement.

3.2 First pipeline

3.2.1 Egocentric video

The decision to use egocentric videos in our study stems from their ability to provide highly detailed and precise information about manual activities, which is essential for clinical and therapeutic applications. However, this choice comes with its own set of advantages and challenges.

Pros

1. **Context-independent information**

   - ***Detail and Precision***: The egocentric perspective captures manual activities in great detail, reducing distractions from the surrounding environment. This is particularly useful in contexts like post-stroke rehabilitation, where it is crucial to observe hand movements accurately.
   - ***Uniformity of Data***: Since the focus is on manual movements and not the environment, the collected data tends to be more consistent, facilitating analysis and comparison between different sessions or subjects.

2. **Comprehensive understanding of manipulation**

   - ***Hand-Object Interaction***: Egocentric videos provide a clear view of how hands interact with objects. This is essential for studying and improving manual dexterity and fine motor skills.
   - ***Clinical and Therapeutic Applications***: The detailed view of movements can help therapists better understand patient progress and personalize rehabilitation programs.

3. **Large availability of datasets**

   - ***Accessibility***: Numerous public datasets of egocentric videos exist, reducing the costs and time required for data collection.
   - ***Training Models***: The availability of large amounts of data facilitates the training of deep learning models, improving their performance and accuracy.

Cons

1. **Sensitivity**

   - ***Changes in Light and Camera Movements***: Egocentric videos are sensitive to changes in lighting and sudden head or body movements, which can introduce noise into the data and complicate analysis.
   - ***Obstructions***: Objects or body parts blocking the view can interfere with the quality of the recordings, making automatic data interpretation difficult.

2. **Limited environmental context**

- ***Restricted Perspective***: The egocentric view offers a limited vision of the surrounding environment, which can be problematic when a broader context is needed to correctly interpret activities, like in training models.
- ***Model Generalization***: The lack of environmental context can make it harder to apply models to scenarios different from those they were specifically trained for.

3. **Requirement for adequately labeled datasets**

   - ***Manual Effort***: Ensuring that the labeled dataset meets the specific needs of the study is critical. Inadequate or poorly labeled data can lead to inaccurate model training and less reliable outcomes. Collecting and labeling large amounts of egocentric video data requires significant manual effort, which can be time-consuming and resource-intensive.
   - ***Data Management***: Managing and processing large datasets necessitates robust infrastructure and advanced data management techniques, posing additional logistical challenges.

Despite the challenges, the decision to utilize egocentric videos is justified by the need to obtain high-quality, detailed data on manual movements, which are crucial for clinical and therapeutic applications. The egocentric perspective provides a unique and detailed view of hand-object interactions, which is difficult to achieve with other angles.

Additionally, the accessibility of existing datasets and the potential to develop robust deep learning models represent significant advantages that can offset the initial costs of labeling and technical difficulties. The key is to balance these factors with innovative solutions to address the limitations, such as the use of weak labeling techniques and automated supervision.

The choice to use egocentric videos is driven by the necessity for precise and detailed data that can significantly enhance the understanding and treatment of clinical conditions, while also recognizing and addressing the associated technical and logistical challenges.

3.2.2 First stage

The videos of six subjects are preprocessed and then passed to the first block of the pipeline.

Stage one comprises two main steps. Firstly, to detect hand activity and assess the most effective model, the videos were analyzed using the Hand Object Detector [27], which employs a FasterRCNN. This model was trained with the dataset

described in [27] and is detailed in the subsequent sections.Dictionary structures are employed to manage the model outputs. At the same time, a subset of video frames underwent manual annotation.

The second phase involves employing two Python scripts to handle the outputs from the initial phase in readiness for the subsequent stage:

1. Conversion from dictionary to pandas structure.
2. The pandas dataframe is synchronized with the manually annotated labels.

Further specifics about these scripts are provided in section 3.2.4.

3.2.3 Hand Object Detector

The first component of the pipeline is the Hand Object Detector [27], a deep learning model capable of providing information about the contact status of hands with objects. It is trained on online videos featuring subjects involved in manual tasks.

Dataset

You can get a sense of the dataset by observing Figure 3.2, noting the interactions of daily living activities sourced from YouTube. It includes 100,000 annotated frames and a significant compilation of unlabeled videos for unsupervised learning.

- **Video Collection**: Approximately 10 million videos were sourced from YouTube, which were then filtered using video thumbnails and 15,000 queries to identify scenes of interaction. The starting point included 11 categories similar to DIY, from which animated content was excluded.
- **Image Collection**: Based on the same categories, the data collected in "100 Days of Hands" (100DOH) consists of approximately 30,000 videos, providing 131 days of footage. Frames without hands were excluded, and 100,000 random frames were selected and annotated. The dataset was divided into training (80%), validation (10%), and test (10%) sets based on the YouTube uploader's ID in order to avoid duplicates and ensure consistency with pre-existing datasets such as VLOG.

Pipeline

You can observe the model development workflow in Figure 3.3: regardless of the size of the RGB image, the framework is capable of performing detection seamlessly.

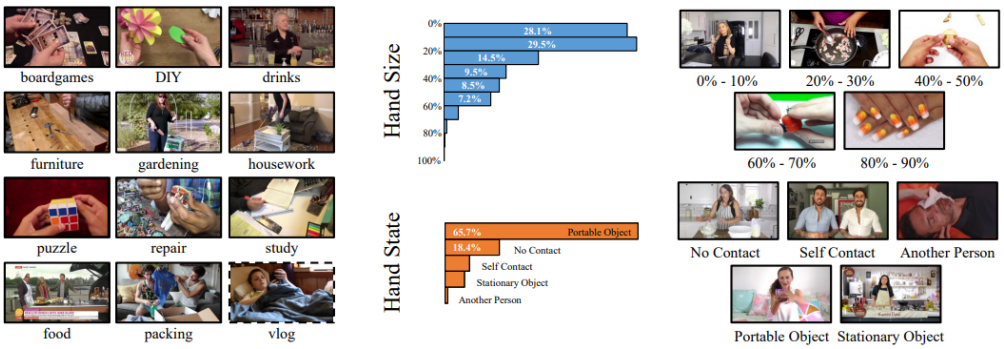

Figure 3.2: Snapshot of the "100DOH" dataset

The model identifies the bounding box that outlines the hand's space and can recognize the hand's side (right or left) and its *state of interaction*:

- No contact,
- Self contact,
- Contact with a person,
- Contact with a portable object,
- Contact with a non-portable object.

Thanks to the integration of hand reconstruction models like [28], it delineates the bounding box solely of the object manipulated by the hand, linking its center with the center of the bounding box of the interacting hand.

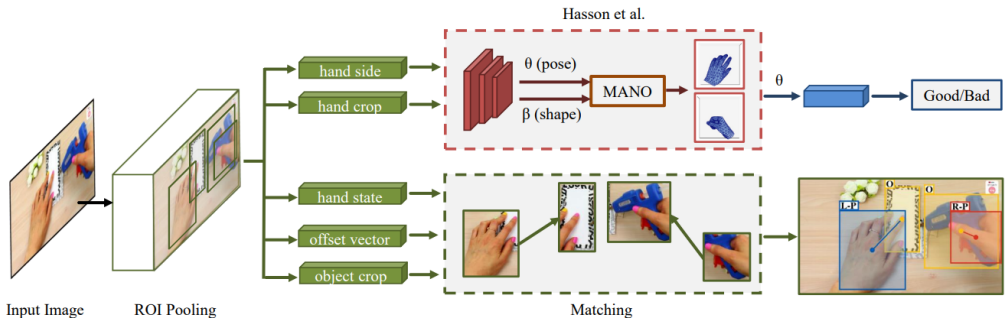

Figure 3.3: HOD workflow.

The framework is based on a Faster-RCNN model [29], a two-module system for object detection (Figure 3.3):

- **RPN** (Region Proposal Network): A deep fully convolutional network that proposes regions of interest in the image, each with a confidence score.
- **Fast R-CNN**: An object detection network that processes the regions proposed by the RPN to identify hands and manipulated objects. Similar to a traditional Fast R-CNN detector [30], it predicts the bounding boxes of objects, their category, and adjusts the bounding box dimensions as necessary. Using ROI-pooling, the system also provides additional outputs such as hand side and interaction state.

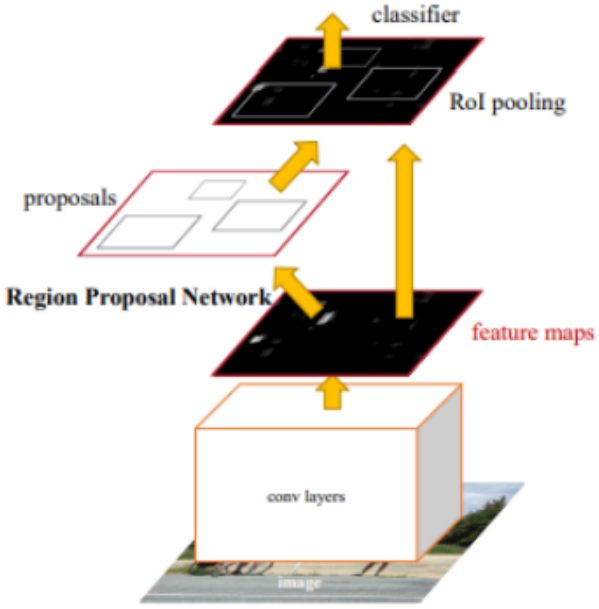

Figure 3.4: Workflow of a Faster-RCNN network

3.2.4 Python scripts

Output correction module

The HOD provides output as dictionaries, so the initial script handles the conversion into a Pandas DataFrame. For each frame, the model provides information on which hand (right or left) is present, whether there is an object being interacted with, the coordinates of the respective bounding boxes, the timestamp in milliseconds of the corresponding frame, and the confidence score. The frame timestamps in milliseconds were used to perform all the following synchronization steps. The object identified as being manipulated by the model is determined by calculating

the distance between the center of the hands and the center of the detected objects, then choosing the one with the minimum distance. *None* elements are retained in the DataFrame to prevent errors in frame counting during the various stages of the pipeline. To the created structure, the script applies the functions described below:

- **correct switch**: To address the persistent issue of models confusing left and right hands across frames, this function divides each frame into a left zone, a right zone, and a neutral zone, which represents the transition area between the other two. The hand's orientation is determined based on the position of the bounding boxes.
- **check side**: Once the hand orientation is defined, this function evaluates the overlap of bounding boxes in successive frames. If the overlap exceeds a threshold and the assigned zone changes, the hand's side is adjusted accordingly. This corrects initial hand side misidentifications.
- **extract hand side**: This function separates the data into right and left hand after ensuring accurate organization. Only right-hand data is used for symmetry and pipeline efficacy evaluation.
- **correct duplicate**: Based on the confidence scores returned by the HOD, this function handles multiple detections of the same hand by identifying and discarding duplicates.
- **check missing**: The function ensures that the total number of frames matches the actual number of frames in the video.

An example of the final Pandas structure for the Hand Object Detector is shown in Table 3.1.

subj	object	subj_bbox	object_bbox	label	score	frame_ms	labels
Right_hand	None	[999, 411, 1249, 662]	None	3	0.997141242	1131831	-1
Right_hand	Object	[1049, 844, 1223, 984]	[1061, 470, 1239, 945]	3	0.995938746	1178278	-1
Right_hand	None	[1019, 788, 1232, 924]	None	3	0.588171126	1246011	-1
None	None	None	None	-1	-1	1325958	-1
Right_hand	Object	[1699, 614, 1850, 787]	[947, 769, 1063, 928]	3	0.999849458	1433999	-1
Right_hand	Object	[1120, 955, 1402, 1060]	None	3	0.997748196	1603569	-1
Right_hand	Object	[1056, 862, 1359, 1071]	[957, 725, 1395, 1041]	3	0.997645086	1630929	-1

Table 3.1: Final Pandas Structure after running the first python script.

ELAN matching

The second script serves to append the real labels to the Pandas structures. After manual annotation as discussed at the beginning of chapter 3, a CSV file from the

ELAN Annotation Software was generated, shown in Figure 3.6. The structure of the CSV file generated by the ELAN Annotation Software is organized as follows:

- **First column**: Represents the hand side (e.g., "right" or "left").
- **Second column**: Indicates the start frame of the annotation, expressed in milliseconds.
- **Third column**: Indicates the end frame of the annotation, expressed in milliseconds.
- **Last column**: Contains the actual annotation, which describes the observed action or state (e.g., "contact," "no contact," "out of frame").

Hand	Start Time (ms)	End Time (ms)	Label
Right_Hand	364678	371657	Contact
Right_Hand	371659	377397	Contact
Right_Hand	377397	379335	No_Contact
Right_Hand	379335	380811	No_Contact
Right_Hand	380811	381278	No_Frame
Right_Hand	381278	383114	Contact

Table 3.2: ELAN Annotation Software CSV file.

The script performs 2 steps:

- **Label encoding**: Labels are converted into numerical format: 'No Contact' is 0, 'Contact' is 1, and 'out-of-frame' is 2.
- **Label extraction**: To assign labels, the timestamp in milliseconds of each frame in the DataFrame is compared with the time intervals defined in the CSV file. Matches add the corresponding label to the set, with row identifiers stored in a list.

3.2.5 Characterization

To evaluate the performance of the Hand Object Detector model, the Intersection over Union (IoU) metric was used by comparing a manually annotated dataset. A custom-built application developed in the MAL was used for manual annotation with bounding boxes. The ground truth dataset was created by segmenting 5-minute videos of 5 randomly selected patients, while performing different tasks.

Intersection over Union (IoU)

Intersection over Union (IoU) is a metric used to evaluate object detection performance by comparing the predicted bounding box to the ground truth bounding box (Figure 3.5). The IoU is calculated using the following equation:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

In this proportion, the numerator denotes the intersection surface between the predicted bounding boxes and the actual ground truth, while the denominator represents the aggregate surface encompassed by the union of each box.

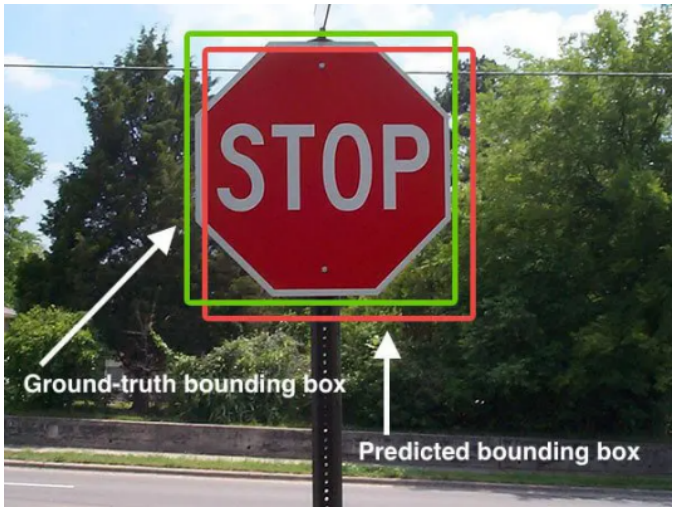

Figure 3.5: Bounding boxes for evaluating IoU.

Bounding Box Labeling Application

Figure 3.8 shows the interface of the custom-developed labeling application. Users can annotate frames by tracing and releasing the cursor around the elements of interest. The process is made efficient by the presence of IDs for hand orientation and objects.

The interface also includes a "draw previous bounding boxes" button, allowing users to replicate bounding boxes from previous frames when the action remains stationary in subsequent frames. Bounding box categories are differentiated by IDs, which are also used for the correction button. The tool's efficiency is enhanced by keyboard shortcuts and the ability to save and load annotations to and from a CSV file.

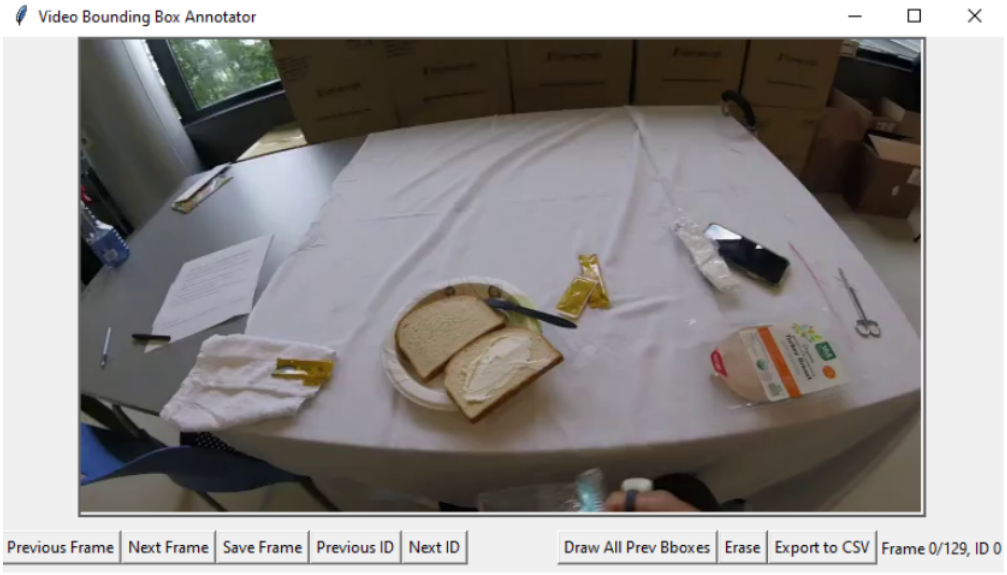

Figure 3.6: Screenshot of the application's user interface.

3.2.6 Stage 2

After preparing the dataset, the processing moves to the second module of the workflow using Snorkel, initially for model training and subsequently for label validation. The main steps of Snorkel are:

1. **Uploading the dataset** using its built-in functions. This involves data manipulation to ensure proper formatting for subsequent processing.

2. **Labeling Functions** are created to identify relationships among bounding box sets by incorporating particular insights. The functions, of Categorical type and Spatial type, enable the detection of existing relationships within the data. In this step, five labeling functions are applied to the dataset to detect *Movement*, *No Movement*, and *Uncertain* scenarios.

3. **Training the Label Model**: The performance of the labeling functions is initially evaluated and then improved to more correctly represent the underlying relationships in the collected data. The revised functions are implemented into the dataset to assign preliminary labels. Subsequently, a generative model is used to further enhance the quality of the labels.

Once the label model is trained, the annotations are integrated into the dataset.An additional script was added to the pipeline to streamline the visualization of outputs in ELAN. The model's performance was evaluated in section [4.1.3] using specific metrics such as F1-score and confusion matrices.

3.2.7 Snorkel

Snorkel [31] is an innovative system that enables users to train advanced models without hand-labeled data. LFs are created by the users, in order to underly patterns in an unlabeled dataset. The noise is effectively discarded from the outputs by Snorkel without the presence of ground truth, utilizing the data programming, a novel ML approach. Figure 3.7 succinctly shows the design of the system. The system operates as follows:

1. Subject matter experts (SMEs) design labeling functions that capture weak supervision sources, incorporating methods like distant supervision, patterns, and heuristics.

2. Snorkel applies these LFs to unlabeled data, generating a probabilistic model that combines the outputs of the LFs into probabilistic labels.

3. These probabilistic labels are then used to train a discriminative classification model, such as a deep neural network.

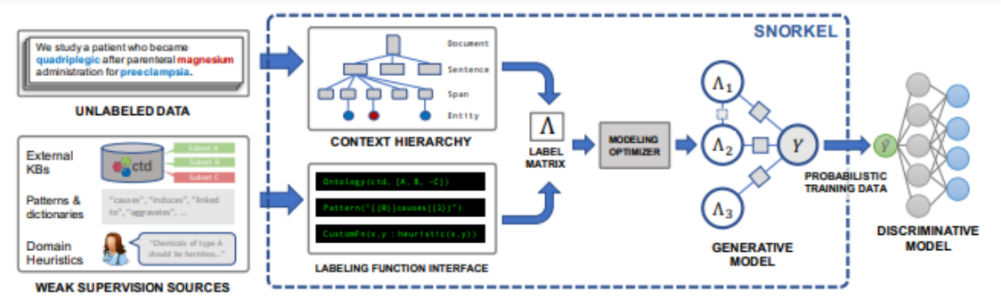

Figure 3.7: Workflow of the Snorkel architecture

Labeling Functions

Users do not manually label training data; instead, they create labeling functions. These functions are divided into two main types: Categorical and Spatial intuitions. For the RingSensor study, five labeling functions were developed, three categorical and two spatial, specifically for the *Movement* label.

- **Categorical LFs**: The categorical intuitions involve knowledge of typical categories of subjects and objects in relationships (e.g., 'person' as the subject for actions like 'ride' and 'carry').
  - If both hands and objects are detected by the deep learning models, it is labeled as '**Movement**'.

  - The '**No Movement**' label is used if no object is detected, but the hand.
  - The '**Uncertain**' label is applied if no hands are in the frame.

- **Spatial LFs**: The Spatial intuitions involve understanding the relative positions of subjects and objects (e.g., the subject is usually above the object in 'ride' actions).
  - The '**Movement**' label is determined by the overlap between bounding boxes of hands and objects, assigned if the overlap percentage surpasses a pre-established breakpoint.
  - To address scenarios where contact could persist despite minimal overlap, a function verifies if the centroid of the hand falls inside the bounding box of the object.

Once the labeling functions are applied, Snorkel produces a table displaying the effectiveness of the LFs. The table can be examined in section [4.1.2].

Model training

Using the probabilistic labels produced by the generative model, Snorkel trains a discriminative classification model, such as a deep neural network. The training process for this model spanned 100 epochs, utilizing a learning rate set at 0.01.

3.2.8 Python Script

Following the labeling process, the performance of the entire pipeline was evaluated using metrics on the data, as detailed in the next chapter. Additionally, the annotated data can be visualized on ELAN. The results from Snorkel consisted of a vector of numeric labels corresponding to each frame in the video. These elements were synchronized using frame timestamps in milliseconds, aligning the row IDs with the newly generated labels. The labels are matched to the rows of the original dataset and transformed from numerical to textual in the final part of the script. Subsequently, the training set was labeled.

One of the challenges to address is the presence of brief label variations: Snorkel labels frame by frame, so there are instances where a label temporarily changes before reverting to its previous value. Therefore, the script checks for consecutive frames labeled with different labels in blocks of fewer than 7 frames (approximately 0.23 seconds) and stabilizes the values by assigning them the label of the preceding frame. This enhancement improves performance accuracy, ensuring that sequences of consecutive frames with the same label consist of at least 7 elements.

We established a new pandas structure to organize consecutive sequences of frames with the same label, recording start and end timestamps in milliseconds along with

their corresponding annotations to ensure proper display in ELAN. Sequences were sorted by their start timestamps. The outcome was saved in a structured table within a CSV file, an example of which is shown in Table 3.3. Applying the same process to Snorkel's results enabled a visual comparison between predictions and ground truth labels using ELAN.

Hand	Start Time (ms)	End Time (ms)	Label
Right_hand	371671	377344	Contact
Right_hand	377344	377544	No_Contact
Right_hand	377544	377711	Contact
Right_hand	377711	377978	No_Contact
Right_hand	377978	378311	No_Frame

Table 3.3: Annotated Frames with Hand Activities

3.2.9 Characterization

The effectiveness of the proposed method was evaluated using the weighted F1-score metric, typically employed in multi-label machine learning frameworks and derived from confusion matrices. Confusion matrices are essential tools for assessing classification model capability.

F1-score

The F1 score assesses the accuracy of a test set by evaluating both precision and recall. Precision denotes the proportion of true positives among all positive predictions, whereas recall represents the proportion of true positives among all positive instances (true positives plus false negatives). The F1 score is computed as the harmonic mean of precision and recall, reflecting the frequency of accurate predictions. In cases involving multiple classes, the metric employed is the micro F1-score, while when the dataset is imbalanced, the weighted F1-score is preferred. The formula for the weighted F1-score is:

$$\textbf{F1 weighted-score} = \sum_{i=1}^{k} w_i \cdot \text{Score}_i \tag{3.1}$$

where w_i is defined as:

$$w_i = \frac{\text{Number of samples in class } i}{\text{Total number of samples}} \tag{3.2}$$

Confusion Matrix

The confusion matrix, also known as a misclassification matrix, provides a visual representation of the accuracy of a statistical classification. Each column of the matrix represents predicted outcomes, while each row corresponds to actual outcomes. The entry at row i and column j shows the number of instances where the classifier predicted class j when the true class was i. This matrix allows for the observation of any discrepancies in the classification across different classes. Figure 3.8 illustrates the structure of a misclassifications matrix in a binary classification scenario:

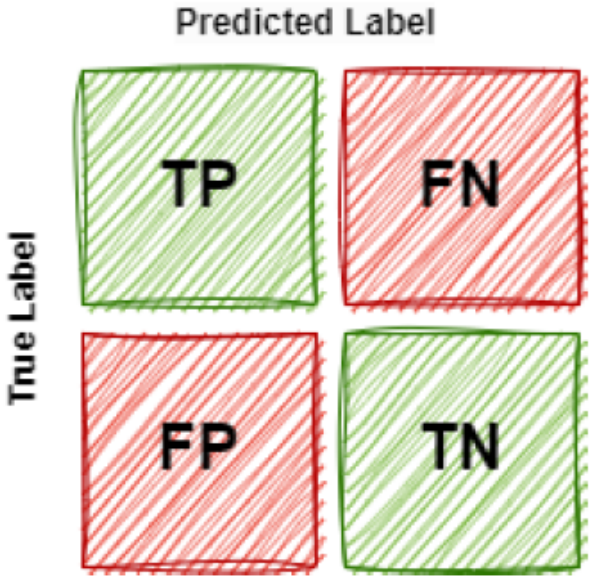

Figure 3.8: Binary Confusion Matrix.

In multi-class classification scenarios, the misclassifications matrix expands to accommodate multiple classes. Rows correspond to the actual (ground truth) classes, while columns represent the predicted classes. Each cell indicates the count of predictions for each class combination. Diagonal entries denote accurate predictions for every class, while off-diagonal entries indicate errors in classification.

- **TP - True Positive**: Correctly predicted positive instances.
- **FP - False Positive**: Incorrectly predicted positive instances.
- **TN - True Negative**: Correctly predicted negative instances.
- **FN - False Negative**: Incorrectly predicted negative instances.

Analyzing the misclassifications matrix allows for a quick assessment of not only the number of correct predictions but also the types and frequencies of errors made by the model. Figure 3.9 presents an example of a multi-class confusion matrix.

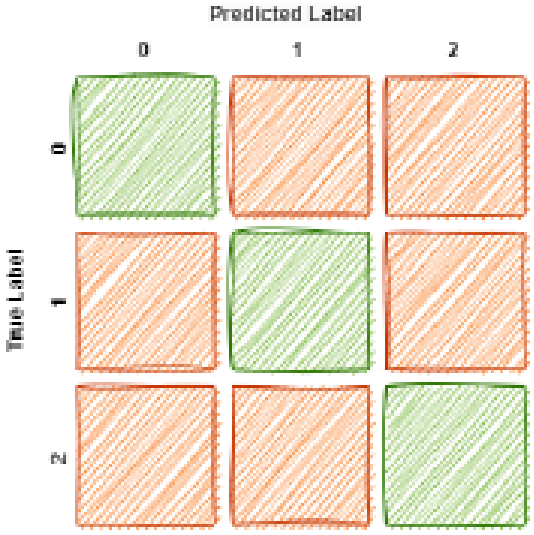

Figure 3.9: Multi-class Confusion Matrix for a Classifier with Three Labels

3.3 Second pipeline

In light of the results obtained from the evaluation of the initial pipeline, it was observed that while the results for movements were promising, those related to *No Movement* were lacking. These results, which will be discussed in detail in the appropriate section, highlighted significant limitations, such as data imbalance with minimal coverage for the *No Movement* label and relatively low accuracy of labeling functions for this category.

The main limitations included data imbalance, unsatisfactory accuracy of labeling functions for the *No Movement* label, the need for continuous fine-tuning specific to the data, and training data constraints that limited the model's generalization to real-world scenarios.

As a result, significant modifications were made to the existing pipeline to address these issues. One of the main innovations introduced was the integration of a Large Language Model (LLM). This change was motivated by the need to improve the accuracy of automatic labeling and to enhance the system's robustness in recognizing various activities, including *No Movement* cases.

The new pipeline was designed to include modified blocks from the base pipeline, described up to this point, along with the addition of advanced components for data processing and analysis. However, the Hand Object Detector (HOD) block was retained, as the characterization results, discussed later, were satisfactory. These modifications aim to optimize the entire process of automatic upper limb activity labeling, ensuring greater accuracy and better adaptability to real-world data.

3.3.1 Chest video

Including chest-mounted videos along with head-mounted ones in our analysis was a strategic decision aimed at enhancing the robustness and accuracy of our activity recognition framework. The visual data from both perspectives provide complementary information, which is crucial for a comprehensive understanding of upper limb activities.

- **Complementary Perspectives**: Head-mounted cameras offer a direct view of the subject's hands and the objects they interact with, but they might not capture the full range of movements, especially those involving the lower body or interactions outside the direct line of sight. Chest-mounted cameras, on the other hand, provide a broader view, capturing movements that head-mounted cameras might miss. This dual perspective ensures that all relevant actions are recorded, enhancing the completeness of the data.
- **Improved Recall Rates**: Comparing the recall rates for head and chest videos highlights the importance of considering multiple viewpoints. While head-mounted video analysis offers valuable insights, chest-mounted video analysis can capture movements and static periods with higher precision on the *No Movement* label, as will be shown in Chapter 4.
- **check consistency**: handles and corrects missing bounding boxes by interpolating positions between consecutive frames. This ensures that the dataset is complete and free from gaps, enhancing the overall quality of activity detection.
- **Enhanced Algorithm Performance**: Including chest videos allows our algorithms to access a richer, more diverse dataset, which improves their ability to generalize. Training the models on data from both head and chest videos results in greater accuracy and reliability in detecting and classifying upper limb activities.
- **Improved Recall Rates**: Comparing the recall rates for head and chest videos highlights the importance of considering multiple viewpoints. While head-mounted video analysis offers valuable insights, chest-mounted video

analysis can capture movements and static periods with higher precision on the *No Movement* label, as will be shown in Chapter 4.

3.3.2 Python scripts

Output correction module

To streamline the process, the Python scripts from the initial pipeline were consolidated into a unified module. The goal of this integration was to improve the efficiency and consistency of the data processing pipeline. Figure 3.10 illustrates the workflow of these newly integrated functions. Additionally, new functions were defined to improve the output of the Hand Object Detector (HOD). These improvements focused on addressing key areas such as correcting hand side detection based on positional data within the frame, filling gaps by identifying and interpolating missing frames and bounding boxes, and ensuring label consistency.

- **assign hand side**: created to ensure that detected hands are correctly labeled as left or right. This is essential for maintaining the consistency of labels with the actual position of the hands within the video, not only checking the position with respect to the center of the frame, improving the accuracy of activity detection.

- **process missing bbox**: ensures the consistency of hand labels across frames. It verifies and corrects any discrepancies, ensuring that the timeline of hand activities is accurate and free from labeling errors.

- **check consistency**: handles and corrects missing bounding boxes by interpolating positions between consecutive frames. This ensures that the dataset is complete and free from gaps, enhancing the overall quality of activity detection.

- **process elan data**: responsible for integrating and aligning external annotation data with the detected hand activities. Specifically, it reads the ELAN annotation file, processes and normalizes the labels, and then aligns these labels with the corresponding frames in the dataset. This function ensures that the ground truth labels from the ELAN annotations are accurately matched with the detected hand activities, providing a reliable basis for evaluating the performance of the detection and labeling process.

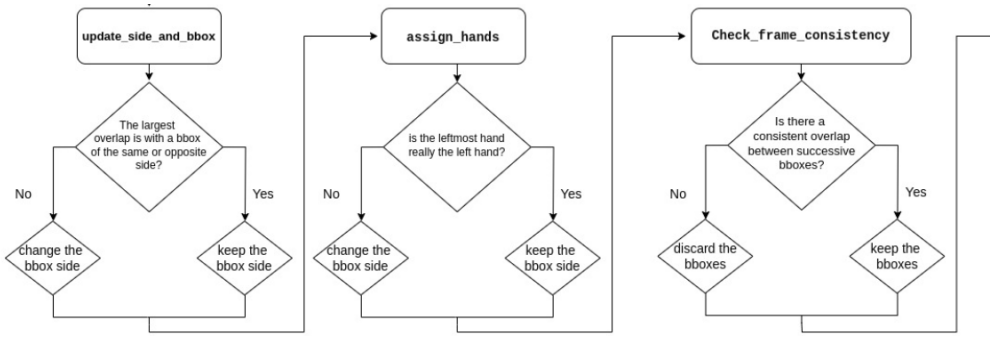

Figure 3.10: Integrated new functions

3.3.3 Large Language Model

The Generative Image-to-Text (GIT) model [32] is an advanced LLM designed to bridge the gap between visual and textual data. It employs a Transformer decoder architecture, which has proven to be highly effective in natural language processing tasks. The GIT model is trained using a method known as "teacher forcing" on pairs of images and their corresponding text descriptions. This training methodology ensures that the model can accurately generate descriptive text based on visual input.

Model Architecture

The model structure is illustrated in Figure 3.11 and defined as described below:

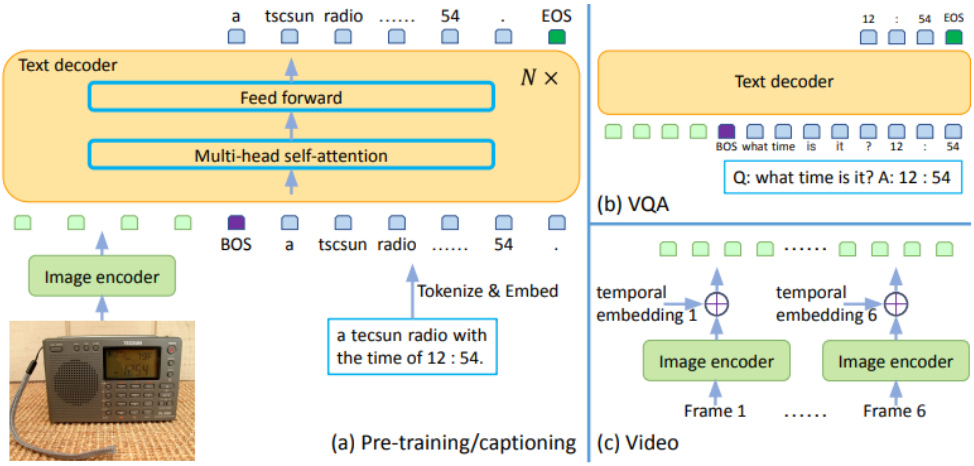

Figure 3.11: The GIT network architecture.

- **Image Encoder**

  The first component, the Image Encoder, is based on a pre-trained model using contrastive tasks. This encoder receives raw images as input and transforms them into a compact 2D feature map. This map is then flattened into a list of features projected into dimension D through a linear layer and a normalization layer. The Image Encoder was selected for its superior performance in object recognition tasks, ensuring a robust visual representation that feeds into the subsequent Text Decoder.

- **Text Decoder**

  The second component, the Text Decoder, is a transformer module designed to predict the textual description of images. It is composed of multiple transformer blocks, each including a self-attention layer and a feed-forward layer. The text is tokenized and embedded into dimension D, with positional encoding and a normalization layer added to maintain sequential coherence. The image features, generated by the Image Encoder, are concatenated with the textual embeddings as input for the transformer module. The text decoding process begins with the [BOS] token and proceeds auto-regressively until reaching the [EOS] token or the maximum number of allowed steps, ensuring smooth and coherent text generation.

- **Capabilities and Performance**

  The GIT model is capable of performing various tasks such as image captioning, visual question answering (VQA), and image classification.

  - ***Image Captioning***: In this task, the model generates a descriptive sentence for a given image, as shown in part (a) of the figure. The image encoder processes the visual data, while the text decoder, equipped with feed-forward and multi-head self-attention layers, generates the corresponding textual description.
  - ***Visual Question Answering*** (VQA): As illustrated in part (b), the model can understand and answer questions related to the visual content. This involves encoding both the image and the question text, allowing the decoder to produce a relevant answer.
  - ***Video Analysis***: In the context of video analysis, shown in part (c), the GIT model handles sequences of frames, applying temporal embeddings to capture the dynamic information across frames. This enables the model to perform tasks like video captioning and activity recognition.

- **Pre-training and Fine-tuning**

  The GIT model undergoes two main training phases: pre-training and fine-tuning.

  - ***Pre-training***: During pre-training, the model is trained to map the input image to the associated text description using the language modeling (LM) objective. In this phase, a cross-entropy loss with label smoothing of 0.1 is applied, enhancing the model's ability to generalize from the training data.
  - ***Fine-tuning***: Fine-tuning varies depending on the specific task. For the image captioning task, the training data format remains the same as in pre-training, applying the same LM objective. For the visual question answering (VQA) task, the question and the correct answer are concatenated as a new special caption during fine-tuning, but the LM loss is applied only to the answer and [EOS] tokens. During inference, the question is interpreted as the caption prefix, and the completed part is the prediction, allowing the model to generate relevant answers based on the provided visual context.

Snorkel integration

We utilized the LLM to analyze images and generate textual descriptions or responses that provide crucial information about the visual data. These results were then integrated into our DataFrames as an additional column called 'GIT_Result'. The results generated by the LLM were used as input for Snorkel's labeling functions, enhancing the accuracy of the produced labels.

We created several labeling functions in Snorkel, each designed to leverage specific characteristics of the data and the information provided by the LLM. Here is an overview of the labeling functions used:

- *LF_no_contact*: This function uses the result generated by the LLM (GIT _Result). If the result is "no", indicating no contact, the function returns the **NO MOVEMENT**label. Otherwise, it returns **ABSTAIN**.

- **LF_area**: This function calculates the overlap area between the bounding boxes of the hand and the object. If the overlap area exceeds a predefined threshold (10%), the function returns the **MOVEMENT** label, indicating contact. Otherwise, it returns **ABSTAIN**.

- **LF_centroid**: This function calculates the distance between the centers of the bounding boxes of the hand and the object. If the distance exceeds a predefined threshold (1000 units), the function returns the **NO_MOVEMENT**label, indicating no contact. Otherwise, it returns **ABSTAIN**.
- **LF_dist**: This function checks the relative position of the bounding boxes of the hand and the object. If the hand is within the bounds of the object, the function returns the **MOVEMENT** label. Otherwise, it returns **ABSTAIN**.

Model training

Leveraging the probabilistic labels generated by the generative model, Snorkel trains a discriminative classification model, trained for 10,000 epochs with a learning rate of 0.01.

3.3.4 Python Script

Smoothing module

This module addresses the issue of fluctuating labels, which can occur due to inherent noise and variability in video data: for each frame, the smoothing module identifies and counts label changes within a 21-frame window. The workflow of the smoothing module is illustrated in Figure 3.12, maintaining data integrity, which in turn improves the accuracy and reliability of the analysis performed on the labeled video data.

1. Extract Label: The module starts by extracting the label for each frame.
2. Identify and Count Label Switches: It then identifies any label switches within the specified window and counts their occurrences.
3. Apply Corrections: If the number of label switches within the window is less than 21, the module uses the previous label to maintain consistency. If the count is 21 or more, it applies the current label.

3.4 ELAN

This annotations tool is used as a final step to provide visual feedback and better interpretation of the results from confusion matrices, highlighting any shortcomings in the model. ELAN allows users to annotate multimedia files with unlimited written notes, such as characterization of attributes and comments, and also

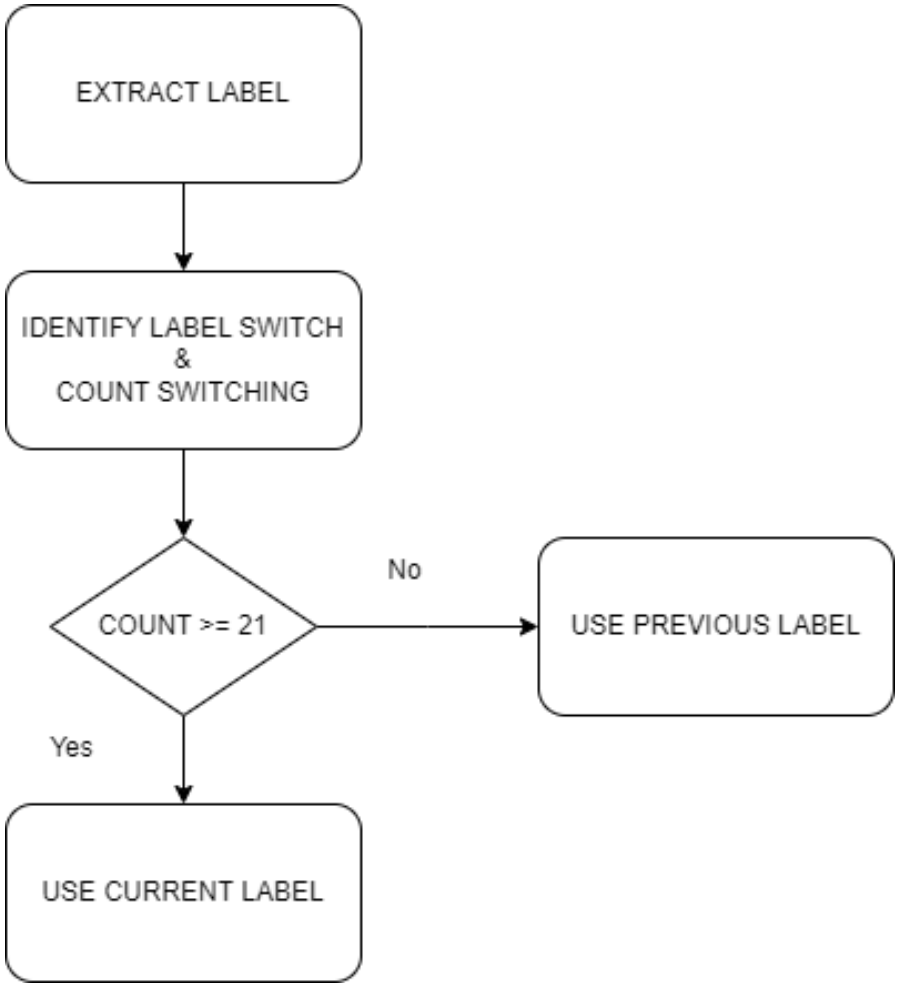

Figure 3.12: Smoothing module workflow.

enables these notes to be organized into tiers that can be hierarchically linked. Additionally, annotations can be made to correspond to specific moments in the media or can be linked to pre-existing annotations. For the RingSensor video data, this synchronization was particularly advantageous, as demonstrated in Figure 3.13. ELAN is a strong but complex tool due to its sequential method of operation. Users must select each label layer within a specific time window and choose the suitable annotation. Additionally, annotations need to be tailored for both limbs, as detailed in section 2.2.4., a detailed step that slows down the labeling process unless more annotations are used.

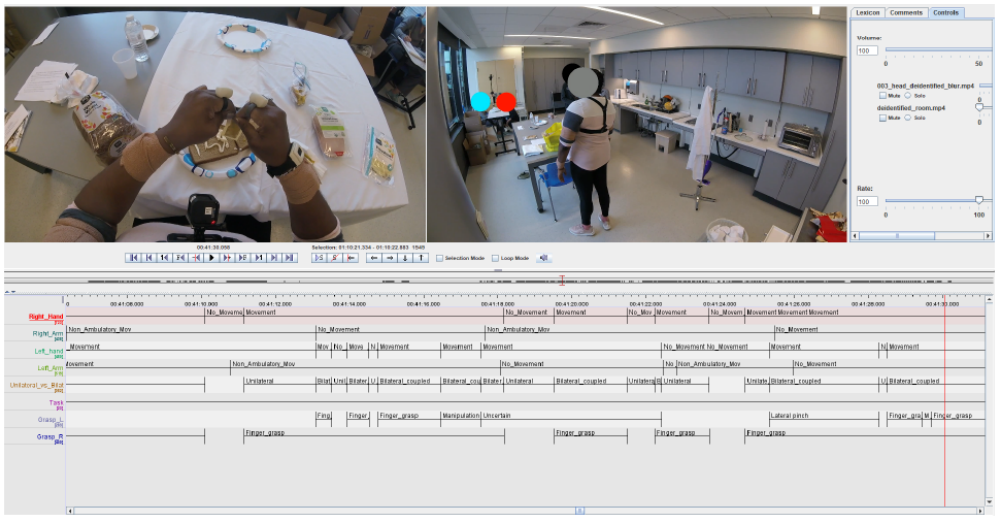

Figure 3.13: ELAN interface

Chapter 4

Results

4.1 First pipeline

4.1.1 Hand Object detection

To assess the stage 1 model, we began by observing how the Hand Object Detector (HOD) processes our video data. This involved reviewing the videos after processing by the HOD model. The outcome, shown in Figure 4.1, demonstrate that the HOD model accurately pinpoints hand bounding boxes and recognizes only the objects they are interacting with, ignoring other items in the frame and focusing exclusively on the book the subject is holding.

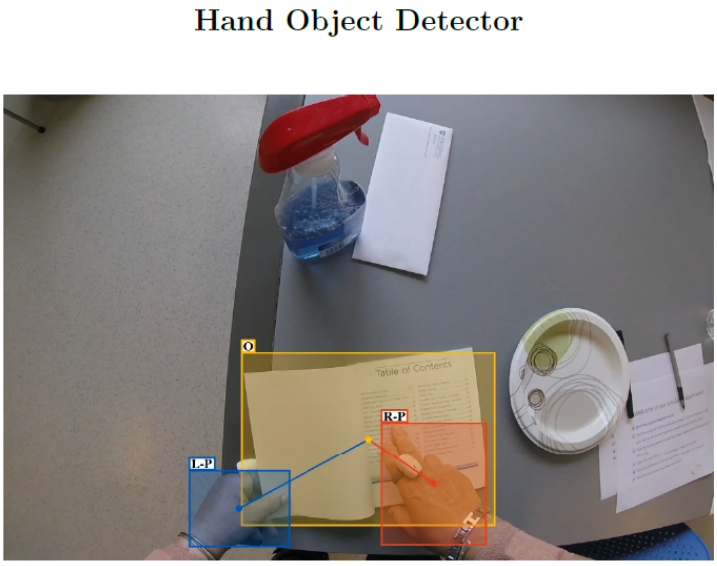

Figure 4.1: Snapshot of an HOD output frame.

We evaluated how well the model performs by comparing manually annotated bounding boxes around hands and objects in contact in casually chosen five-minute video segments from different patients. Specifically, we computed the Intersection over Union (IoU) to measure the overlap between these annotations and the predictions of the HOD model.

Table 4.1 presents the precision of the right-hand bounding boxes identified by the HOD model. The "Hand" column lists the number of accurately identified right-hand frames out of the total frames showing that hand. The "IoU" column provides the mean IoU score for the accurately recognized frames depicting the right limb, demonstrating the model's predictive accuracy and reliability.

Subjects	**Hand det.**	**IoU**
002	901/1284	0.87
004	951/1063	0.86
005	965/1033	0.87
014	1285/1377	0.93
019	798/842	0.91
Avg.	4732/5598	0.90

Table 4.1: IoU for HOD.

4.1.2 Labeling Function - Snorkel

In stage 2, the Snorkel labeling function (LF) evaluation tool gave a comprehensive overview of how effectively the crafted LFs aligned with the validation dataset. This analysis yielded valuable insights into the performance of the LFs and identified if modifications were needed. The evaluation encompassed the succeeding indicators:

- **Polarity**: The group of exclusive labels generated by this LF.
- **Coverage**: The ratio of the dataset annotated by the LF.
- **Overlaps**: The ratio of the dataset where this LF and at least one other LF have provided labels.
- **Conflicts**: The ratio of the dataset annotated by both this LF and another LF where their labels differ.
- **Correct**: The count of data points correctly labeled by this LF (if ground truth labels are available).
- **Incorrect**: The count of data points incorrectly labeled by this LF (if ground truth labels are available).

- **Empirical Accuracy**: The observed accuracy of this LF (given ground truth labels).

Labeling Function	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_obj	0	[1]	0.51	0.50	0.0	19154	2761	0.87
lf_overlap	1	[1]	0.50	0.50	0.0	18894	2574	0.88
lf_distance	2	[1]	0.31	0.31	0.0	11409	1948	0.85
lf_no_obj	3	[0]	0.10	0.0	0.0	1454	3041	0.32
lf_out_frame	4	[2]	0.39	0.0	0.0	11593	5097	0.69

Table 4.2: LF Analysis on a small subset.

Labeling Function	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_area	0	[1]	0.66	0.51	0.20	55707	25073	0.69
lf_no	1	[0]	0.03	0.03	0.03	2319	1377	0.63
lf_centroid	2	[0]	0.0055	0.0055	0.0055	355	313	0.53
lf_no_contact	3	[0]	0.18	0.18	0.18	6905	15027	0.31
lf_dist	4	[1]	0.42	0.42	0.11	36038	15518	0.70

Table 4.3: LF Analysis with GIT on the entire dataset.

4.1.3 Confusion matrix & F1-score

When evaluating the pipeline results, we used a confusion matrix for visualization, created by combining the data from all present subjects. Each column of the confusion matrix represents the predicted values, while each row represents the actual values.

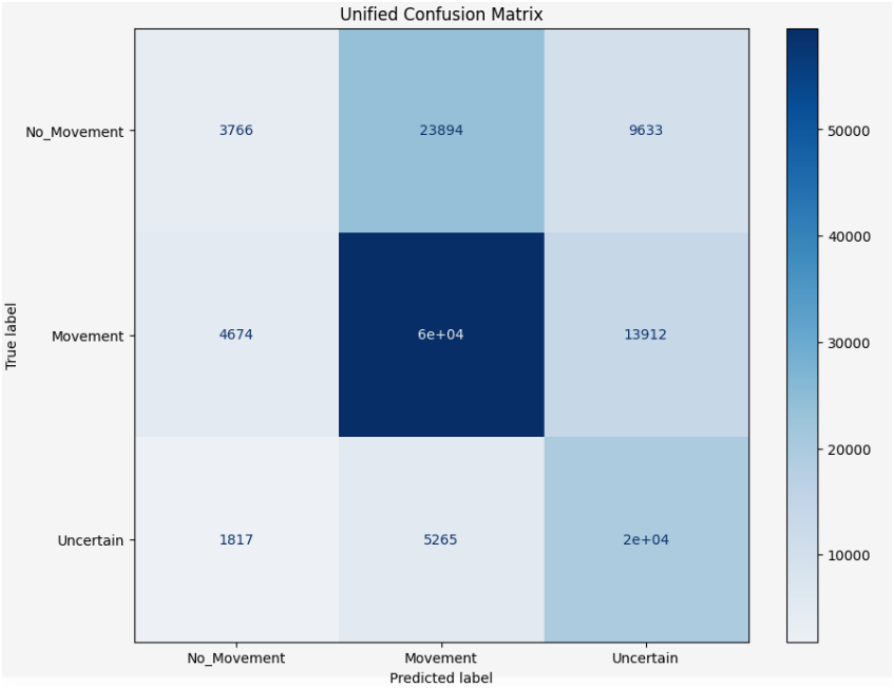

Figure 4.2: Confusion Matrix - Head.

Subject	F1 Score
004	0.69
005	0.42
011	0.66
013	0.57
014	0.75
019	0.56
combined	**0.49**

Table 4.4: F1 Score by Subject - Head.

Subject	F1 Score
004	0.44
005	0.46
011	0.57
013	0.36
014	0.52
019	0.37
combined	**0.45**

Table 4.5: F1 Score by Subject - Chest.

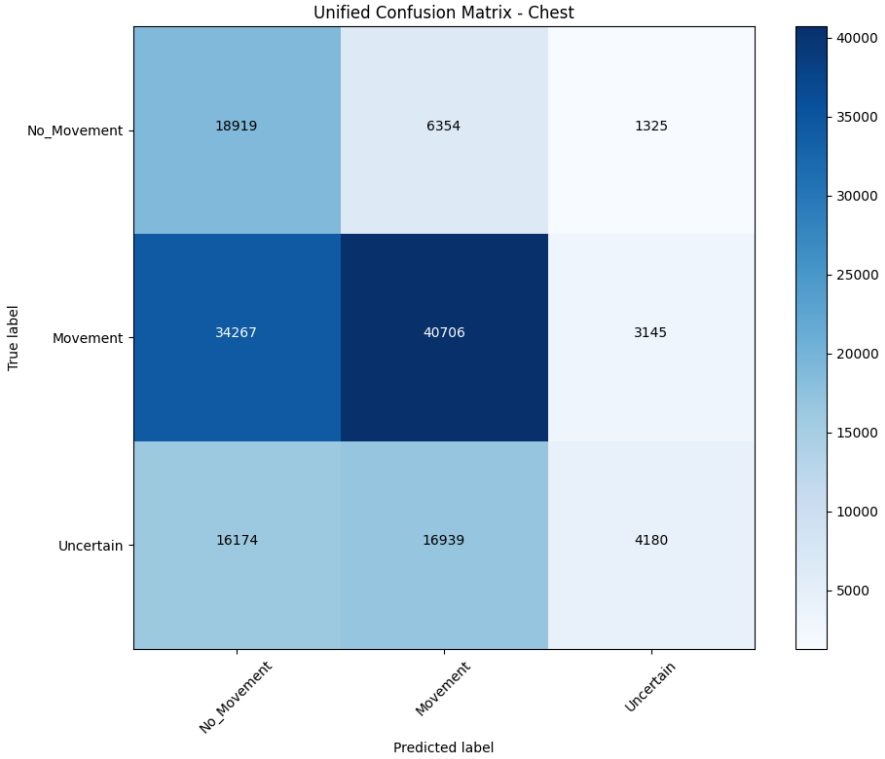

Figure 4.3: Confusion Matrix - Chest.

Subject	F1 Score
004	0.68
005	0.41
011	0.65
013	0.60
014	0.75
019	0.62
combined	**0.60**

Table 4.6: F1 Score by Subject - Combined.

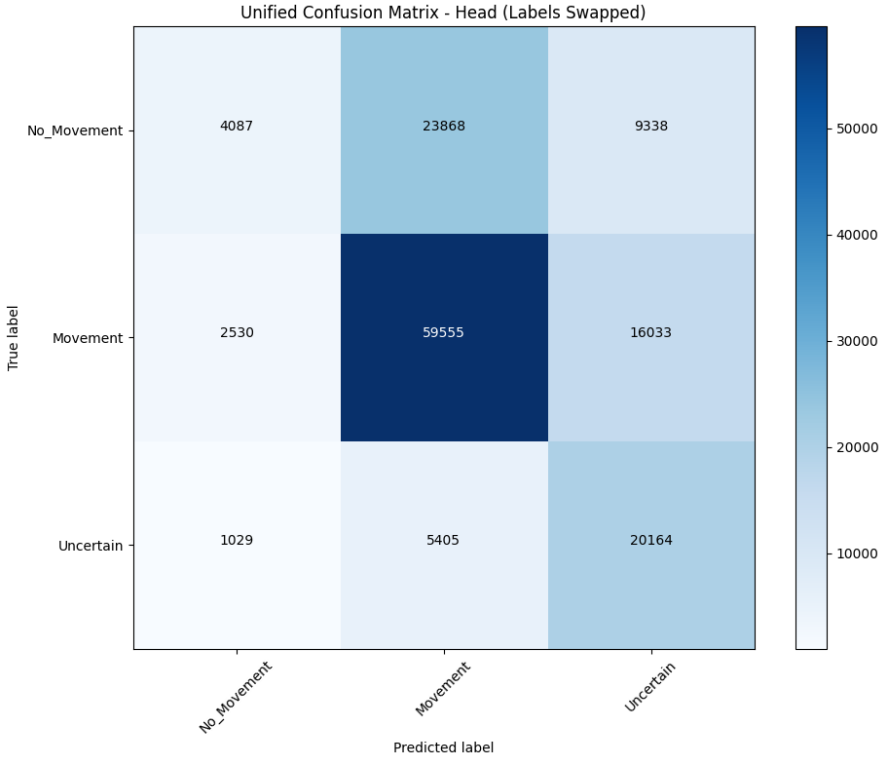

Figure 4.4: Confusion Matrix - Combined.

Subject	**F1 Score**
004	0.71
005	0.47
011	0.71
013	0.58
014	0.75
019	0.63
combined	**0.64**

Table 4.7: F1 Score by Subject - correction module - Head.

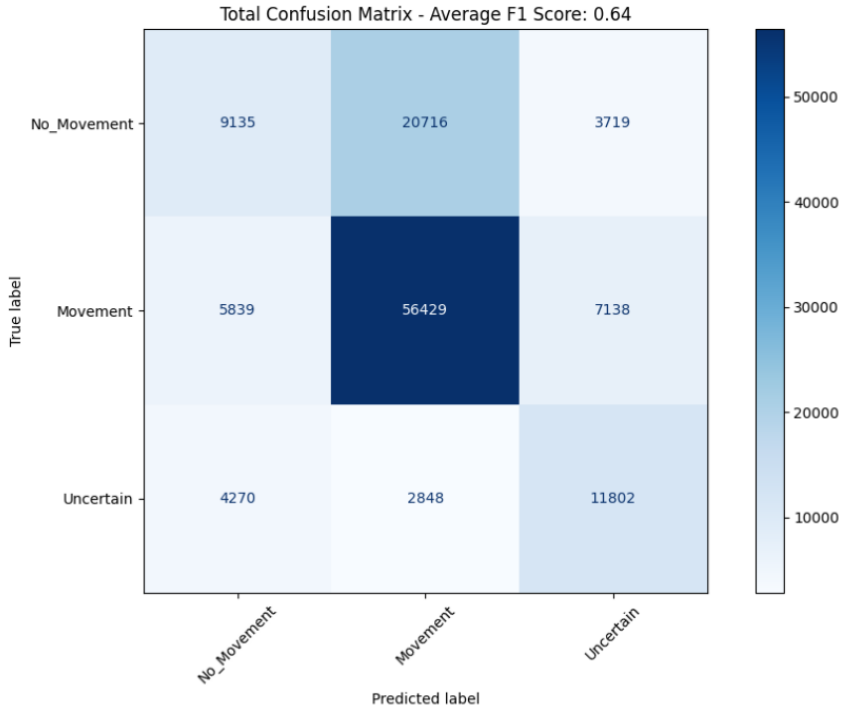

Figure 4.5: Confusion Matrix - correction module - Head.

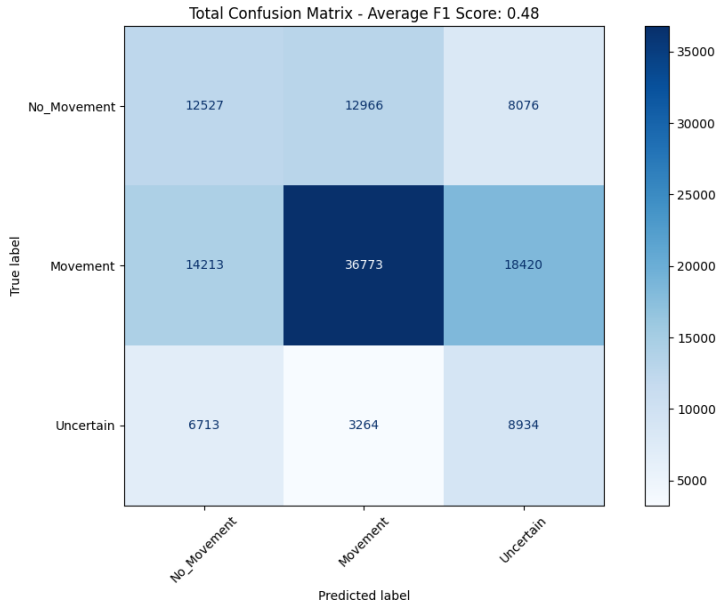

Figure 4.6: Confusion Matrix - correction module - Chest.

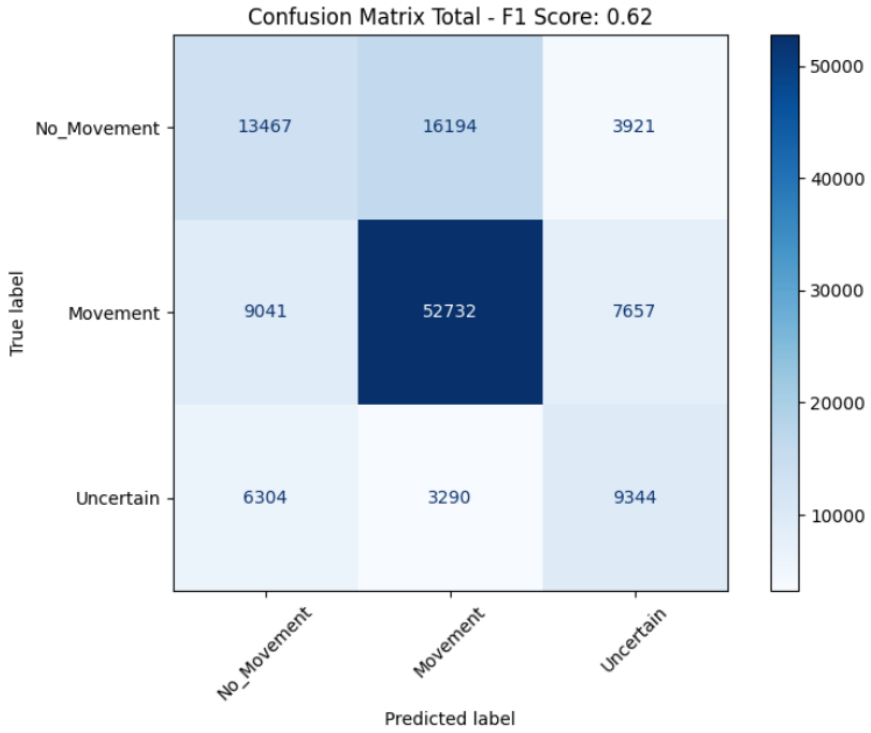

Figure 4.7: Confusion Matrix - GIT - Head.

Chapter 5

Discussion

5.1 First pipeline

5.1.1 Hand Object detection

Hand Object Detector model was trained to evaluate its ability to recognize hands in RingSensor videos, a crucial step to determine the feasibility of the project. Figure 4.1 illustrates how the HOD identifies hands in the videos. Although these images alone are not sufficient for a comprehensive assessment of the model's performance, they helped us confirm that the model was functioning correctly on our video data. We observed that, despite the presence of other objects in the video, the model exclusively highlighted the objects in contact with the hands.

To obtain a more detailed evaluation of the model's performance, we manually annotated bounding boxes for hands and objects in contact in casually chosen five-minute video segments from 5 different patients, performing the simulated activities described in Chapter 2. The results are shown in Table 4.1, where the *Hand* column indicates the number of correctly detected right-hand frames out of the total frames containing the right hand. The IoU column provides the mean IoU score for the accurately recognized frames depicting the right hands. Overall, the HOD model proved to be highly effective, with an average IoU of 90% across subjects, confirming its high performance for our purpose.

5.1.2 Snorkel

Following the creation of the labeling functions (LFs), the results are compiled into a DataFrame, impacting phase 2 of the pipeline. The LFs must be evaluated to ensure accurate and consistent labels, considering coverage, precision, and conflicts. The results are used to train a supervised labeling model. The quality of the labels

is essential for the effectiveness of the final model, making the correct writing and evaluation of LFs crucial.

In Table 3.3, which presents the performance of the LFs on the model, a high coverage of the *Movement* label (Polarity 1) is highlighted. The Overlaps column indicates that the three LFs for the *Movement* label are highly consistent and possess superior accuracy compared to the others, demonstrating the greater focus of the pipeline on detecting hand/object contact, while the coverage for the *No Movement* label is very low.

We prefer having a broader coverage because it simplifies the creation of time intervals for the *Movement*. This approach provides clinicians with predefined time intervals, significantly speeding up the grasping labeling process. Initially, having greater coverage for the *Movement* label is advantageous as it ensures that most significant interactions are captured, providing a solid starting point for further analysis. It is crucial to avoid false positives for both *Movement* and *No Movement*. False positives for *Movement* can lead to misinterpretations of irrelevant actions as significant contacts, while false positives for *No Movement* can overlook important events, reducing the model's overall accuracy. Although prioritizing broader coverage initially is beneficial, refining the model to minimize these errors is essential for accurate and reliable classification. Misclassification between *Movement* and*No Movement* results in more time spent manually correcting these inaccuracies. This concept can be better understood by analyzing the frames in the validation set. Approximately 59% of the frames are labeled as *Movement*, and the Hand Object Detector aligns closely with this figure, indicating comprehensive coverage of *Movement* labels. The *Uncertain* label represents about 30% of the dataset, but the model might mistakenly label instances as *Out of frame* even when hands are present, due to its excessive coverage. Additionally, the *No Movement* label accounts for roughly 25% of the frames, but it has a high error rate of approximately 89.90%, indicating frequent misclassifications. These issues suggest that while the model has strong coverage in some areas, significant refinements are needed to improve overall accuracy and reduce mislabeling.

Figure 4.2 shows the combined confusion matrix for all subjects, based on the head-mounted video recordings. This confusion matrix, together with the F1 scores reported in Table 4.3, provides a detailed picture of the model's detection performance. The combined confusion matrix highlights that while the model is generally effective at detecting *Movement*, there is significant confusion between *No Movement* and *Movement*. Additionally, the individual F1 scores suggest that the model's performance varies considerably across subjects, indicating the need for further optimization to improve the model's consistency.

After training the labeling model and calculating the weights, the observed performance showed a significant relationship between the analysis of labeling functions

and the patterns in the confusion matrices. The *No Movement* label demonstrated lower precision and recall compared to the other labels, while the *Uncertain* label achieved moderate precision and high recall. As expected, the *Movement* label emerged as the most effective, showing high precision and recall.

Label	Precision	Recall
No_Movement	0.367	0.101
Movement	0.858	0.763
Uncertain	0.459	0.739

Table 5.1: Precision and Recall - Head.

Analyzing the metrics reveals that the:

- *No Movement* label falls short of expectations, showing both low precision and recall. This indicates that the model is not effective in accurately detecting *No Movement* instances.
- *Uncertain* label, it exhibits moderate precision and high recall, suggesting that the model often assigns this label even in uncertain situations. Despite this, the 'Uncertain' label is useful in the annotation software as it establishes a time interval that aids clinicians in the labeling process. However, it is clear that this label requires further refinement to improve its accuracy.
- *Movement* label demonstrates high performance with both high precision and medium-high recall, indicating that the model is generally accurate in identifying movement. This reliability makes the annotation process much more efficient, with time intervals determined more accurately, increasing the likelihood of correct annotation and speeding up the workflow.

5.1.3 Chest camera

A significant support for the decision to integrate the chest camera is provided by the results of the confusion matrix for the chest (Figure 4.3) and the F1 scores per subject reported in Table 4.4. The F1 scores range from 0.36 to 0.57, with a combined score of 0.49, indicating overall better performance compared to the results obtained with the head camera alone. This suggests that the integration of the chest camera can indeed contribute to improving the accuracy and consistency of the model's detections.

The integration of the chest camera was considered based on a deeper analysis of the confusion matrix results. The precision for the *No Movement* label in the chest set was found to be about 70%, indicating that when the model predicts *No*

Label	Precision	Recall
No_Movement	0.700	0.277
Movement	0.521	0.635
Uncertain	0.112	0.483

Table 5.2: Precision and Recall - chest.

Movement, it is correct 70% of the time. However, the recall for the same label was only 28%, showing that the model misses a significant number of *No Movement* instances.
Given these observations, the decision to integrate the chest camera aims to address this discrepancy. By incorporating additional data from a different perspective, it is anticipated that the model's ability to detect *No Movement* instances will improve, both in terms of identifying true positives and reducing false negatives.
Additionally, the F1 scores among the various subjects in the head set, ranging from 0.36 to 0.57, highlight the variability in the model's performance. The combined F1 score of 0.45 further underscores the need for improvement. By integrating the chest camera, we expect more consistent and reliable detections across different subjects, thereby enhancing the overall robustness of the model.

5.2 Second pipeline

5.2.1 Chest and head integration

The results of combining the chest camera and head camera show a significant improvement in the overall performance of the Hand Object Detector (HOD) model. Analyzing the combined confusion matrix (Figure 4.3) and the F1 scores per subject reported in Table 4.4, we can observe a clear enhancement in the model's detection capabilities.

Label	Precision	Recall
No_Movement	0.109	0.422
Movement	0.763	0.670
Uncertain	0.702	0.443

Table 5.3: Precision and Recall - head.

Analyzing the metrics reveals that:

- *No Movement*

The combination of the two perspectives has led to a reduction in false positives and negatives, as evidenced by the values in the confusion matrix. The precision for *No Movement* was calculated to be approximately 0.109, while the recall was 0.422. This shows that the model has improved its ability to correctly identify instances of*No Movement*.

- *Uncertain*

  The precision for *Uncertain* was calculated to be approximately 0.702, while the recall was 0.443.

- *Movement*

  The precision and recall for *Movement* show improvement over the separate sets. The number of true positives has increased, while false positives and negatives have decreased. The precision for *Movement* was calculated to be approximately 0.763, while the recall was 0.670.

The integration of the chest camera with the head camera has provided a more complete and accurate view of the activities performed. This has led to a reduction in classification errors and an improvement in precision and recall for all labels. The combined F1 scores show a significant increase compared to the scores obtained using the individual perspectives. The combined F1 score of 0.60 represents a substantial improvement, suggesting that using both perspectives allows the model to better detect different activities and movements.

5.2.2 Output correction module

Head

The application of the output correction module to the HOD model for the head camera data has yielded significant improvements. As illustrated in Figure 4.5, the confusion matrix shows notable enhancements in the model's detection capabilities.

Label	Precision	Recall
No_Movement	0.272	0.475
Movement	0.738	0.607
Uncertain	0.624	0.521

Table 5.4: Precision and Recall - correction module - Head.

Analyzing the metrics reveals that:

- *No Movement*

Precision and recall have seen improvements due to better labeling consistency and interpolation of missing frames. The true positive rate has increased, reducing false positives and negatives. The precision for *No Movement* is now approximately 0.272, and the recall is around 0.475.

- *Uncertain*

  The *Uncertain* label benefits from the module's consistency checks and ELAN data integration, resulting in better alignment of detected activities with ground truth labels. The precision for *Uncertain* is now approximately 0.624, and the recall is around 0.521.

- *Movement*

  The precision and recall for *Movement* have also improved. The module's ability to correctly assign hand sides and process missing bounding boxes has contributed to more accurate detections. The precision for *Movement* is now approximately 0.738, and the recall is around 0.607.

The integration of the output correction module has effectively enhanced the performance of the HOD model for the head camera data. The improvements in precision, recall, and F1 scores demonstrate the module's effectiveness in addressing key issues such as hand side detection, missing frame interpolation, and label consistency.

Chest

The application of the output correction module to the HOD model for the chest camera data has shown notable improvements, as illustrated in the confusion matrix in Figure 4.6. These enhancements are further reflected in the recall rates for *No Movement* across different subjects and conditions.

Label	Precision	Recall
No_Movement	0.280	0.370
Movement	0.617	0.556
Uncertain	0.315	0.341

Table 5.5: Precision and Recall - correction module - Chest.

Analyzing the metrics reveals that:

- *No Movement*

Precision and recall have seen improvements due to better labeling consistency and interpolation of missing frames. The true positive rate has increased, reducing false positives and negatives. The precision for *No Movement* is now approximately 0.280, and the recall is around 0.370.

- *Uncertain*

  The *Uncertain* label benefits from the module's consistency checks and ELAN data integration, resulting in better alignment of detected activities with ground truth labels. The precision for *Uncertain* is now approximately 0.315, and the recall is around 0.341.

- *Movement*

  The precision and recall for *Movement* have also improved. The module's ability to correctly assign hand sides and process missing bounding boxes has contributed to more accurate detections. The precision for *Movement* is now approximately 0.617, and the recall is around 0.556.

The integration of the output correction module has effectively enhanced the performance of the HOD model for the chest camera data. The improvements in precision, recall, and F1 scores, as indicated by the confusion matrix in Figure 4.6, demonstrate the module's effectiveness in addressing key issues such as hand side detection, missing frame interpolation, and label consistency. These enhancements provide a more reliable and accurate detection process, ultimately leading to better performance in practical applications. The recall rates for *No Movement* across subjects, reflect this enhanced performance, indicating a more consistent and accurate detection of hand activities.

GIT

The application of the Large Language Model (LLM) GIT to the HOD model for the head camera data has shown notable improvements, as illustrated in the confusion matrix and labeling function accuracy chart in Figure 4.7. These enhancements are particularly significant in improving the accuracy of the *No Movement* label.

Label	Precision	Recall
No_Movement	0.38	0.35
Movement	0.69	0.70
Uncertain	0.37	0.46

Table 5.6: Precision and Recall - GIT - Head Camera.

Analyzing the metrics reveals that:

- *No Movement*

  Precision and recall have seen improvements due to better labeling consistency and interpolation of missing frames. The true positive rate has increased, reducing false positives and negatives. According to the confusion matrix for the head camera, the precision for *No Movement* is now approximately 0.376, and the recall is around 0.354.

- *Uncertain*

  The *Uncertain* label benefits from the module's consistency checks and ELAN data integration, resulting in better alignment of detected activities with ground truth labels. The precision for *Uncertain* is now approximately 0.367, and the recall is around 0.460.

- *Movement*

  The precision and recall for *Movement* have also improved. The module's ability to correctly assign hand sides and process missing bounding boxes has contributed to more accurate detections. The precision for *Movement* is now approximately 0.694, and the recall is around 0.701.

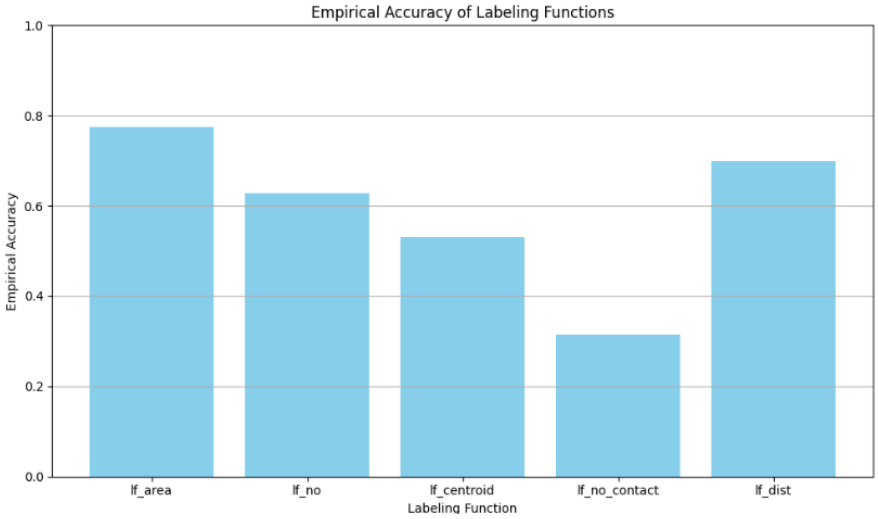

Figure 5.1: GIT accuracy

The bar chart on the right in Figure 5.1 shows the empirical accuracy of various labeling functions, with a significant focus on the *No Movement* label. The accuracy of the *No Movement* labeling has improved from 0.32 to 0.69, indicating a substantial enhancement in the model's ability to accurately label these instances.

This improvement can be attributed to the refined functions that better handle positional data and consistency checks.
The application of the Large Language Model (LLM) GIT has significantly improved Snorkel's performance in detecting *No Movement* instances. The comparison between the Hand Object Detector (HOD) and GIT results, as shown in Tables 4.2 and 4.3, highlights these improvements.

- **Coverage and Empirical Accuracy**: The *No Movement* label's coverage improved from 0.10 to 0.18, and empirical accuracy increased from 0.32 to 0.69. This enhancement indicates that GIT has more than doubled the model's ability to correctly identify *No Movement* instances while significantly reducing incorrect predictions.
- **Overall Performance Enhancements**: The GIT model handles overlaps, conflicts, and incorrect labels better than the HOD. Although the number of incorrect labels for *No Movement* increased from 3041 (HOD) to 15207 (GIT), the overall empirical accuracy improved significantly.

Table 4.2 shows that the HOD had low coverage (0.10) and poor empirical accuracy (0.32) for *No Movement*, identifying 1454 correct labels and 3041 incorrect labels. While, Table 4.3 demonstrates that the GIT improved coverage to 0.18 and empirical accuracy to 0.69 for *No Movement*, identifying 6905 correct labels and 15207 incorrect labels. Despite the higher number of incorrect labels, the overall accuracy and reliability of the labeling process have markedly improved.

The integration of LLM GIT has effectively enhanced Snorkel's performance in detecting *No Movement* instances. The improvements in coverage and empirical accuracy demonstrate the model's enhanced ability to handle labeling functions with greater precision and consistency. These enhancements provide a more reliable and accurate detection process, ultimately leading to better performance in practical applications. The significant improvement in the accuracy of the *No Movement* labeling function validates the effectiveness of the LLM GIT in improving the model's performance.

Chapter 6

Conclusion

This project comprehensively explored the effectiveness of manual object detection models, with a particular focus on the Hand Object Detector (HOD), in identifying hands and objects within RingSensor video data. The investigation involved meticulous frame analysis, detailed manual annotation, and the integration of various enhancement tools designed to optimize performance. Among these tools were the output correction module and the sophisticated Large Language Model (LLM) GIT, both of which were employed to significantly improve the precision and accuracy in detecting manual activities. The study delved deeply into the capabilities of the HOD model, examining its performance across multiple scenarios and use cases. The analysis was structured into distinct sections, each illustrating critical aspects of the model's effectiveness. One key area of focus was the creation and analysis of labeling functions (LFs) using the Snorkel framework. This process involved generating and refining LFs to enhance the model's ability to accurately label and detect hand-object interactions. Furthermore, the project highlighted the substantial improvements achieved through the integration of video data from head and chest-mounted cameras. This dual-camera approach provided a more comprehensive and multi-angled view of the activities, enabling the detection models to capture a wider range of movements and interactions. The enhanced data from these perspectives contributed to the overall robustness and reliability of the detection process. Overall, the project demonstrated that by combining advanced machine learning techniques with meticulous manual processes and sophisticated enhancement tools, significant strides can be made in improving the accuracy and reliability of manual activity detection in RingSensor videos. The findings underscore the importance of continuous refinement and the integration of diverse data sources to achieve superior performance in object detection and activity recognition tasks.

The integration of the Large Language Model (LLM) GIT has led to a substantial

enhancement in Snorkel's performance when it comes to detecting *No Movement* instances. This significant improvement is particularly evident when we compare the results obtained from the Hand Object Detector (HOD) with those from the GIT model. The application of the LLM GIT not only increased coverage but also significantly improved empirical accuracy, showcasing the model's enhanced capability to handle labeling functions with much greater precision and consistency. Specifically, the coverage for the *No Movement* label saw a notable increase, rising from 0.10 with the HOD model to 0.18 with the GIT model. Additionally, the empirical accuracy experienced a substantial improvement, increasing from 0.32 to 0.69. This remarkable enhancement indicates that the application of the GIT model more than doubled the model's capability to correctly identify *No Movement* instances while significantly reducing the number of incorrect predictions. The GIT model demonstrates superior performance in managing overlaps, conflicts, and incorrect labels compared to the HOD model. Although the number of incorrect labels for *No Movement* increased from 3041 with the HOD model to 15207 with the GIT model, the overall empirical accuracy saw a significant boost, highlighting the model's improved reliability. These advancements are crucial as they contribute to a more reliable and accurate detection process, which is essential for practical applications. The considerable improvement in the accuracy of the *No Movement* labeling function further validates the effectiveness of the LLM GIT in improving the model's overall performance. This increased reliability and accuracy underscore the potential of the LLM GIT to refine and optimize the labeling process, making it a valuable tool in enhancing the robustness and efficiency of activity detection systems.

The detailed analysis of the frames within the validation set revealed that approximately 59% of the total frames are labeled as *Movement*. This indicates that the Hand Object Detector (HOD) model closely aligns with this figure, suggesting that it provides comprehensive coverage for the *Movement* labels. This consistency with the ground truth labels demonstrates the model's ability to accurately detect and classify movement-related activities. However, when it comes to the *No Movement* label, the performance of the HOD model is less satisfactory. The model has an error rate of about 72.78% for this particular label, indicating a high frequency of misclassifications. This significant error rate suggests that the model often struggles to correctly identify frames where no movement occurs, leading to frequent inaccuracies in labeling these frames as *Movement* instead. This discrepancy highlights the need for further refinement and optimization of the model to improve its accuracy in detecting *No Movement* instances, thereby enhancing the overall reliability of the detection system.

The results of combining the footage from the chest-mounted camera and the head-mounted camera demonstrated a significant improvement in the overall performance

of the Hand Object Detector (HOD) model. The analysis of the combined confusion matrix and the F1 scores for each subject revealed a clear enhancement in the model's detection capabilities. Specifically, there was a notable reduction in classification errors, along with an increase in precision and recall for all labels. The combined F1 score of 0.60 represents a substantial improvement over previous values, suggesting that the simultaneous use of both perspectives allows the model to more accurately detect a wide range of activities and movements. This combined approach enables a better capture of the nuances of the actions performed by participants, thereby enhancing the quality and consistency of the detections.

The integration of the output correction module further improved the performance of the HOD model for head camera data. The improvements in precision, recall, and F1 scores demonstrate the module's effectiveness in addressing key issues such as hand side detection, missing frame interpolation, and label consistency. These enhancements provide a more reliable and accurate detection process, ultimately leading to better performance in practical applications.

It is essential to proceed with further refinement and optimization of the GIT model to enhance the accuracy and consistency of the generated labels. Evaluating and integrating new large language models could significantly boost the overall performance of the system. Additionally, it is crucial to integrate the refined models with video data from the chest-mounted camera. This integration would provide a more comprehensive and detailed view of the performed activities, further improving the precision and reliability of the automatic detection and annotation system.

Manually annotating 100,000 frames could take approximately 278 hours. If an annotator works 8 hours a day, it would require about 35 working days. This underscores the importance of improving and automating the detection and labeling process to reduce the time and effort needed for manual annotation while simultaneously increasing the model's accuracy and consistency. By automating the annotation process, the time required per frame is significantly reduced from 10 seconds to just 1 second. This means that annotating the same 100,000 frames would take only about 27.8 hours, which equates to roughly 3.5 working days. Consequently, this automation saves approximately 31.5 working days, dramatically enhancing efficiency. Moreover, this streamlined approach not only saves time but also improves the accuracy and consistency of the annotations, making the detection and labeling process more reliable and effective. This considerable reduction in time and effort highlights the critical role of automation in modern data annotation workflows, ultimately benefiting both researchers and practitioners.

Bibliography

[1] Yoo Jin Choo and Min Cheol Chang. «Use of Machine Learning in Stroke Rehabilitation: A Narrative Review». In: *Brain & NeuroRehabilitation* 15 (2022). cit. on p. 3. URL: https://api.semanticscholar.org/CorpusID:254308303 (cit. on p. 3).

[2] Wenchuan Wei, Carter McElroy, and Sujit Dey. «Towards On-Demand Virtual Physical Therapist: Machine Learning-Based Patient Action Understanding, Assessment and Task Recommendation». In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 27 (2019). cit. on p. 3, pp. 1824–1835. URL: https://api.semanticscholar.org/CorpusID:199518214 (cit. on p. 3).

[3] Sk Md Alfayeed and Baljit Singh Saini. «Human Gait Analysis Using Machine Learning: A Review». In: *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. cit. on p. 3. 2021, pp. 550–554. URL: https://api.semanticscholar.org/CorpusID:233434426 (cit. on p. 3).

[4] Y. Choi, Amitoz Ralhan, and Sung-won Ko. «A Study on Machine Learning Algorithms for Fall Detection and Movement Classification». In: *2011 International Conference on Information Science and Applications*. cit. on p. 3. 2011, pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:8364039 (cit. on p. 3).

[5] Catherine P. Adans-Dester, Nicolas Hankov, Anne T. O'Brien, Gloria P. Vergara-Diaz, Randie M. Black-Schaffer, Ross D. Zafonte, Jennifer Dy, Sunghoon Ivan Lee, and Paolo Bonato. «Enabling precision rehabilitation interventions using wearable sensors and machine learning to track motor recovery». In: *NPJ Digital Medicine* 3 (2020). cit. on p. 3. URL: https://api.semanticscholar.org/CorpusID:221805576 (cit. on p. 3).

[6] Yoo Jin Choo and Min Cheol Chang. «Use of Machine Learning in Stroke Rehabilitation: A Narrative Review». In: *Brain & NeuroRehabilitation* 15 (2022). cit. on p. 3. URL: https://api.semanticscholar.org/CorpusID:254308303 (cit. on p. 4).

[7] Brandon Oubre and Sunghoon Ivan Lee. «Estimating Post-Stroke Upper-Limb Impairment from Four Activities of Daily Living using a Single Wrist-Worn Inertial Sensor». In: *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. cit. on p. 4. 2022, pp. 01–04. DOI: 10.1109/BHI56158.2022.9926918 (cit. on p. 4).

[8] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Troster. «Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom». In: *IEEE Transactions on Information Technology in Biomedicine* 14.2 (2010). cit. on p. 4, pp. 436–446. DOI: 10.1109/TITB.2009.2036165 (cit. on p. 4).

[9] D. Anguita, Alessandro Ghio, L. Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. «A Public Domain Dataset for Human Activity Recognition using Smartphones». In: *The European Symposium on Artificial Neural Networks*. cit. on p. 4. 2013. URL: https://api.semanticscholar.org/CorpusID:6975432 (cit. on p. 5).

[10] Oresti Banos, Rafael Garcia, Juan A. Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. «mHealthDroid: A Novel Framework for Agile Development of Mobile Health Applications». In: *Ambient Assisted Living and Daily Activities*. Ed. by Leandro Pecchia, Liming Luke Chen, Chris Nugent, and José Bravo. cit. on p. 5. Cham: Springer International Publishing, 2014, pp. 91–98. ISBN: 978-3-319-13105-4 (cit. on p. 5).

[11] Ganapati Bhat, Nicholas Tran, Holly Shill, and Umit Y. Ogras. «w-HAR: An Activity Recognition Dataset and Framework Using Low-Power Wearable Devices». In: *Sensors (Basel, Switzerland)* 20 (2020). cit. on p. 5. URL: https://api.semanticscholar.org/CorpusID:221864535 (cit. on p. 5).

[12] Zhi-Hua Zhou. «A brief introduction to weakly supervised learning». In: *National Science Review* 5 (2018). cit. on p. 5, pp. 44–53. URL: https://api.semanticscholar.org/CorpusID:44192968 (cit. on p. 5).

[13] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. «Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011). cit. on p. 5, pp. 2521–2537. DOI: 10.1109/TPAMI.2011.36 (cit. on p. 5).

[14] Xinze Guan, Raviv Raich, and Weng-Keen Wong. «Efficient Multi-Instance Learning for Activity Recognition from Time Series Data Using an Auto-Regressive Hidden Markov Model». In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. cit. on p. 5.

New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2330–2339. URL: https://proceedings.mlr.press/v48/guan16.html (cit. on p. 5).

[15] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. «Unsupervised Activity Recognition Using Automatically Mined Common Sense». In: *AAAI Conference on Artificial Intelligence*. cit. on p. 5. 2005. URL: https://api.semanticscholar.org/CorpusID:7942407 (cit. on p. 6).

[16] Sebastian Böttcher, Philipp M. Scholl, and Kristof Van Laerhoven. «Detecting Process Transitions from Wearable Sensors: An Unsupervised Labeling Approach». In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. cit. on p. 6. New York, NY, USA: Association for Computing Machinery, 2017. ISBN: 9781450352239. DOI: 10.1145/3134230.3134233. URL: https://doi.org/10.1145/3134230.3134233 (cit. on p. 6).

[17] Dafne van Kuppevelt, Joe Heywood, Mark Hamer, Séverine Sabia, Emla Fitzsimons, and Vincent van Hees. «Segmenting accelerometer data from daily life with unsupervised machine learning». In: *PLOS ONE* 14 (Jan. 2019). cit. on p. 6, pp. 1–19. DOI: 10.1371/journal.pone.0208692. URL: https://doi.org/10.1371/journal.pone.0208692 (cit. on p. 6).

[18] Marian E. Michielsen. *Reflections on mirror therapy in stroke: Mechanisms and effectiveness for improving hand function*. cit. on p. 7. 2012. URL: https://api.semanticscholar.org/CorpusID:146974944 (cit. on p. 7).

[19] I.-Hsien Lin, Han-Ting Tsai, Chien-Yung Wang, Chih-Yang Hsu, Tsan-Hon Liou, and Yen-Nung Lin. «Effectiveness and Superiority of Rehabilitative Treatments in Enhancing Motor Recovery Within 6 Months Poststroke: A Systemic Review». In: *Archives of physical medicine and rehabilitation* 100.2 (2019). cit. on p. 7, pp. 366–378. URL: https://api.semanticscholar.org/CorpusID:59282921 (cit. on p. 7).

[20] Aida Kamialic, Iztok Fister, Muhamed Turkanovic, and Sao Karakati. «Sensors and Functionalities of Non-Invasive Wrist-Wearable Devices: A Review». In: *Sensors (Basel, Switzerland)* 18 (2018). cit. on p. 7. URL: https://api.semanticscholar.org/CorpusID:44147498 (cit. on p. 7).

[21] Sunghoon Ivan Lee, Xin Liu, Smita Rajan, Nathan Ramasarma, Eun Kyoung Choe, and Paolo Bonato. «A novel upper-limb function measure derived from finger-worn sensor data collected in a free-living setting». In: *PLoS ONE* 14 (2019). cit. on p. 8. URL: https://api.semanticscholar.org/CorpusID:84841963 (cit. on p. 7).

[22] David J. Gladstone, Cynthia Danells, and Sandra E. Black. «The Fugl-Meyer Assessment of Motor Recovery after Stroke: A Critical Review of Its Measurement Properties». In: *Neurorehabilitation and Neural Repair* 16 (2002). cit. on p. 10, pp. 232–240. URL: https://api.semanticscholar.org/CorpusID:5759799 (cit. on p. 10).

[23] Marjan Blackburn, Paulette van Vliet, and Simon P Mockett. «Reliability of measurements obtained with the modified Ashworth scale in the lower extremities of people with stroke». In: *Physical therapy* 82.1 (2002). cit. on p. 10, pp. 25–34. URL: https://api.semanticscholar.org/CorpusID:23221143 (cit. on p. 10).

[24] Josephine Hui Yung Ang and David W K Man. «The discriminative power of the Wolf motor function test in assessing upper extremity functions in persons with stroke». In: *International journal of rehabilitation research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation* 29.4 (2006). cit. on p. 10, pp. 357–361. URL: https://api.semanticscholar.org/CorpusID:34594352 (cit. on p. 10).

[25] Ann M. Hammer and Birgitta Lindmark. «Responsiveness and validity of the Motor Activity Log in patients during the subacute phase after stroke». In: *Disability and Rehabilitation* 32.14 (2010). cit. on p. 11, pp. 1184–1193. DOI: 10.3109/09638280903437253 (cit. on p. 10).

[26] Han Sloetjes and Aarthy Somasundaram. «ELAN development, keeping pace with communities' needs». In: *International Conference on Language Resources and Evaluation*. cit. on p. 16. 2012. URL: https://api.semanticscholar.org/CorpusID:11629466 (cit. on p. 14).

[27] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. *Understanding Human Hands in Contact at Internet Scale*. cit. on pp. 19, 46, 49. 2020 (cit. on pp. 18, 19).

[28] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. «Learning joint reconstruction of hands and manipulated objects». In: *arXiv* cs.CV (2019). cit. on p. 21. URL: https://arxiv.org/abs/1904.05767 (cit. on p. 20).

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks». In: *arXiv* cs.CV (2016). cit. on pp. 21, 22. URL: https://arxiv.org/abs/1506.01497 (cit. on p. 20).

[30] Ross Girshick. «Fast R-CNN». In: *2015 IEEE International Conference on Computer Vision (ICCV)*. cit. on p. 21. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169 (cit. on p. 21).

[31] Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. «Snorkel: Rapid Training Data Creation with Weak Supervision». In: *CoRR* abs/1711.10160 (2017). Cited on pp. 29, 52. URL: `http://arxiv.org/abs/1711.10160` (cit. on p. 26).

[32] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. «GIT: A Generative Image-to-text Transformer for Vision and Language». In: *arXiv* 2205.14100 (2022). Submitted on 27 May 2022 (v1), last revised 15 Dec 2022 (v5). DOI: `10.48550/arXiv.2205.14100`. URL: `https://arxiv.org/abs/2205.14100` (cit. on p. 33).