POLITECNICO DI TORINO

Master's Degree in Electronic Engineering

Master's Degree Thesis

Development and evaluation of a Machine Learning pipeline for the generation of video annotations

Supervisors

Prof. Danilo DEMARCHI

Prof. Paolo BONATO

Dr. Giulia CORNIANI

Candidate

Luca SPAGNUOLO

OCTOBER 2023

POLITECNICO DI TORINO

Department of Electronics and Telecommunications
Master's Degree in Electronic Engineering

Master's Degree Thesis

Development and evaluation of a Machine Learning pipeline for the generation of video annotations

---

Luca Spagnuolo

Supervisors: Prof. Danilo Demarchi, Prof. Paolo Bonato, Dr. Giulia Corniani

OCTOBER 2023

Abstract

The rapid growth of machine learning techniques has promoted innovative advancements across various domains, including health, medicine, and rehabilitation. However, the effectiveness of these methods heavily relies on the availability of large and well-annotated datasets. In particular, in the realm of medicine, a common challenge lies in the existence of extensive datasets, which often lack comprehensive labeling. This limitation hampers the progress and deployment of automated algorithms across various medical applications.

This thesis addresses the challenge of unlabeled video datasets by proposing a novel approach for generating automatic labels in the context of monitoring the upper limb activity in stroke patients. Leveraging developments in deep learning and computer vision, the proposed framework extracts relevant features from video sequences and inputs them into Snorkel [1] to generate labeled data of hand activity. By utilizing weakly supervised learning techniques, the framework is designed to effectively learn from limited annotated samples and generalize to unlabelled data. The study begins by exploring state-of-the-art machine learning architectures that can learn from scarce data and focus on weakly supervised machine learning along with the generative model used by Snorkel. To evaluate the effectiveness of the proposed approach, a comprehensive dataset of upper limb activity of stroke patient video recordings [2] is analyzed, and quantitative and qualitative assessments are conducted to compare the performance of the automated label generation framework against manual annotations. The metrics include accuracy, Intersection over Union, F1-score, and confusion matrices. Visual comparisons of generated labels and ground truth annotations provide insights into the system's interpretability. Overall, the pipeline achieved an F1-score of 76%. The results of this study offer an effective solution to the issue of limited labeled data in stroke patient video analysis. The proposed framework showcases the potential of harnessing the growth of machine learning in rehabilitation, even when confronted with large unlabeled datasets. Furthermore, the methodologies developed can serve as a blueprint for addressing similar challenges in other medical domains requiring video data analysis with limited labeled samples.

Acknowledgements

First of all, I want to thank prof. Demarchi and prof. Bonato for the opportunity to carry out my Master's thesis at the Spaulding Rehab Hospital in Boston. I not only had a wonderful time but I also met lifelong friends and amazing colleagues. A big thank you goes to my parents that supported me through my journey and my brother that, even with six hours difference, was always there for me . Throughout my staying in Boston, I had the pleasure to meet lots of people who I share amazing and unforgettable memories, from pasta and pizza nights to weekly barbecues and road trips. From wrongly translating Italian saying in English to helping each other out with our thesis's projects. I also want to thank my roommate Luke that made my staying in Boston even more enjoyable. I also want to thank my friends in Italy, that are always there for me since High School. Finally, I want to thank my girlfriend Anna for all the moral support she gave me while I was stressing out to write this thesis.

Table of Contents

List of Tables

List of Figures

Acronyms

3D
Three Dimensional

ADL
Activity of Daily Life

AI
Artificial Intelligence

BSN
Body Sensor Networks

CSV
Comma-Separated Values

DIY
Do It Yourself

ECG
Electrocardiogram

FMA
Fugl-Meyer Assessment

HAR
Human Activity Recognition

HSMM
Hidden Semi-Markov Model

IMU

Inertial Measurement Unit

IoU

Intersection over Union

IRB

Institutional Review Board

LiDAR

Light Detection and Ranging

MGB

Mass General Brigham

MA

Massachusetts

MAL

Motion Analysis Laboratory

MAS

Modified Ashworth Scale

MIL

Multi-instance Learning

MMSE

Mini-Mental State Examination

PI

Principal Investigator

RCNN

Region-Based Convolutional Neural Network

ROI

Region Of Interest

RPN

Region Proposal Network

SRH

Spaulding Rehabilitation Hospital

SME

Subject Matter Expert

TC

Tai Chi

Yolo

You Only Look Once

The Motion Analysis Lab

This thesis was conducted at Harvard Medical School's Motion Analysis Lab at Spaulding Rehabilitation Hospital in Boston, MA. The lab primarily focuses on utilizing robotics and wearable technology to analyze the bio-mechanics of human movement. The ultimate goal of this research is to advance the understanding and treatment of conditions that limit mobility, including cerebral palsy, stroke, and Parkinson's Disease.

During my time at the lab, I had the opportunity to explore various ongoing projects. This experience allowed me to delve into different aspects of research and gain a comprehensive perspective on how technology and biomechanics intersect to address mobility challenges.

Recovery-on-Track

The aim of this project is 3D reconstruction of rehabilitation exercises using human pose estimation and lidar data to build a stroke tele-rehabilitation system. My contribution was to explore different depth cameras available on the market to enable the tracking of the body movements in 3D. I tested different depth cameras such as, OAKD-pro W, Intel RealSense D455, Microsoft Azure Kinect and iPad LiDAR technology, compared the quality of the depth images and helped in designing the optimal setup for the data collection.

Tai Chi

The TaiChi projects focuses on testing the delivery of a novel Tele-Tai Chi (TC) intervention in a single-arm feasibility study for community-dwelling TC-naive older adult and invstigate meaningful changes in areas like physical activity, (self-efficacy), overall well-being, balance, walking abilities (gait), and assess improvements of TC proficiency. I was involved in the ongoing data collection.

Smartwatch

For the SmartWatch project I was involved in validating the app introduced in [3], along with a new app based on the Timed Up and Go task , using the Vicon Motion System.

I was involved in writing the script to synchronize Vicon and watch data and in analyzing data collected on physical therapist while simulating chronic stroke patients. I also submitted a 1 page abstract for the BSN conference on this project and got accepted.

Posture Check phase II

Posture Check is a project on detecting compensatory movements of stroke survivors while using a robotic device for arm rehabilitation and a camera system that can provide feedback about those undesired movement while doing therapy.

For phase II, they wanted to test the possibility of introducing the third dimension while gathering data. At first, my work was to test different Human Pose Estimation algorithms such as, MoveNet, MediaPipe, OpenPose and LightBuzz, on the 2D dataset.

Then, I tested two different hardware to enable 3D pose estimation. Inter Realsense, based on active stereo technology, and Microsoft Azure Kinect, based on LiDAR technology.

Finally, I was involved in exploring the possibility to use OpenCap [4] to have 3D pose estimation along with a biomechanic model developed in OpenSim.

Introduction

The fast growth of machine learning methods has driven significant advancements in various areas. Machine learning is progressively integrating into our daily lives, enhancing the ease, effectiveness, and personalization of our interactions and experiences. In the field of rehabilitative medicine, machine learning offers the potential to tailor treatments, enhance patient outcomes, and provide more efficient and effective care [5]. Some examples are:

- Physical Therapy Assistance [6]: Machine learning algorithms analyze patients' movement patterns during physical therapy sessions, providing real-time feedback to both patients and therapists. This aids in ensuring correct exercises are performed and tracking progress.

- Gait Analysis [7]: Machine learning is employed to analyze the gait of individuals recovering from injuries or surgeries. Sensors and cameras capture movement data, which is then processed to assess changes in gait and provide insights for rehabilitation plans.

- Fall Prevention [8]: Machine learning algorithms analyze movement data to predict the likelihood of falls in elderly patients, allowing caregivers to implement preventive measures.

- Recovery Monitoring [9]: Wearable sensors and devices equipped with machine learning continuously monitor patients' movements and vital signs, aiding healthcare providers in tracking progress and adjusting rehabilitation plans accordingly.

The effectiveness of these methods heavily relies on the availability of large and well-annotated datasets. A labeled dataset serves as the backbone of machine learning applications in rehabilitation. It empowers models to learn, predict, and guide rehabilitation processes with a level of precision and personalization that can greatly enhance patient outcomes and the overall quality of care [5]. This

personalized approach enables rehabilitation programs to be tailored to individual needs, optimizing the likelihood of positive outcomes.

Nevertheless, it's important to recognize that the labeling process significantly affects the time required for the deployment of these models. The time-consuming task of annotating data directly impacts the efficiency of the entire pipeline, and any improvements made to expedite the labeling process can have a cascading effect, accelerating the deployment of machine learning models.

This thesis presents a comprehensive pipeline designed to significantly expedite the video labeling process. By combining deep learning strategies and weakly supervised machine learning, the framework can correctly recognize hand activity in video and automatically label frame by frame. These generated labels will subsequently be employed for annotating data gathered from wearable sensors, specifically wrist and ring sensors [2], accelerating the deployment of algorithms for sensor data analysis.

In the following of Chapter 1, previous works are reviewed to give a comprehensive outlook on the current approaches to automatically label sensor data. Then, the RingSensor study is introduced and described as the data collected in that study are used to train and test the framework proposed in this manuscript for the self-labeling of video recordings and, consequently, of sensor data.
Chapter 2 will focus on an in-depth description of the methods and materials used, including deep learning models and weakly supervised framework, along with the metrics selected to analyze the results.
Then, results are analyzed and discussed in chapters 3 and 4. Finally, a summary of the pipeline is provided in Chapter 5, along with a discussion on limitations and possible future work.

1.1 Related work

In previous studies on Human Activity Recognition utilizing wearable sensors, manual techniques like video recordings and direct observations were widely employed to gather annotations.

For example, *Plotnik et al.*[10] designed a wearable assistant for Parkinson's disease patients with freezing of gait symptoms. In this research, two annotators were assigned to conduct on-site annotations. One used a digital video camera to record subjects' activities, while the other assigned real-time labels to the acceleration data transmitted from a wearable device on a laptop. Finally, a physiotherapist pinpointed the endpoints of freezing of gait events in the collected data through post-analysis of video recordings.

Similarly, *Anguita et al.*[11] employed a smartphone (Samsung Galaxy S II) to collect data about human movements and identify different human activities

using ambient information. They gathered acceleration and angular velocity data covering daily activities like standing, sitting, lying, walking, and going downstairs and upstairs. Each subject performed two rounds of every activity set, with 5 seconds of rest in between. Following the data collection, manual labeling was carried out based on the video footage of the subjects' activities.

Banos et al.[12] attached two IMUs to the subject's right wrist and left ankle, along with an additional sensor on the chest for two-lead ECG measurements. This configuration allowed the collection of acceleration, angular velocity, geomagnetic data, and ECG signals from 12 distinct outdoor human activities among 10 volunteers. The entire data acquisition procedure was documented via video recording and subsequently manually annotated.

Furthermore, numerous other publicly available HAR datasets [13] centered on wearable sensors or portable devices are accessible online. These datasets mainly include acceleration, angular velocity, and geomagnetic signals. Detailed annotation with high accuracy can be obtained by manual annotation methods with intensive labeling efforts.

In the realm of long-term human activity monitoring, acquiring a comprehensively labeled dataset for supervised algorithms poses a significant challenge. Given this difficulty, an increasing number of researchers are gravitating towards weakly supervised methodologies, aiming to mitigate the laborious task of manual labeling [14].

By integrating experience sampling, multi-instance learning (MIL) gains knowledge from a rather weakly labeled dataset. Here, labels are connected to sets, known as bags, of instances, as opposed to individual instances. This approach allows sensor data to be labeled at a coarser level, significantly reducing the burden of annotation. A bag is considered positive if it contains at least one positive instance and negative if all instances within the bag are negative.

The pioneering work in applying MIL to time series data for Human Activity Recognition was described in [15]. This extensive study and comparative assessment demonstrated the effectiveness of MIL-based methods in significantly reducing the effort required for annotation. Expanding upon the work outlined in [15], Guan et al. [16] introduced a novel MIL model for offline activity recognition using multivariate time series data. This model employs a generative graphical approach based on an Auto-Regressive Hidden Markov Model HMM, enabling the prediction of both bag and instance labels.

Unsupervised learning methods are typically employed to uncover hidden patterns in activity data without the need for predefined labels. For instance, Wyatt et al. [17] treated activity data as a series of natural language terms, essentially sequences of object usage. They utilized generic models derived from everyday activities found on the internet, which served as a form of common knowledge in Human Activity Recognition.

Another example is the approach proposed by Bottcher et al. [18], which introduced an unsupervised framework that employs clustering algorithms to identify transitions between various stages of manual work that follow a semi-predefined procedure. This framework, even when the order and number of steps are known in advance, eliminates the requirement for labeled data.

Similarly, van Kuppevelt et al. [19] delve into the segmentation of accelerometer data from daily activities using unsupervised machine learning techniques. A Hidden Semi-Markov Model (HSMM), configured to identify a maximum of ten behavioral states from five-second averaged acceleration with and without the addition of x, y, and z-angles, was used to segment and cluster the data. The analysis of the data revealed the patterns of daily physical activity and sedentary behavior in individuals. By harnessing unsupervised techniques, the need for manually labeling activity data is unnecessary.

The methods mentioned earlier tackle the challenge of data annotation by primarily employing two types of annotation techniques, which involve either extensive manual labeling or learning-based approaches. As research transitions from controlled laboratory settings to real-world environments, acquiring detailed labeled data becomes more challenging.

Consequently, the fundamental concept underlying techniques like Multi-Instance Learning (MIL) and unsupervised learning is to utilize a limited amount of labeled data along with existing knowledge about the target activities [20]. This approach aims to train a learning model capable of accurately classifying Activities of Daily Life. However, it's important to note that even within these advanced methods, there remains a certain degree of necessity for manual labeling of the initial data. This initial labeling effort serves as a foundation upon which these sophisticated techniques are built to enhance the efficiency and accuracy of activity recognition.

Thus, this thesis focuses on a novel approach to annotating hand activity through video data collected in laboratory settings. The recordings are provided by the RingSensor study, whose end goal is to monitor the use of the affected upper limb in stroke survivors through the use of finger and wrist-worn sensors. To do this, videos were recorded to enable the manual labeling of sensor data.
To speed up the annotations process, the proposed framework uses deep learning to identify hands and objects in frames and weakly supervised learning to generate labels.

1.2 RingSensor Study

1.2.1 Aims

The specific aims of this study are:

- **Aim 1:** To lay the groundwork for the development of a machine learning-based algorithm (Aim 2). Preliminary data will be collected using finger-worn sensors while participants (Up to 20 stroke survivors) perform a variety of activities of daily living.

- **Aim 2:** To validate the suitability of a finger-worn sensor to accurately measure real-world upper-limb performance in stroke survivors (up to 60 subjects) via the development of machine learning-based algorithms, using measures derived from the sensor data. The proposed finger-worn sensor is hypothesized to capture and accurately illustrate measures of real-world upper-limb performance.

- **Aim 3:** To obtain both stroke survivors (users) and clinicians (prescribers) feedback on the functionality and usability of the proposed system.

At present, Aim 1 has been successfully achieved, and the video data has been integrated into the proposed pipeline, while Aim 2 is currently in progress.

1.2.2 Background and significance

Upper-limb paresis ranks as the primary impairment post-stroke, afflicting up to 75% of those who experience a stroke [21]. This paresis considerably hinders an individual's ability to perform a variety of essential everyday tasks. Especially even with rehabilitation, nearly half (49%) of stroke survivors still encounter challenges using their affected limb, even five years after the incident. Given this, a more tailored, systematic approach to individual rehabilitation plans is crucial to ensure the best clinical outcomes. Solid scientific evidence points to the efficacy of rehabilitation interventions in enhancing motor skills [22], stemming primarily from motor learning processes.

The rise of wearable sensors offers a promising approach for objectively tracking motor performance in real-world scenarios. Currently, wrist-worn sensors dominate wearable technology, mainly aiming to quantify the extent of arm usage, like the duration and intensity of daily upper-limb movements[23]. Yet, there's a drawback: these wrist-based sensors predominantly track gross arm movements, like the passive arm swings during walking. This often leads to an overestimated assessment of motor performance.

In contrast, emerging research paints finger-worn sensors in a promising light, highlighting their potential for more precise monitoring of upper-limb activities[24]. Preliminary data from healthy individuals, as shown in 1.1, indicates a strong correlation between acceleration data from finger-worn sensors and their real-world upper-limb activities, both within and beyond lab environments. Arising from these findings, Ring sensor study aims to explore an innovative approach to accurately gauge the upper-limb movements of chronic-stage stroke survivors during daily activities using both finger-worn and wrist-worn sensors. Additionally, the study aims to delve into potential feedback mechanisms that could be integrated based on these sensor technologies.

Figure 1.1: Displays the proposed finger-worn sensor and ring-sensor.

1.2.3 Research design and methods

In the proposed study, up to 60 subjects were recruited. Up to 20 out of the 60 subjects recruited for Aim 2 were asked to participate in Aims 1 and 3. Additionally, 10 clinicians will be recruited for Aim 3.

Notably, this study doesn't include any interventions. Preliminary screening occurs via phone by the study staff, followed by a final assessment at Spaulding Rehabilitation Hospital's Motion Analysis Laboratory (MAL). An in-person preliminary screening is also available upon request.

Stroke Survivors Recruitment Criteria

Inclusion Criteria:

- Stroke survivor (ischemic or hemorrhagic), $>$ 6 months post-stroke at the time of consent

- Residual mild to moderate upper-limb impairments with a score > 35 on the Fugl-Meyer assessment (FMA) without severe range of motion limitations
- Age between 18 and 80

Exclusion Criteria:

- Inability to lift upper-limb against gravity ($>$ 30 degrees of flexion and abduction).
- Severe upper-limb spasticity preventing passive finger movement (MAS > 3).
- Unable to put on/take off sensors independently or with caregiver assistance.
- Cognitive impairments affecting comprehension and instruction following (score < 23 in the MMSE).
- Possessing implantable medical devices not compliant with ISO 14117:2012 or ANSI/AAMI PC69 Bluetooth compatibility standards. Subjects will provide their medical device record card for verification.

Clinician Recruitment Criteria for Aim 3

Inclusion Criteria:

- Clinicians with a minimum of one year of experience in stroke rehabilitation.
- At least 21 years of age.

Study procedure for aim 1

From the designated group of 60, approximately 20 subjects are invited for a preliminary face-to-face meeting before the procedures outlined in Aim 2. This session is hosted at SRH in Charlestown, MA, potentially extending over three hours.

During this session, after obtaining consent and finishing the initial screening process (as detailed in Aim 2, Visit 1), participants are equipped with sensors on both hands, upper limbs, and torso. They then undertake a set of tasks (refer to Table 1) monitored by the research team. The entire session is captured on available recording devices such as GoPros or handheld cameras, facilitating the synchronization of the accelerometer data from the sensors.

The recorded video data are then annotated by the research team, significantly contributing in completing Aim 1.

Study procedure for aim 2

Study activities are scheduled at the SRH MAL, the participant's home, or through virtual sessions. The location for the visit is decided after consulting with the prospective study participant and study PI. For participants preferring virtual sessions who lack the necessary hardware or internet access, an encrypted device is provided throughout the course of the study. The entire study is structured to last a maximum of 2 weeks, incorporating 2 to 3 sessions at SRH. These sessions are separated with a 2 to 7-day period of sensor data gathering at the participant's home.

Figure 1.2: Study procedure pipeline

Visit 1 - Consent and Screening

If participants have already taken part in Aim 1 and wish to continue, they would have consented to the subsequent aims of the study. Once initially screened, candidates meet with a researcher to understand and complete the consent process. Their cognitive functionality and instruction comprehension are assessed using the Mini-Mental State Examination (MMSE). Prospective participants who don't meet this criterion are not included.

For those who clear this stage, their capacity to make decisions and consent is evaluated via the UCSD Brief Assessment of Capacity for Consent Questionnaire. They must comprehend the research nature of the study, differentiating it from treatment, and be aware of the associated risks and advantages. Failing to understand these aspects result in their exclusion. After clearing these evaluations, participants either sign the consent form or give verbal agreement if participating remotely.

Following this, a Fugl-Meyer assessment (FMA) [25] assesses their upper-limb functionality, giving a score between 0 and 66 to determine upper extremity motor impairment. Those scoring below 35 on this scale are not eligible for the study. Post FMA, participants have their muscle tone evaluated via the Modified Ashworth Scale (MAS) [26] or medical records review. Their upper limb functionality is measured by performing tasks corresponding to the Wolf Motor Function Test [27].

They will also self-evaluate using the Motor Activity Log [28], a clinically approved method to assess the usage and movement quality of their affected limb during daily activities. This evaluation is anticipated to take around 2 hours.

- Participants can choose to combine both Visit 1 and Visit 2 on the same day.

Visit 2 - Lab-based Assessments

Sensors will be positioned bilaterally on participants' fingers and wrists, along with one on the torso by the research team. Commercially available silicone rings will be used for the finger-worn sensors, whereas Velcro straps will secure the wrist-worn and torso sensors. This set-up is estimated to be completed within approximately 10 minutes.

Following the sensor set-up, participants start the laboratory phase of the study. This phase can occur in a simulated home setting located at the Spaulding Rehabilitation Hospital or the participant's own house virtual sessions. All tasks are captured on video for offline analysis, employing GoPro cameras and a handheld videorecorder or a webcam for virtual sessions. GoPros, strapped to the participant's trunk and head, offers researchers a first-person view, facilitating the later synchronization of movement with data from the wearable sensors. The handheld recorder serves a similar purpose, offering supplementary angles that might elude the GoPros' field of view. Only those approved by the IRB are allowed to record and access the video data, and participants' consent for recording is secured in advance. Participants are directed to execute a set of tasks, each to be performed three times, to capture within-subject variability of the motor patterns. To avoid participant fatigue, task performance is divided across two sessions.The activities executed are detailed in Table 1.1.

Each in-lab session may last approximately 1.5 hours. Once the first session is completed, subjects are asked to return after 2 to 7 days to complete the second session.

Home Monitoring

Subjects are required to wear sensors on both the wrist and index finger for a period ranging from 2 to 7 days. They are asked to take off the sensors every night for charging using the provided charger. Care must be taken to ensure that the sensors don't come into contact with excess amounts of water; for instance, they should be removed prior to activities like showering or swimming. Nonetheless, hand washing is permissible with the sensors on.

Each participant receives a pre-paid smartphone preinstalled with Google Timeline, a custom application to oversee the condition of the sensors (Sensor Monitoring app), and another for annotating day-to-day activities (Activity Annotation app).

Tasks
Walking (level, incline, decline)
Sit to Stand transitions
Stair Walking
Drinking a glass of water
Opening a door
Apply Makeup or Shave
Buttoning a shirt
Don/ Doff a sweater
Pick-up and place a two-handled basket (e.g. laundry basket)
Put a table cloth on table
Apply toothpaste on toothbrush
Make a sandwich
Wash Hands
File Nails
Unload bag of groceries
Sweep Floor
Remove money from wallet

Table 1.1: List of Tasks

These specially designed apps have undergone a security assessment by the MGB IT department. The functionalities of these apps are outlined below:

- Google Timeline is intended to record movement patterns and types (such as walking or driving). This data aids in filtering out sensor signals collected during passive mobility actions. Although participants are advised to carry the phone, it isn't mandatory for them to take it everywhere they travel.

- The Sensor Monitoring app ensures time-synchronization of the wearable sensors, oversees their operational status and facilitates communication between the patient and researchers for troubleshooting. Furthermore, this app provides daily notifications to participants in the morning and in the evening. These alerts remind subjects to charge the devices, put on/take off the sensors, and record their activities.

- The Activity Annotation app allows participants to note their activities roughly every 90 minutes (with reminders sent about five times daily) or any time they want to add a new activity that involves significant upper limb movements.

Visit 3 - Lab-based Assessments

After completing the home-monitoring phase (spanning 2 to 7 days), participants will be requested to revisit the MAL. Here, they will hand back the sensors and smartphone and take part in the second lab-based evaluation, structured similarly to Visit 2.

Study procedure for aim 3

Interview and Observation (Stroke Survivors)

A selection of study participants (up to 20 individuals) are invited for an interview, either face-to-face or virtually via platforms such as UMass, UMD, or MGB Enterprise Zoom. Additionally, a brief observation of their ADLs performance is conducted. The aim behind these activities is to gain insights into the participants' unique environmental contexts and natural behaviors, including any challenges or obstacles they might face.

The interview are recorded using a voice recorder (Sony) or Zoom (in case of a remote interview).

Interview(Clinicians)

An interview is conducted with up to 10 clinicians, either in-person or through MGB Enterprise Zoom, to gather insights on the acceptance and opinions regarding the use of ring sensors for monitoring patients' in-home activities. Clinical participants is presented with an information sheet that outlines the specifics of this protocol, and verbal consent is secured before initiating the interview.

1.2.4 Video Annotation

We assembled a team of clinicians to determine and propose labels that would be clinically relevant for the video recordings captured during the second aim of the study. Beyond just analyzing hand activity, the team of clinicians introduced a grasp ontology. This was aimed at understanding which types of grasps stroke survivors utilized most frequently. The goal behind this was to provide insight that could guide interventions, encouraging survivors to use their fingers more extensively in grasping activities. The labels used to annotate videos are summarized in the following.

- No Movement
  - Arm
  - Hand
- Movement
  - Ambulatory Movements (arm swing during gait)
  - Non-Ambulatory Movements (goal/task oriented)
    - Unilateral
      - Gross Arm
      - Fine Hand
        1. Full hand grasp
        2. Finger grasp
        3. Lateral pinch
        4. Flag for uncertainty
    - Bilateral (coupled manipulation-single object)
      - Gross Arm
      - Fine Hand
        1. Full hand grasp
        2. Finger grasp
        3. Lateral pinch
        4. Flag for uncertainty
    - Bilateral (uncoupled manipulation-separate objects)
      - Gross Arm
      - Fine Hand
        1. Full hand grasp
        2. Finger grasp
        3. Lateral pinch
        4. Flag for uncertainty

Examples for clarity: Full hand grasp (gross use of hand) includes:

- Hook – fingers flexed around small diameter object (holding bucket, hang from bar above)
- Power – cylindrical object between fingers and thumb (holding bottle)
- Palmer – object diagonally across palm with thumb stabilizing (hammer)
- Spherical – fingers shaped around round object (ball, round lid)

Finger grasp (fine manipulation between fingers) includes:

- Pincer/Precision – object between forefinger and thumb (pick up small object)

In Table 1.2, we provide a summary of the labels applied to the tasks. This offers a comprehensive view of how the annotations correlate with the dataset.

Tasks	Ambulatory	Non-Ambulatory	Unimanual	Bimanual	Fine Hand Movements	Gross Arm Movements
Walking (level, incline, decline)	X			X		X
Sit to Stand transitions	X			X		X
Stair Walking	X	X				X
Drinking a glass of water		X	X			X
Opening a door		X	X			X
Apply Makeup or Shave		X	X		X	
Buttoning a shirt		X		X	X	
Don/ Doff a sweater		X		X		X
Pick-up and place a basket		X		X		X
Put a table cloth on table		X		X		X
Apply toothpaste on toothbrush		X		X	X	
Make a sandwich		X		X	X	
Wash Hands		X		X		X
File Nails		X	X			X
Unload bag of groceries		X		X		X
Sweep Floor		X		X		X
Remove money from wallet		X		X	X	

Table 1.2: Summary of tasks performed

The pipeline outlined in this thesis aims to expedite the grasp annotation process signaling the frames where the hand(s) is in contact with any object. Given that contact annotations are already in place within ELAN, clinicians can streamline their workflow by selecting the corresponding time intervals and just focus on assigning the grasp label. This approach saves considerable time as the clinicians don't need to go through the whole set of acquired frames but can inspect just a subset.

Materials and Methods

2.1 Overview

We developed a two-stage pipeline for the self-labeling of hand activity in egocentric videos, as illustrated in Figure 2.1. Briefly, in the first stage, videos are processed through a deep learning model that identifies and produces rectangular bounding boxes around detected hands and objects they are in contact with. Notably, this framework doesn't require ground truth labels for the training set. However, to characterize the performance of the first stage of the pipeline, a subset of frames in the dataset was manually annotated to create both validation and test sets and compare with the generated bounding boxes.

In the second stage, the data split into training, validation, and test sets were processed using Snorkel. Through the application of labeling functions, the training set was annotated with the following labels:

- **Contact label:** Denoting instances where there's contact between hands and objects.
- **No Contact label:** This label marks moments where there was no contact between hands and objects.
- **Out-Of-Frame label:** This denotes instances where hands were not captured by the GoPros.

After the labeling process, the labeled dataset was loaded into ELAN [29], an annotation software, for visualizing and reviewing the annotation.

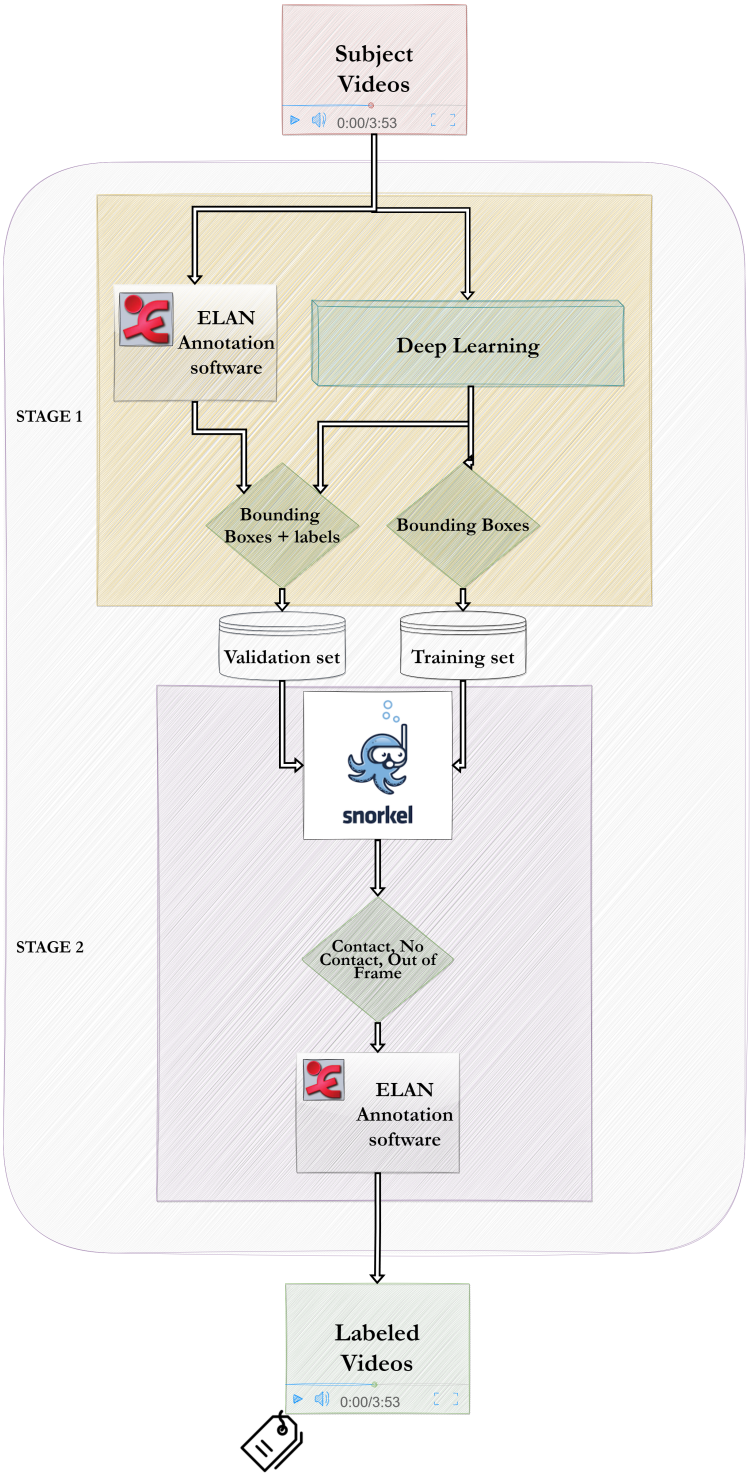

Figure 2.1: High-level block diagram of the two-stages pipeline used for the self-labeling of hand activity in videos.

2.2 RingSensor Study - Video Data

Before going into details of the stages, a brief overview of the dataset is presented. As previously mentioned, the video recordings incorporated in the pipeline were obtained within aim 1 of the RingSensor study, as detailed in section 1.2.3. Overall, the recordings capture the activity of 20 subjects from 4 distinct views within a simulated kitchen setting, as illustrated in Figure 2.2. In particular, the 4 views are

1. **Head Video:** GoPro camera attached to subjects' heads via a head strap.

2. **Chest Video:** GoPro camera attached to subjects' torso using a chest strap.

3. **Room:** Recordings obtained from a standalone GoPro camera positioned on a tripod to provide a general view of the room.

4. **RoomDoor** Similar to the Room view but offers a vantage point from the doorway, using another tripod-mounted GoPro camera.

All the video recordings were synchronized with a remote controller that triggered the start.

For the purpose of developing the self-labeling pipeline presented in this manuscript, only the head videos of two participants (namely participant 001 and participant 003) were considered. To prepare the videos for the deep learning model in stage 1, frames recorded while clinicians were attaching sensors to the subjects were trimmed out.

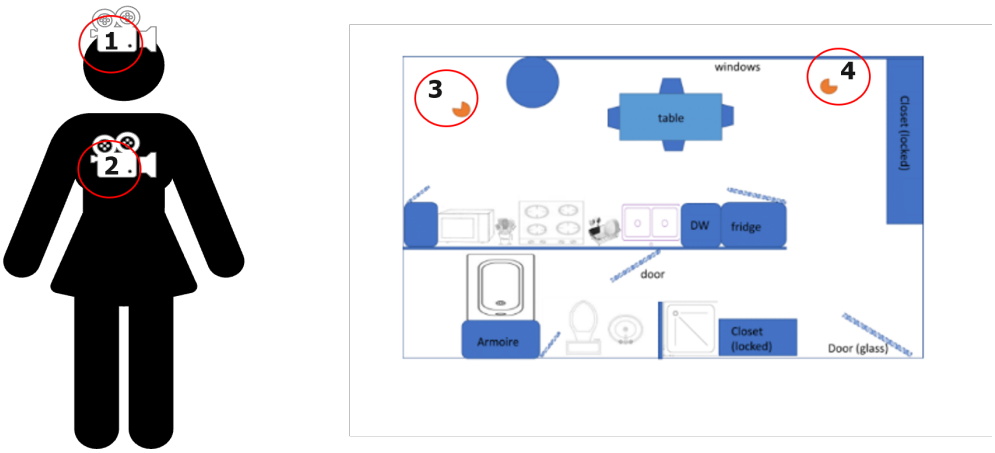

Figure 2.2: Different views of the RingSensor recordings

2.3 Stage 1

After the pre-editing, the head videos of two subjects were used as input for the first stage. A detailed block diagram of the pipeline employed for this initial stage can be found in Figure 2.3.

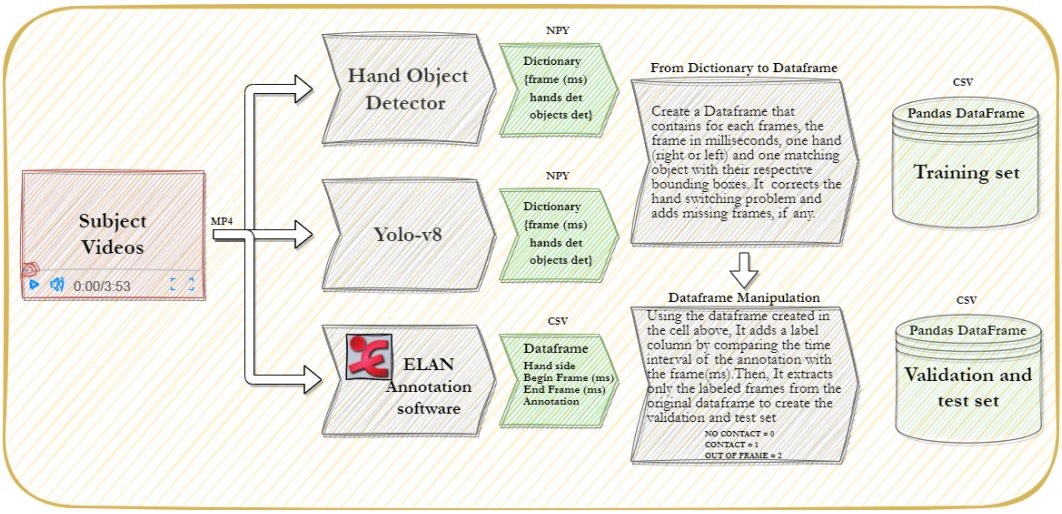

Figure 2.3: Block diagram of the first stage of the self-labeling analysis pipeline

Stage one consists of 2 steps. In the first step, to identify hand activity and determine the most effective model, the same videos are analyzed using two different deep learning models. One is the Hand Object Detector [30], based on a Faster-RCNN, and the other is Yolo-v8 [31]. Both models were trained with the dataset in [30] and are described in details in the following sections. The results produced by these models were formatted as dictionaries. Concurrently, a manual annotation of a subset of the video frames was carried out.

The second step consists of two Python scripts to process the output of step one and make it suitable for the forthcoming stage two. The first script is employed for data manipulation and converts the dictionary format into a pandas dataframe. The second script incorporates the manually annotated labels into the pandas dataframe, thus forming the validation and test sets. More details of the contents of the scripts are provided in section 2.3.3.

2.3.1 Hand Object Detector

The first deep learning model employed is the Hand Object Detector [30], developed to reliably extract hand state information from Internet videos of humans engaged in activities involving their hands.

Dataset

The dataset consists of a diverse set of everyday interactions sourced from YouTube. An overview of the dataset is illustrated in Figure 2.4. It can be divided into two main parts: a vast collection of unlabeled videos employed for unsupervised learning and a subset of 100K annotated frames.

- **Collecting Video Dataset:**Starting with 11 categories, such as DIY and cooking, 13.2K search queries were used to source around 6.5M YouTube videos. The objective was to identify videos showcasing hands interacting with objects. To filter this vast dataset, a model based on video thumbnails was employed. This model identified videos featuring hands and human interactions while omitting those with cartoons.

- **Image Dataset:** The "100 Days of Hands" (100DOH) dataset consists of 27.3K videos from 11 categories, providing 131 days of everyday interaction footage. This data was utilized to produce a 100K frame-level dataset. Random frames were selected, discarding images without hands. There are 189.6K annotated hands in these images, interacting with 110.1K objects. The dataset was divided into training, validation, and testing sets (80/10/10%) based on the YouTube uploader's ID. This ensures no overlap and maintains compatibility with existing data like VLOG.

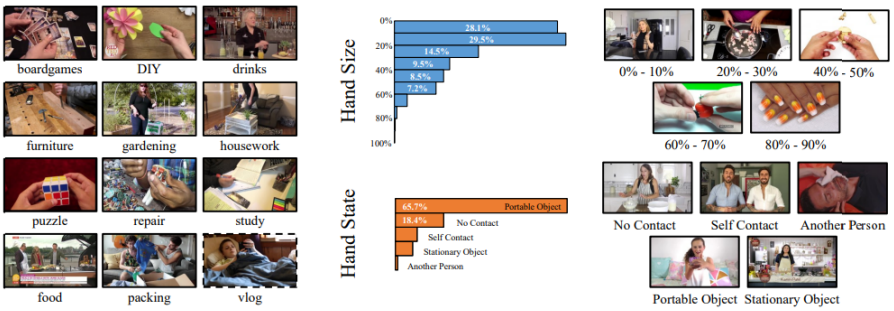

Figure 2.4: Snapshot of the "100DOH" dataset

Pipeline

The pipeline used to build the model is illustrated in Figure 2.5. The system processes an RGB image to detect hands regardless of their size. For each detected hand, the system predicts a bounding box, determines its side as left or right, and identifies its contact state (such as none, self, person, or with a portable/non-portable object). Additionally, it specifies a bounding box for any object the hand

interacts with and establishes a link between the hand and the corresponding object. This output is suitable for direct integration into hand reconstruction tools like [32].

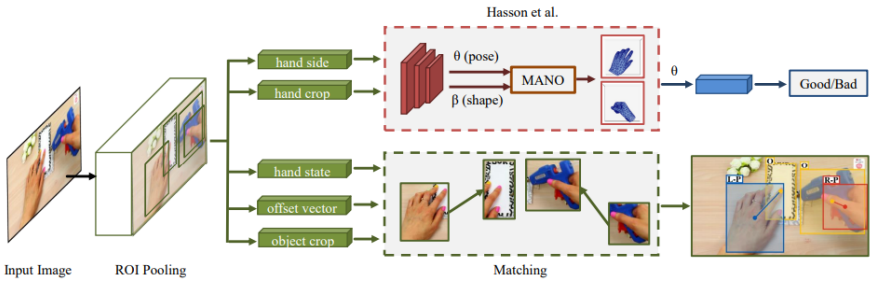

Figure 2.5: Workflow of the Hand Object Detector framework

The backbone of the model is a Faster-RCNN [33]. It's a two-module object detection system (Figure 2.6). The initial module utilizes a deep, fully convolutional network to propose regions. An RPN (Region Proposal Network) accepts an image and output a set of rectangular object proposal, each with a score. This mechanism is modeled utilizing a fully convolutional network. The ultimate aim is to merge computation with a Fast R-CNN object detection network. Following this, the second module, which is the Fast R-CNN detector [34], operates on these proposed regions.

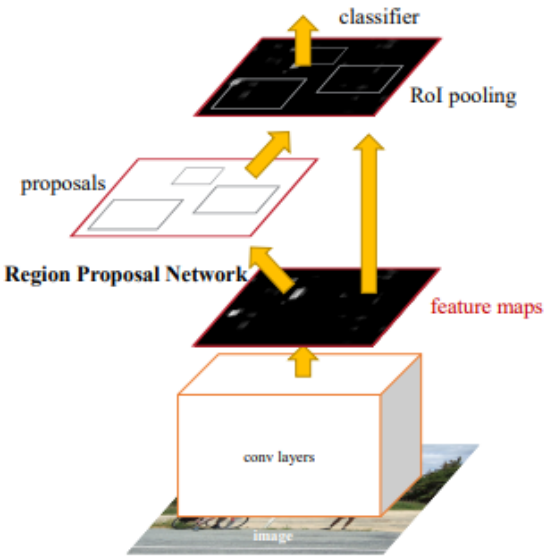

Figure 2.6: Workflow of a Faster-RCNN network

The Hand Object detector is built upon Faster-RCNN [33], which is trained to detect hands and the objects they are in contact with. Like the traditional Faster-RCNN, this network predicts whether each anchor box represents an object, its specific category, and any necessary adjustments to the anchor box's bounding dimensions. Additionally, the system predicts a range of auxiliary outputs sourced directly from the same ROI-pooled features as the standard classification outputs like hand side and contact state.

2.3.2 Yolo-v8

The second deep learning model used was Yolo-v8, the latest version of the Yolo family [31]. Yolo stands for You Only Look Once and is presented as a new approach to object detection. They conceptualized it as a regression challenge, focusing on spatially separated bounding boxes and their correlated class probabilities. A unified neural network produces predictions for bounding boxes and class probabilities straight from complete images in a singular evaluation. As the entire detection process is integrated into one network, it facilitates end-to-end optimization based on detection efficacy. A representation of the Yolo pipeline is illustrated in Figure

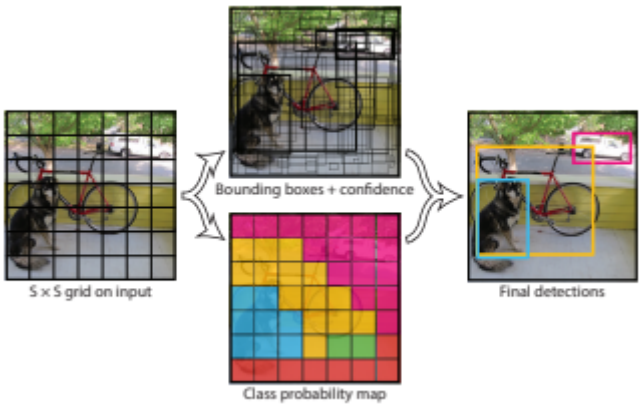

Figure 2.7: Processing step of the yolo model

2.7. The system processes the input image by dividing it into an S × S grid. A grid cell is tasked with detecting an object if the object's center lies within that cell. Each cell is designed to predict B bounding boxes along with their confidence scores. These scores denote the model's confidence in the box containing an object and the predicted box's accuracy. If a cell doesn't contain an object, its confidence score is zero. If it does, the confidence score is expected to match the intersection over union (IOU) of the predicted box and the actual object's position. Moreover, each grid cell predicts C conditional class probabilities, which depend on the presence

of an object within that cell. By multiplying these conditional class probabilities with the box's confidence scores, the system derives class-specific confidence scores for every box. These final scores capture both the likelihood of a particular class being in the box and how well the predicted box around the object fits.

Yolo-v8 is the latest release of the Yolo family and is easily deployable through UltraAnalytics. After getting the annotation from the author of the hand object detector, three Yolo-v8 models were trained with the same dataset: medium, large, and extra-large models. The metrics used to compare the 4 models are described in Section 2.3.5. To obtain a dictionary structure for the Yolo-v8 models, additional lines of code were used in the original script.

2.3.3 Python scripts

First Script

Given that the outputs from both models were in the form of dictionaries, the initial script was designed to transform this dictionary into a Pandas dataframe. The script iterates through the dictionary, retrieving details like hands, objects, and their respective bounding boxes for each frame. Other parameters, such as confidence scores, were also saved. To ensure that the frame count matched the original video's, frames labeled as "None" were inserted to facilitate synchronization throughout the various stages. Additionally, this process involved determining the side of the hand and pairing it with the appropriate object, achieved by minimizing the distance between the centers of the detected hands and objects within a frame.

After acquiring an initial data structure, the following functions were implemented:

- **correct_switch:** There was a persistent issue with both models where they intermittently confused the left and right hands across frames. To address this, two distinct functions were employed. The first function, named correct_switch, partitions the frame into three distinct regions: left, right, and neutral. Depending on the location of the bounding boxes, the hand's orientation (left or right) was determined and assigned accordingly. The neutral zone was designated for instances where a hand traverses across the frame.

- **check_side:** Once the three zones were delineated and initial hand side assignments were made, the function then assessed the overlap between bounding boxes of the current and previous frame. In instances where the overlap exceeded a certain threshold and there was a change in the assigned zone from the previous frame, it was interpreted as a movement across zones. Consequently, the hand's side was adjusted to either left or right. By implementing

this strategy, any hands initially marked as neutral underwent correction, effectively addressing the hand side misidentification challenge.

- **extract_hand_side:** After ensuring the dataset was accurately organized, this function separated the data into right and left hand. For the purposes of symmetry and evaluating the pipeline's efficacy, only data related to the right hand was utilized.
- **correct_duplicate:** Although confidence thresholds could be established for both models, situations where multiple detections of the same hand (either right or left) occurred. To address this, the function leveraged the previously stored confidence scores to discern and eliminate duplicate hand detections.
- **check_missing:** Lastly, a verification step ensured that the total number of frames matched the original video's frame count.

All the synchronization was performed considering frame timestamps in milliseconds. The only difference between the models and their final pandas structure were the additional features from the hand object detector described in section 2.3.1. The snapshot of the final structure, the training set, is shown in Figure 2.8.

1	subj	object	subj_bbox	object_bbox	label	score	frame_ms	labels
2	Right_hand	None	[999, 411, 1249, 662]	None	3	0.997141242	1131831	-1
3	Right_hand	Object	[1049, 844, 1223, 984]	[1061, 470, 1239, 945]	3	0.995388746	1172872	-1
4	Right_hand	None	[1019, 788, 1232, 924]	None	3	0.588112116	1246011	-1
5	None	None	None	None	None	-1	1325958	-1
6	Right_hand	Object	[1699, 614, 1850, 787]	[947, 769, 1063, 928]	0	0.999849558	1433999	-1
7	Right_hand	None	[1120, 955, 1402, 1060]	None	3	0.997748196	1603569	-1
8	Right_hand	Object	[1056, 862, 1359, 1071]	[957, 725, 1395, 1041]	3	0.99764508	1630929	-1

Figure 2.8: Snapshot of final Pandas Structure after running first script with Hand Object Detector as the model.

Second script

The subsequent script served two purposes: it appended the appropriate labels to the previously created Pandas structure and extracted both the validation and test sets. After manually annotating the labels discussed in section 2.1, a CSV file was generated as the output from the ELAN Annotation Software. This file is depicted in Figure 2.9.

The script follows a three-step process:

1. **Label encoding:** In this phase, the labels are transformed into a numerical format. Specifically, 'No Contact' is represented by 0, 'Contact' by 1, and 'out-of-frame' by 2.

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

Figure 2.9: ELAN Annotation Software CSV file, first row is hand side, second and third rows are starting and ending frames in millisecond, last row is the annotation.

2. **Label extraction:** For accurate label assignment, each entry in the 'frame in millisecond' column of the pandas dataframe is matched against the time intervals in the CSV file. When a match is identified, the corresponding label is added to the training set, and the row's identifier is stored in a list.

3. **Creating the validation set:** Using the list of identifiers generated in the previous step, the validation set is extracted from the main training dataset. Subsequent to this extraction, the corresponding rows are removed from the training set. The test set is prepared using the same procedure.

2.3.4 Training, Validation and Test set

Table 2.1 provides an overview of the dataset used to evaluate the pipeline, using subject 001 and 003 of the RingSensor study.

	001 # of frames	003 # of frames	Total # of frames	% of total frames
Train	$71k$	$107k$	$188k$	80%
Validation	$40k$		$40k$	15%
Test		$13k$	$13k$	5%

Table 2.1: Training, Validation and Test set summary

Training

The training set contains approximately 180k frames of two subjects carrying out activities in a simulated kitchen. For this pipeline, the training set is not labeled.

Validation

The validation set consists of about 20 minutes (40k frames) from subject 001. After labeling the right-hand activity manually, this data is used to test the labeling function mentioned in section 2.4.1.

Test

Similarly to the validation set, the test set is a subset of subject 003 of roughly 7min (13k frames), created to evaluate the pipeline's performance.

2.3.5 Characterization

To determine the optimal model for the initial stage, Intersection over Union (IoU) served as the evaluation metric for comparing the performance of two deep learning models employed in the first stage of the proposed pipeline: Yolo-v8 and Hand Object Detector. To enable this comparison, a manually annotated dataset was created. Annotations were made using a custom-developed app designed for generating bounding box annotations. Five minutes of video, randomly extracted from 15 subjects, were utilized to generate the labeled dataset.

Intersection over Union (IoU)

Intersection over Union (IoU) evaluates the performance of object detection by comparing the ground truth bounding box with the predicted bounding box (Figure 2.10). The equation used to calculate IoU is the following:

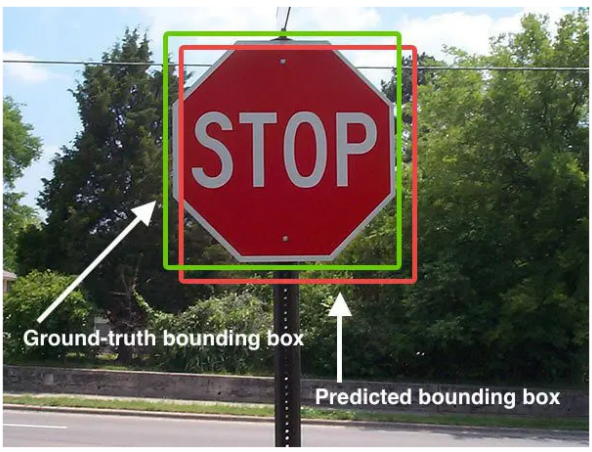

Figure 2.10: IoU uses the ground-truth and predicted bounding boxes to compute the performance of a object detection framework.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2.1}$$

Intersection over Union is a simple ratio where the numerator calculates the overlapping area between the predicted and the ground-truth bounding boxes and the denominator represents the combined area covered by both the predicted and the ground-truth bounding boxes. To facilitate this calculation, a bounding boxes annotation tool was designed to produce the dataset required for comparison.

Bounding Box Annotation Tool

The interface of the bounding box annotation tool is illustrated in Figure 2.11.

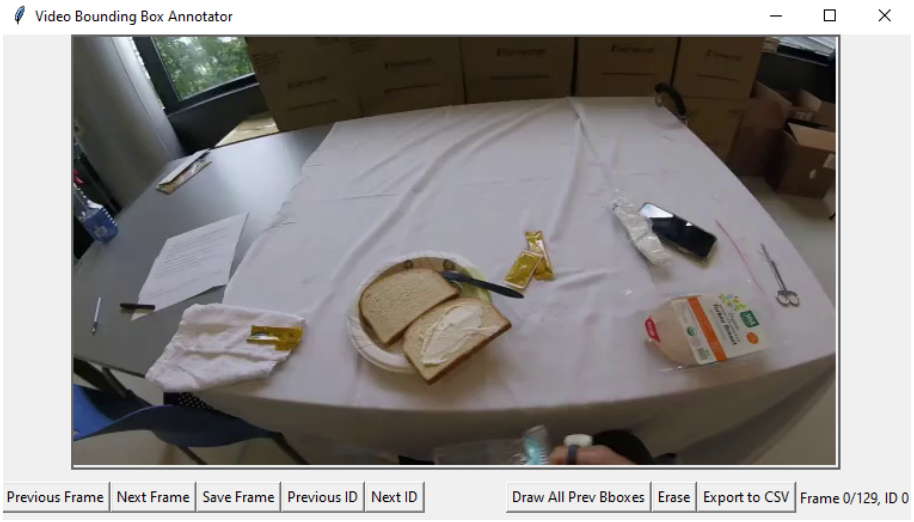

Figure 2.11: Interface of the custom-developed app for the annotation of bounding boxes

Annotations were conducted frame by frame, with users dragging and releasing the mouse cursor around objects or hands. The tool was equipped with IDs to toggle between left/right hands and objects, streamlining the process. To expedite the annotation, a *draw previous bounding boxes* button was integrated, allowing for the replication of bounding boxes from previous frames when objects remained stationary. For error correction, a function to erase bounding boxes based on their ID was introduced. For enhanced efficiency, keyboard shortcuts were assigned to each button, facilitating quicker annotations. Furthermore, features such as saving and loading current annotations were incorporated. Upon completion of the annotations, the data can be exported into a CSV file. The annotation tool was employed to label 5 minutes of footage from 15 random subjects engaging in various activities.

2.4 Stage 2

After finalizing the dataset, the training, validation, and test sets were fed to the second stage, with the training set remaining unlabeled. The pipeline employed in stage 2 is depicted in the block diagram shown in Figure 2.12. The three splits

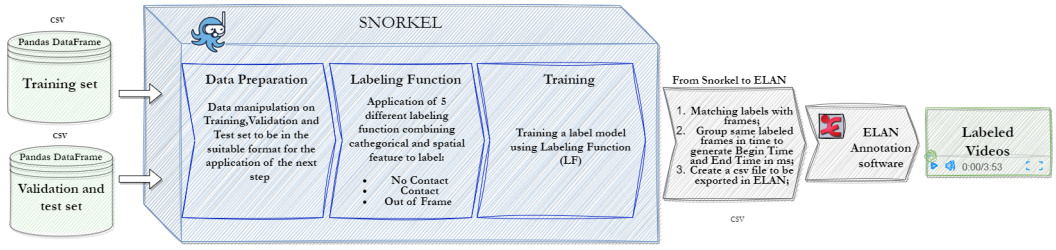

Figure 2.12: Block diagram for the second stage of the self-labeling pipeline

of the dataset underwent initial processing by Snorkel to train the labeling model. Snorkel's workflow can be segmented into three primary steps:

1. **Loading Data:** Through Snorkel's inherent functions, the dataset is integrated into the Snorkel framework. This phase involves data manipulation to ensure it is formatted appropriately for the next step.

2. **Labeling Function:** Labeling functions are designed to determine the relationships between pairs of bounding boxes. By encoding specific intuitions into these functions, existing relationships can be detected. There are two primary types of labeling functions: Categorical and Spatial. This step applies to the datasets five labeling function coded to detect contact, no contact and out-of-frame.

3. **Label model training:** The final step involves evaluating the performance of the labeling functions on the validation set and refining them to more accurately capture the relationships present in the RingSensor data. Ultimately, the labeling functions are applied to the training set, and through a generative model, the dataset is annotated.

After training the label model, annotations were added to the training set. To facilitate visualization of the results in the ELAN Annotation software, an additional script was developed to modify the outputs accordingly. A comprehensive breakdown of this script is available in section 2.4.2. Both the Validation and Test sets were annotated using the previously trained label model to assess performance, leveraging specific metrics such as the micro F1-score and Confusion matrices, described in section 2.4.3.

2.4.1 Snorkel

Snorkel [1] is a pioneering system allowing users to train cutting-edge models without the need for manual data labeling. Users craft labeling functions that capture the underlying relationships of an unlabeled dataset. Snorkel efficiently filters out noise from these outputs without relying on ground truth. This is achieved through the system's complete integration of a newly proposed machine learning approach, data programming. The design of the system is concisely

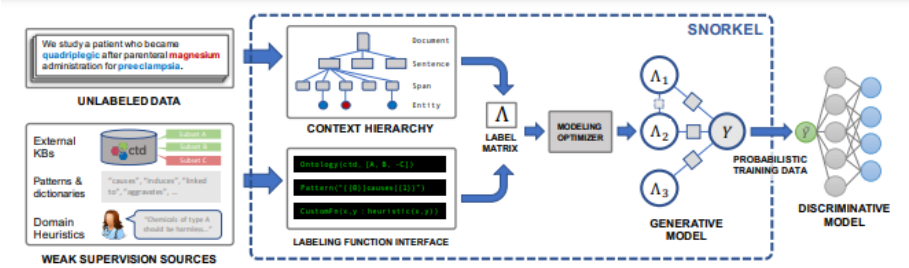

Figure 2.13: Workflow of the Snorkel architecture

depicted in Figure 2.13 and it can be can be outlined as follows::

1. Subject matter expert (SME) users craft labeling functions (LFs) that capture weak supervision sources, incorporating elements like distant supervision, patterns, and heuristics.

2. Snorkel applies these LFs to unlabeled data, subsequently learning a generative model. This model amalgamates the outputs of the LFs into probabilistic labels.

3. Using these probabilistic labels, Snorkel can train a discriminative classification model, which could be a deep neural network.

Labeling Functions

Instead of manually labeling training data, Snorkel users write labeling functions. The labeling functions used for this pipeline can be grouped into two primary category. Categorical intuition, knowledge regarding the typical categories of subjects and objects involved in such relationships (for instance, 'person' is often the subject for actions like 'ride' and 'carry'). Spatial intuition, Understanding the relative positions of the subject and objects (for instance, the subject is generally positioned above the object in the context of the action 'ride'). For the RingSensor study data, five labeling functions were created. Out of these, three are categorical, corresponding to each label, while the remaining two are spatial, specifically

designated for the 'Contact' label.

Categorical LFs:

- Given that the deep learning models identify bounding boxes of objects when hands are nearby or in contact, if both hands and objects are detected, it's labeled as 'contact'.
- The "No contact" label proved challenging due to the dataset used to train the deep learning models. If the hand is detected but not the object, then it's labeled as "No contact".
- Regarding the "out-of-frame" label, if no hands are detected, it is labeled as "out-of-frame".

Spatial LFs: Spatial labeling functions were applicable solely for the 'contact' label and served to reinforce the categorical 'contact' labeling function.

- The 'contact' label is assigned based on the overlapping area of the bounding boxes of hands and objects, specifically when the overlap exceeds a predefined threshold.
- To counteract situations where contact might still exist even if the overlapping area is below the threshold, a labeling function is employed that checks if the center of the hand is within the bounding box of the object.

Following the application of the labeling functions, Snorkel produces a table detailing the performance of each LF. This table is available for review in section 3.2.1.

Training the model

After defining the LFs, a multi-class LabelModel was employed to assign training labels to the unlabeled training set. The output from the generative model comprises a set of probabilistic labels, along with the probability associated with each labeling function. The model underwent training for 100 epochs with a step size of 0.01.

2.4.2 Python Script

The labeled data generated by Snorkel was used to get a visual representation in ELAN. Additionally, it was used to produce metrics, which are described in the next section, to assess the performance of the entire pipeline.

The output of snorkel was a vector of numeric form labels with length equals to the amount of frames in the video. Given that all elements were synchronized using

frame timestamps in milliseconds, synchronization was achieved by utilizing the row IDs. By extracting the row IDs, the script aligned these newly generated labels with the corresponding rows in the original training dataset. After the alignment, the scripts decode the numeric form labels back into text form. Thus, the training set was labeled.

Given that labels were assigned frame by frame, there were instances where labels switched momentarily before reverting to their prior values. To address this inconsistency, a function was developed to correct such switching. This function checks if the consecutive number of frames labeled with the same annotation were less than 7 ($\approx 0.23\,\text{sec}$) and adjusted these labels to match their preceding values. In this way, the smallest sequence of consecutive frames was set to 7. This corrective measure enhanced the accuracy metrics, as evidenced in section 3.2.3.

To maintain compatibility with the annotation software, a new pandas dataframe was utilized. In this revised structure, for every sequence of frames with the same label, only the start and end frame timestamps in milliseconds were recorded alongside the corresponding annotation. This led to the creation of the table illustrated in Figure 2.14. After sorting the rows by the beginning of each time interval, the script exported the updated data into an ELAN-compatible CSV file. By applying the same script to the labels generated for the validation set by Snorkel, a visual comparison between the ground-truth labels and the predicted labels was carried out.

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

Figure 2.14: ELAN-compatible CSV file

2.4.3 Characterization

To evaluate the pipeline's performances, two metrics used in multi-label machine learning frameworks were employed: the F1-score micro and Weighted F1-score. To derive these metrics, we utilized confusion matrices, which are commonly employed to evaluate the performance of a classification model.

F1-score

The F1 score is an evaluation metric in machine learning that gauges a model's accuracy by combining its precision and recall scores. This accuracy metric calculates the frequency with which a model made accurate predictions across the entirety of the dataset. For data distributions with multiple classes, the F1-score micro is calculated. Given the dataset's imbalanced nature, the weighted F1-score is preferred for the thesis. The formula is:

$$Weighted\ F1 - score = \sum_{i=1}^{N} w_i * F1 - Score_i \tag{2.2}$$

where w_i is:

$$w_i = \frac{N.\ of\ samples\ in\ class\ i}{Total\ numebrs\ of\ samples} \tag{2.3}$$

Confusion Matrix

A confusion matrix provides a table-based visualization of a prediction model's performance. Each entry in the matrix represents the count of predictions where the model accurately or inaccurately predicted the classes. Figure 2.15, shows how a confusion matrix is calculated for a binary classification problem.

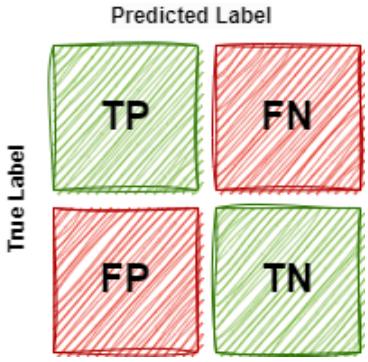

Figure 2.15: Binary Confusion Matrix: **TP:** Correctly predicted positive values. **FP:** Negative values incorrectly predicted as positive. **TN:** Correctly predicted negative values. **FN:** Positive values incorrectly predicted as negative.

The multi-class confusion matrix extends the binary confusion matrix to handle cases where there are more than two classes. Rows represent the actual or ground-truth classes. Columns represent the predicted classes by the classifier. Each cell in the matrix corresponds to the number of times a particular class was predicted for a given actual class. The diagonal elements of the matrix represent correct predictions

for each class, while off-diagonal elements indicate where misclassifications occurred. By examining the matrix, one can quickly see not only how many predictions were correct, but also the nature of the errors being made. The multi-class confusion matrix is shown in Figure 2.16

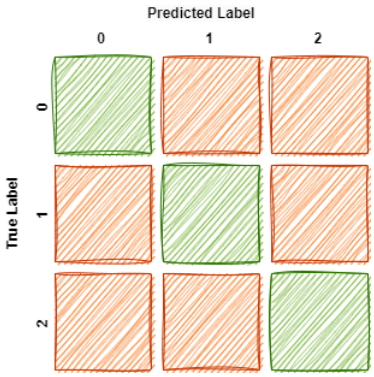

Figure 2.16: Multi-class Confusion Matrix for a three label classifier

2.5 ELAN Annotation Software

Upon completing the execution of the entire pipeline, ELAN was employed as a visualization tool to facilitate a clearer interpretation of the confusion matrices. This visual representation helped in pinpointing the discrepancies between the ground-truth labels and the predictions generated by the framework. This not only enhanced the comprehension of the matrix results but also provided insights into the areas where the pipeline might have faltered.

ELAN is an annotation tool for audio and video recordings. This annotation software allows users to extensively annotate audio and video recordings with limitless textual notes. These annotations can range from individual words, sentences, or glosses to comments, translations, and descriptions of observable features in the media. Users can organize these annotations on different layers, known as tiers. These tiers have the capability to be connected in a hierarchical manner. Moreover, an annotation can be synchronized with specific media timings or be linked to previously existing annotations. Thus, for the RingSensor video data, synchronization proved to be incredibly beneficial as showed in Figure 2.17.

However, the labeling process within the software is tedious due to its multi-step nature. Users must initially choose the time interval for each label tier, followed by selecting the type of annotation. As detailed in section 1.2.4, each annotation type must be selected twice – once for the right side of the body and once for the left. This considerably slows down the labeling process unless there's an increase in the

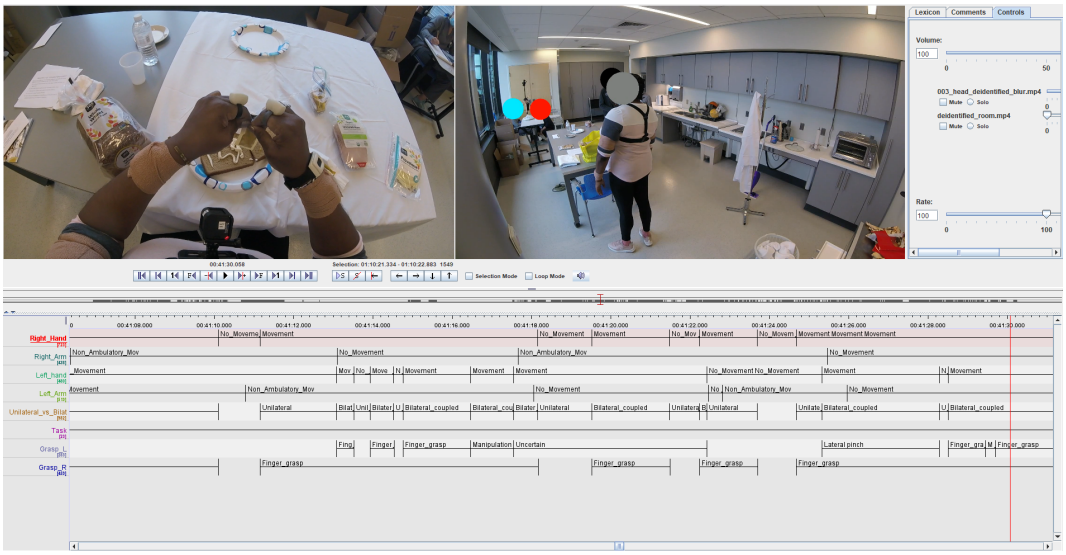

Figure 2.17: ELAN interface for RingSensor video data annotation with the labels described in section 1.2.4

number of annotators.

Results

3.1 Stage 1

To select the optimal model for stage 1, we began by visually assessing how each model processes our video data. This involved observing the videos after they had been processed by the two deep learning models. The outcomes are depicted in figures 3.1 and 3.2.

Hand Object Detector

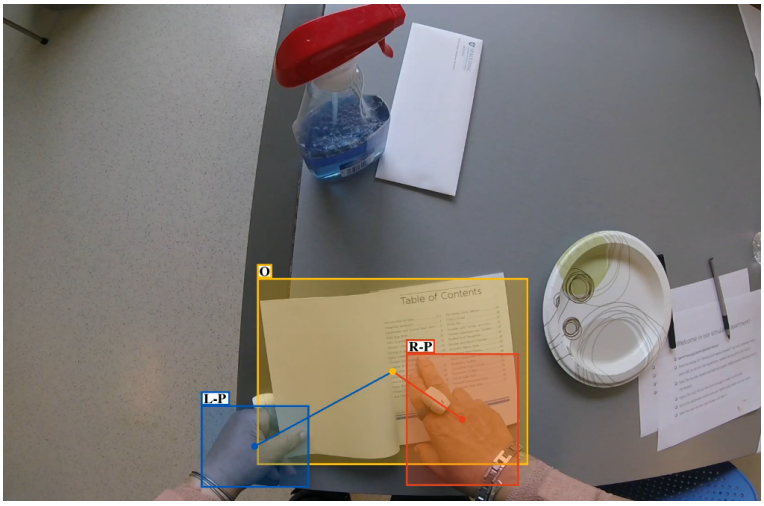

Figure 3.1: Screenshot of subject 001 with the Hand Object Detector

Yolo-v8

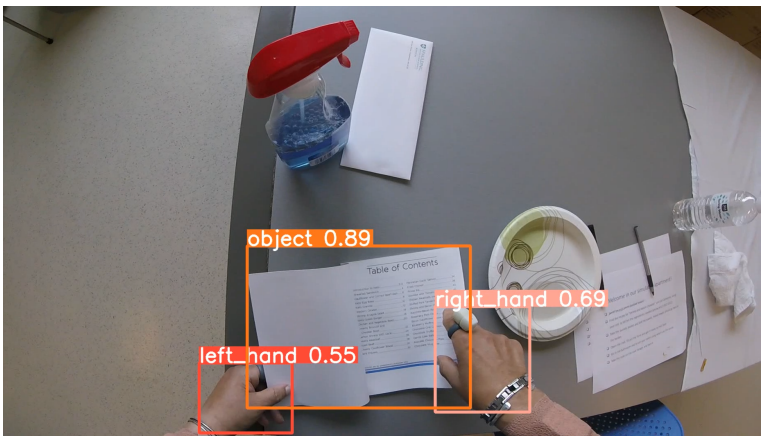

Figure 3.2: Screenshot of subject 001 with Yolo-v8 model

Both models accurately predict hand bounding boxes and exclusively identify the objects they are in contact with. As illustrated, they both ignore other objects in the frame, concentrating solely on the book the subject is holding.

3.1.1 Characterization

The Intersection over Union was computed for the subset of the RingSensor's video data to gain a deeper insight into the performance of the models. After manually annotating bounding boxes for hands and objects in contact in randomly selected five-minute video clips from various subjects, we evaluated the performance of both the Hand Object Detector and YOLO-v8 models. The tables 3.1 for the Hand Object Detector and 3.2 for YOLO-v8 report the accuracy of right-hand bounding boxes. In the *Hand* columns, we present the # of correctly detected right-hand frames out of the total frames containing the right hand. The *IoU* columns showcase the average IoU score for the correctly detected right hand frames.

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.	4733/5599	0.9

Table 3.1: IoU for Hand Object Detector

Subjects	**Hand(Medium)**	**IoU(Medium)**	**Hand(Large)**	**IoU(Large)**	**Hand(XL)**	**IoU(XL)**
002	905/1284	0.92	884/1284	0.86	870/1284	0.86
004	932/1063	0.84	963/1063	0.86	950/1063	0.84
005	817/1033	0.78	827/1033	0.8	845/1033	0.8
014	1316/1377	0.94	1254/1377	0.94	1261/1377	0.94
019	741/842	0.84	729/842	0.84	726/842	0.84
Avg.	4711/5599	0.864	4657/5599	0.86	4652/5599	0.856

Table 3.2: IoU for Yolo-v8 models

3.2 Stage 2

3.2.1 Snorkel - Labeling Function

The labeling function analysis tool provided an overview of how well the crafted labeling functions (LF) matched the validation set. This offers insights into the efficacy of the LFs and determines if there's a necessity to make adjustments to them. The analysis tool provides the following metrics:

- **Polarity:**The label for which the LF is written.
- **Coverage:** The percentage of frames addressed by the LF.
- **Overlaps:**Percentage of potential overlaps with other labeling functions.
- **Conflicts:**Percentage of instances where the LF disagrees with other labeling functions.
- **Correct:**Instances covered by the LF that are correctly labeled.
- **Incorrect:**Instances covered by the LF that are incorrectly labeled.
- **Empirical Accuracy:**Accuracy measured on the data covered by the LF.

Hand Object Detector

	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_object	0	[1]	0.51	0.50	0.0	19154	2761	0.87
lf_area	1	[1]	0.50	0.50	0.0	18894	2574	0.88
lf_dist	2	[1]	0.31	0.31	0.0	11409	1948	0.85
lf_no	3	[0]	0.10	0.0	0.0	1454	3041	0.32
lf_no_frame	4	[2]	0.39	0.0	0.0	11593	5097	0.69

Table 3.3: LF Analysis with Hand Object Detector on Validation set

Yolo-v8 Medium Model

	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_object	0	[1]	0.46	0.45	0.0	17078	2906	0.85
lf_area	1	[1]	0.45	0.45	0.0	16824	2761	0.86
lf_dist	2	[1]	0.25	0.25	0.0	9271	1687	0.85
lf_no	3	[0]	0.16	0.0	0.0	1653	5056	0.25
lf_no_frame	4	[2]	0.38	0.0	0.0	11204	5203	0.68

Table 3.4: LF Analysis with Yolo-v8 Medium on Validation set

Yolo-v8 Large Model

	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_object	0	[1]	0.46	0.45	0.0	17078	2906	0.85
lf_area	1	[1]	0.45	0.45	0.0	16824	2761	0.86
lf_dist	2	[1]	0.25	0.25	0.0	9271	1687	0.85
lf_no	3	[0]	0.16	0.0	0.0	1653	5056	0.25
lf_no_frame	4	[2]	0.38	0.0	0.0	11204	5203	0.68

Table 3.5: LF Analysis with Yolo-v8 Large on Validation set

Yolo-v8 XL Model

	j	Polarity	Coverage	Overlaps	Conflicts	Correct	Incorrect	Emp. Acc.
lf_object	0	[1]	0.51	0.50	0.0	18401	3475	0.84
lf_area	1	[1]	0.50	0.50	0.0	18175	3324	0.85
lf_dist	2	[1]	0.28	0.28	0.0	10130	2091	0.83
lf_no	3	[0]	0.12	0.0	0.0	1363	4024	0.25
lf_no_frame	4	[2]	0.37	0.0	0.0	10828	5009	0.68

Table 3.6: LF Analysis with Yolo-v8 XL on Validation set

3.2.2 Confusion matrices

When assessing the pipeline results and visualizing them, we employed confusion matrices. For each model, we generated two confusion matrices, one for the validation set and another for the test set. Confusion matrices can be visualized in various ways, but in our case, we arranged them with the true labels on the left vertical side and the predicted labels along the bottom side. The following figures illustrate the eight confusion matrices we generated.

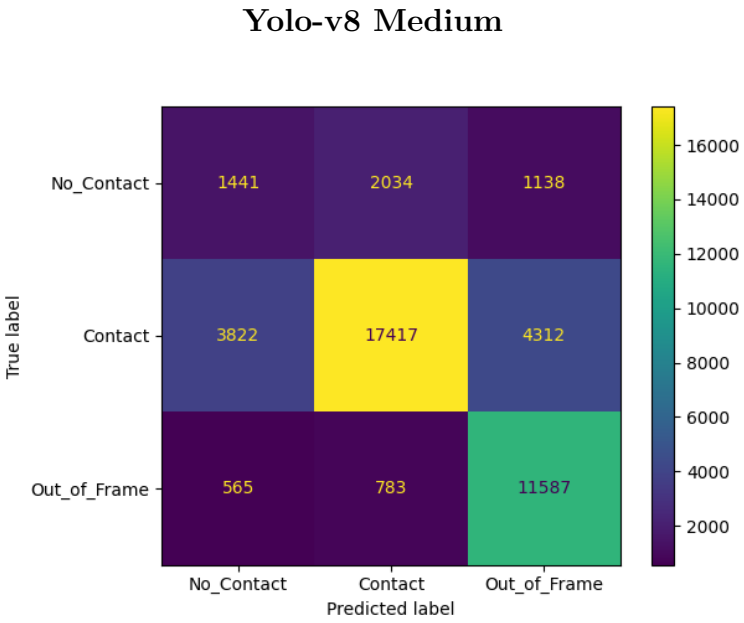

Figure 3.3: Medium model confusion matrix for validation set

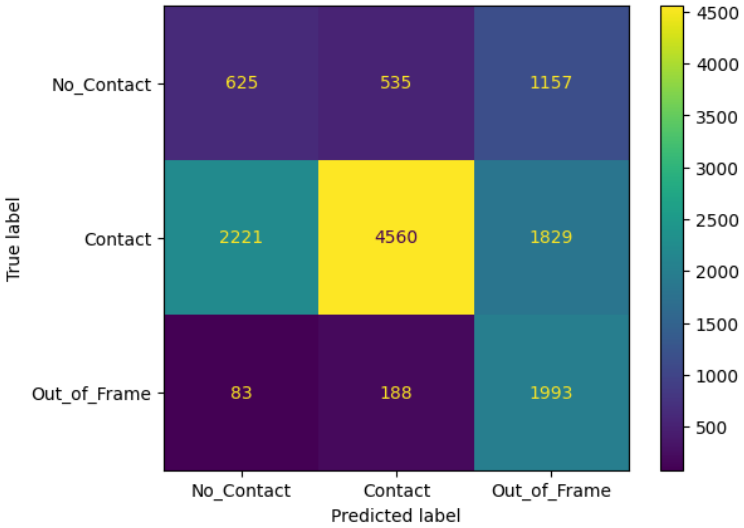

Figure 3.4: Medium model confusion matrix for test set

Yolo-v8 Large

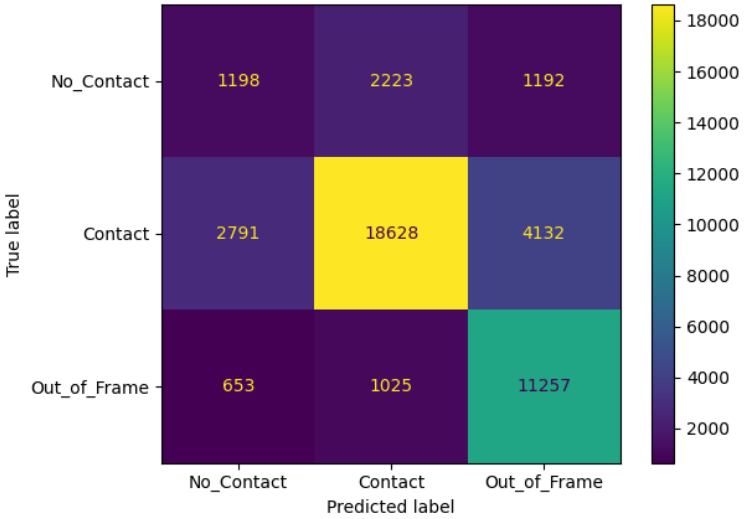

Figure 3.5: Large model confusion matrix for validation set

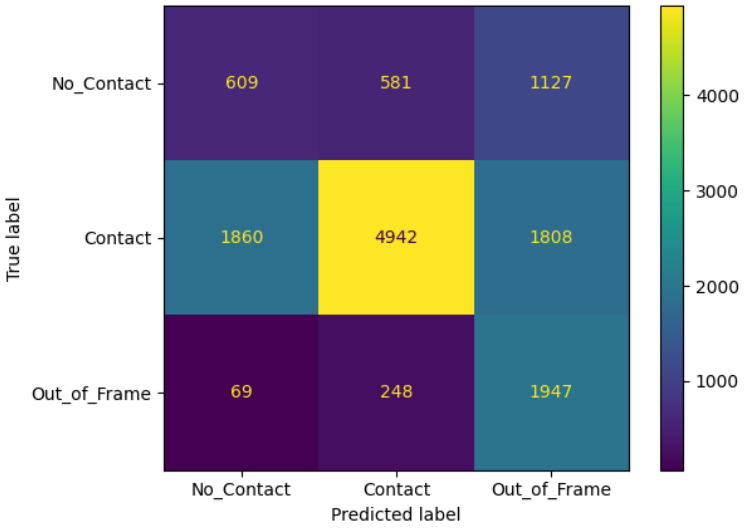

Figure 3.6: Large model confusion matrix for test set

Yolo-v8 XL

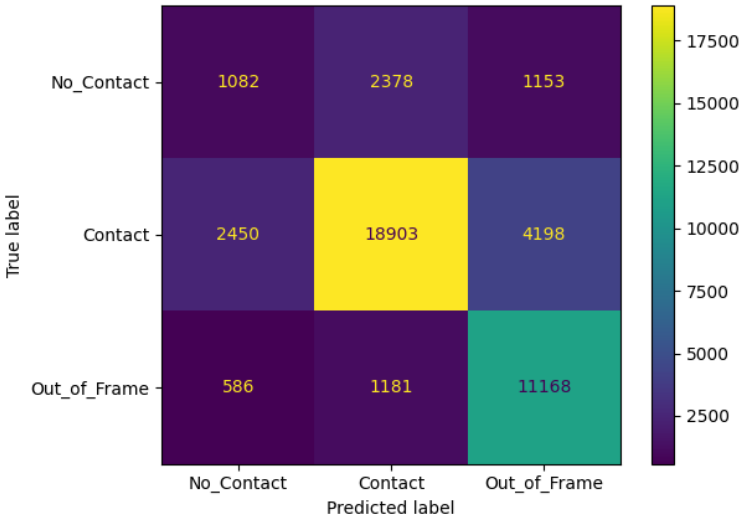

Figure 3.7: XL model confusion matrix for validation set

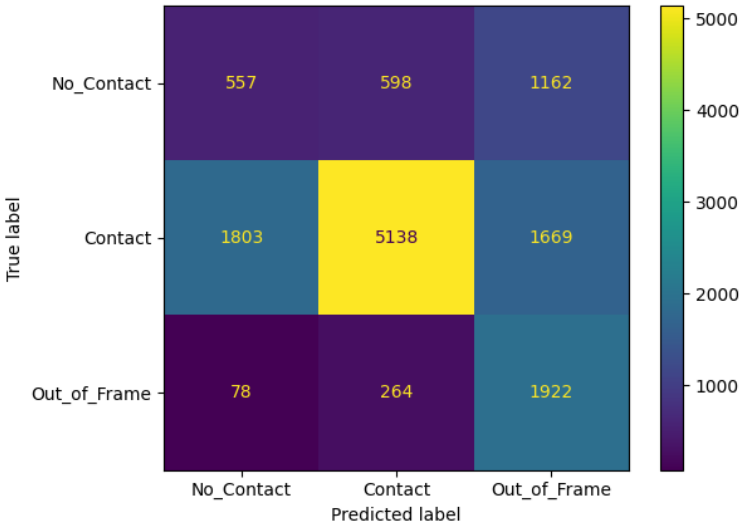

Figure 3.8: XL model confusion matrix for test set

Hand Object Detector

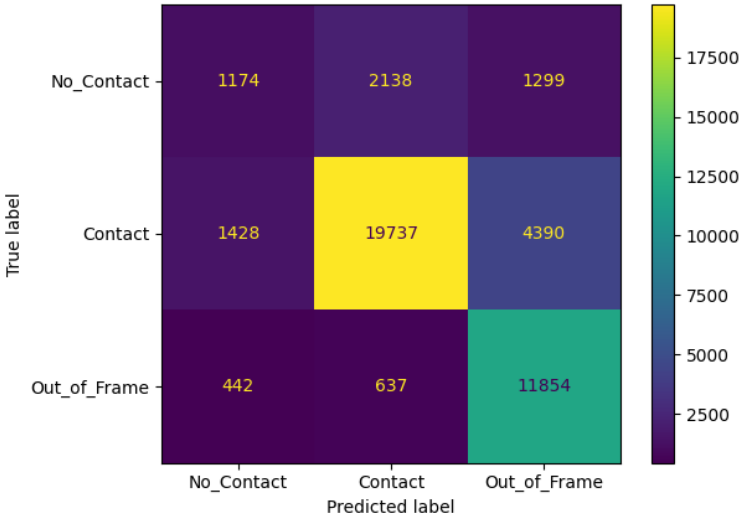

Figure 3.9: Hand Obj. Det. model confusion matrix for validation set

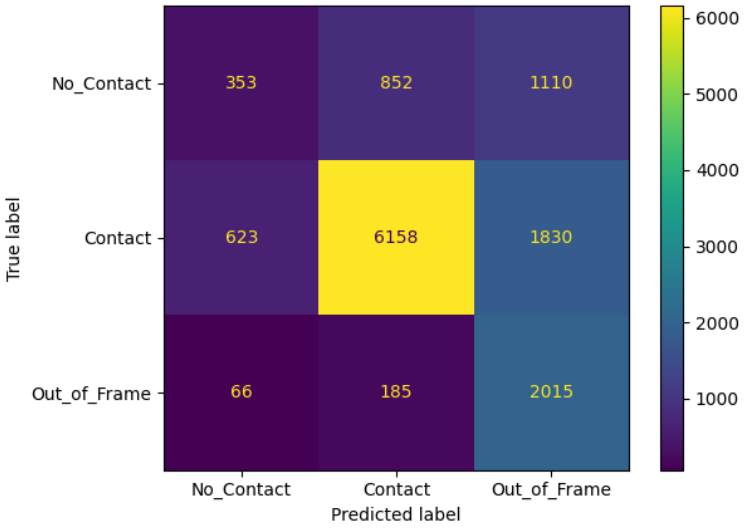

Figure 3.10: Hand. Obj. Det. model confusion matrix for test set

3.2.3 F1-score

To comprehensively assess the pipeline's performance across all labels, we opted to calculate two F1 scores: one for micro and another for weighted evaluation. Tables 3.7 and 3.7 display these scores for the validation and test sets, respectively. The final column represents the micro F1 score evaluated prior to the implementation of the 7-frame minimum non-switching label algorithm.

Validation set	F1-score Micro	Weighted F1-score	No 7 Frames check
Hand Obj. Det.	0.76	0.753	0.74
Yolo-v8 Medium	0.71	0.71	0.69
Yolo-v8 Large	0.72	0.72	0.71
Yolo-v8 XL	0.72	0.72	0.70

Table 3.7: F1-scores Validation set table

Test set	F1-score Micro	Weighted F1-score	No 7 Frames check
Hand Obj. Det.	0.65	0.64	0.63
Yolo-v8 Medium	0.544	0.56	0.546
Yolo-v8 Large	0.57	0.59	0.56
Yolo-v8 XL	0.58	0.59	0.57

Table 3.8: F1-scores test set table

3.3 Visual Comparison Results

Finally, we imported all the labels generated for the test set from the four different models into ELAN to demonstrate each model's proficiency in creating the time intervals for annotations and the correct labeling. A comprehensive discussion of the results and outcome of this phase is offered in the following Chapter.

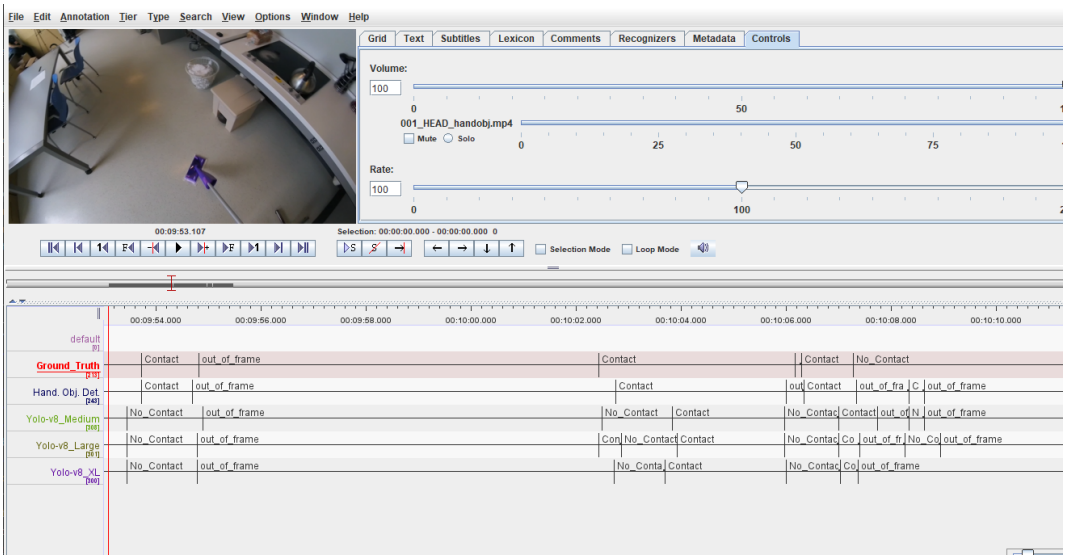

Figure 3.11: Visual Comparison of the different models on the Test set

Discussion

4.1 Stage 1

We trained two different deep-learning models to compare their performance on the RingSensor video data, namely the Hand Object Detector and YOLO-v8. Given that they were trained on the dataset described in [30], this phase was essential to determine the project's feasibility. Figures 3.1 and 3.2 display screenshots of the two deep learning models when applied to the RingSensor video data. Yet, these visuals alone weren't sufficient to comprehensively assess the models' performance. Nevertheless, this was useful for determining if the models functioned as intended on our video data. We observed that, even with other objects present in the frame, the highlighted bounding boxes were exclusively around objects in contact with the hand as described in [30]. Consequently, we manually annotated bounding boxes for 5 minutes of footage from 15 random subjects engaged in the simulated activities detailed in section 1.2.3 to produce performance metrics for the deep learning models. We also picked the activities randomly, and the results are illustrated in tables 3.1 and 3.2. The *Hand* columns represent the # of frames of right hand correctly detected over the total amount of right hand frames per video. The IoU column shows the average Intersection over Union for all the detected right hands.

Let's delve into the analysis of the YOLO-v8 models. When examining the IoU, we primarily focused on hand detections. The performance of the YOLO models was notably consistent, registering nearly identical IoU scores across different implementations of the model. Interestingly, there wasn't a direct relationship between the model size and improved detection capabilities on our video dataset. For instance, in the right-hand detection column, there are scenarios where the medium model outperforms both the Large and XL versions. For three out of five subjects, the medium model correctly detected more right hands. As initially observed from the tables, the medium model appears to be slightly better in hand detection, boasting both a higher overall hand detection rate and IoU score compared to the other two models. Nonetheless, after executing the complete

pipeline, this distinction became less significant, with the XL model outperforming the others when integrating the Yolo models into the full pipeline. For a more comprehensive understanding of this result, we plan to include more manually labeled frames across all subjects.

In terms of the Hand Object Detector, it is undeniably the superior model for our first stage. It outperformed the Yolo-v8 Medium by detecting a greater number of hands and achieving an average IoU across subjects of 90%. This distinction plays a significant role in the final results of the full pipeline; with this model, we achieve the best performance.

4.2 Stage 2

4.2.1 Labeling Function

After crafting the labeling function, the outcomes are consolidated by the LF analysis tool and presented in a DataFrame, as illustrated and detailed in section 3.2.1. This phase is crucial for stage 2, as it will significantly influence the overall performance of the pipeline. Inadequately written LFs will lead to an overall low accuracy throughout the pipeline.

Tables 3.3, 3.4, 3.5, and 3.6 display the performance of the Labeling Function across the four deep learning models employed in this thesis. From the tables, it is evident that the Hand Object Detector excels in coverage for the 'Contact' label (Polarity 1). In contrast, it demonstrates the least coverage for the 'No Contact' label, emphasizing the model's behavior toward detecting hands in contact with objects. However, an interesting observation is that even with its minimal coverage among the four models, its accuracy remains the highest. The three labeling functions tailored for the 'Contact' label are remarkably aligned, as evidenced by the *Overlaps* column, and exhibit superior accuracy compared to the rest. This further reaffirms that the Hand Object Detector's primary emphasis is on hand and object contact detection.

Concerning the Yolo models, they exhibit a similar pattern to the Hand Object Detector, with the three LFs for the 'Contact' label consistently outperforming in both Accuracy and Coverage for each model. This trend can be attributed to the fact that the same dataset was employed to train all four models. Nonetheless, this isn't a setback, as the primary objective of the thesis is to annotate the 'Contact' label with the utmost accuracy. Among the three models, the Yolo-v8 XL model exhibits the highest Coverage for the 'Contact' LFs, displaying impressive accuracy, though not outperforming the Hand Object Detector. Conversely, its coverage for the 'out of frame' and 'No Contact' labels is the least, but its accuracy remains on par with the other models. In sum, the XL model stands out with superior performance within the Yolo models.

We favor a model with superior coverage as it aids in streamlining the creation of time intervals for the contact label. This approach enables clinicians to expedite the grasp labeling process significantly, as they are provided with pre-defined time intervals. This idea can be further elucidated by examining the frames from the validation set. Out of the 43,100 total frames, 25,556 are labeled as 'Contact', indicating that 'Contact' instances constitute 59% of the set. Both the Hand Object Detector and the Yolo-v8 XL model closely align with this figure, suggesting that the coverage of the LFs encompasses almost all the 'contact' labels within the validation set. A parallel observation can be made for the 'out of frame' label. Here, there are 12,933 instances, making up 30% of the dataset. Given this, it's evident

that both models might be mislabeling instances as 'out of frame' even when hands are in the frame, seeing as their coverage surpasses the actual count of 'out of frame' labels. Addressing this discrepancy would likely require the integration of ringSensor Video data into the validation set.

4.2.2 Characterization

After training the label model and deriving the model weights, I evaluated its performance using both the validation and test sets. The resulting figures are detailed in section 3.2.2. An immediate observation is the relationship between the results of the labeling function analysis and the patterns in the confusion matrices. The 'Contact' label, along with the 'out of frame' label, emerges as the best-performing one across the four models, at least when considering the validation set. But before delving into the details of the confusion matrices, let's first address the best-performing model.

Focusing solely on the 'Contact' label, given its primary relevance to our study, it becomes abundantly clear that among the Yolo models, the XL model takes the lead, a prediction consistent with our LF analysis. Yet, a reaffirmation from this evaluation is that the Hand Object Detector consistently eclipses the other models in performance, evident across both the validation and test datasets.

When it comes to the 'No Contact' label, it's evident that none of the models effectively detect instances where there's no contact between hands and objects. An examination of the training dataset utilized for the deep learning models, as outlined in [30], reveals a distinct imbalance skewed against 'No Contact' instances. This observation is further solidified by looking at the confusion matrices. Within the validation set, the 'No Contact' label accounts for 4,611 frames out of a total of 43,100, and 2,315 frames out of 13,192 in the test set. Given the objectives of this thesis, we have chosen to bypass the 'No Contact' label. However, it would be interesting as future steps of this research to leverage the RingSensor video data to annotate bounding boxes capturing more 'No Contact' instances.

Finally, concerning the 'out of frame' label, while it manifests promising outcomes in the validation set, its performance doesn't mirror the same in the test set. A closer examination of the test set reveals a notably reduced count of 'out of frame' instances, consisting of merely 2,266 of the 13,192 frames. Given that both the validation and test sets consist of randomly chosen everyday activities, as referenced from table 1.1, such disparities might be expected with other subjects in the RingSensor video data. Consequently, the reliability of this label is analyzed. Indeed, this outcome aligns with our anticipations, especially when reviewing the LF analysis related to the 'out of frame' label.

To harness the full potential of the 'Contact' label, other than just looking at its accuracy, we must delve deeper into Precision and Recall metrics. By thoroughly

understanding these aspects, we can understand the true value of the label in real-world applications. This deeper analysis will provide a comprehensive perspective on how clinicians can effectively streamline and enhance their annotation process using this framework. We'll focus exclusively on the Hand Object Detector, given its superior performance.

Label	Precision	Recall
No Contact	0.39	0.25
Contact	0.87	0.77
Out of frame	0.67	0.92

Table 4.1: Precision and Recall on the Validation set

Label	Precision	Recall
No Contact	0.34	0.15
Contact	0.85	0.71
Out of frame	0.40	0.88

Table 4.2: Precision and Recall on the Test set

Table 4.1 and 4.1, summarizes the Precision and Recall metrics obtained for the Hand Obj. Det. for the validation and test. By looking at these metrics, we can say that the no-contact label is not working as expected, justified by the low Precision and Recall in both the validation and test set.

The 'Out of frame' label exhibits low precision and high recall, indicating that our model tends to be generous in its predictions. It frequently assigns this label even when uncertain. However, we utilized the 'Out of frame' label for instances where we understood that our model couldn't reliably tell whether there was contact with an object, primarily because the hands were not visible within the frame. Therefore, this label was inherently meant for further examination. Even though a review of this label is necessary, its presence still establishes a time interval within the annotation software. This, in turn, facilitates a more streamlined labeling process for clinicians.

The 'Contact' label exhibits both high precision and medium to high recall. This signifies that when our model assigns the 'Contact' label, it's usually correct. With such robust performance, the annotation process becomes much more efficient. Given that the designated time intervals are established with higher accuracy, the likelihood of them being correct is high, further expediting the clinician's workflow.

To enhance the model's accuracy, we employed F1 scores as a metric to gauge the overall performance of the pipeline. This was used in determining the optimal number of consecutive frames to consider without encountering a label switch,

effectively determining the minimum sequence of frames required to achieve the highest F1 scores. One of the challenges previously addressed was the frequent label switching due to inaccuracies in bounding box predictions. Leveraging the F1 score, we established that the ideal minimum sequence of non switching frames to enhance performance is seven. This adjustment led to an improvement across all label metrics. Tables 3.7 and 3.8 depict the performance enhancements achieved through this strategy.

The outcomes of the complete pipeline are illustrated in section 3.3. In Figure 3.11, the outputs from all models, post-frame correction, have been imported into the annotation software. The figure distinctly reveals that the Hand Object Detector outperforms others when over imposed with the ground truth. Its time intervals align more closely with the original ones. The 'out of frame' label, while requiring further verification, still contributes to streamlining the annotation process, successfully fulfilling the thesis's aim.

Conclusions

In this thesis, we confronted the prevailing issue of unlabeled video datasets, presenting an innovative methodology tailored to auto-generate labels, especially within the realm of observing upper limb activities in stroke patients. After reviewing existing approaches to annotating wearable sensor data through manual video recordings and various machine learning techniques, we presented our novel framework. By capitalizing on recent advances in deep learning and computer vision, our devised system extracts bounding boxes of hands and objects from videos and collaborates with Snorkel [1] to generate hand activity labels.

The primary objective of this thesis centered around accurately annotating the 'Contact' label to expedite the annotation process for clinicians. Within the context of the RingSensor project, the insights of medical professionals are essential. They will annotate the specific grasp types of post-stroke survivors. This is key to customizing treatments, especially if patients exhibit a preference for a particular grasp type due to their condition rather than using conventional ones. Nonetheless, manually selecting the time interval and then annotating contact and grasp type is a tedious process for clinicians. Therefore, our framework streamlines this by automatically creating the time intervals and accurately labeling the 'Contact' label.

To achieve this, we implemented a two-stage framework for our study. In the first stage, we evaluated various deep learning models. The hand object detector emerged as the top performer, accurately identifying hands and the objects they interact with. We faced a significant challenge in this stage with the switching of the left and right sides. However, we tackled this issue by using a customized script to process the outputs, correcting and preparing them for the next stage.

The core of our framework lies in the second stage. For the Snorkel [1] model, correctly crafting the labeling functions was central. By deeply analyzing the datasets and leveraging spatial and categorical intuition on the RingSensor video data, we were able to correctly define LFs for the 'Contact' label with high accuracy. Following the data processing and correcting the label switching issues, our model achieved a micro F1-score of 76% across all labels. After examining the Precision

and Recall metrics across all the labels, we chose to exclude the 'No Contact' label because of its low precision, recall, and limited data presence. On the other hand, the 'Contact' label stood out with high precision and recall, attesting to our framework's ability to accurately pinpoint when hands and objects are in contact and delineate the associated time intervals. The 'out of frame' label, despite its low precision, was retained due to its high recall. Meaning that when the framework labels 'out of frame', we need to double-check it. We had originally incorporated this label to signal instances when the hands were not visible within the frame. Hence the deep learning model was unable to generate bounding boxes during such instances, necessitating a manual review and annotation using an alternative camera angle. Retaining this label ensures the creation of the time interval, and, although a review is mandatory, it accelerates the overall annotation procedure.

The results of this study offer an effective solution to the issue of limited labeled data in stroke patient video analysis. The existing framework offers room for improvement. By incorporating more instances of the 'No Contact' label and deploying separate deep learning models for hands and contacts, we can better identify movements like reaching and more accurately delineate the time intervals for contact.

Additionally, It can be used as a blueprint for addressing similar challenges in the wearable sensors annotation process. The approaches outlined can be extended to annotate additional labels within the RingSensor data. For instance, leveraging a room camera's perspective combined with Human Pose Estimation could provide valuable features for Snorkel, facilitating the labeling of non-ambulatory arm movements.

In conclusion, the proposed framework shows the potential of the expanding domain of machine learning within rehabilitation, especially when faced with vast unlabeled datasets, and presents a viable alternative to traditional manual annotation and machine learning methodologies.

Bibliography

[1] 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). arXiv: 1711.10160. URL: http://arxiv.org/abs/1711.10160 (cit. on pp. i, 29, 52).

[2] 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)*. 2022, pp. 01–04. DOI: 10.1109/BHI56158.2022.9926918 (cit. on pp. i, 4).

[3] Emil Jovanov, Shelton Wright, and Harsha Ganegoda. «Development of an Automated 30 Second Chair Stand Test Using Smartwatch Application». In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2019, pp. 2474–2477. DOI: 10.1109/EMBC.2019.8857003 (cit. on p. 2).

[4] Scott D. Uhlrich, Antoine Falisse, Łukasz Kidziński, Julie Muccini, Michael Ko, Akshay S. Chaudhari, Jennifer L. Hicks, and Scott L. Delp. «OpenCap: 3D human movement dynamics from smartphone videos». In: *bioRxiv* (2022). DOI: 10.1101/2022.07.07.499061 (cit. on p. 2).

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

[6] 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), pp. 1824–1835. URL: https://api.semanticscholar.org/CorpusID:199518214 (cit. on p. 3).

[7] 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)* (2021), pp. 550–554. URL: https://api.semanticscholar.org/CorpusID:233434426 (cit. on p. 3).

[8] 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* (2011), pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:8364039 (cit. on p. 3).

[9] 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). URL: https://api.semanticscholar.org/CorpusID:221805576 (cit. on p. 3).

[10] 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), pp. 436–446. DOI: 10.1109/TITB.2009.2036165 (cit. on p. 4).

[11] 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*. 2013. URL: https://api.semanticscholar.org/CorpusID:6975432 (cit. on p. 4).

[12] 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. Cham: Springer International Publishing, 2014, pp. 91–98. ISBN: 978-3-319-13105-4 (cit. on p. 5).

[13] 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). URL: https://api.semanticscholar.org/CorpusID:221864535 (cit. on p. 5).

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

[15] 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), pp. 2521–2537. DOI: 10.1109/TPAMI.2011.36 (cit. on p. 5).

[16] 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. 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).

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

[18] Sebastian Böttcher, Philipp M. Scholl, and Kristof Van Laerhoven. «Detecting Process Transitions from Wearable Sensors: An Unsupervised Labeling Approach». In: 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).

[19] 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), pp. 1–19. DOI: 10.1371/journal.pone.0208692. URL: https://doi.org/10.1371/journal.pone.0208692 (cit. on p. 6).

[20] Ruo Du, Qiang Wu, Xiangjian He, and Jie Yang. «MIL-SKDE: Multiple-instance learning with supervised kernel density estimation». In: *Signal Process.* 93 (2013), pp. 1471–1484. URL: https://api.semanticscholar.org/CorpusID:29628104 (cit. on p. 6).

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

[22] 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), pp. 366–378. URL: https://api.semanticscholar.org/CorpusID:59282921 (cit. on p. 7).

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

[24] 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). URL: https://api.semanticscholar.org/CorpusID:84841963 (cit. on p. 8).

[25] 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), pp. 232–240. URL: https://api.semanticscholar.org/CorpusID:5759799 (cit. on p. 10).

[26] 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), pp. 25–34. URL: https://api.semanticscholar.org/CorpusID:23221143 (cit. on p. 10).

[27] 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), pp. 357–61. URL: https://api.semanticscholar.org/CorpusID:34594352 (cit. on p. 10).

[28] 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), pp. 1184–1193. DOI: 10.3109/09638280903437253 (cit. on p. 11).

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

[30] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. «Understanding Human Hands in Contact at Internet Scale». In: 2020 (cit. on pp. 19, 46, 49).

[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV] (cit. on pp. 19, 22).

[32] Yana Hasson, Gül Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. *Learning joint reconstruction of hands and manipulated objects*. 2019. arXiv: 1904.05767 [cs.CV] (cit. on p. 21).

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV] (cit. on pp. 21, 22).

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