Smart Rehabilitation

MASTER THESIS

Master in Innovation and Research in Informatics - Data Science

Facultat d'informàtica de Barcelona

Universitat Politècnica de Catalunya

Author:
Víctor Sendino García

Advisor:
Andrés Perez-Uribe

Tutor:
Marta Arias Vicente

October 20, 2021

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

FIB

HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD
www.heig-vd.ch

Acknowledgements

First of all, I would like to use this section to thank professor Andrés Perez-Uribe for giving me the opportunity to work on this project and for guiding me. It is a research field a bit unusual but interesting at the same time. It has forced me to do a really varied research. I think I have learned a lot during this time and I feel that now I have a much better picture of computer vision and Deep Learning in general.

I would also like to express my gratitude to Yasaman Izadmehr and Clemente Irigaray who helped me during my stay and made my time there much more enjoyable.

Even in these strange pandemic times, my experience in Switzerland has been wonderful. I think it has helped me to grow up as a person, I have met really nice people and I really enjoyed the country at many different levels.

I would also like to thank Marta Arias for her help and advice regarding the project, the report and the presentation.

To end, I would like to thank my family and friends who have been always there supporting me and making me feel better. In these strange and difficult times, they are much more important than ever.

Abstract

This thesis is born from a collaboration project between the HEIG-VD and the CHUV hospital in Lausanne, Switzerland. We study the problem of human grasp recognition from first-person RGB video input data. Grasping is the action of seizing and holding firmly an object and there exist many different types. The objective is to use grasp recognition for automating the monitoring of the rehabilitation sessions of patients with upper-limb neurological disorders.

We compared three different approaches based on Deep Learning. Firstly, a naive image model that is trained with the entire images. Secondly, a video model, so apart from the spatial features it also takes advantage of the temporal dimension. Lastly, an image model that is trained with images cropped around the hands, so it focuses only on the part that determines the grasp. We used the Yale Grasping Dataset for training the models. To enhance the interpretability of the results we proposed a coarse-grained grasp grouping based on the Feix grasp taxonomy. We also captured our own small first-person video grasp dataset to test the applicability of the models to our setup, which differs from the training dataset in the camera location and angle.

Considering the intrinsic challenges of the data such as the frequent hand-object occlusions or the dataset difficulties like its real-world setting and the low video quality, the results are relatively good. Nevertheless, they are insufficient for deploying a satisfactory system at the hospital and remark the difficulty of grasp recognition from just egocentric RGB data. It would be interesting to further research other data modalities such as depth data or to study the problem from the perspective of hand pose estimation and object detection. It is also clear that the field lacks a more modern and large dataset.

Keywords: Grasp recognition, egocentric camera, upper-limb rehabilitation, deep learning

Contents

List of Figures

List of Tables

Listings

1 Introduction

In recent years, we have witnessed the continuous rise of Artificial Intelligence, embodied particularly by Deep Learning algorithms. The advances in technology have allowed us to progressively work with bigger amounts of data and more complex models. This has led to very interesting and promising applications. Computer vision is one if not the most important of them and can be applied to many different fields like the medical one, in which is enclosed this project. Its aim is the usage of computer vision and sensors for different applications related to the automation of the monitoring of patient's rehabilitation process at a local hospital in Switzerland.

1.1 Motivation

This master thesis is the result of my internship at the Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud (HEIG-VD). This university is located in the city of Yverdon-les-Bains, in the Vaud Canton in Switzerland.

There, I have been working on a project named Smart Rehabilitation under the tutorship of professor Andrés Perez-Uribe. There were also other students inside the project that worked alongside me: Yasaman Izadmehr, a Ph.D. student from Iran, and Clemente Irigaray a Master's student from the University of Granada (UGR). However, we focused on different research directions and our tasks didn't coincide much.

This project is done in collaboration with the biggest hospital of the region, the Centre Hospitalier Universitaire Vaudois (CHUV) [1] in the city of Lausanne, the capital of the Vaud Canton. The project was born because the hospital wanted to explore the possibility of capturing data during the rehabilitation sessions of their patients to later exploit it and gain some insights that can help with the monitoring of the patients.

The hospital has some facilities where they treat and rehabilitate patients with upper-limb neurological disorders. One illustrative example of them (but not the only one) is ataxia, a clinical neurological sign (caused by other illnesses) that consists of a lack of coordination of the muscle movements. It can affect the hands, the fingers, the arms, the legs, other parts of the body and even the eye movement or cause trouble when eating and swallowing. It is typically a manifestation of the dysfunction of the parts of the nervous system that coordinate movement, like the cerebellum. It can have multiple causes like alcohol abuse, head trauma, certain medications, strokes, tumors, cerebral palsy, brain degeneration, and multiple sclerosis. Inherited faulty genes can also cause the condition.

The patients can present this condition with different levels of severity, being traduced often into having big difficulties to proceed with normality in their day-to-day life. This means that they have to adapt to their situation and in the most extreme cases even learning again to do some of their typical daily chores, which in the rehabilitation literature are often referred to as "Activities of Daily Living" (ADLs). A typical symptom like presenting a high degree of hand shakiness is a good sample of the difficulties that these people can suffer for doing something as common as interacting with different objects.

The subjects are sometimes elder people, so they may present other conditions that are

natural due to the process of aging and hence increasing their difficulties. However, currently at the hospital, the most frequent group is for middle-aged people.

For rehabilitation, the hospital has some facilities full of equipment that mimic the different chores that a person does at home. This includes things like a kitchen to cook, a bed, a bathroom... The idea is that through repetition the patients regain slowly some autonomy and the capacity of going back to their normal day-to-day life. If the patient did a certain manual action as a job or as a hobby, the rehabilitation often focuses on it.

The patients go daily to the hospital to proceed with their rehabilitation. There they are asked to do some tasks while they are monitored by some expert. This is useful for assessing the state and development of every individual and to allow them to find their main difficulties and focus especially on them. However, this approach has some limitations. First of all, it requires active work by the hospital members. To supervise the state of the patient, some therapists must be there watching closely following their progress and then give a subjective opinion. Secondly, the monitoring can only be done at the hospital. Therefore, once the patients go back to their respective homes it is not possible to know how they are doing.

The collaboration project was born with these suggestions in mind. The main idea was the integration of multiple technologies to capture and store data from the rehabilitation sessions so that it can be used for different applications focused on easing and automating the monitoring of the patient's rehabilitation giving some quantifiable results.

After discussing and exploring different technologies and devices, it was decided to require the patients to wear two devices:

- Wrist wearable PHYSILOG® 5 - MOTION SENSOR [2]: It captures different biometric variables with its sensors. It has a high-quality 3D accelerometer, a 3D gyroscope and measures the barometric pressure.
- GoPro Hero 6 camera: The patient will wear a camera located in their chest. It will capture RGB video frames from a first-person perspective. This is also called an egocentric camera.

The hospital wanted to work with the least invasive setup possible, so they didn't want these devices to have an impact on the way the patients perform their actions, even if that implies capturing less or worst quality data.

Initially, two separate applications were thought of, one for the sensor data and another for the video. Given that the project is still in an early stage, these are still being subject to some exploration and changes.

For the wrist sensor, the original idea is to capture biometric data while the patients do their rehabilitation sessions to assess the quality of their movements. The focus is that patients can progressively improve in aspects such as precision and smoothness. This data can help to find the particular difficulties of each individual and keeping historical reference can be very useful for tracking the evolution of the treatment objectively.

My work was exclusively focused on working with the data captured with the egocentric video camera. Unlike with the sensor data, the proposed task wasn't so clear from the

beginning because there were many available possibilities to explore. Wearable cameras can be great tools for patient monitoring (even at home). The original idea was to capture the rehabilitation sessions of the patients and use Deep Learning Networks to identify the activity being performed by the person (action recognition). Thus, we could create a log that indicated the action performed and at which time.

Once the two separate applications work appropriately, they could be combined so that we have a system where we can characterize the quality of the movement associated with a particular action. This kind of statistics could be useful for the rehabilitation process to find which actions suppose the biggest difficulty for a certain patient.

The idea of performing action recognition with the egocentric video had some potential but it was quite challenging, mainly due to the potentially large number of labels that we can have. Apart from that, from the hospital, they decided that studying the interactions at the action level was not very appropriate. As an alternative, they found another task more interesting for them, grasp recognition. It consists of classifying the different ways a person can grab/hold an object.

Grasping could be considered as a special case of action or gesture recognition. It is a way to study the patient's interactions at a lower level than actions (more physical), and with a smaller set of possible labels. Knowing the kind of grasp that patients attempt can be quite useful for ergo-therapists, especially if we know the grasped object and we have the sensor data so we can give some information about the quality of the interaction.

In the end, my work was centered on exploring the usage of Deep Learning Networks for performing human grasp recognition from an egocentric RGB video data input.

1.2 Purpose and Objectives

Grasping objects is one of the most general tasks that we do in day-to-day life and hence being able to perform successful grasps again is an important part of the rehabilitation process of the aforementioned patients.

The grasps that we perform are defined in great measure by the characteristics of the objects that we interact with. Those are mainly their weight, shape and deformability. That implies that for certain objects some grasps are more natural and hence more typical than others.

This would be the general case for a healthy person, but not for someone with difficulties. Depending on the case, there could be some grasps that represent a bigger challenge and therefore be used less or with a minor accuracy.

In that sense, the ability to perform successfully determined grasps can be a good indication of the state of a patient. Therefore, identify accurately those grasps tried by the patients could provide useful information for ergo therapists.

That information can be especially illuminating if it is accompanied by knowing the grasped object (object detection) and if the interaction was successful or not. That combination can open the door to give statistics regarding the accuracy of the interactions of a patient concerning a certain grasp or object, which was one of the final goals given by the hospital.

Then, the general purpose of the thesis is to explore the usage of Deep Learning Networks to perform grasp recognition from video input. In an ideal scenario, the final objective would be to train a model that could distinguish with high accuracy between all the different existing human grasps. However, this is a very challenging task to achieve for many different reasons. First of all, that would imply working with a very fine-grained taxonomy, where very subtle variations exist. Secondly, the kind of data that we work with is very complex. Occlusions between fingers and objects are very common and there can be many variations regarding position or rotation. Lastly, we are working only with the RGB frames captured from a video camera, so in some cases, it will be impossible to avoid occlusion or even being out of the frame. Therefore, the final objective needs to be more realistic.

Our first objective is to explore the grasp literature to know which are the most common grasps. Given that there is not a 100% consensus on them, we will have to stick with one of the existing taxonomies. We will have to understand the similarities between the different grasps and the existent sub-categories to group them. We will then need to define a new suitable grouping that includes the most typical grasps and a reasonable amount of classes so that our model can achieve acceptable performance and be informative at the same time.

The second goal that we have is to find or generate the appropriate training data. We should explore the existing publicly available video datasets to see if someone fulfills our requirements. We work in a setup where the patients wear the cameras on the chest, hence capturing their actions in first person. Therefore, we must find an egocentric human grasp video dataset. Since this is something quite specific, there are few available. We will also need to do some pre-processing to adapt the data before we can use it.

Once we have the data, we need a suitable model for grasp classification. We assume that we have a method that given a video is able to detect the start and end frame of each grasp. Therefore our model will just receive as input a sequence of one or more frames of a subject grasping an object and will have to predict to which grasp class it belongs.

Initially, we don't know whether an image model or a video model will perform better. Grasping is not necessarily an action that is defined by a temporal sequence, but a prediction after seeing several frames could be more robust to occlusion. Also, given that grasping is an action completely defined by the usage of hands, a model that receives cropped images around the hands instead of the whole frame might work better. The idea is to compare which of these three approaches obtains better results.

Given the complexity of the data and the models, we will not program all the code. In particular, we will use implementations from third-party libraries for the models and the transformations. The rest of the code, including the pre-processing phase, the reading of video data and the scripts used during train and test, will be implemented by us. Since the available grasping datasets are not big enough, we will fine-tune pre-trained models rather than training from scratch.

After we have trained these models, we should check that they generalize to unseen test data. In particular, we are interested to see if they work with data generated with our setup. That is because the dataset we will use to train is generated with a camera located on the head, while in our setup we will locate it in the chest. This difference in perspective and height

can be significant for the accuracy of the models. It may be required to fine-tune the models with our own data or to train from scratch. For this reason, we will also capture our own small dataset.

1.3 Thesis structure

In section 2 we will cover the state of the art of concepts that are more relevant for its development. In section 2.1 we will examine Deep Learning video models while in section 2.2 we will cover with quite detail several important grasp concepts and the main research directions.

In section 3 we review the few existent egocentric video grasp datasets and we justify the selection of the Yale Grasp Dataset. We also examine in more detail the dataset and describe its composition. In section 4 we rely on the fine-grained Feix grasp taxonomy to propose an alternative coarse-grained classification of 7 grasp classes.

In section 5 we describe the different Deep Learning approaches that we propose for performing grasp classification. In section 5.1 we comment on several decisions that we do regarding the treatment of the temporal dimension of the data. In sections 5.2, 5.3, and 5.4 we define the image, video and hand models respectively. In section 6 we justify our setup and why is necessary to capture our own dataset. We also describe the process of recording and labeling.

In section 7 we comment on the most important things regarding the training process. First of all, in 7.1 we comment on the hardware that we use for training. In 7.2 we describe the preprocessing of the dataset and the generation of the auxiliary hand grasp dataset. In 7.3 we explain the two ways in which we split our data, while in 7.4 we describe the data augmentation process done at training time to slightly compensate the relatively small dataset. In the remaining subsections, we explain the hyperparameter choice and details about the architecture of the three approaches.

In section 8 we present and compare the results obtained for the three approaches and the different experiments. To end, in section 9 we present the conclusions that we draw from the thesis and the different possibilities to explore regarding future work.

2 State of Art / Related work

In this section, we are going to review the most important research in the areas that are more relevant to the thesis.

For one side, in section 2.1 we cover several aspects of Deep Learning video models for action recognition. This encloses why they are important, the importance of the different data modalities and the architecture evolution from the origins to this day. We also mention some of the most important video datasets because of their relevance as a baseline or to be used for pre-training.

On the other hand, in section 2.2 we cover the grasping literature. Grasping is a broad concept and as such, there are many different fields of study inside it but not all of them are useful for this thesis. This is the case of sections 2.2.4 and 2.2.5 about grasp affordance and grasp contact. Even if they are not directly related to our work, we have decided to include them to illustrate grasping a bit better with two of the most prominent study directions.

2.1 Deep Learning applied to video data

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. It seeks to replicate parts of the complexity of the human visual system to enable computers to understand and automate tasks that humans can do.

It is a discipline that has existed for a long time. However, the technological resources at that time were not powerful enough to keep pace technically with the problems that were being investigated. Thanks to the advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to evolve very quickly and achieve success in several tasks.

Inside computer vision, we can distinguish different tasks. One of the most popular is image recognition, which consists of determining if a certain object feature or activity appears in an image or not.

At the same time, image recognition can be further divided into other subtasks [3]:

- Image Classification: Predict the type or class (label) of an object in an image.
- Object Localization: Locate the presence of objects in an image and indicate their location with a bounding box.
- Object Detection: Locate the presence of objects with a bounding box and types or classes (label) of the located objects in an image. It combines the image classification and objects localization problem.
- Object/Instance segmentation: It is a variation of object detection where instead of a coarse bounding box, the instances of the recognized objects are indicated by highlighting the specific pixels.

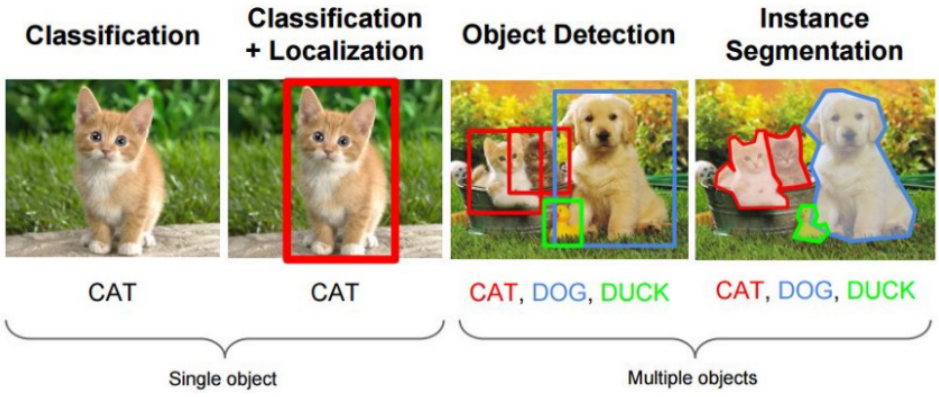


Figure 1: Image recognition subtasks. Figure extracted from [4]

These tasks are typically used with still images, but can also be applied to videos. The most simple way to adapt them is to use a **rolling average**. It consists of using an image model to predict individually every single frame of the video and then average the predicted probabilities and select the class with the highest one. The problem with this approach is that this way we are not really exploiting the temporal dimension that is present on video data. There are some instances where time is relevant for the prediction task. That could be the case of Action Recognition, a subcase of image classification where we want to predict human activities based on image/video. Some actions are defined by their temporal sequence. If we had for instance one label consisting of a human backflip, we would need a series of data points (frames) to predict the action correctly. Otherwise, we may predict individual frames as jumping or falling.

That is why there exist some contexts where we need models specific for video. These are more complex and may need more data than their image counterparts. In the next sections, we will cover the evolution and state of art of Deep Learning video models for action recognition as well as some of the most important existing datasets.

2.1.1 Action Recognition video models

The first approaches for video focused on hand-crafted features formed from sparsely or densely sampled trajectories. A popular approach was Improved Dense Trajectories (iDT), which consisted of extracting trajectories and features for a dense set of interest points, encoding them in a fixed-sized video description, and then train an SVM with the bag of words representation. The problem with this approach is that it was quite limited and it needed to apply preprocessing for each frame.

After 2014, deep learning architectures started to prevail as state of the art on video action recognition datasets. The first approaches explored how to fuse temporal data with a single stream 2D convolutional neural net. They tested the following architectures:

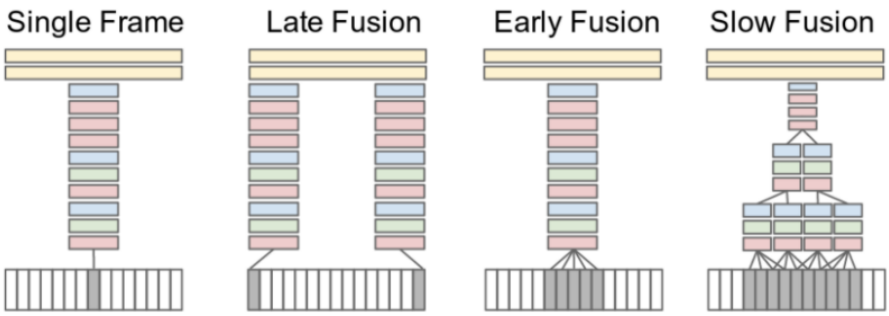


Figure 2: Single stream network architectures. Figure extracted from [5]

- Single Frame: It is basically an image classification network with no temporal features. It is the base for applying video rolling average.
- Late Fusion: A Fusion layer is used to merge the output of separate networks that operate on temporally distant frames.
- Early Fusion: It is the opposite of late fusion. The temporal dimension and the channel (RGB) dimension of the video are fused at the start before passing it to the model. It learns video descriptors via 2D Convolution on the entire stack of frames.
- Slow Fusion: Attempts to concatenate features hierarchically from a stack of frames, so as the network gets deeper more temporal features are learned.

The strength of these single-stream strategies is that we can use transfer learning from models trained on large-scale image datasets. Moreover, since they use only RGB image data it is not necessary to preprocess the images to compute the optical flow. Hence, they could be used for real-time processing. The big problem with these architectures is that they failed to improve significantly the accuracy from the single-frame approach. Their main weakness was that they didn't capture correctly motion features.

More or less at the same time, two-stream architectures, which use two different data modalities (RGB and optical flows) that run through two parallel streams of convolutional networks, appeared as a more precise alternative.

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene. Most of the time is computed as a shift between two gray images. Two types of optical flows exist: sparse and dense. Sparse optical flows track distinct pixels among the images using feature selectors and compute their displacement. On the other hand, dense optical flows are a per-pixel-motion estimate method. Optical flows are effectively used in motion tracking applications and therefore were appropriate for capturing temporal information.

In the two-stream architecture, the first one is called spatial stream. It requires a single RGB frame that is passed through some CNN kernels that capture its spatial information to then make a prediction. The other stream is called Temporal and it takes as input a stack of the frame's adjacent optical flows merged with the early fusion technique, that is passed through its own 2D convolution network. The final prediction is obtained by averaging the

probabilities of the separate streams.

The upside of this approach is that it matched the results obtained by other state-of-the-art methods like iDT. The main drawback is that it is not end-to-end trainable. Optical flows need to be calculated separately and require to be pre-computed, so it is less suitable for real-time. Also, both streams need to be trained separately (the spatial with images datasets and the temporal with videos) so transfer learning is less applicable.

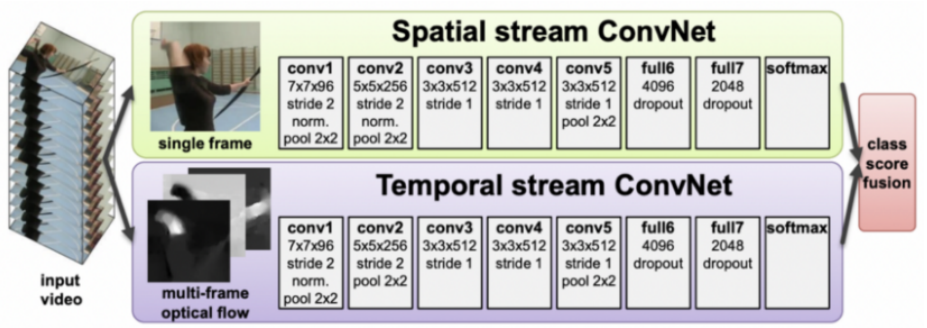


Figure 3: Two stream network architectures. Figure extracted from [5]

These original studies opened the door for further research into deep learning for video classification. The architectures developed in the last years are generally a variation of the following:

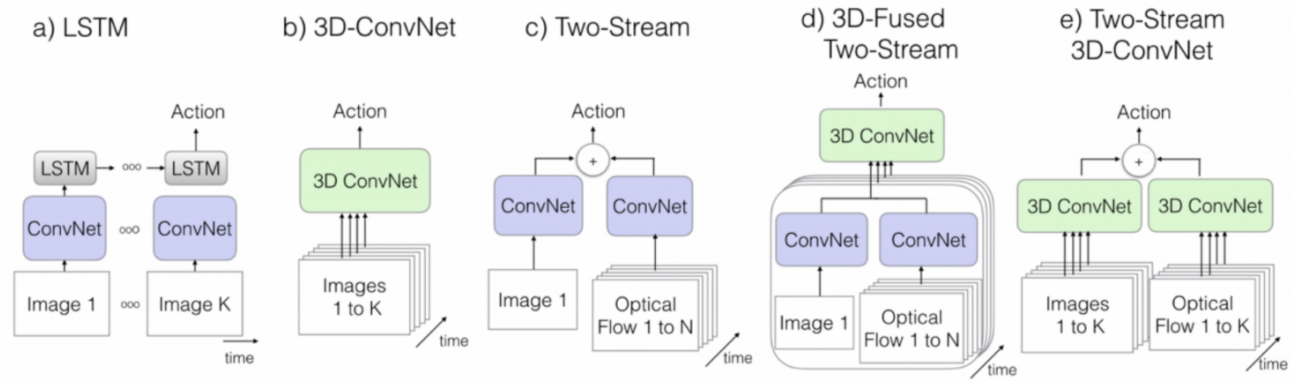


Figure 4: Modern video action recognition architectures. K refers to all the frames in a video. N stands for a subset of neighbouring frames. Figure extracted from [5]

- LSTM: In this approach, CNN's are used to extract local features of each frame. These independent outputs are then fed to a many-to-one multilayer LSTM network to fuse this extracted information temporarily.
- 3D-ConvNet: The introduction of 3-dimensional convolutional networks supposed a new step forward. This method fuses the temporal and spatial information slowly at each CNN layer throughout the entire network with just the RGB video input. They

are like the before mentioned Slow Fusion approach, but with 3D convolutions instead of 2D.

- Two-Stream: It has two independent streams of 2D CNN's. One for individual RGB images and another for multi-frame optical flow. The output of each stream is averaged to obtain the final result.
- 3D-Fused Two-Streams: The base is the same as for the Two-Stream architecture, but the outputs obtained for several frames are then passed through a 3D CNN.
- Two-Stream 3D-ConvNet: It is very similar to the base Two-Stream, but it uses 3D ConvNets instead of 2D for both the image and the optical flow streams.

The strength of approaches a and b is that they are end-to-end trainable and real-time capable, unlike c, d and e that require optical flow calculations over the raw data. Furthermore, b, d and e use 3D convolutions. This tremendously increases the computational and memory requirements and significantly affects their training cost.

The latest video architectures are improvements over the bases shown at Fig. 4. We have reached a point where we have moved beyond optical flow, and we instead architect networks that can natively learn temporal embeddings and are end-to-end trainable. Architectures a few years old like I3D or R(2+1)D already obtained great accuracy in several baseline datasets. Introduced a bit later, Slowfast networks [6] [7] outperformed by a considerable margin other approaches in all the reference datasets and became up to this day, the architecture with the greatest accuracy overall.

Slowfast networks work with two parallel streams. The first stream, called the slow pathway is a Spatio-temporal residual network that operates on a temporarily low-resolution video (few frames of the action) and is oriented to capture semantic information. On the other hand, the fast branch has low channels and operates with a higher temporal frame rate of the same video that is more oriented towards capturing motion information. Both streams are connected to merge the information from the fast branch to the slow branch at multiple stages. The slow branch obtains a reasonably good accuracy as a standalone architecture, but its accuracy gets boosted when its complemented by the fast pathway. On the other hand, the fast branch by itself obtains much poorer results. They obtain very good results at the expense of having a larger number of parameters. However, despite its high temporal rate, this pathway is made very lightweight, taking only around 20% of the total computation cost.

The most recent architectures, CSN and X3D, don't achieve an accuracy as high as Slowfast networks, but they are more lightweight approaches, especially X3D.

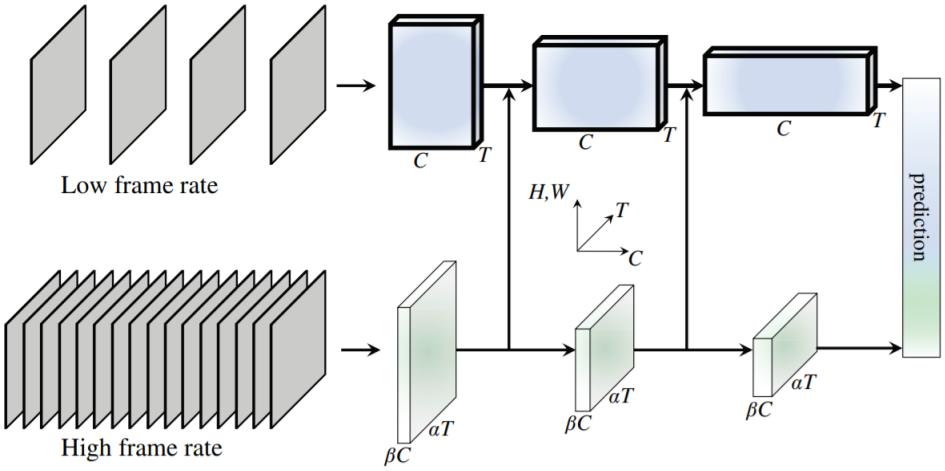


Figure 5: Slowfast network architecture. The slow branch is the one on the top, while the fast one is on the bottom. Figure extracted from [6]

2.1.2 Action Recognition video datasets

No element is more essential in deep learning than quality training data. There are many publicly available datasets and video labeled data is not an exception. Some of them, like Sports-1M or Kinetics, are incredibly large datasets that are often used as a benchmark or for pre-training purposes. Sports-1M is a dataset with more than 1 million video URLs from YouTube which have been annotated with 487 Sports labels. On the other hand, Kinetics contains up to 650,000 URL links of video clips that cover 400/600/700 human action classes, depending on the dataset version. Each action class has at least 400/600/700 video clips that last around 10 seconds.

There also exist many egocentric video datasets but not so many as for third person and generally not as large. One of the most relevant is Epic Kitchens [8]. It is a dataset recorded by researchers of the University of Bristol. In 2018 they released Epic Kitchens 55 and in 2020 Epic Kitchens 100.

Epic Kitchens is a large-scale egocentric video benchmark where 32 participants belonging to 10 different nationalities recorded themselves while cooking or doing other kitchen activities. It is especially interesting since the videos were non-scripted and therefore are more natural. On the other hand, this produces a heavy class imbalance.

In Epic Kitchens 55, they recorded 55h of video, which consisted of 11.5M frames. A total of 39.6K action segments were labeled, while 454.3K object bounding boxes were highlighted. They set some challenges in action recognition, action anticipation and object detection. In Epic Kitchens 100, they extended the total footage to reach 100h of video. This added up to 20M frames and 90K action segments

The dataset is quite challenging since a kitchen can be a very diverse and messy environment with heavy class imbalance, a lot of objects together and many occlusions. The labels of the actions to be predicted consist of a verb plus a noun, having in total 97 verb classes and 300 noun classes. A sample of the difficulty of the task is that the candidates that obtained the

best results in the 2020 challenge obtained an accuracy of 70.41%, 52.85% and 42.57% for the verb, the noun and the action (verb + noun) respectively. The average precision and recall for the actions were 24.94% and 26.93%, which demonstrates the difficulty to predict the less frequent classes.

2.2 Grasping

The hands are one of the most important parts of the human body. They are an incredibly multi-functional tool and the analysis of its diverse functionality is a very interesting field of study for understanding human manipulation behaviors as well as motivating design choices for objects or for mechanical hands intending to replicate its abilities such as robotic or prosthetic hands.

In our case, we focus on the concept of grasping, which is commonly defined as every hand posture used for holding an object stably during hand manipulation tasks.

The grasping literature is quite large and varied due to its multiple utilities. Next, we will explain some of the most popular research areas.

2.2.1 Taxonomy research

Understanding the way that the human grasps objects and more precisely knowing the kinematic implications and limitations associated with each grasp as well as knowing common use patterns is important for very different domains. These are medicine and rehabilitation, psychology, product design (e.g. grip of objects) and robotics, among many others.

That is why one of the main research goals is the definition of an exhaustive taxonomy that covers and classifies the most frequent human grasps. This is a challenging task because of the complexity and variety of uses of the human hand that makes it really difficult to cover the whole spectrum or to avoid ambiguity.

The hand has 15 joints, resulting in more than 20 degrees of freedom, which makes it difficult to directly model hand shapes. Moreover, its movement and function is not only a product of its internal degrees of freedom, but also the movement of the body and the arms, and most importantly the contact with the environment. However, the combination of ways in which the hand interacts with grasped objects is much more limited and can be effectively sub-classified.

The problem some years ago was that there were many different taxonomies defined, influenced partly by its underlying field of study (robotics, medicine, biomechanics...). Nevertheless, it didn't exist one that was a big figure of consensus.

Luckily, this has changed considerably with the taxonomy described by Feix et al. in 2009 [9]. It is also often referred to as the GRASP taxonomy because of the name of the funding project. They compared 22 existent grasp taxonomies in the literature, then they found the largest set of distinct grasps, and finally, they synthesize them into a single classification with 33 different grasps.

Some key concepts are necessary to understand the classification. These are the PIP classification, the Opposition type, the Virtual finger and the Thumb position.

PIP is a shortening of Power Intermediate and Precision, which is a way to classify the grasps depending on its need for precision or power to be properly executed. In a power grip, there is a rigid relation between object and hand, which means that all movements of the object are evoked by the arm. Usually, the palm of the hand is used. In contrast, in precision handling the hand is able to perform intrinsic movements on the objects without having to move the arm. The third category, the intermediate class, incorporates grasps that have elements of precision and power in a similar proportion.

Opposition type refers to the three basic directions relative to the hand coordinate frame, in which the hand can apply forces on the object to hold it securely. They differ in terms of the force direction that is applied between the hand and object.

- Pad Opposition (Fig. 6.a): occurs between hand surfaces along a direction generally parallel to the palm. Examples include holding a needle or a small ball. Is the X-axis in Fig. 6.d.
- Palm Opposition (Fig. 6.b): occurs between hand surfaces along a direction generally perpendicular to the palm. Examples include grasping a large hammer or screwdriver. Z-axis in Fig. 6.d.
- Side Opposition (Fig. 6.c): occurs between hand surfaces along a direction generally transverse to the palm. Examples are holding a key between the volar surface of the thumb and the radial sides of the fingers or holding a cigarette between the sides of the fingers. Y-axis inset in Fig. 6.d.

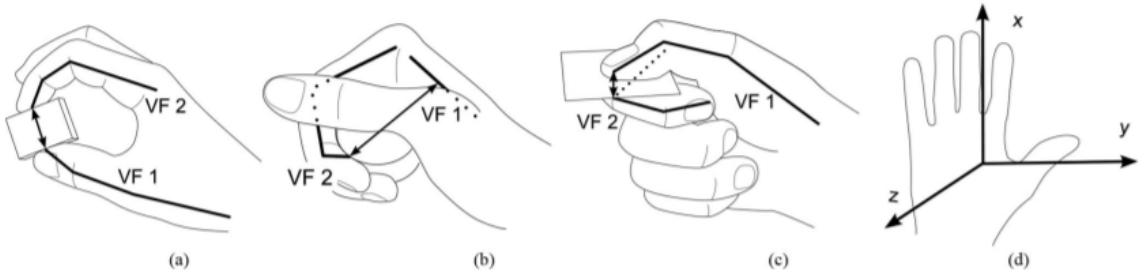


Figure 6: Opposition types along its virtual fingers. (a) Pad Opposition. (b) Palm Opposition. (c) Side Opposition. (d) Hand Coordinate System. Figure extracted from [9]

A **Virtual Finger (VF)** is a concept that applies when several fingers work together as a functional unit. Fingers belong to the same virtual finger if they apply forces in a similar direction and act in unison.

Depending on the grasp type, one or more fingers or hand parts can be assigned to one VF. The VFs oppose each other in the grasp, as would be the case for a simple gripper or vice.

The last concept used in the taxonomy is the thumb position. The **thumb** can be either **abducted** or **adducted**. In an abducted position the thumb can oppose the fingertips. On

the other hand, adducted position allows to apply forces on the side of the fingers or to push forward with the thumb.

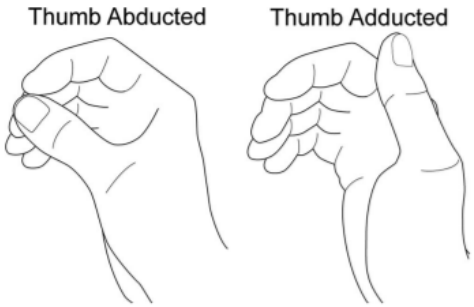


Figure 7: Thumb position. Figure extracted from [9]

The Feix taxonomy is quite exhaustive but has some casuistry that is not included. In particular, they defined a grasp as: “every static hand posture with which an object can be held securely with one hand, irrespective of the hand orientation.”

These cases studied fall into the static and prehensile grasp class [10]. Prehensile means that there is more than a single contact point between the hand and object ("virtual finger") and the contact forces from the hand alone can stabilize the object without the need for external forces such as gravity or the ground. Static refers to that the object is in fixed relation with the hand. This excludes in-hand motion like reorientation of the objects. Other grasps excluded from this definition are two-handed grasps or non-prehensile grasps, like those that depend on gravity (e.g. Holding a book with the palm open).

The Feix taxonomy of 33 grasps is present in Fig. 8. Apart from the original grasps of the taxonomy, two more are depicted, being them number 34 (lift) and 35 (push).

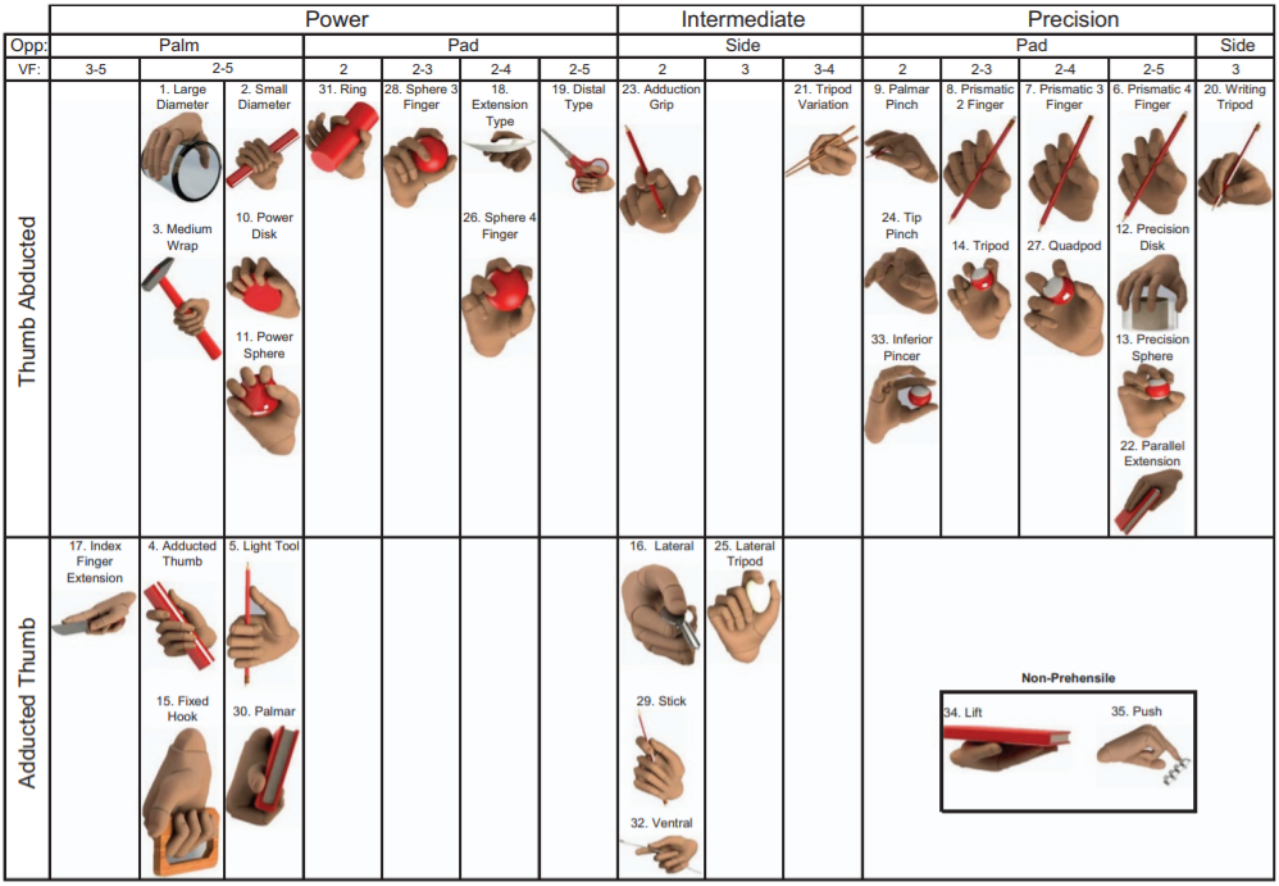


Figure 8: Feix Grasp taxonomy. Figure extracted from [11]

The Feix taxonomy has become the standard for many of the literature's subsequent publications. Since our work is mainly based on it we will describe it in section 4 with greater details.

2.2.2 Egocentric Human Grasping Video Datasets

The human grasping video datasets are generally recorded from a first-person perspective with a camera located on the head. Most of them are captured in unstructured environments, while people do their day-to-day chores. This makes them very useful to determine the most used grasps and other statistical properties. They are also suitable to perform grasp recognition from either images or video.

There are few datasets like that available. That is because most of the grasping research is related to robot grasping. Some of these datasets were recorded specifically for grasping and others are adapted. They require to be manually labeled by some experts. It is not an easy task due to the existent similarities between some grasps and the frequent hand-object occlusions.

The most relevant grasping dataset for our environment is the **Yale human grasping**

dataset [12] [13]. They used a head-mounted camera to record in an unstructured environment the actions of 2 housekeepers and 2 machinists while they were working. The full dataset contains 27.7 hours of tagged video and represents a wide range of manipulative behaviors spanning much of the typical human hand usage. The authors tagged the grasps with the Feix taxonomy and only those done with the right hand. The camera was oriented downwards to mainly capture their hands and interactions with the environment. The videos were recorded at 25 FPS, they had a 640×480 resolution and a field view of approximately 140º.

Figure 9: Camera setup and example images from the Yale Grasp Dataset. Figure extracted from [12]

The authors provided the original videos and a spreadsheet with the grasp labeling. For each instance of grasp in the video, the data is tagged with grasp type, time information, properties of the object including size, shape, stiffness, and mass parameters, and task properties including force, movement constraints, and general class parameters.

Due to the complexity of the tagging task, there are different subsets of labeled grasps available. This translates to having in total 18,210 labeled grasps. If we are interested only in right-handed grasps labeled under one of the 33 categories of the Feix taxonomy (like our case), the number is reduced to 11,539. If we are interested in knowing also the properties of the object or the task, the number gets further reduced.

This dataset has been used as a base for several subsequent studies and up to this day is probably the most complete video human grasp database in first-person. However, it has several drawbacks for our purposes. The image quality of the videos is a bit poor for today's standards and they have the time information on the upper-right part, which is not useful. Also, some of the tagged grasps do not include video or there are blackened frames because of privacy concern issues. This reduces a bit the amount of useful labeled grasps. Even with these flaws, the Yale human grasping dataset is the one that we are going to mainly use.

The **dataset** from the **Nazarbayev University** in Kazakhstan [11] [14], is one of the most interesting recent additions for our purposes. They collected a dataset with three different modalities: color images from a head-mounted action camera (GoPro Hero 4), distance data from a depth sensor (SoftKinetic DS325 RGB-Depth camera) on the dominant (right) arm

and upper body kinematic data acquired from an inertial Xsens MVN motion capture suit. They captured 9 hours of video while doing three different tasks, being food preparation, housekeeping and laundry. They identified up to 3826 grasps that were grouped according to the Feix taxonomy with the addition of two non-prehensile grasps (push and lift). Then they studied the statistical properties of the grasps in terms of duration and frequencies.

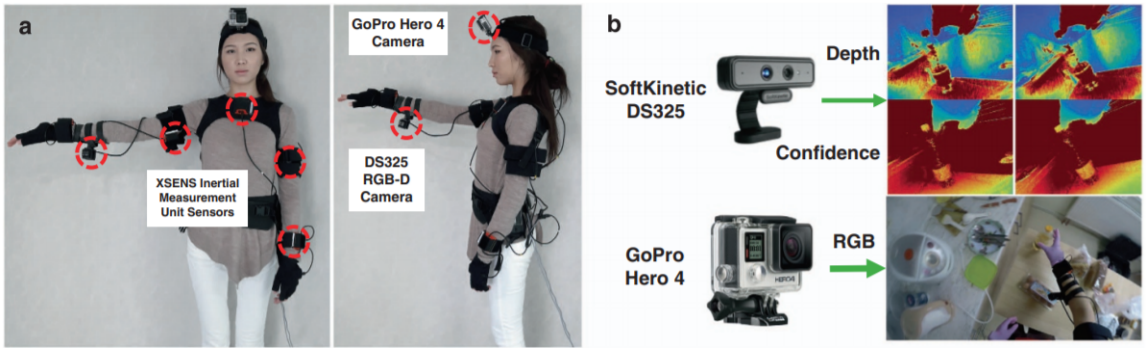


Figure 10: Kazakh grasp dataset camera setup and example snapshots from RGB and Depth sensors. Figure extracted from [11]

This dataset is quite interesting due to the 3 modalities of data that they capture. Also, the image quality is better compared with the Yale dataset. It has two main downsides. The first is its lower number of grasps. The second is that they wear pink gloves to protect the kinematic sensors. This differentiates from our setup, where we won't use gloves, and makes it difficult to use hand detectors.

The University of Tokyo Grasp Dataset (**UT grasp dataset**) is another existent egocentric human grasp video dataset [15] [16]. Videos were recorded by a head-mounted camera (GoPro Hero2) at 30 fps and downsized to 960×540 pixels per frame. It was captured in a controlled laboratory environment, so it doesn't have value for statistical analysis. The authors focused on a set of 17 distinct grasp types from the Feix taxonomy, that were selected based on their statistical prevalence. 5 Subjects were asked to just grasp different objects placed on a desktop. The objects belong to five unique sets which are commonly used in different tasks (cleaning, cooking, office work, bench work and entertainment). In each video recording, the subject performed all 17 grasp types on one object set. The same subject did the same graspings with the same objects in two trials. In total, 50 videos were recorded, each one being close to 5 minutes. This adds to over 4 hours of video.

The **GUN-71** (from Grasp UNderstanding Dataset) is another existent egocentric grasp dataset. They captured a balanced human grasp dataset with an Intel Senz3D located on the chest with a GoPro harness. For the labeling of the grasps, they used a fine-grained taxonomy with 71 different grasps. They captured in total 12,000 RGB-D images (they don't share the full video sequences), with 28 objects per grasp. For each hand-object configuration, they took between 5 or 6 frames, hence representing some steps of the action and the different 3D locations and orientations with respect to the camera.

2.2.3 Statistical properties of grasping

With a clear taxonomy that acts as a base to classify the different grasps and some captured data (mainly video), many papers just expand on studying certain grasp statistical aspects. These are very useful to understand human behavior and to translate it to robotics or prosthetics.

One of them [17], uses a subset of the Yale Human Grasping dataset for finding the most used grasps in the household and machine shop environments. From the results, it can be seen that only a small number of grasp types comprise the majority of those used. For the housemaid, nearly 80% of the time was spent utilizing six grasp types: medium wrap, index finger extension, power sphere, lateral pinch, precision disk, and thumb-index finger. On the other hand, nine grasps consumed nearly 80% of the machinist's grasping time: lateral pinch, light tool, tripod, medium wrap, thumb-3, thumb-4, index finger extension, thumb-2, and thumb-index.

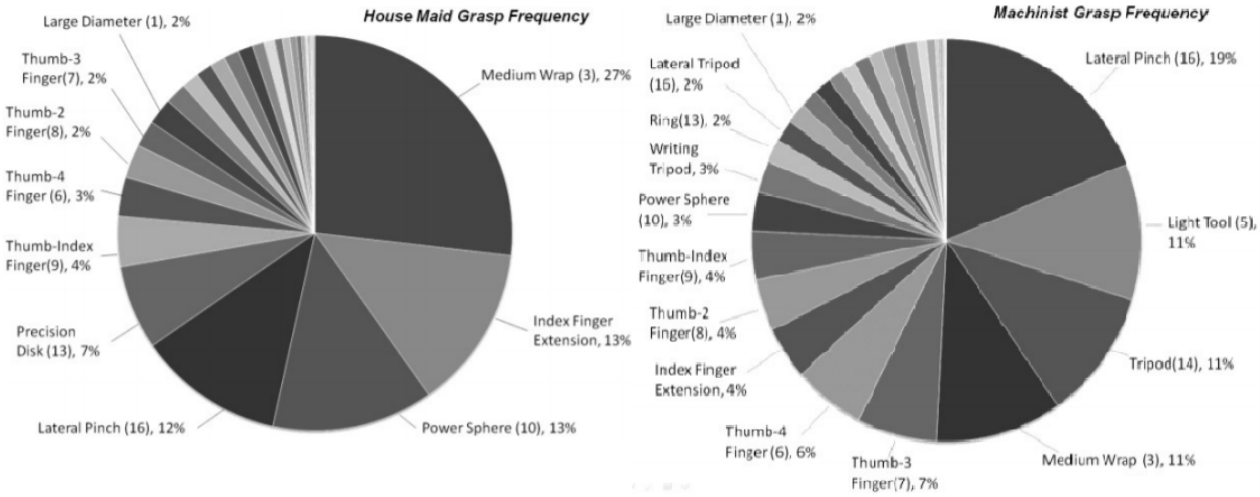


Figure 11: Grasp frequency results for the maid and the machinist. Figure adapted from [17]

The papers [18] [19] also work with the Yale Human Grasping Dataset. In [18], the authors study the properties of the objects and correlate them with the grasps. A better appreciation of the types and properties of objects that humans commonly manipulate is important in many domains. In hand rehabilitation, it can be used to focus on grasping the objects with the most typical shapes, sizes and masses. Those insights could also influence the design of prosthetic or robotic hands.

The authors assign to each object properties from a set of seven classes, including mass, shape and size of the grasp location, grasped dimension, rigidity, and roundness. They draw some interesting conclusions:

- A big part of the objects cannot physically be grasped from their largest dimension (55% of objects have a dimension larger than 15 cm).

- 92% percent of objects had a mass of 500 g or less, which means that a high payload capacity is not so important for having a large subset of human grasps.
- 96% of grasps had a width equal or smaller than 7 cm, which could be a requirement for hand rehabilitation or the aperture size of a robotic hand.
- In 94% of the cases, the subjects grasped the smallest overall major dimension of the object. This could be a default behavior for a grasp planner.

The authors also examined the smaller subset of grasp types that can be used to grasp the most objects, given the fact that many grasps can be used for effectively the same purpose. From a subset of 1 to 5, these are Medium wrap, Lateral pinch, Thumb-2-finger, Power sphere and Tripod.

The results showed the significant differences between the objects handled by housekeepers (often squeezable and floppy) and those used by the machinist (always rigid). However, there were also some common characteristics, like the major proportion of cylindrical objects grasped along the circumference.

On the other hand, [19] focuses on correlating the task being performed with the grasp types and the object attributes. The task is classified according to the force required, the degrees of freedom, and the functional task type. The authors found some interesting insights:

- 46% of the tasks are constrained (the manipulated object is not allowed to move in a full 6 degrees of freedom). This emphasizes that many real-world human tasks are not as simple as object transport.
- The authors used decision trees to try to predict the grasp type from its several characteristics (object and task). The results show that the best predictors are the object size, task constraints, and object mass. The grasp type can be predicted with a 47 % accuracy. Those parameters could make useful heuristics for grasp planning systems.
- Further results suggest that the common sub-categorization of grasps into power, intermediate, and precision categories may not be appropriate since it couldn't be effectively predicted with the same previous predictors. Grasps are generally more multi-functional than previously thought, especially the power ones, that are also practical for grasping small and lightweight objects.

The paper [20] focuses on the choice of grasp type and location when handing over an object in two different settings. In the first, one participant (passer) was asked to grasp the objects from a table and perform two different tasks on them. In the second session, the passer was asked to hand the objects over to a partner (receiver) who subsequently performed the same two tasks the passer had performed in the previous setting. They noted the changes in the grasp types and locations that naturally we do to accommodate the handover to the receiver.

2.2.4 Grasp affordance

Grasp affordance is a problem that consists of given an object or group of them, detect and suggest the possible grasps to grab the object(s). This can be solved in two different ways. The simplest version is to just predict the grasp contact area box and its orientation. The

most advanced approaches predict a complete hand mesh model with the grasp pose. Grasp affordance is clearly the most important research field about grasping and its applications are thought mainly to be applied for robotics or prosthetics. Robotic grasping of household objects has made remarkable progress in recent years despite the challenge of conformation of the hand with the surface of the object in a semantically and physically plausible manner.

Work like [21] is an example of models that use Deep Neural networks for predicting the grasp contact area box from a single image. For training, they use datasets like the Cornell grasp dataset. Its extended version comprises 1035 RGB-D images with a resolution of 640×480 pixels of 240 different real objects with 5110 positive and 2909 negative grasps. The annotated ground truth consists of several grasp rectangles representing grasping possibilities per object.

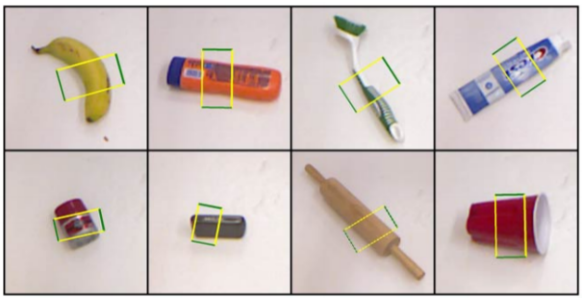


Figure 12: Examples of correct grasps boxes from a regression model. Figure extracted from [21]

There is also plenty of papers that deal with the more complex problem of predicting the 3D hand grasping mesh. That is the case of [22]. The authors focused on predicting how a human would grasp one or several objects, given a single RGB image of these objects. For that, they released a large-scale dataset of manually annotated grasps on the 58 objects of the YCB Benchmark Set. For each grasp, they annotated the hand position, the hand pose and the grasp type according to the Feix taxonomy. Then, they trained GanHand, a model that takes a single RGB image of one or several objects and predicts how a human would grasp these objects naturally having as output a 3D hand mesh. This kind of work has considerable potential in fields like augmented reality, robotics or prosthetic design.

Figure 13: GanHand predicts hand shape and pose for grasping multiple objects given a single RGB image. The figure shows sample results on the YCB-Affordance dataset. Figure extract from [22]

One of the most complete and promising research is the GRAB dataset [23] (GRasping Actions with Bodies), a dataset of full-body grasps. It contains the full 3D shape and pose sequences of 10 subjects interacting with 51 everyday objects of varying shape and size. Given MoCap markers, they fit the full 3D body shape and pose, including the articulated face and hands, as well as the 3D object pose. This gives detailed 3D meshes over time, from which they infer contact between the body and object. They illustrate the utility of this dataset with one particular application. They trained GrabNet, a conditional generative network, to suggest 3D hand grasps for unseen 3D object shapes.

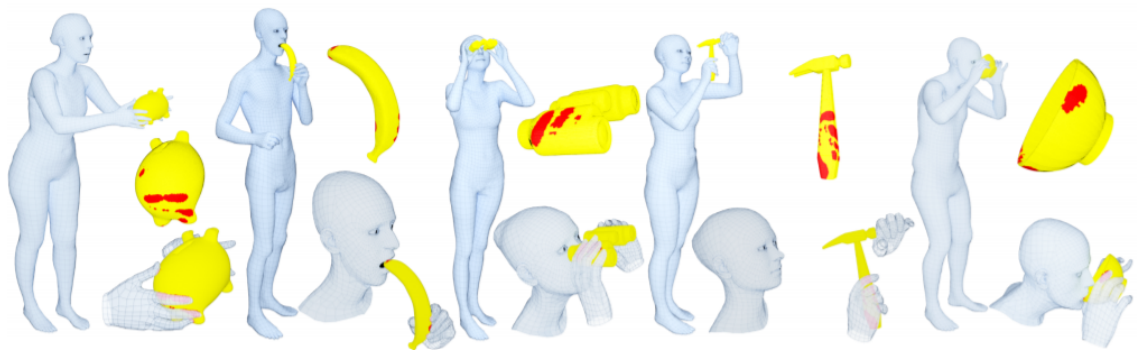


Figure 14: Example of "whole-body grasps" with contact maps from the GRAB dataset. Figure extract from [23]

Something quite different is [24]. The authors worked on the idea of suggesting grasps (only the label) for prosthetic hands that had a camera on them. They wore at the same time one camera in the head and another in the right hand. They captured paired images of several objects from those 2 different perspectives, eye-view and hand-view. From the eye-view photos they ranked which were the most suitable grasps among a set of 5 possible. The idea was to train a CNN to predict from the hand-view images the labels for the eye-view images.

2.2.5 Grasp contact

Another field of study with quite a momentum is based on studying the contact surfaces when grasping.

One of the most representative works in that direction is ContactDB [25]. It is a database based on the idea of using thermal cameras to estimate precisely the most important contact points during a grasp. This is possible because when a participant grasps an object, heat from the hand transfers onto the object's surface, which can be captured afterwards. The intensity in those thermal images is related to the heat of the skin, duration of the contact, heat conduction and contact pressure. They recorded functional human grasping for a set of 50 household objects in 2 different setups, use and handoff. The dataset consists of 3750 meshes textured with contact maps and 375K frames of paired RGBD-thermal data. The grasped objects are 3D printed PLA representations to ensure uniform heat dissipation

properties. To finish, they trained image translation and 3D convolutional algorithms to predict diverse contact patterns from object shape.

Another example is the paper called the Grasping Field [26]. The authors proposed an expressive representation for human grasp modeling to effectively represent the contact between hand and object. This novel interaction representation is based on regressing a continuous function that they call Grasping Field. The main insight is that every point in a three-dimensional space can be characterized by the signed distances to the surface of the hand and the object, respectively. Then, the hand, the object, and the contact area can be represented by implicit surfaces in a common space, in which the proximity between them can be modeled explicitly. They further utilize a deep neural network to parameterize the grasping field and learn it from data, which serves as a powerful representation to facilitate hand-object interaction modeling.

2.2.6 Hand-pose estimation: A related problem

Hand pose estimation is the process of modeling the human hand as a set of some parts (e.g. palm and fingers) and finding their positions in a hand image (2D estimation) or the simulation of hand parts positions in a 3D space [27]. The hands are modeled as a number of joints and the task is equivalent to finding their position. In the latest times, the hand model of 21 joint has become the most popular.

Hand pose estimation is not strictly a grasping problem, but in a way, it can be considered a similar idea. After all, grasping could be considered as the hand pose when grabbing an object. Therefore we should consider studying some of its literature. Mainly the one specialized for a first-person setup.

One popular tool that is open and available to use is the hand pose detector from Google MediaPipe [28]. It is a high-fidelity hand and finger tracking solution that employs machine learning to infer 21 3D landmarks of a hand from just a single frame.

Contact pose [29] is the first dataset that includes hand-object contact paired with hand pose, object pose and RGB-D images. ContactPose has 2306 unique grasps of 25 household objects grasped with 2 functional intents by 50 participants, and more than 2.9 M RGB-D grasp images. The dataset is thought to improve the modeling of hand-object contact, which is a difficult task since it involves complex hand configurations and soft tissue deformation that may result in complicated regions of contact. It is similar to ContactDB dataset [25], but instead of just including contact maps and turntable RGB-D images, which are often not enough to fully interpret the grasp, it includes 3D joint locations and multi-view RGB-D grasp images that allows to better associate the contacted areas to the hand parts.

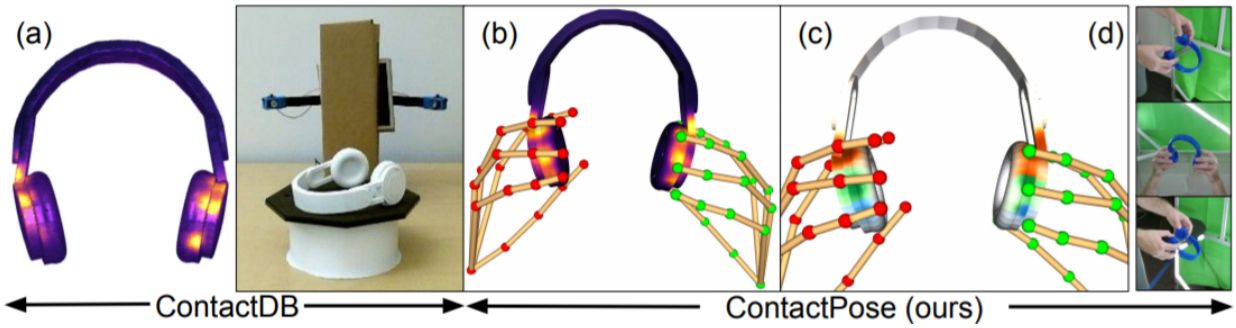


Figure 15: Comparison between ContactDB (Contact maps+RGB-D) and ContactPose (Contact maps+RGB-D from 3 viewpoints+3D hand joints and object pose). Figure extracted from [29]

The most related hand pose papers to our problem are the ones that estimate the hand pose in a first-person scenario when grabbing the objects. In [30], the authors study the use of 3D hand poses to recognize first-person dynamic hand actions interacting with 3D objects. They collected a dataset with RGB-D video sequences that comprise more than 100K frames of 45 daily hand action categories and 26 different objects in several hand configurations. To obtain the hand pose annotations, they used a mo-cap system that automatically infers the 3D location of each of the 21 joints of a hand model via 6 magnetic sensors and inverse kinematics.

Then they experimentally evaluated several RGB-D and pose-based action recognition approaches. From the results, they saw clear benefits of using hand pose as a cue for action recognition compared to other data modalities. The models that used the hand sensor data outperformed the RGB and depth models. When they swapped the input for the one generated by the model that estimates the 3D hand pose the accuracy dropped, but it still was remarkable (78.73% vs 72.06%).

Similarly, on the paper [31] the authors propose a method that allows to automatically annotate with accurate estimates the 3D poses of both hand and object from input images despite the large mutual occlusions. To deal with this challenge, they capture sequences with one or several RGB-D cameras and jointly optimize the 3D hand and object poses over all frames simultaneously. With this method, they created HO-3D, the first markerless dataset of color images with 3D annotations for both the hand and object. Using this dataset they train a model that with a single RGB image can predict the hand and object pose under severe occlusions and that generalizes to objects not seen in the dataset.

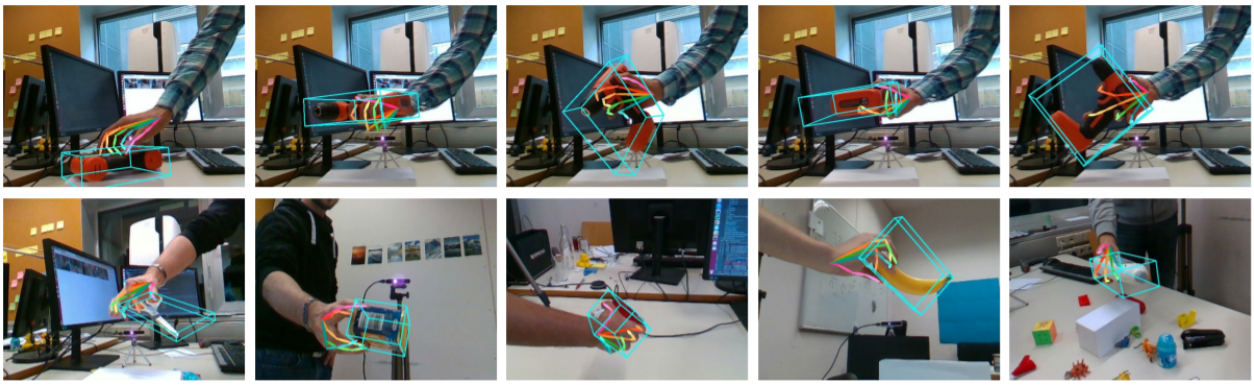


Figure 16: Proposed HO-3D dataset. Figure extracted from [31]

Another related task is hand detection. In contrast to hand pose estimation, the problem only requires detecting the hands on the image and enclose them with a bounding box. One of the most interesting works that is compatible with an egocentric environment is [32]. They introduced a model that identifies for every single hand in a single RGB image: a hand box; its side (left/right); its contact state (none/self/other person/ non-portable object/portable object); and, for the hand in contact, an object box around the object or person in contact. A significant part of their training data is from a first-person perspective.

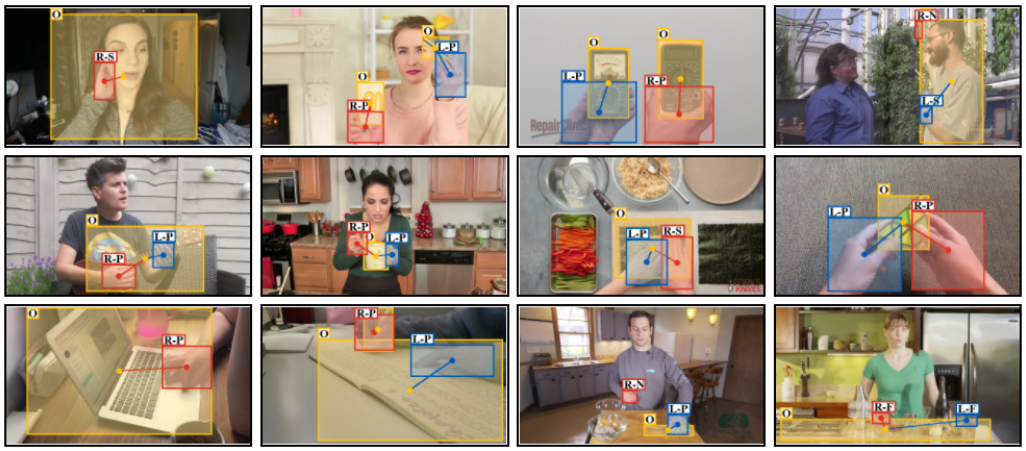


Figure 17: Examples of detected hands and objects. Figure extracted from [32]

The authors on [33] worked also on hand detection. Their objective is the definition of an effective and efficient method for detecting hands in egocentric videos for rehabilitation applications. They captured with a head-mounted GoPro camera at 30 FPS with 1080p resolution a large egocentric dataset of individuals with spinal cord injury (SCI) performing a variety of 35 ADLs and manipulating over 30 objects. Those actions were performed in many different environments including the kitchen, washroom, living room, dining room, bedroom, and hallway. They tested several object detection and tracking algorithms or

combinations of them to detect the hands. Hand detection is an essential step before further analysis can be conducted, including hand segmentation, activity recognition, interaction detection, or grip posture analysis

2.2.7 Grasp recognition

Grasp recognition or grasp classification, is the task in which given some input grasp with a certain data modality (sensor data, image, video...) we predict to which class it belongs. Being able to classify between several different types of grasps with great accuracy can be very useful across different domains such as neuromuscular rehabilitation, robotic arm design and motor control analysis.

The traditional approaches for grasp analysis had been developed mostly in controlled environments that often included intrusive hand contact sensors (may inhibit free hand-object interactions) or calibrated cameras (require to record in a limited workspace). In contrast, wearable cameras suppose a great tool for a more natural, larger scale and less effort grasp analysis. The first-person view allows recording quite precisely the hand and object interactions, that are often in the center of the visual field. However, once the data has been recorded, it is quite tedious to manually visualize and tag each individual grasp. That is why it is so interesting to develop methods that can automatically recognize the different hand grasps.

The authors on [34] propose an egovision system that can recognize different hand grasp types and learn visual grasp structures automatically from large scale of data recorded with a wearable camera.

They use as input images from the UT grasp video dataset [16]. They use a subset of some of the 17 most used grasps of the Feix taxonomy. First, they crop the right hands from the images with a model that they train to do so. After, they extract several feature descriptors (HOG, HOG-PCA, HandHOG and BlockHOG-SIFT). Finally, they trained three types of one-versus-all multi-class grasp classifiers (SVM-linear, SVM-rbf and Exemplar SVM). They used 5-fold cross-validation. They compute the average F1 score from the weighted average of the F1 score of each grasp type. The best results were obtained for the E-SVM trained with the HOG, with an average F1 score of 0.89. They also tried the same approach with a more real-world scenario, using the Yale grasp dataset. In this case, the results are much worse, with an average F1 score of 0.42 for the HOG-PCA and SVM-rbf combination. The big performance gap between SVM-linear and SVM-rbf showed that hand grasps have a wide variance in pose and are therefore not linearly separable.

The same authors published [15] following their previous work. They extended the UT grasp dataset and compared the grasp recognition performance of their previously used methods with a 5-layers CNN and Dense Hand Trajectories (DHT). These two clearly outperformed SIFT and HoG.

They had three different setups for testing:

- Cross-trial: Train grasp classifiers for each subject and object set on one trial and test on another trial. CNN had the best results, with 92% of average accuracy.

- Cross-task: Train grasp classifiers on a set of objects and test on the rest. DHT-based feature achieves the best average accuracy with 76.4%. The drop in performance is reasonable since the different object appearance undermines the discrimination ability of appearance-based classifiers.
- Cross-User: Train grasp classifiers with one subject, test on the others. Best performance is achieved from CNN-based feature and DHT-based feature with an average accuracy of 73% and 72% respectively. Two important reasons can explain the performance degradation. One reason is the difference in the skin color and hand size of different users. The other is the difference in grasping styles even in doing the same grasp type.

Regarding the performance on the Yale Machinist Dataset, using 5-fold cross-validation the CNN obtained an accuracy of 49%, while DHT 59%. The authors believe that trajectory-based features (DHT) outperform appearance-based features (CNN, HoG) because hand motion information is also captured in trajectory-based features and also due to its higher robustness to unreliable hand detection.

The most interesting part of their study was that they computed the correlation index between all pairs of grasp types for the Machinist Grasp Dataset. Then they followed an iterative supervised hierarchical clustering algorithm to construct a dendrogram of grasp types by iteratively clustering the two most correlated grasp types after each iteration of supervised learning. This learned visual structure gives researchers the flexibility of finding a good balance between better performance and more detailed grasps analysis. From the analysis of the clustering, they observed that for the low levels the grasps were clustered consistently with known divisions of the expert-designed grasp taxonomies.

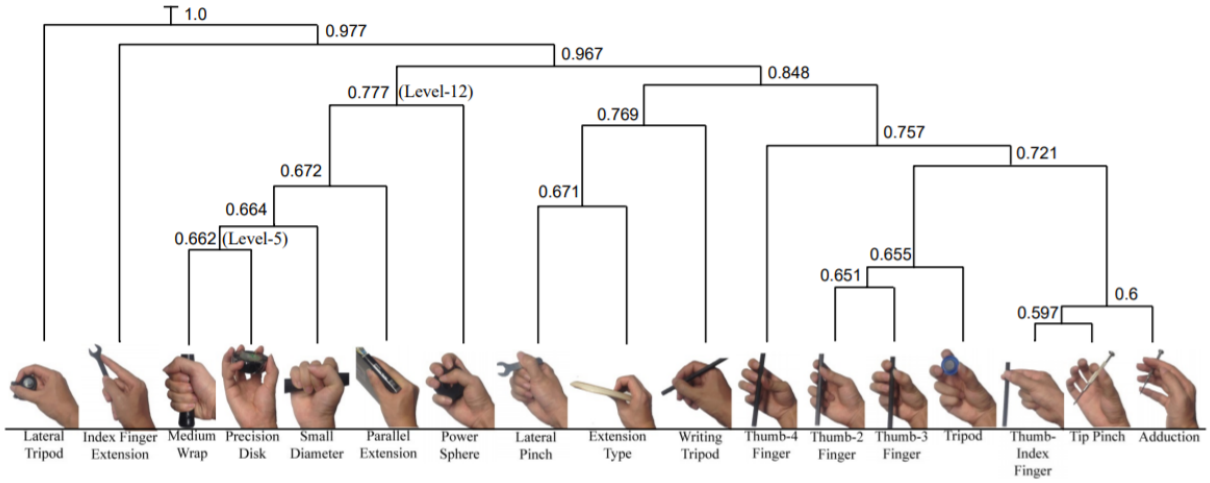


Figure 18: Automatically learned grasp dendrogram (taxonomy tree) for DHT. Classification accuracy obtained at different clustering levels are shown. Figure extracted from [15]

The increase in the average accuracy at the different levels of the dendrogram Fig. 18 is not constant. This is because new clustered groups can become more dissimilar so there is limited room for improvement of recognition performance.

The same authors also proposed a unified model to recognize grasp types, object attributes and actions from a single image [35]. Grasp types and object attributes are key elements for understanding hand manipulation. Grasp types determine the patterns of how a hand grasps an object, while object attributes indicate possible hand motion of the interactions. In addition, both concepts together characterize the manipulation actions.

The approach is composed of three components. The first is a visual recognition layer that recognizes hand grasp types and attributes of the manipulated objects. They train linear SVM classifiers for recognizing nine different grasp types selected from the Feix taxonomy. They use the GTEA Gaze Dataset. The hands are cropped following the same methodology as in [34]. They compare the performance of HoG and CNN, being the second far superior (50% vs 61.2% of accuracy for 9 grasp types). Regarding the object's appearance, they consider four classes (prismatic, round, flat and deformable). First, they train an SVM target regressor for predicting the relative location and scale of the grasped object based on hand appearance. After cropping the grasped part of the object, they train an SVM classifier for predicting the object properties. Again, the CNN method is superior, with an accuracy of 72.4%.

The second element is a Bayesian network that models the mutual context of grasp types and object attributes to boost the recognition of both. It is based on the idea that the information of one side facilitates the recognition of the other. Its usage improves in a 12.9% and 9.5% grasp and object attribute prediction respectively. Lastly, they add an action modeling layer based on the belief distribution of grasp types and object attributes.

The authors on [36] trained with the GUN-71 dataset (of 12,000 RGB-D images) a grasp recognition system. Its pipeline had 2 stages, first a depth-based hand segmentation and then the fine-grained classification. Once the hand is detected and segmented from the image, they use CNN's to extract off-the-shelf features from the entire RGB image, a cropped window around the detected hand and a segmented RGB image. They resize each window to a canonical size (of 224 x 224 pixels) before proceeding. The final concatenated descriptors of dimension 3096 are fed into a linear multi-class SVM for processing.

For all the experiments of this section, they use a leave-one-out approach. The results show that the best modality of image data is Cropped data, with a mean accuracy of 13.67 %. On the other hand, just using the segmented hand/object obtains an accuracy of 11.10%. This suggests that some amount of local context around the hand and object helps. Lastly, using the entire RGB image obtains an accuracy of 11.31 %. The best model is obtained with the concatenation of the 3 modalities, with an accuracy of 17.97%.

The authors observed that the easy cases tended to be characterized by limited variability in terms of viewpoint and to exhibit limited occlusions of the hand. On the other hand, misclassified classes often presented occlusions and similarities to other classes.

When they limited their taxonomy to the 17 grasps from [16] (i.e. evaluating only the subset of 17 classes), they obtained an accuracy of 20.53%. When they used the Feix taxonomy of 33 classes, they obtained an accuracy of 20.50%. This marginal improvement when evaluating grasps from smaller taxonomies suggests that the new classes are not much harder to recognize. Rather, the overall performance may be low because of the challenge present on

the dataset due to diverse subjects, scenes, and objects.

Similarly, the authors on [37] studied the usage of grasp classification as an intermediate step for hand pose estimation during the interaction with unknown objects. First of all, they produced a synthetic depth map grasp dataset. It consisted of 330K synthetic depth maps and included the 33 grasps of the Feix taxonomy. In their pipeline, they use first a ConvNet model to regress the confidence map for the hand and the object and obtain their centroids on the depth map and crop them. Then they use autoencoders to refine their synthetic depth images and give them a more realistic look. After, they input each crop to a separate CNN. The hand crop is sent to a hand-oriented network that focuses on the loss of hand information caused by occlusions due to the object. The object crop is used as input for the object-oriented network, that extracts potential pose information even from the unseen object. The two feature vectors are then fused into a decision network CNN that outputs two things: the grasp type and the global orientation. This information is then used for hand pose estimation.

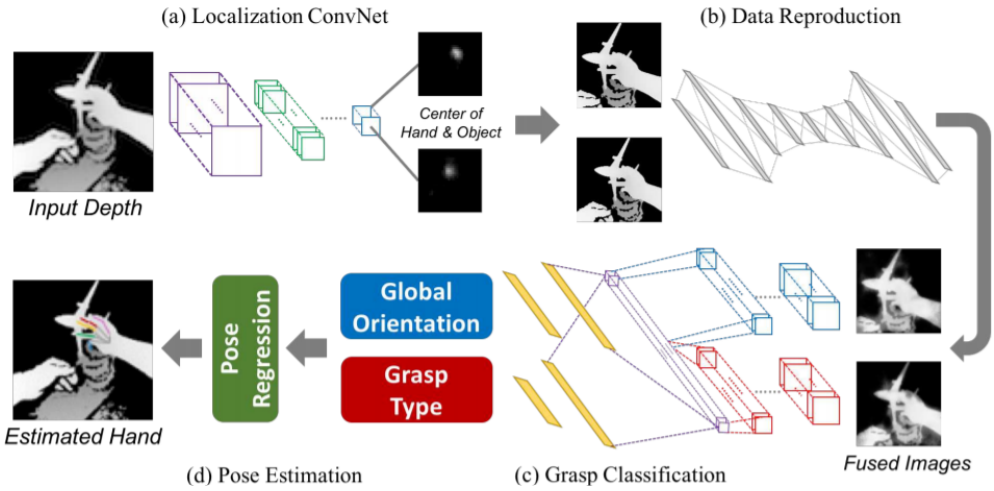


Figure 19: Pipeline for action recognition. Figure extracted from [37]

Regarding the grasp recognition, they obtained an accuracy of 55.56% on their dataset. When they used only the hand stream, they obtained a 43.87% while for the object only stream a 49.12%. The better results of the Object-only stream may indicate that the shape of the object infers better the hand grasp. To compare their approach with other baselines, they trained their model with their dataset and tested it with the depth maps from the GUN-71 dataset. They obtained a 41.00 % of accuracy (using only depth maps), which clearly outperforms the 20.50 % from the original authors [36] (which use depth maps to detect hands and then use CNN with RGB).

3 Dataset

3.1 Dataset Selection

To train our grasp recognition models we need to choose between the 4 available egocentric video grasp datasets. None of them is perfect, having each one its pros and cons.

Dataset	Modality	Position	Balanced	Quality	Grasps	Taxonomy
Yale [12]	RGB (V)	Head	No	640×480 25FPS	11,539	33
Kazakh [11]	RGBD+S (V)	Head	No	1280×720 30FPS	3,826	35
UT [16]	RGB (V)	Head	Yes	960×540 30FPS	1,000	17
GUN-71 [36]	RGBD (I)	Chest	Yes	640×480	2,000-2,400	71

Table 1: Egocentric Human Grasp Video Datasets information

In our setup we have a GoPro Camera fixed to the chest of the subject by a harness, in what is called an egocentric point of view. The only dataset that complies with this requirement is the GUN-71. However, this dataset has a big drawback, it doesn't include the whole video sequence of each grasping (gives only 5 or 6 RGB-D frames), and we want to treat each grasping as a sequence of frames rather than a single image. This automatically discards the GUN-71 dataset.

The other datasets are fairly similar in the sense that all of them capture RGB video frames from a camera located in the head and using a taxonomy that is derived from [9]. Our GoPro camera only captures RGB data so the depth and sensor modalities from the Kazakh dataset are not really useful. What really discards its use is the fact that they wear pink gloves to protect the sensors on the hands. This confronts the politics of the hospital of having a setup as least invasive as possible. It also may complicate the recognition of the hand with some of the existing hand detectors.

This leaves to choose between the UT and the Yale datasets. The UT is superior in video quality and its grasp classes are balanced because it was captured in a lab environment. This makes the scene cleaner and therefore easier to predict. The big problem is that is really small. On the other hand, the Yale dataset is captured in an unstructured environment, which produces a class unbalance and a messier environment that nevertheless represents better a real-world situation. Despite its lesser video quality and some inconsistencies in the labeling, it is far superior in the number of grasp sequences (11,539 vs 1,000). That is the main reason that we will opt to use the Yale Grasp Dataset to train our models.

This decision will have an implication for the future. We use a camera located on the chest, while the Yale dataset is captured from a head-mounted camera perspective. This is an important change because there is a big difference in terms of height and angle that makes vary notably the appearance of the scene, the hands and the objects. A model trained on the Yale dataset will perform much worse if used directly with data captured with our perspective. Therefore it will probably be necessary to further fine-tune the models with our own data.

Another important thing to keep in mind is that the Yale dataset only tagged the grasps with the dominant hand. Therefore, if we want to train a model that also works with left hands, we will need to flip the images.

3.2 Dataset Description

The Yale grasp dataset is composed of 179 videos, but we only have access to 136. They capture 2 machinists and 2 housekeepers while doing their job in a natural environment. The labels of each grasp are given in a CSV file, where each row represents a certain grasp instance. There are 25 columns, but not all of them are interesting for us:

- Video: Number of the video file from 1–179.
- Timestamp: Start timestamp of the grasp in the video file in hh:mm:ss format.
- Duration: Length of the grasp instance in seconds
- Subject: Participant profession and number. Machinist 1/2, Housekeeper 1/2.
- BlackRatio: Proportion of frames blacked out because of privacy concerns. Ratio between 0 (all visible) and 1 (all black).
- Grasp: Column that has the grasp label. It uses the Feix taxonomy. The label is either no-grasp or one of 33 grasp types.
- OppType: Opposition type of the grasp. It can be either Pad, Palm, Side or NG (no grasp)
- PIP: Power, Intermediate or Precision grasp. The label can be Power, Intermediate, Precision or NG (no grasp)

The columns that are not interesting for us refer to the object properties (Dimensions in cm, Rigidity, Mass...) or to the task of the grasp.

In terms of PIP, the Power grasps are the most present class with 4551 appearances. They are followed by the Precision group with 4348 instances. Intermediate is the least frequent one, with only 2640 grasps.

The grasps are slightly unbalanced regarding the Subject. Housekeeper 1 performed 2552 grasps, Housekeeper 2 2369, the Machinist 1 2807 and the Machinist 2 3811.

In table 2 we have the grasp proportion in the dataset. As we can see, due to its unstructured environment there is a high-class imbalance. Grasps as Medium Wrap accounts for 12.61% of the total number of grasps, while others like tripod variation appear one single time (0.008%).

In Fig. 20 we can see the histogram of the grasp durations. We observe that the durations tend to be really short. The average is 6.52s while the median is 2.64s. The 1st and 3rd quartile are 1.04 and 6.48 respectively. There is a respectable number of very large grasps in time, with 575 grasps going over the 0.95 percentile. There are 153 instances of grasps with a duration of 0 seconds. We will have to discard these cases during preprocessing.

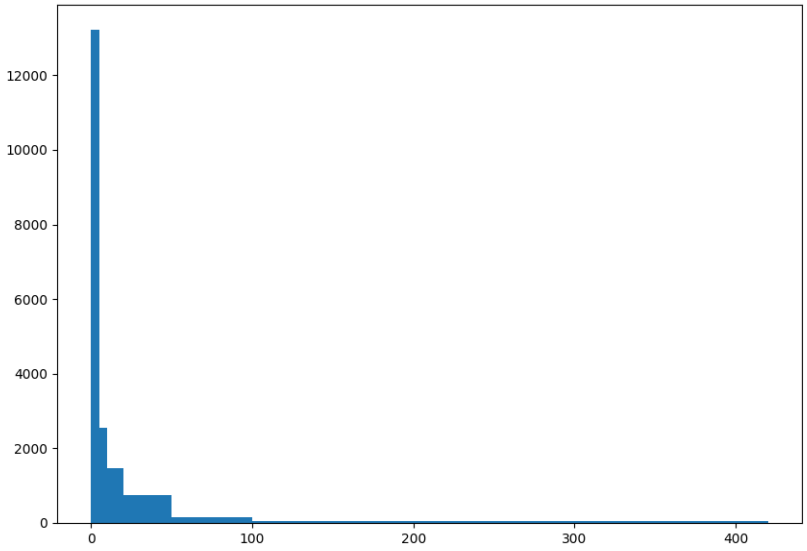


Figure 20: Histogram of grasp duration in seconds

Another important concept for the quality of data is the black ratio. In Fig. 21 we have the histogram of this proportion. We observe that very few grasp sequences have blackened frames since The average is 0.022 while the median is 0. There are 365 grasps that have at least one black frame. We will consider removing these cases during the preprocessing phase.

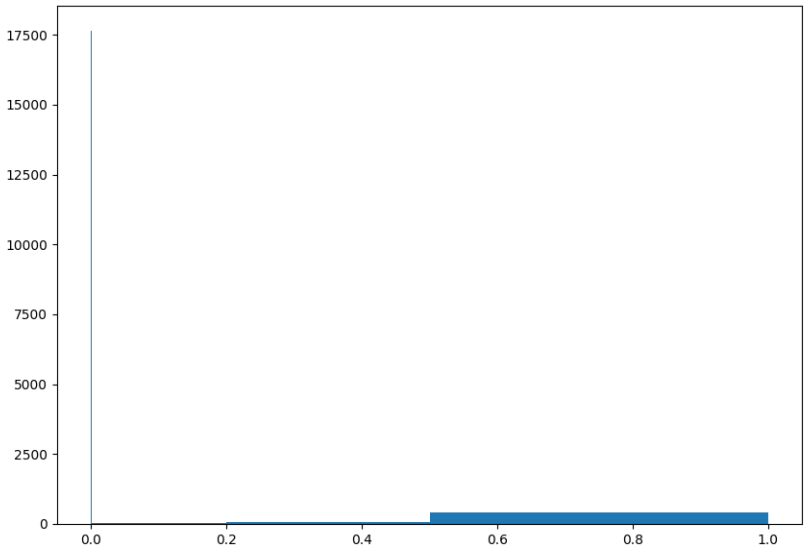


Figure 21: Histogram of the proportion of frames blacked out

4 Proposed Grasping Taxonomy

As stated during the introduction, fine-grained grasp recognition from egocentric RGB data is not an easy task. Due to the occlusions between hand and object sometimes it is almost impossible to differentiate between two very similar grasps even for an expert human annotator. Like many other recent studies (included the Yale Grasp Dataset) we will use the Feix Taxonomy [9], which considers 33 different grasps, as our main reference.

We could train our models with these fine-grained taxonomies, but there are going to be some instances that will be very difficult to predict and in retribution, to this high risk of error we will receive a very small insight. To be realistic, if we want to obtain a system that achieves enough accuracy to be deployed in a real-world setting we must consider the usage of more coarse-grained taxonomies. To define it, it is important that we find a balance between having high accuracy while being informative enough.

We will depart from the 33 grasps Feix taxonomy and we will reduce its number by either merging grasps or by discarding a class if its use is marginal. For the grasp merging, we will take into account the similarity of the grasps according to the characteristics mentioned in section 2.2.1, PIP, opposition type, virtual finger, thumb position. The clustering done in [34] and [15] (Fig. 18) will also be useful.

We depart from the 33 classes taxonomy. In table 2 we have the grasp proportion in the dataset. The first thing we do is discard the Platform/lift grasp because it doesn't really belong to the Feix taxonomy since it strictly depends on gravity. We then begin our discrimination with grasps that appear very rarely (less than 1%), which are 14 classes that represent only 6.15% of the total number of grasps. If we are able to merge them with another similar (and relevant) grasp, we will keep them. If not, they will be good candidates to be eliminated, and on paper, we will discard them. Once we have removed some irrelevant grasps, we start to group the remaining based on their similar characteristics. The need for either precision or power (PIP), the object type and the thumb position are well preserved in our classification. The opposition type is also considered even though there are some cases (abducted prismatic and power circular) where we have some mixing. The concept of virtual finger is the one we sacrifice the most for doing our grouping. We also try to relieve class imbalance.

We end up having 7 different categories, corresponding 3 of them to power grasps, another 3 for precision and lastly 1 for intermediate grasps. They are depicted in Fig. 22 They account for the 90.962% of the grasp instances in the dataset.

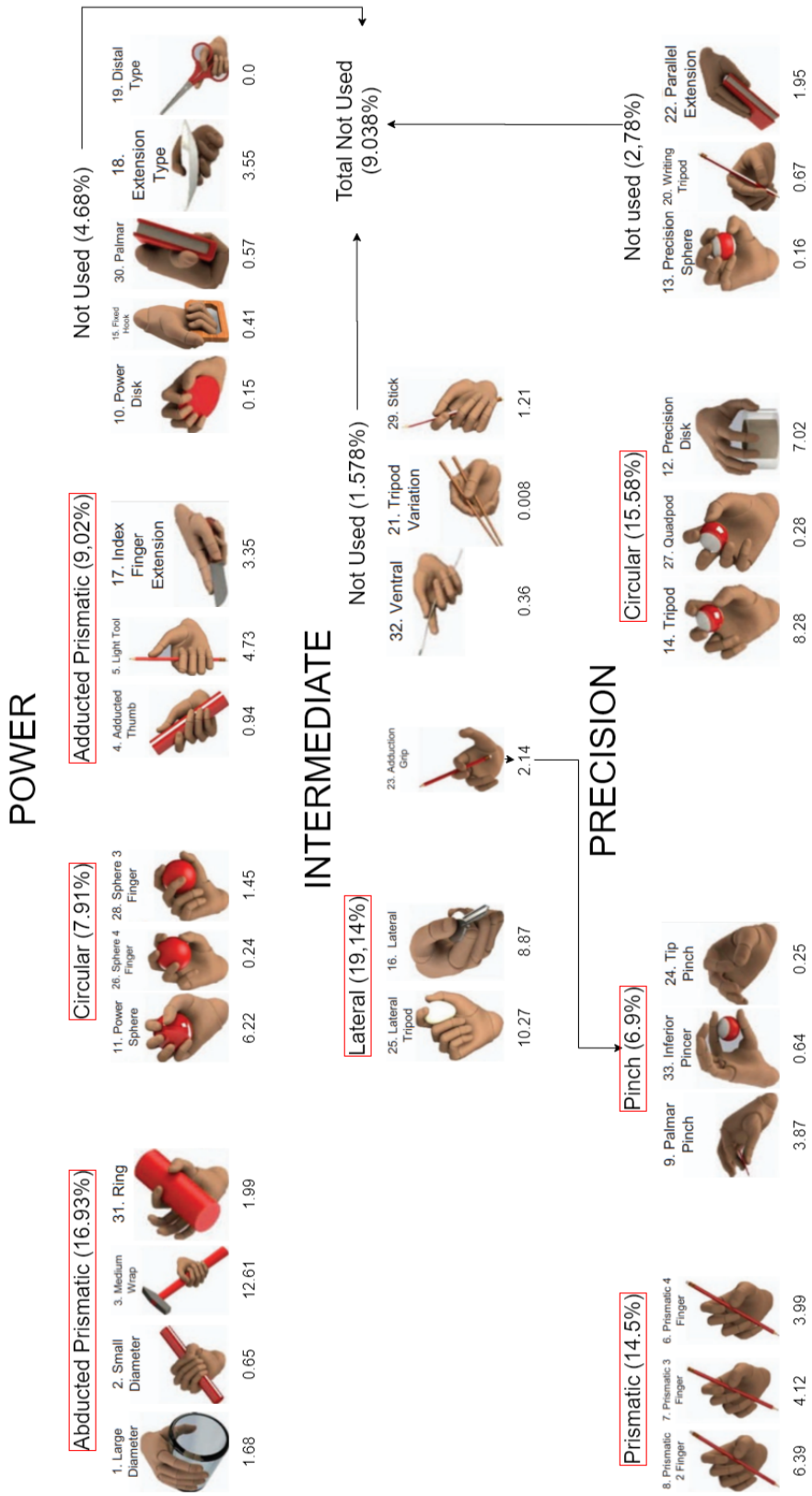


Figure 22: Proposed Grasp Classification for training our model

The power grasps are grouped into 3 classes.

- Power Abducted Prismatic (a.k.a. power cylindric) (16.93%): It refers to grasping of prismatic objects with the thumb in abducted position. It is composed of Large Diameter, Small Diameter, Medium Wrap and Ring.
- Power Adducted Prismatic (a.k.a. power oblique) (9.02%): These grasps are also prismatic but the thumb is in adducted position. Adducted Thumb, Light Tool and Index Finger Extension compose it. Palmar and Fixed Hook could also be considered but due to their few appearances, we discard them.
- Power Circular (7.91%): It groups the power grasps of rounded/circular objects. It is formed by Power Sphere, Sphere-3 Finger and Sphere-4 Finger. Power disk should also be classified in this category, but since it is infrequent, we don't consider it.

We further discard Extension Type and Distal Type. These grasps do not fall well within our categories and visually they may be closer to intermediates grasps.

For the intermediate grasps, we have a single category that incorporates the two most frequent intermediate grasps.

- Intermediate (19.14%): It is composed of Lateral Pinch and Lateral Tripod.

We discard Stick, Ventral and Tripod Variation because they are very infrequent and do not resemble too much the others. Adduction Grip is the only grasp that crosses the PIP barrier and is grouped in the Precision Pinch class because is similar functionally and otherwise the class would be too small.

Lastly, we have 3 groups for precision grasps.

- Precision Pinch (6.9%): It is constituted by Palmar Pinch, Inferior Pincer, Tip Pinch and Adduction Grip. These grasps use only the finger and the thumb (except Adduction Grip).
- Precision Prismatic (14.5%): Composed by Prismatic-2 Finger, Prismatic-3 Finger and Prismatic-4 Finger. These are precision grasp of prismatic objects with either 2,3 or 4 fingers.
- Precision Circular (15.58%): It groups the precision grasps of rounded/circular objects. It is formed by Tripod, Quadpod and Precision Disk. Precision Sphere would also fall in this category, but it is not considered because it is very rare.

Writing tripod is not used because it is infrequent and even if can be similar to Precision Prismatic it has side opposing type. Parallel Extension, on the other hand, is quite different from the rest of precision grasps.

Grasp (number in Fig. 8)	Instances	Percentage %
adducted thumb (4)	109	0.94
adduction (23)	247	2.14
extension type (18)	410	3.55
fixed hook (15)	48	0.41
index finger extension (17)	387	3.35
inferior pincer (33)	74	0.64
large diameter (1)	195	1.68
lateral pinch (16)	1024	8.87
lateral tripod (25)	1186	10.27
light tool (5)	546	4.73
medium wrap (3)	1456	12.61
palmar (30)	66	0.57
parallel extension (22)	226	1.95
platform (34)	96	0.83
power disk (10)	18	0.15
power sphere (11)	718	6.22
precision disk (12)	811	7.02
precision sphere (13)	19	0.16
quadpod (27)	33	0.28
ring (31)	230	1.99
small diameter (2)	76	0.65
sphere-3 finger (28)	168	1.45
sphere-4 finger (26)	28	0.24
stick (29)	140	1.21
thumb-2 finger (8)	738	6.39
thumb-3 finger (7)	476	4.12
thumb-4 finger (6)	461	3.99
thumb-index finger (9)	447	3.87
tip pinch (24)	29	0.25
tripod (14)	956	8.28
tripod variation (21)	1	0.008
ventral (32)	42	0.36
writing tripod (20)	78	0.67

Table 2: Grasp proportion of the original 33 types in Yale Grasp Dataset

5 Proposed Methodology

5.1 Temporal Dimension Treatment

Since we are working with video data, there are several issues that we must address if we want to take advantage of its temporal dimension.

The first important thing is to define at which level we will work with the dataset. The CSV has many rows, where each one represents a certain grasp instance in the videos. The grasps have different duration and therefore, are composed of a variable number of frames. Since we want to deal with video models we will split the data at the grasp sequence level. This means that if a certain grasp sequence (row in the CSV) is selected for training data, then all its frames will be part of this set. When training the video model it is very natural to do the split at the grasp sequence level. To be consistent, we do the same for the image models. This means that even if we treat the frames individually, all the frames of a grasp sequence must either belong to the training or the test set. If we instead split the dataset at the image level, we would have frames in the test set that are almost identical to the ones we used to train, and in accordance, the recognition rate increases drastically. However, that would be an unrealistic way of measuring the accuracy.

The second thing is about the number of frames that we will use for predicting. This is an important issue of video models, where a single prediction is given for several input frames. Using a higher or lower number changes the model and impacts its complexity. It is not very clear the improvement extent of using more images, but the intuition says that the more frames we use, the more information we would have, and the better we could discriminate. However, there must exist a cliff where adding more frames doesn't improve the results We decide to use 8 frames because it is the number that employs the pre-trained video model that we will use. There is also available another model that uses 4 frames, but we prefer to use the one with 8 because it had better results in the pre-training dataset.

The last thing to consider is the way that we sample the frames (i.e. sampling strategy or sampling rate). There are many different possibilities, but we will opt for a simple approach. We will draw the same number of frames from each grasp sequence regardless of their real duration. Since our video models work with 8 frames, we will always draw 8 frames.

When training, for each grasp sequence we will select randomly (without repetition) 8 frames between its start and end frames. Then we put them in order so that they make sense temporally. For validation and test, we draw uniformly 8 frames, including the start and end frame, so that the results are reproducible.

The problem with this approach is that depending on the duration of the grasp, we are feeding data with different temporal resolutions because the sampling rate varies with the grasp duration. As an example, we assume that our video model accepts as input 8 frames. If our dataset has a short grasp sequence that only lasts 8 frames, then we would use all its frames. These frames are very close temporally and therefore we are feeding data with a high temporal resolution. On the other hand, if we have a long grasp that lasts 256 frames, the same model would receive 8 frames with a uniform stride of 32. Given that the dataset frame rate is 25FPS this would mean that each frame has a difference of more than 1 second with

respect to its predecessor. In these cases, the frames are more distant in time than before, and in accordance, the temporal resolution is lower. This makes our system rather inconsistent but is a simple way to deal with sequences that have different duration. Furthermore, given that most part of the grasps are short, the overall temporal stride will not be so different. The 1st quartile of grasp duration is 1.04s, while the third is 6.48s. If our model works with 8 frames, then the temporal stride between one frame and the next will mostly range between 0.13s and 0,81s. If we are really concerned about this issue, we could potentially discard long grasps.

Given that the dataset has a framerate of 25FPS and the video model uses 8 frames, this means that the grasping sequences must take at least 0.32 seconds. This reduces the number of useful grasp rows in the dataset to 11,004.

5.2 Image model with rolling average

We want to compare three different approaches to determine which one seems to be better for the task of grasp recognition from RGB egocentric video.

Our first model will act as a baseline since on paper it is the most simple approach. It will consist of an image model that given the entire image will predict the grasp class. This approach doesn't have the ability to extract temporal features, and therefore the prediction is merely done with spatial information.

We will use a ResNet-50, which is a residual convolutional neural network that is 50 layers deep. We will use the implementation from the Torchvision library of Pytorch. Given that our dataset is quite small, we will not train the model from scratch. We will use pre-trained weights from the ImageNet dataset and we will use the grasp dataset for fine-tuning. Since the dataset used for pre-training (ImageNet) is quite different from the Yale Grasp Dataset, we will need to fine-tune some layers from the convolutional base.

For training, we will take our grasp video sequences, and we will break them into individual images. Logically all the frames of the same sequence will have the same grasp label and will be either used for training or validation (we split the data at the video sequence level, not at the individual image). With this data, we will train the model.

During test we will predict at the video level using a rolling average methodology. For each test video sequence, we will pass individually each image to the model, which will be predicted as one of the output classes with a certain probability. Then we will take the predicted probabilities of the whole sequence and do the average. The class with the highest probability will be selected for that video sequence.

The problem of this approach is that because of the complexity of the input data (frequent hand-object occlusions) it is very likely that some of the individual frames of a sequence are not clear. Then its prediction may not be accurate. That is why we use the rolling average. We expect to attenuate the effect of bad frames with the average prediction of the whole sequence.

5.3 Video model

We want to see how a video model would perform for grasp recognition. Grasping is not an action that is really determined by a succession of hand poses, but the temporal dimension may have some relevance for its prediction. Its extent is unclear, but it is worth trying. As far as we know, this may be the first time that video models have been used for grasping recognition.

There exist many different video architectures in the literature, like I3D, R(2+1)D, SlowFast (or only its Slow path), CSN or X3D. In Kinetics-400, a good action recognition video dataset benchmark they obtain similar accuracy so any of them could be a reasonable choice.

We finally decide to choose the Slow architecture with a ResNet50 backbone. It is basically a 3DConvNet. It is an average architecture in terms of accuracy but its design goes well with our expectation that in grasping the spatial component is more important than the temporal. Another reason to choose it is that we will use an implementation from Facebook's library Pytorchvideo [38]. They have several pre-trained models with different video datasets. According to [39] pre-training labels that overlap the most with the target labels improve performance. That is why we wanted to use weights coming from an egocentric dataset such as Charades. Interestingly, we found out that the pre-trained weights from the Kinetics-400 dataset lead to better results, so we finally used them.

The model that we use has a temporal dimension of 8 frames, so for each prediction we need to pass to the model 8 frames from the same grasp sequence. The model is quite complex, having a total of 32.45 million parameters. As the Yale Grasp Dataset is relatively small and it is quite different from the one that was used for pre-training (Kinetics-400) we will fine-tune a few layers from the convolutional base.

5.4 Hybrid Hand detection model

Lastly, we will have a specialized hybrid model. We have seen that a considerable part of the models for grasp recognition have a first step where they detect the hands in the scene.

Following that idea, we will train a grasp recognition image model that receives as input images cropped around the hand and the object grasped. The first step is to create this dataset of hands while grasping. We generate it offline using the hand detector from [32]. This is a nice hand detector that uses FasterRCNN as a base. Given an image, it detects the appearing hands and their sides. Apart from that, it can also detect the objects being interacted with by the hands. This is a very interesting feature that could be useful for grasping, but its accuracy is not good enough for automatically generating the dataset. On the other side, the hand detector is not infallible but is much more consistent, so it is more suitable for our purpose.

The basic idea is to use this hand detector to detect and crop the right hands from the Yale Grasp Dataset video frames where some grasping is occurring (the same frames that we use to train the image or video models). The main problem is that sometimes it doesn't detect any hand (which in this domain is not a big problem), and few times it confuses the left and

right hands. We crop the hands around the hands leaving some margin so that the object can also be seen (at least partially).

Once we have created the auxiliary hand grasp dataset, we use it to train a ResNet50 image model to predict the grasp class. We use the same architecture and pre-trained weights as in 5.2. They only differ on the datasets that we use for fine-tuning them.

Then our idea is that for each test image, we pass it through the hand detector from [32]. If it detects a right hand, we will crop it and pass it to the hand image model. Otherwise, we pass it to the grasp recognition model that works with the whole image from 5.2. With this kind of hybrid approach we ensure that if we do not find a right hand in a certain frame, we can still predict the image.

We will predict individually every frame of the video sequence and then we will apply the rolling average strategy to predict at the video sequence level.

6 Capturing our Grasp Dataset

We have recorded our own small egocentric video human grasp dataset. We do it mainly because in our setup we mount the first-person camera in the chest with a harness instead of locating it in the head as they do in the other datasets. The only one that has the camera positioned in the chest is [36], but as we have already mentioned it is not useful for several reasons.

The discussion about where to locate the camera is an interesting one. The two main options are wearing the camera in the head or in the chest. When we perform actions we tend to accommodate our body position to do it in a natural/comfortable way. One of the things we do is orient our body towards the action direction. In the case of grasping, we have the hands mostly in front of the body. That is why in general locating a camera in the chest will capture with good detail those sequences. There are some cases where we would place the hands out of frame, but it will work in general. Apart from tilting our body, we frequently look or move the head in the direction of the action. Locating a camera in the head pointing downwards would have both the body and head benefits, hence it would probably capture better the grasps.

The problem with head cameras is that they are less practical to wear. Putting a headband in a patient is a bit less comfortable than putting a harness in the chest. Another downside is that the head is a less stable position so the captured video would have more motion. This could be a problem for us given that our patients have some mobility problems.

With this balance of advantages and downsides for both methods, it was decided to use the chest located camera. The comfort and the video quality prevailed over a richer orientation.

We used a GoPro Hero 6 camera positioned in the center of the chest with a harness. We tilted the camera downwards around 20°. We captured all the videos in the same controlled environment at 120FPS and a 1920x1080 resolution. The lab room of the university had a yellow table that we cleared to have a clean setting. This setup resembles the one from the UT Grasp dataset. We put two marks with duck tape on the table. In each video, we capture one specific grasp type and object. We start with the object located over one of the marks. The idea was to grab the object with the appropriate grasp and move it to the other mark to then drop it. We do it from a stand-up position. We repeat this process back and forth so that we do the same grasp 10 times. In every attempt, we try to describe different trajectories with the hand and the arm so that we have some orientation and position variety.

We have 60 videos, with 10 grasps in each, for a total of 600 grasps. We used 22 different objects. It includes a beer bottle, an apple, a pen, a coffee mug, a roll of duck tape, a foam circular object, a ceramic plate, a small cable, a beer can, a paper coffee cup, a spatula, a water bottle, a gripper, a spoon, a knife, a box of mints, a box, a wallet, a window cleaner, a plastic plate, a glass and a battery. We labeled the grasps according to our taxonomy of 7 classes. For each object, we captured between 1 and 4 different grasps types. We basically performed the grasps that were more plausible according to each object. The class composition is as follows.

Grasp	Instances	Percentage %
Intermediate Lateral	20	3.33
Power Circular	79	13.16
Power Prismatic Abducted	131	21.83
Power Prismatic Adducted	19	3.16
Precision Circular	51	8.50
Precision Pinch	160	26.66
Precision Prismatic	140	23.33

Table 3: Grasp proportion in our dataset

As we can see, there is an important class imbalance that is mainly produced because of the intrinsic nature of the objects. In particular, there are very few instances of intermediate and power adducted grasps.

The way we capture the videos is mainly focused to ease the task of labeling. At the beginning of the video, we say out loud which type of grasp we were performing and the object name. We counted every grasp in the video so that we couldn't miscount. The labeling task was quite simple, we created a CSV file where each row represented a grasp sequence. We have 6 different columns.

- ID: The grasp number.
- Video: The filename where the grasp appears.
- Start Frame: Start frame of the grasp.
- End Frame: End frame of the grasp.
- Object: The object that we grasp.
- Class: The category to which the grasp belongs.

For finding the start and end frames of each grasp we used a command from FFmpeg that given an input video it creates a copy that has at the bottom the frame number. We reproduced this copy and stopped the video at the frames where the grasping started or ended. We got those frame numbers and manually added them to the CSV file. Given that the videos have a high framerate, we didn't focus too much on finding the exact frames where the grasps start/end.

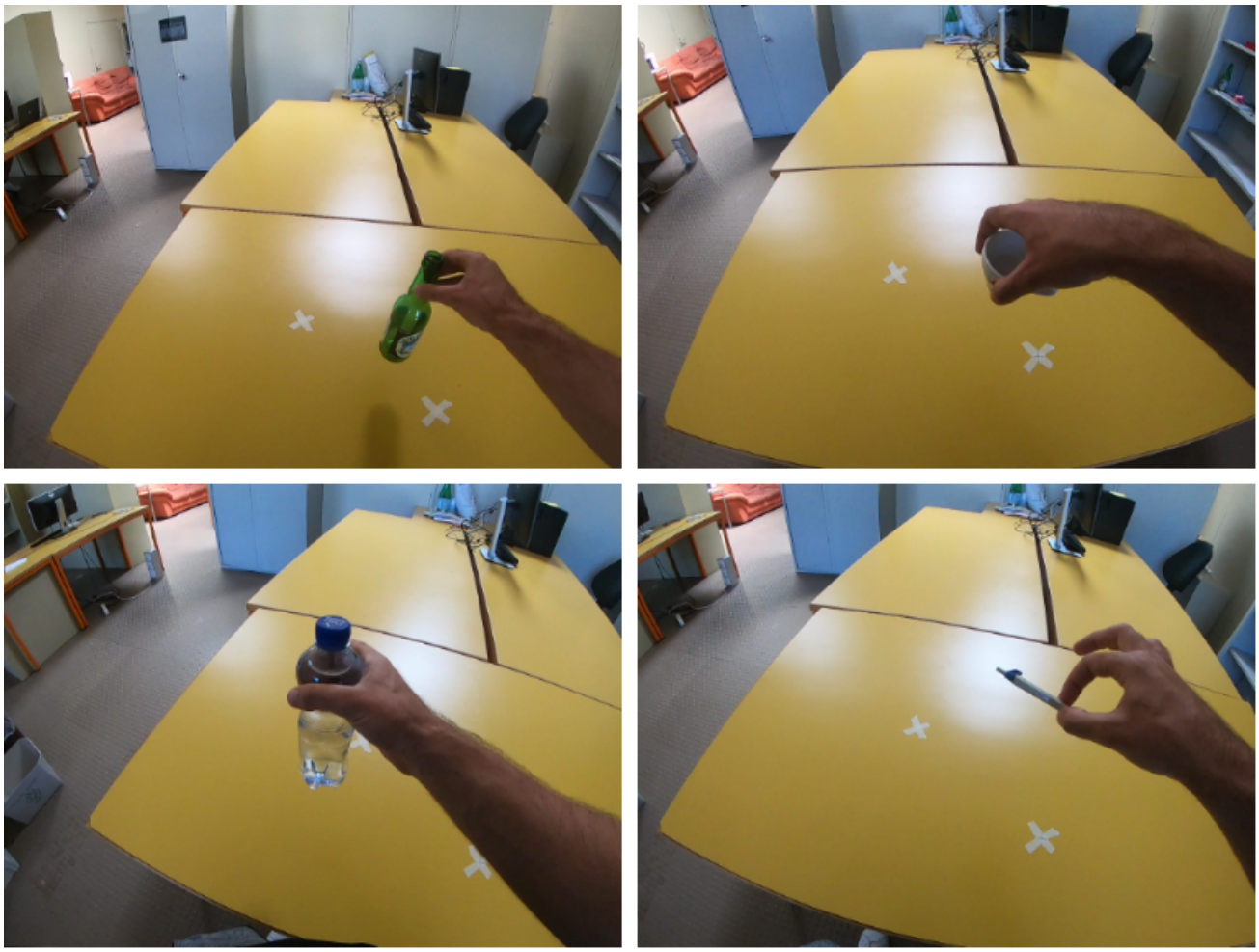


Figure 23: Example frames from our own grasp dataset

```
ffmpeg -i video.mov -vf "drawtext=fontfile=Arial.ttf: text=%{n}: x=(w-tw)
   /2: y=h-(2*lh): fontcolor=white: box=1: boxcolor=0x00000099" -y output.
   mov
```

Listing 1: ffmpeg command to add frame number to video

7 Training

We will use the multi-class cross-entropy loss function as it is the default loss function for the multi-class classification problem.

Cross-entropy will calculate a score that summarizes the average difference between the actual and predicted probability distributions for all classes in the problem. The score is minimized and a perfect cross-entropy value is 0.

$$Cross\_entropy = -\sum_{x} (p(x) * q(x))$$

We will use validation cross-entropy loss as a criterion for stopping training (several epochs without improving validation loss) and for selecting the epoch with the best model weights (minimum validation loss).

7.1 Hardware and code implementation

For the training process, we used a server from the HEIG-VD. The processor was a Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz with 8 cores. The GPU was an NVIDIA GeForce GTX 1080 with 8GB.

All the code has been programmed using Python language. The different scripts used and the results can be found on the following Github link.

7.2 Preprocessing

7.2.1 Yale Grasp Dataset Preprocessing

Before we can train our models, we must do some preprocessing of our data to have it in an appropriate format and ensure its quality. The first thing we do is to select only the right-handed grasp sequences that are annotated under one of the 33 categories of the Feix taxonomy. This reduces the dataset length from 18,210 to 11,539 labeled grasps.

After, we will keep only the grasp sequences that don't have any blackened frame. We could have decided to also use sequences with a low percentage of blackened frames, such as a 10%, but since the difference was really small (only 24 grasps), we opted for the cleanest option. This lowers the length of the dataset from 11,539 to 11,174 grasp sequences. Even after this step, we observed during training that some sequences still had some blackened frames. This is probably because of annotation errors and it is really difficult to identify. Given that the number of instances where this occurs is really low, we will obviate this phenomenon.

Then, we dropped all the grasps that had a duration inferior to 0.32 seconds. That is because we wanted to have at least a minimum of 8 frames per grasp sequence and the videos were captured at 25FPS. After, we compute the start and end frame of each sequence since this way it will be easier to read the videos during training. First, we have to convert the start timestamp that is given in hh:mm:ss format to the numeric start frame. After, we compute

the end frame as $start\_frame + duration * fps$. Our CSV file gets further reduced to 10,723 instances.

Next, we check that the grasp sequences of the dataset are actually present in the videos. This means that the file and the associated frames exist (i.e. the video is not cut before). With these checks, we have 10,518 valid grasps.

Lastly, we apply our proposed grouping of 7 classes. In accordance, we drop all the instances that do not belong to it. The final dataset length is 9,452. We observe that the proportions do not vary too significantly from the original dataset, which means that the preprocessing affects similarly all the classes.

Grasp	Instances	Percentage %
Intermediate Lateral	1995	21.10
Power Circular	808	8.54
Power Prismatic Abducted	1776	18.78
Power Prismatic Adducted	988	10.45
Precision Circular	1619	17.12
Precision Pinch	733	7.75
Precision Prismatic	1533	16.21

Table 4: Grasp proportion in Yale Grasp Dataset after preprocessing

In table 4 we have the proportions of our proposed classes after the preprocessing step. As we can see, there is some class imbalance but is not too heavy.

In Fig. 24 we have a boxplot of the grasp duration grouped by their categories. As we can see, the durations are similar, but power-cylindric (power prismatic abducted) and precision-circular grasps, tend to be slightly longer.

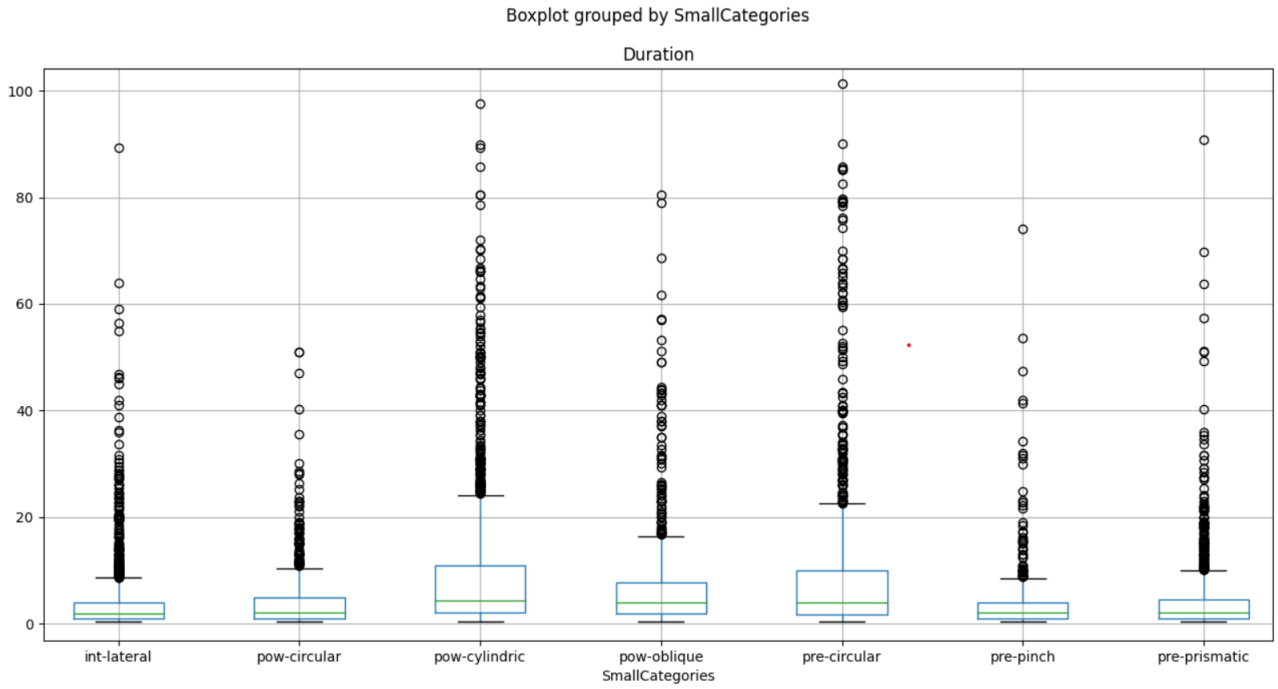
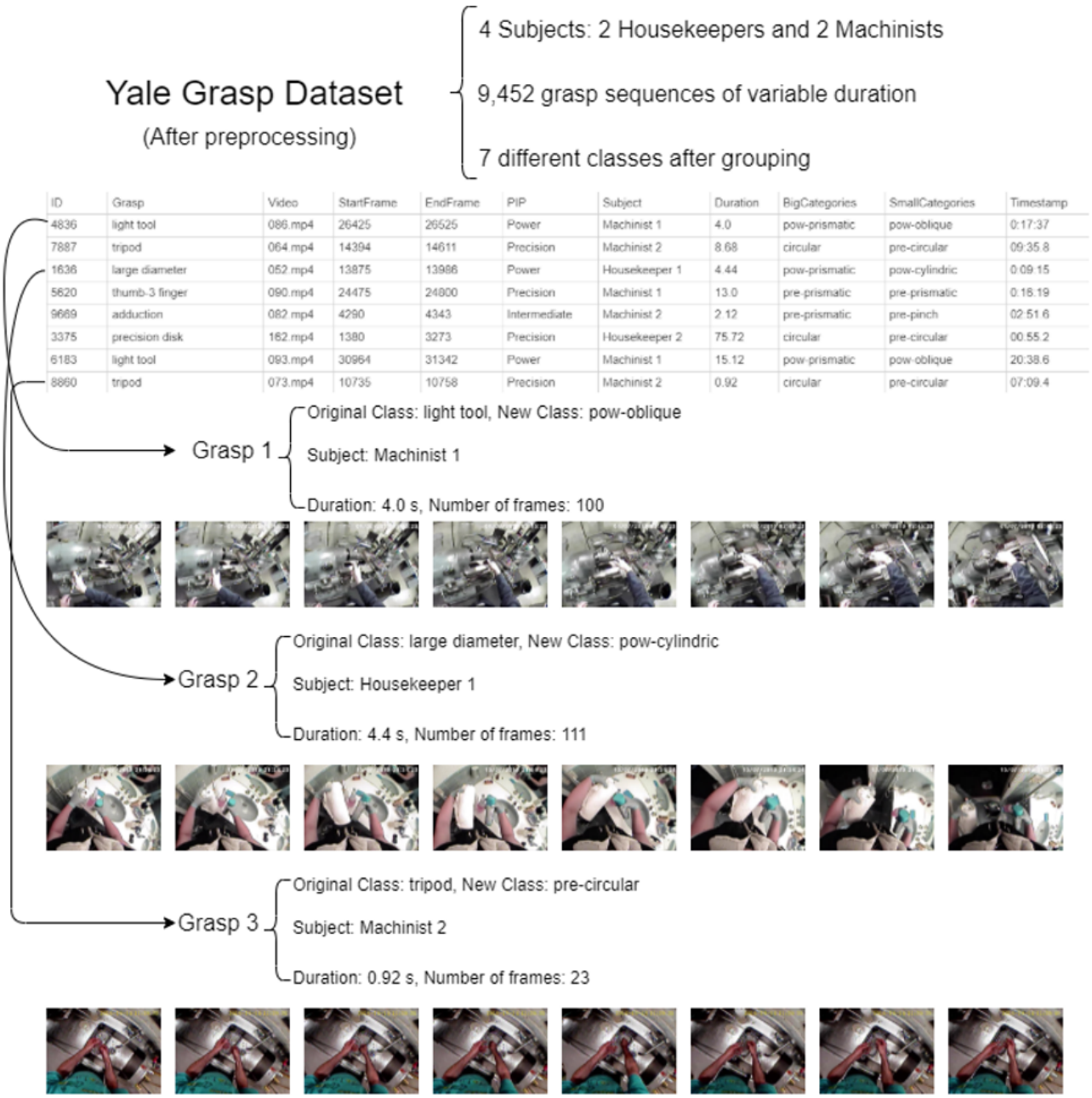


Figure 24: Grasp duration boxplot grouped by categories

In Fig. 25 we show some examples of grasp examples to try to illustrate better the dataset.



ID	Grasp	Video	StartFrame	EndFrame	PIP	Subject	Duration	BigCategories	SmallCategories	Timestamp
4836	light tool	086.mp4	26425	26525	Power	Machinist 1	4.0	pow-prismatic	pow-oblique	0:17:37
7887	tripod	064.mp4	14394	14611	Precision	Machinist 2	8.68	circular	pre-circular	09:35.8
1636	large diameter	052.mp4	13875	13986	Power	Housekeeper 1	4.44	pow-prismatic	pow-cylindric	0:09:15
5620	thumb-3 finger	090.mp4	24475	24600	Precision	Machinist 1	13.0	pre-prismatic	pre-prismatic	0:16.19
9669	adduction	082.mp4	4290	4343	Intermediate	Machinist 2	2.12	pre-prismatic	pre-pinch	02:51.6
3375	precision disk	162.mp4	1380	3273	Precision	Housekeeper 2	75.72	circular	pre-circular	00:55.2
6183	light tool	093.mp4	30964	31342	Power	Machinist 1	15.12	pow-prismatic	pow-oblique	20:38.6
8860	tripod	073.mp4	10735	10758	Precision	Machinist 2	0.92	circular	pre-circular	07:09.4

Figure 25: Yale Grasp Dataset grasp examples

7.2.2 Hand Dataset Generation

As a previous step before training the model that predicts grasps from images cropped around the hand, we have to generate an auxiliary hand dataset.

To do that, we use the hand detector from [32]. Basically, we loop through all the grasp train sequences of the Yale Dataset and we apply the hand detector every 15 frames. Originally we wanted to apply the detector to every frame, but it was too slow (every 15 frames took

more than 4 days).

If we detect a right hand, we crop it and we save it to disk. If it is possible (due to the image coordinates), we add 20 pixels in each of the 4 directions to ensure that in the hand image, a significant part of the object will appear. In case we detect a left hand or we don't detect any hand, we skip the frame and do nothing. While doing this, we are generating a CSV file with the label of the grasps and the references to the files.

The problem with our approach is the uncertainty about hand detection. First of all, for a certain grasp, we are not guaranteed the detection of any hand. Furthermore, each grasp sequence has a different length, and therefore, the longer the grasp, the more possibilities we have to detect a hand in one of its frames. This can distort the grasp type proportion from the one that we had originally. Nonetheless, we preferred this approach to have the largest possible hand dataset subject to our time constraints.

The auxiliary hand dataset ended up having a total of 51,141 images. The grasp composition changed with respect to the one that we had originally. We observe that power-prismatic-abducted, power prismatic-adducted and precision circular classes have increased their presence while the others have decreased. This is probably related to their slightly longer duration. There is no clear evidence that the hand detector has more trouble detecting hands during certain grasps, but this can be the other factor that explains the change in grasp proportion.

Grasp	Instances	Percentage %
Intermediate Lateral	7885	15.39
Power Circular	2666	5.20
Power Prismatic Abducted	13211	25.78
Power Prismatic Adducted	6795	13.26
Precision Circular	10819	21.11
Precision Pinch	2987	5.83
Precision Prismatic	6867	13.40

Table 5: Grasp proportion in the hand auxiliary dataset

The images have very variable dimensions but generally, the height is slightly higher than the width. They are rather small, with both dimensions moving generally between 70 and 130 pixels. Given that the original source has quite low resolution, the resulting crops have very low quality. This can be an important factor that can affect negatively the accuracy of our models.

In Fig. 26 we have several examples of the cropped hands that compose the auxiliary dataset that we generate. In the last row, we included examples of incorrect detections. As we can see there are several left hands detected as right hands and one arm detection instead of the hand. In any case, this faulty data represents a small part of the data, accounting for less than 5% of the total.

Figure 26: Examples of cropped hands generated as auxiliary dataset

7.3 Data split

We want to test the generalization capacities of our models in two different ways.

In the first one, we just want to ensure that our model is able to predict unseen grasp sequences during the training phase. For that, we will split randomly the grasps between training and test set. We use a certain seed so that the split can be replicated. Since we don't have much training data, we will use 16% of the data for test and the rest for training. We have 7939 grasps for training and 1513 for test. The proportion of grasp types in the train and test sets are very similar.

In the second data split, we want to test the cross-user predictive capabilities. This is important because people have different heights, hand sizes, grasp styles or skin colors. We have 4 different subjects (2 machinists and 2 housemaids) so we can create 4 different splits, where we use 3 users for train and the remaining for test. Due to time restrictions, we will just use 1. The problem of this approach is that the class proportion between subjects varies significantly due to the nature of their activities. We decide to use Housekeeper 2 for test set because it is the smallest group, and at the same time is the second split with the most similar class composition between train and test set. In this case, we will have 7109 training grasps and 2343 for test.

7.4 Data Augmentation

Since we have a relatively small dataset, it will be important to use data augmentations to elevate the number of training samples and their variety. These transformations will be done

on the fly at training time.

For each training grasp sequence sample, we draw randomly 8 frames (without repetition) and we put them in order. We do this because this way at each new epoch we will potentially read different frames that can enrich our training. After that, we normalize and standardize the video. The values used are the mean and std of the dataset used for pretraining. For the case of the video model, that is pre-trained with the Kinetics 400 dataset, those values are $mean = [0.45, 0.45, 0.45]$ and $std = [0.225, 0.225, 0.225]$. On the other hand, the image model is pre-trained with the ImageNet dataset, with values of $mean = [0.485, 0.456, 0.406]$ and $std = [0.229, 0.224, 0.225]$.

Next, we scale the data from 640×480 pixels to 320x320. We use a short side scale, which means that first we determine the shorter spatial dimension of the video and we scale it to the given size. To maintain the aspect ratio, the longer side is then scaled accordingly. After that, we do a random crop of size 244. This way, for each epoch we will crop a different part of the image.

Then, we perform a random rotation between -90 and 90 degrees. With this, we look for small changes in the angle for the grasp appearance. Lastly, we perform a random horizontal flip with a probability of 0.5, which means mirroring half of the grasps. With this, we seek for training the model so that is also able to classify left-hand grasps.

For validation and test, the transformations we do are not stochastic. This is because the results must be consistent across the different epochs. For each grasp sequence, we will always draw uniformly the same 8 frames. We will standardize them as before. Next, we will do a short side scale of 320 pixels. Finally, we will do a center crop of size 244.

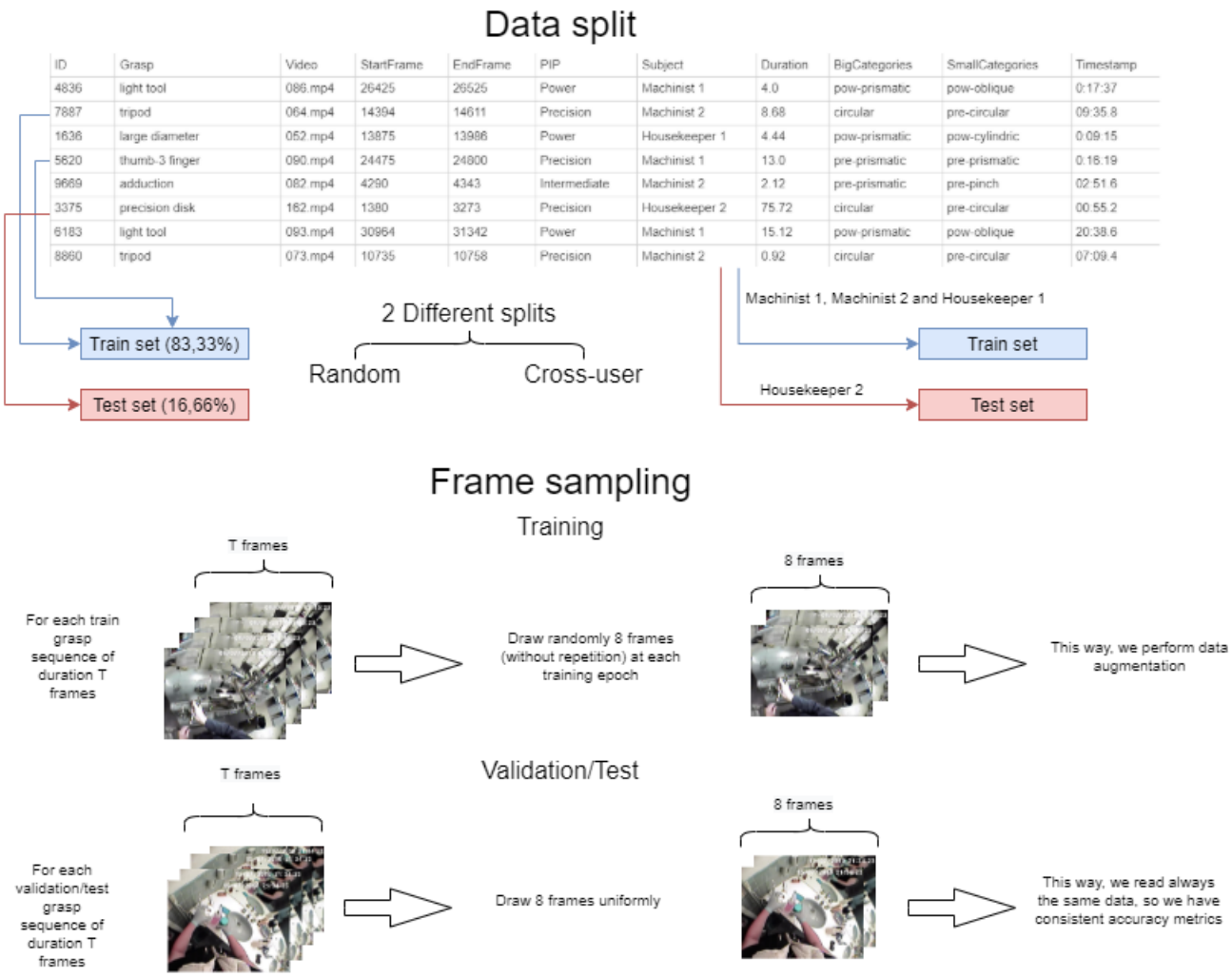


Figure 27: Dataset split and frame sampling

7.5 Hyperparameter choice

Due to the many different experiments, tasks and changes that we had to do and the long time that it can take to train, it was hard to set up a rigorous methodology for hyperparameter selection. The values that we have selected are based on a more informal manual experimentation made along the different stages of the project.

Regarding the hyperparameters related to the Network structure, we started by just adding at the end of the convolutional network a fully connected layer that had as many output neurons as the number of classes. Given that it suffered underfitting, we added a prior hidden layer of size 256.

Then we tested fine-tuning several layers of the convolutional network. We tested several values, but we ended up selecting a low value to avoid overfitting.

Overfitting was still a problem, so we decided to add a dropout layer between the two fully connected layers. We tested values of 0.2 and 0.5, selecting the latter as it produced better results. Since we used pre-trained models we didn't have to bother about network weight initialization.

Regarding Hyperparameters related to the training algorithm, we used 100 training epochs, but we set up a control method to stop training if the validation loss didn't improve in 20 epochs. Even if we do that, we select the model weights from the best validation epoch, not the last one.

We selected the Adam optimizer as it is one of the best among the adaptive optimizers. We didn't compare with other optimizers.

As learning rate, we tried several values spaced with a logarithmic scale. Those were 0.1, 0.01, 0.001 and 0.0001. We didn't observe much of a difference, but we decided to select the lowest value to try not to change very fast the weights that we were fine-tuning from the convolutional network. For the momentum, we used the default value of 0.9 for the Adam optimizer.

For the batch size we wanted to select a small value because as they are noisy, they offer a regularizing effect and lower generalization error. We also wanted a value that didn't slow down the training too much. We found out that 32 was a good value for video input. For image input, we used 256 as it corresponds to 32 videos batches of 8 images each one.

7.6 Image model

We use a Torchvision implementation with pre-trained weights from the ImageNet dataset. The convolutional network has an image input size of 224-by-224 and an output size of 1000 (because it classified images into 1000 categories). After this convolutional base, we add some extra layers for classification. In particular, a fully connected layer of size 256 with ReLU activation function, a dropout layer with probability 0.5 and a final fully connected layer that has as many output neurons as the number of classes (7). We optimize the weights of the classification layer and the last 8 elements (not layers) of the convolutional base. There are 5,720,047 trainable parameters in total.

We split randomly the data between training (80%) and validation set (20%). We used a fixed seed so that we can replicate the same split in all the subsequent experiments. We split the data at the grasp sequence level so that the images of a certain action belong completely to either the train or the validation set.

We use a batch size of 256 images. Given that we use 8 frames per grasp sequence, it corresponds to 32 videos. We use a learning rate of 0.0001 and the Adam optimizer.

7.7 Video model

We use the Slow architecture with a ResNet50 backbone. We use a Pytorchvideo implementation with pre-trained weights from the Kinetics video dataset. The convolutional network has an image input size of 224-by-224 and an output size of 400 (because it classified videos

into 400 categories). As in the case of the image model, we add a fully connected layer of size 256 with ReLU activation function. A dropout layer with a probability of 0.5 and a final fully connected layer that has as many output neurons as the number of classes (7). Apart from the weights of the classification layers, we also fine-tune the last 8 elements of the convolutional base. There are 4,337,047 trainable parameters.

We use the same random data split (80% train and 20% validation) as for the image model. This means that the training and validation images used in both cases are exactly the same except for the random augmentations.

As we use 8 frames per grasp, this time we use a batch size of 32 grasp sequences. Again, the learning rate is 0.0001 and we utilize the Adam optimizer.

7.8 Hand image model

The base model is exactly the same as for the image model, a ResNet50. The classification layer is again as before. The difference is that it will be trained with a hand grasp auxiliary dataset that we previously created.

Similarly, we randomly take 80% of the data for training and 20% for validation (20%). Due to the aforementioned uncertainties during the creation of the hand dataset, it is not possible to map it completely with all the train grasps of the original dataset, and therefore the split will be different from the one that we had for the image and video models. Anyway, we still split the data at the grasp sequence level, so all the detected hands of a certain grasp sequence are either for training or for validation.

We still use a batch size of 256 images. We employ the Adam optimizer and a learning rate of 0.0001.

As we use a different dataset, this time we will perform different augmentation transformations. For training, we start by normalizing and standardizing the pixel values of the images that we read from the disk. After that, we apply a resize so that all the images have the same size of 244x244 pixels. To end, we will perform a random rotation between -90 and 90 degrees and a random horizontal flip with a probability of 0.5. For validation, the transformations will not have a random component. We will standardize the images and then we will resize them so that they have a dimension of 224x224.

8 Results

In this section, we present the results obtained. For each approach, we will present the results for the random and cross-user splits of the Yale dataset. Apart from that, we also use the models to predict our own dataset. At the end of the chapter, we will present a table summary with the most important results.

It is important to mention that it is quite difficult to compare our results with other baselines. The only other authors that have tried to do something similar with the Yale Grasp Dataset are [34] and [15] but because of several reasons, it will be hard to contrast it with our work. First of all, they use a subset of 17 grasps from the Feix taxonomy, while we use a more coarse-grained classification with 7 different types. Using the same 17 grasps for the sake of comparison is a doable task. The problem is that their methodology is not very clear. They use a model to crop the right hands prior to train 17 one-versus-all multi-class grasp classifiers. It seems that they only use frames where they detected a hand, so they have potentially removed difficult cases. Then it is not evident the final number of images used for training. They may use one image per grasp sequence or all of them. Additionally, it is not clear if they allow for the same subject/scene to be included across the train and test set. This is not trivial, because doing that boosts dramatically the accuracy results.

8.1 Image model

8.1.1 Random Split

The image model is supposed to be the simplest approach to deal with this problem and in accordance, we expected to obtain inferior results compared with the other two approaches.

We trained the model during 71 epochs. In Fig. 28 we show the training and validation loss plot. The best validation loss was obtained for epoch 50, so we will use the weights of this step for testing. We observe that the training loss decreases steadily during the whole training process, even by the time that we stop, so we could have kept the training going. The problem is that the validation loss doesn't follow that pattern. In the beginning, it decreases similarly to the training loss, but after some epochs it gets stuck. This is a sign of overfitting. We tried to alleviate it by reducing the model capacity, adding some regularization and adding more data augmentation. The problem is that it doesn't get much better after it. We suspect that the problem is that there are significant differences between the grasp sequences of the training and validation sets. Even if two grasps have the same label, they can have different objects, backgrounds, hand sizes and orientations...

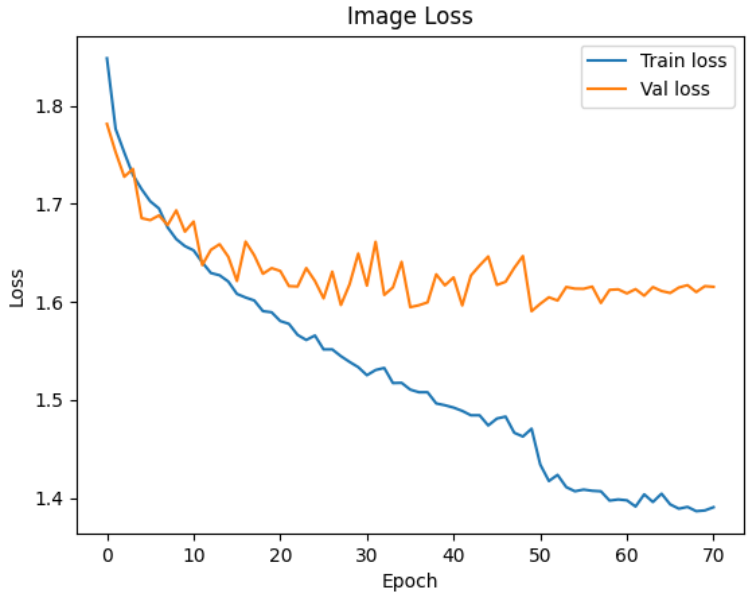


Figure 28: Training and validation loss of the image model with random split

During the training process, each train/validation image is predicted individually. We obtain a validation accuracy of 42%, which is an acceptable result given that we have 7 different classes.

We then try with the test set. When we predict each frame separately, we obtain an **accuracy** of **38.40%**. This suggests that the model has a certain ability to generalize to unseen grasps during test. When we do the **rolling average** of the **8 frames** of each grasp sequence we obtain an accuracy of **42.96%**. The Top 2 accuracy is of 63.18%, which means that the model is likely to predict the correct class as either the first or the second option. The results are far from being perfect, but they are fairly superior to a random (14,28%) or a majority class prediction (21.87%).

It is clear that with the averaging process we can have more robust predictions because bad predictions from not favoring frames are softened by the averaging. With a deeper look into the results, we see that sometimes incorrect predictions have large probabilities, while the correct predictions of several frames do not win by a large margin. In those cases, the averaging process could lead to incorrect predictions. In those circumstances using majority voting could behave a bit better, especially if we predict a large number of frames. In our case, majority voting for 8 frames achieved an accuracy of 41.50%, which is a bit worse.

In Table 6 we have the confusion matrix of the test set. In Table 7 we have more detailed results in terms of precision, recall and F1 score of each class.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	191	4	45	16	46	2	27	331
pow-cir	21	16	51	4	24	0	4	120
pow-cyl	23	1	189	6	38	1	12	270
pow-obl	17	0	59	47	25	2	5	155
pre-cir	45	4	55	5	133	0	19	261
pre-pin	26	1	26	6	25	7	25	116
pre-pri	73	4	66	9	40	1	67	260
total	396	30	491	93	331	13	159	1513

Table 6: Test confusion matrix of the image model (8 frame rolling average) with random split

Class	Precision	Recall	F1 score
Intermediate Lateral	0.4823	0.5770	0.5254
Power Circular	0.5333	0.1333	0.2133
Power Prismatic Abducted	0.3849	0.7000	0.4967
Power Prismatic Adducted	0.5054	0.3032	0.3790
Precision Circular	0.4018	0.5096	0.4493
Precision Pinch	0.5385	0.0603	0.1085
Precision Prismatic	0.4214	0.2577	0.3198
AVG	0.4668	0.3630	0.3560

Table 7: Precision, Recall and F1 score of the image model (8 frame rolling average) with random split

From the results, we observe that the average class precision is superior to the average recall. The power-circular and precision-pinch classes are predicted rarely, but when they are, the prediction is often correct. That is why they have high precision but very low recall. On the other hand, int-lateral, pow-cylindric and pre-circular are predicted often, so they have a recall value higher than the precision one. Intermediate lateral is the class that is detected better in general since it has the highest F1 score. On the opposite side, the pinch precision grasps are the ones that behave worse.

We also study the confusion matrix to try to understand the similarities between grasps from the misclassifications. We see that power-oblique grasps are really often misclassified as power-cylindric grasps. This makes sense since they can look similar because they only differ significantly with respect to the thumb position. We also observe that pinch precision is often confused with prismatic precision grasps.

8.1.2 Cross-User Results

We train the model with the cross-user data split for 63 epochs. The best validation results are from epoch 42, in which the validation accuracy was 41.40%.

Figure 29: Training and validation loss of the image model with cross-user split

We observe a very significant drop in accuracy when we use the test data. In tables 8 and 9 we present more detailed results, with the confusion matrix and the precision, recall and F1 score metrics.

When we predict individually each frame the accuracy is 19.50%. If we predict by averaging the probabilities of the 8 frames of each grasp, we obtain an accuracy of 21.04% and a top-2 accuracy of 39.15%. If we instead predict the most frequent class of the 8 frames, the accuracy is 20.33%.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	66	17	68	0	60	0	74	285
pow-cir	61	25	56	0	63	0	47	252
pow-cyl	107	33	125	0	42	0	84	391
pow-obl	39	11	50	0	20	1	20	141
pre-cir	108	39	98	0	102	1	71	419
pre-pin	20	2	23	0	15	0	13	73
pre-pri	35	9	29	0	35	0	42	150
total	436	136	449	0	337	2	351	1711

Table 8: Test confusion matrix of the image model (8 frame rolling average) with cross-user split

Class	Precision	Recall	F1 score
Intermediate Lateral	0.1514	0.2316	0.1831
Power Circular	0.1838	0.0992	0.1289
Power Prismatic Abducted	0.2784	0.3197	0.2976
Power Prismatic Adducted	0.0000	0.0000	0.0000
Precision Circular	0.3027	0.2434	0.2698
Precision Pinch	0.0000	0.0000	0.0000
Precision Prismatic	0.1197	0.2800	0.1677
AVG	0.1480	0.1677	0.1496

Table 9: Precision, Recall and F1 score of the image model (8 frame rolling average) with cross-user split

It is evident that the model doesn't generalize well to the grasps done by unseen users. There is probably a big difference between the training and the test set in terms of scene and hand appearance, object set and grasp style. The accuracy is superior to a random prediction (14.28%), but not to majority class voting (24.48%). The precision, recall and F1 score metrics are also quite poor. For instance, the power-oblique and precision-pinch classes have precision and recall of 0.

This lack of generalization is an important drawback because it would limit severely its applicability in a real scenario, where the patients have different hand sizes and grasp styles.

8.1.3 Own dataset

Now we use the model trained with the random split of the Yale Grasp Dataset to predict the grasps from the dataset that we captured.

In table 10 we present the confusion matrix when we predict our own data with the image model and a rolling average of 8 frames. In table 11 we show other metrics.

We obtain very poor results. When we predict individually the frames, we obtain an accuracy of 13.27%. If we average the prediction for 8 frames, the accuracy is reduced to 12%. Its top-2 accuracy is 29.66%. If we instead select the most predicted class, the accuracy is slightly higher, with a 13.16%.

From the confusion matrix, we see clearly that our model doesn't generalize to the new data. We only predict power-cylindric and precision-circular grasps apart from just 3 exceptions. Even for those classes, the performance metrics are very poor.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	0	0	7	0	13	0	0	20
pow-cir	0	0	28	0	51	0	0	79
pow-cyl	0	0	59	0	72	0	0	131
pow-obl	0	0	4	0	15	0	0	19
pre-cir	0	0	38	0	13	0	0	51
pre-pin	0	0	98	0	62	0	0	160
pre-pri	2	0	77	1	60	0	0	140
total	2	0	311	1	286	0	0	600

Table 10: Test confusion matrix of the image model (8 frame rolling average) with our own dataset

Class	Precision	Recall	F1 score
Intermediate Lateral	0.0000	0.0000	0.0000
Power Circular	0.0000	0.0000	0.0000
Power Prismatic Abducted	0.1897	0.4504	0.2670
Power Prismatic Adducted	0.0000	0.0000	0.0000
Precision Circular	0.0455	0.2549	0.0772
Precision Pinch	0.0000	0.0000	0.0000
Precision Prismatic	0.0000	0.0000	0.0000
AVG	0.0336	0.1008	0.0492

Table 11: Precision, Recall and F1 score of the image model (8 frame rolling average) with our own dataset

8.2 Video model

8.2.1 Random Split

The video model is supposed to be a more complex approach since it deals with the temporal dimension. That is why we expected it to behave slightly better than the two other approaches, especially the simple image model.

We trained for 37 epochs. In figure 30 we have the training and validation loss. The best results in terms of validation loss were obtained in epoch 16. We observe that the training and validation loss curves are very similar to the ones obtained before. The training loss keeps decreasing, while the validation one stabilizes after a small number of epochs. We hence suffer again from overfitting. In epoch 16 the validation accuracy is 43.99%. We observe that at this epoch there is a drop between the training and validation loss, but is not as big as in the next epochs.

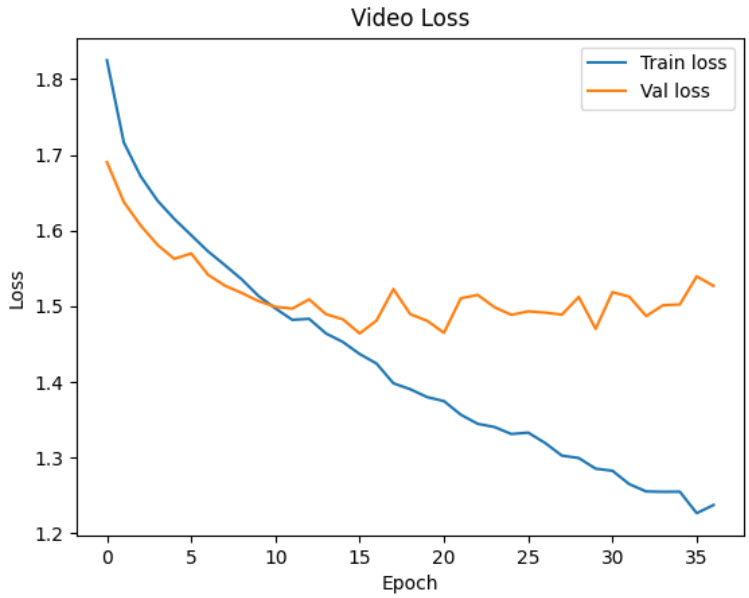


Figure 30: Training and validation loss of the video model with random split

With the test set, we predict 8 frames for each grasp sequence. The **accuracy** is **44.15%**. Again we see that the model is able to generalize to unseen grasp sequences. This time, the Top 2 accuracy is 65.56%. These metrics are a bit superior to the ones obtained in the image model.

In table 12 we present the precision, recall and F1 score values of each class. In table 13 we have the test confusion matrix.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	174	19	29	21	36	6	46	331
pow-cir	20	49	17	5	14	3	12	120
pow-cyl	40	17	157	11	31	1	13	270
pow-obl	14	6	45	65	13	2	10	155
pre-cir	40	19	37	11	125	6	23	261
pre-pin	34	6	11	11	17	7	30	116
pre-pri	74	19	26	17	30	3	91	260
total	396	135	322	141	266	28	225	1513

Table 12: Test confusion matrix of the video model with random split

Class	Precision	Recall	F1 score
Intermediate Lateral	0.4394	0.5257	0.4787
Power Circular	0.3630	0.4083	0.3843
Power Prismatic Abducted	0.4876	0.5815	0.5304
Power Prismatic Adducted	0.4610	0.4194	0.4392
Precision Circular	0.4699	0.4789	0.4744
Precision Pinch	0.2500	0.0603	0.0972
Precision Prismatic	0.4044	0.3500	0.3753
AVG	0.4108	0.4034	0.3971

Table 13: Precision, Recall and F1 score of the video model with random split

We see that with the exception of the precision pinch class, the rest of the classes obtain relatively good results in terms of precision, recall and F1 score.

8.2.2 Cross-User Split

We test the cross-subject generalization capabilities of the video model. We train during 41 epochs. The best validation results are obtained in epoch 24. At that step, the validation accuracy was 48.17%.

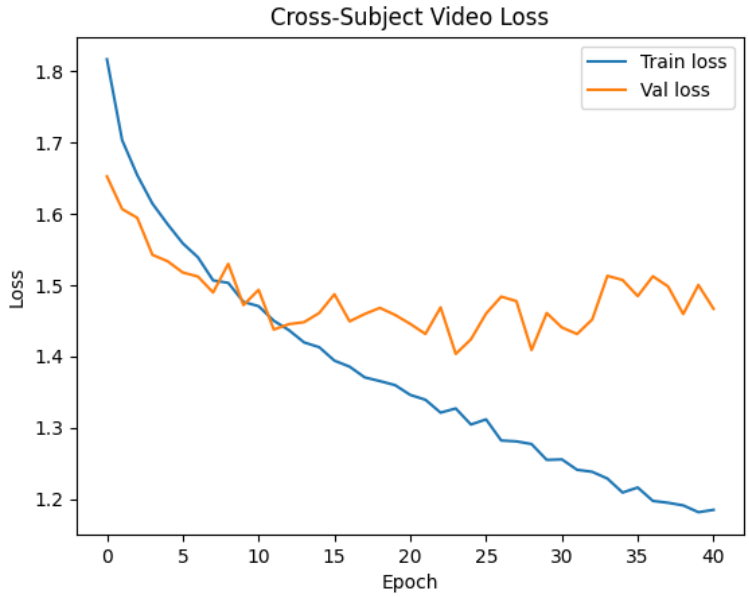


Figure 31: Training and validation loss of the video model with cross-user split

Then, we predict the test data, which is composed only by the grasps from the Housekeeper 2. The results are a top 1 accuracy of 29.69% and a top 2 accuracy of 48.50%. In tables 14 and 15 we present more detailed metrics.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	54	30	57	0	110	4	30	285
pow-cir	35	44	37	0	102	3	31	252
pow-cyl	36	42	100	1	182	5	25	391
pow-obl	21	10	38	0	64	0	8	141
pre-cir	35	26	46	0	292	5	15	419
pre-pin	13	7	14	1	25	2	11	73
pre-pri	32	12	18	0	65	7	16	150
total	226	171	310	2	840	26	136	1711

Table 14: Test confusion matrix of the video model with cross-user split

Class	Precision	Recall	F1 score
Intermediate Lateral	0.2389	0.1895	0.2114
Power Circular	0.2573	0.1746	0.2080
Power Prismatic Abducted	0.2558	0.1832	0.2853
Power Prismatic Adducted	0.0000	0.0526	0.0000
Precision Circular	0.3476	0.6969	0.4639
Precision Pinch	0.0769	0.0274	0.0404
Precision Prismatic	0.1176	0.1067	0.1119
AVG	0.1944	0.2073	0.1887

Table 15: Precision, Recall and F1 score of the video model with cross-user split

We observe again a significant drop in performance compared with the random split. Unlike with the image model, the accuracy is superior to a basic approach of predicting the most frequent class (29.69% vs 24.48%). In any case, the performance metrics are again quite poor. In particular, we observe that a really big part of the grasps are predicted as the precision-circular class.

8.2.3 Own dataset

Next, we test our own dataset with the video model trained with the random split. In table 16 we present the confusion matrix while in table 17 we show the precision, recall and F1 score metrics.

We get again very poor results. Unlike the image model, the video model predicts all the classes but with very low accuracy. We have a top 1 accuracy of 13.33% and a top 2 accuracy of 35%. We see again that the model doesn't generalize well to our captured video data.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	10	0	0	0	0	8	2	20
pow-cir	45	1	2	0	1	0	30	79
pow-cyl	81	1	24	3	2	3	17	131
pow-obl	8	0	0	1	0	2	8	19
pre-cir	36	0	0	0	0	0	15	51
pre-pin	101	3	1	0	0	16	39	160
pre-pri	63	6	17	0	1	25	28	140
total	344	11	44	4	4	54	139	600

Table 16: Test confusion matrix of the video model with our own dataset

Class	Precision	Recall	F1 score
Intermediate Lateral	0.0291	0.5000	0.0549
Power Circular	0.0909	0.0127	0.0222
Power Prismatic Abducted	0.5455	0.1832	0.2743
Power Prismatic Adducted	0.2500	0.0526	0.0870
Precision Circular	0.0000	0.0000	0.0000
Precision Pinch	0.2963	0.1000	0.1495
Precision Prismatic	0.2014	0.2000	0.2007
AVG	0.2019	0.1498	0.1127

Table 17: Precision, Recall and F1 score of the video model with our own dataset

8.3 Hand image model

8.3.1 Random Split

This approach is quite different compared with the two others. First of all, we train an image model with an auxiliary dataset composed of cropped images around the right hand while grasping. We trained for 45 epochs but we could have stopped earlier. In Fig. 32 we have the training and validation loss. Once again we see important overfitting in the loss plots. In this case, it is much heavier than the one we had previously. We use the weights from epoch 24 because is the one with lower validation loss. In that epoch, the validation accuracy was 42.77%.

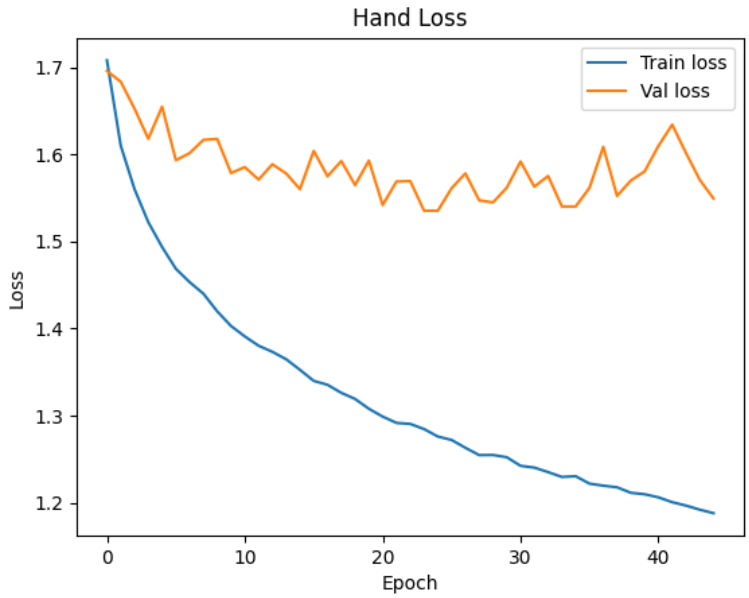


Figure 32: Training and validation loss of the hand model with random split

Next, we use the same test images as in the other two approaches. First, we apply the hand detector. If we detect a hand, we crop it and we pass the image to the hand grasp model predictor. Otherwise, we fit the image to the simple image model. Predicting at the single image level we obtain an accuracy of 29.78%. If we use an averaging window of 8 frames we obtain an accuracy of 36.61% and a top 2 accuracy of 57.96%. We observe that the results are significantly worse than the ones obtained with the simple image model and the video model.

Assuming that our hand detector doesn't find any hand, then the system would behave exactly as in 8.1. The improvement of the system depends fully on the precision of our hand detector and the accuracy of the hand grasp predictor.

The hand detector detects the hands in 63.21% of the images, which is a relatively good value given the difficulty of the dataset and the low resolution of the images. This means that 63.21% of the predictions are done by the hand grasp predictor, while the remaining 36.78% are done by the image predictor.

However, it is clear that our hybrid approach produces worse results than using just the simple image model. Without deeper analysis, it is difficult to apportion the blame. There are some factors that can hinder the performance of the hand model. First of all, since the hand detector used is not perfect, a small percentage of the cropped images that are fed into the model will be inaccurate (incorrect hand detection or left and right-hand confusion). Apart from that, the hand model may be worse than the image model, so a considerable part of the predictions (63.21%) have a bigger uncertainty. Lastly, it is possible that the predictions from both models do not complement appropriately.

In Table 18 we have the test confusion matrix, while in Table 19 we present the precision,

recall and F1 score values of each class.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	167	4	68	4	47	0	41	331
pow-cir	18	10	56	3	26	0	7	120
pow-cyl	41	3	171	2	34	0	19	270
pow-obl	25	0	67	17	16	0	30	155
pre-cir	53	1	70	3	110	0	24	261
pre-pin	33	1	31	2	20	2	27	116
pre-pri	83	5	54	2	38	1	77	260
total	420	24	517	33	291	3	225	1513

Table 18: Test confusion matrix of the hand model (8 frame rolling average) with random split

Class	Precision	Recall	F1 score
Intermediate Lateral	0.3976	0.5045	0.4447
Power Circular	0.4167	0.0833	0.1389
Power Prismatic Abducted	0.3308	0.6333	0.4346
Power Prismatic Adducted	0.5152	0.1097	0.1809
Precision Circular	0.3780	0.4215	0.3986
Precision Pinch	0.6667	0.0172	0.0336
Precision Prismatic	0.3422	0.2962	0.3175
AVG	0.4353	0.2951	0.2784

Table 19: Precision, Recall and F1 score of the hand model (8 frame rolling average) with random split

Even if the results are worse than with the other two approaches, we think that the evidence is not strong enough to completely discard the cropped hands as a data modality. In particular, the low video quality of the dataset makes it difficult to train a model of these characteristics. It would be very interesting to investigate further this approach with other data more appropriate for this purpose.

8.3.2 Cross-User Split

Given that the results were quite inferior to the other two approaches and that it takes a long time to generate the auxiliary dataset (around 4/5 days), we opted to skip this experiment due to time issues. Anyway, we expected a similar pattern to the other two models, with a performance that degraded significantly.

8.3.3 Own dataset

Lastly, we test our own dataset with the hand model. In table 20 we present the confusion matrix while in table 21 we show the precision, recall and F1 score metrics.

The results are again very poor. When we predict individually every frame the accuracy is 19%. If we average the prediction of 8 frames, the top 1 accuracy is 17.08% and the top 2 accuracy is 26.83%. If we instead pick the most selected class among those 8 frames, the accuracy is 17.5%.

The hand model predicts mostly the same 2 classes as the image model, but this time it predicts more often the power cylindric class, which is more present in the dataset and therefore it obtains better accuracy results.

The most interesting thing is that in our dataset, the hand detector that we use finds the hand in 90.93% of the images. This means that with higher quality images and a cleaner setup the detector works much better than with the Yale Grasp Dataset.

	int-lat	pow-cir	pow-cyl	pow-obl	pre-cir	pre-pin	pre-pris	total
int-lat	0	0	20	0	0	0	0	20
pow-cir	0	0	66	0	13	0	0	79
pow-cyl	1	0	111	0	19	0	0	131
pow-obl	2	0	12	0	5	0	0	19
pre-cir	5	0	46	0	0	0	0	51
pre-pin	1	0	133	0	26	0	0	160
pre-pri	6	0	117	0	14	0	3	140
total	15	0	505	0	77	0	3	600

Table 20: Test confusion matrix of the hand model (8 frame rolling average) with our own dataset

Class	Precision	Recall	F1 score
Intermediate Lateral	0.0000	0.0000	0.0000
Power Circular	0.0000	0.0000	0.0000
Power Prismatic Abducted	0.2198	0.8473	0.3491
Power Prismatic Adducted	0.0000	0.0000	0.0000
Precision Circular	0.0000	0.0000	0.0000
Precision Pinch	0.0000	0.0000	0.0000
Precision Prismatic	1.0000	0.0214	0.0420
AVG	0.1743	0.1241	0.0559

Table 21: Precision, Recall and F1 score of the hand model (8 frame rolling average) with our own dataset

8.4 Summary

As we have seen from the results, the achieved metrics are not particularly good in any of the approaches that we tried, but we appreciate some performance differences that are interesting to analyze.

Data	Metric	Image model	Video model	Hand model
Random split				
	Accuracy	0.4296	**0.4415**	0.3661
	Top-2 accuracy	0.6318	**0.6556**	0.5796
	Avg F1 score	0.3560	**0.3971**	0.2784
Cross-user split				
	Accuracy	0.2104	**0.2969**	-
	Top-2 accuracy	0.3915	**0.4850**	-
	Avg F1 score	0.1496	**0.1887**	-
Own dataset				
	Accuracy	0.12	0.1333	**0.1708**
	Top-2 accuracy	0.2966	**0.35**	0.2683
	Avg F1 score	0.0492	**0.1127**	0.0559

Table 22: Results summary

We see that the video model obtains the best results across all the metrics of the different data, with the exception of the accuracy in our dataset, which is not very reliable. The difference is not very big but it seems enough to justify that is the superior approach. This is interesting because it suggests that the use of the temporal dimension of the data can be beneficial for grasp recognition. Even if the video model is the more complex model (more parameters), during the training phase it had fewer trainable parameters than the two others.

The image model obtains the second best results but they are only closer to the video model if we use the rolling average method. Lastly, the hybrid hand model seems to be the worst one. However, there are some external factors that can hinder its performance.

We have seen that none of our models generalized to our own dataset. Data is probably too different. First of all, the camera position is different (head vs chest). This leads to different camera angles that change the appearance of the hands and objects. Secondly, there is a big difference in the scene looking. Also, there are small differences in the grasp style. Moreover, the object set is very different.

To have a model that can predict more accurately human grasps with our setup we would need to do something different. One possibility would be to further fine-tune our trained models with our own data. The other would be to train or fine-tune a new model from scratch with our own data. The second option would be preferable but first it would require to capture a lot of data, which takes time and effort.

9 Conclusions and future work

9.1 Conclusions

In our work, we have focused on the problem of grasp recognition to be applied in the medical field as a tool for easing the monitoring process of patient rehabilitation with upper-limb neurological disorders. We have tried three different approaches for performing grasp recognition and as far as we know this may be the first time that video models have been used for that purpose.

We have used the Yale Grasp Dataset for training our models. It is the largest existing egocentric human grasp dataset with RGB video. It is captured in an unstructured environment so the accuracy of the models is a reflection of their applicability in a real-world environment and not in a controlled scenario. This dataset is a bit outdated in terms of image quality but very probably it was the best available option for our purposes. Anyway, the lack of a really large and modern egocentric human grasp video dataset is probably the biggest point that hinders the possibility of having better models and it is certainly the first issue that must be addressed if we want to keep investigating.

We have proposed a grasp classification with a coarse-grained grouping of a subset of the most important grasps of the Feix taxonomy. We have 7 different classes that reflect the need for the grasping (power or precision) and the object shape (circular, prismatic...). We believe that our grouping is consistent across those characteristics but it is heavily influenced by the grasp proportion and availability in the dataset. The possibilities of doing something different are then quite limited because of it.

Despite the reduced number of classes, we have seen from the results that the task was quite difficult. The best accuracy was obtained with the video model (44.15%), which emphasizes the importance of the temporal dimension for grasp recognition. On the other hand, the image model showed closed results when we used an averaging strategy with several frames (42.96%). Lastly, the hand model obtained much worse results (36.61% of accuracy), but these can be conditioned to some of the handicaps that we have already mentioned, so it is not completely prudent to discard this approach.

When we tested the cross-user generalization capabilities of the models we saw that the accuracies dropped significantly. This gives an idea of the difficulty of the task and that maybe the way we are dealing with the problem is not the most appropriate.

To end, we have captured our own small first-person human grasp video dataset with a GoPro camera located in the chest. We have labeled the grasps with our proposed grasp classification. It has been recorded in a simple controlled environment but is a good entrance to see if the models that we have trained with a dataset with different camera location (chest vs head) work well directly or if it is necessary to perform some fine-tuning or training from scratch with new data. From the results, it has been really clear that the models didn't generalize at all to the new data so we must try something different if we want to achieve something that can be applicable to the hospital setup.

In conclusion, the results showcase that our work is not enough to deploy in a production

environment a successful grasp recognition system from egocentric RGB input. Our final objective even if it was too optimistic, was to train a model with enough accuracy so that it could be used in the rehabilitation sessions at the hospital for performing grasp recognition. We see that with a reasonable number of classes we are quite far from obtaining that. Even with those mixed feelings mainly because of the results, we are conscious that this work was a good starting point for understanding better the several different aspects of grasping and tackle the quite unusual problem of grasp recognition. We have faced several challenges that can encourage us to explore many different directions to further improve the solution.

9.2 Future Work

Grasp recognition is a really complex interdisciplinary task, and as such we have just seen the tip of the iceberg, so there is plenty of things that can be explored in order to improve the performance of the system.

One of the biggest limitations that we have is the absence of a really good human grasping dataset. All the existent grasp datasets have some shortcomings that limit considerably the possibility of having a better model. We should create from scratch a new large dataset that addressed some of the limitations of the existing ones. First of all, it would be nice if we can capture it in the hospital facilities with a camera located in the chest. It will probably be easier to capture and label it if we do it in structured sessions. We could dispose of a large object set from which we can grasp each object many times and in many ways. It would be really nice if we also can label some precise information about the object such as its size, weight, deformability... The dataset should be really large and as much balanced as possible. It would be really interesting to have several subjects of different ages, height, hand size, color skin... We should use a fine-grained labeling such as the original Feix taxonomy so that if it is necessary we can later group in more coarse-grained classes. Lastly, it would be really profitable if we can capture different data modalities. We need at least RGB video, but capturing depth data would enrich the dataset. If we want to open the possibility to study grasp recognition from a hand pose perspective, it would be interesting to put discreet sensors in both hands so that we know the position of the 21 joints of the hands while we do the grasps.

The problem of capturing a dataset from scratch is that even if it is preferable, it can take a long time and a lot of resources. Instead, we could do the labeling of an existing one. One good candidate could be Epic Kitchens, which is a really complete egocentric dataset that with some effort could be labeled to have a nice grasp recognition dataset. Another example would be to add grasp labels to the dataset by [30]. This would allow to perform grasp recognition on a dataset with RGB-D and hand sensor data (hand-pose), which is a really interesting combination.

Apart from that our model has been trained with data that labeled only the right-hand grasps, which limits its applicability to a real-world scenario, where left-hand grasp, both-hand grasp or multi-grasping (one object in each hand or more than one object per hand) are also very important. It could be interesting to consider adding also these modalities in future datasets.

As pending work, we have to really see if it is worth fine-tuning the trained models with our own dataset from the chest, or if it is better to train from scratch. In any case, we would need to extend the dataset that we have captured.

We have worked with our own grasp grouping, which uses a subset of grasps of the Feix taxonomy. We could try to classify them differently or to select fewer of them if we want to focus only on a restricted group. Additionally, we could study other existent taxonomies that define the concept of grasp differently.

Our models have been trained with frame sequences with a certain (only one) grasp class. We didn't have a case for no-grasp nor we have considered the case where our input sequence contains a transition between two grasps. Therefore our models could only work if they are fed with video data that follow those assumptions. Of course, it is always possible to make someone watch the video and manually annotate the start and end frame of each grasp (not the class) so that this data can be fed into our models. However, this is not a scalable approach. That is why we should automate this task. The solution could be to train a video segmentation model. This model would receive as input an egocentric video with a rehabilitation session of a patient interacting with objects. As output, we desire to temporally segment the video between the grasp and no-grasp classes. This would consist of giving the start and end frame of each grasp performed. After, we can crop the video with those frames and pass them to the models we have trained. Temporal segmentation of videos is one of the fields of study for Deep Learning Video Models and in the literature there exist several proposals that trained with the appropriate data could be worth trying. In fact, instead of having a binary temporal segmentation video model (grasp and no-grasp classes), we could have one that segmented the video between our 7 classes plus the no-grasp class. However, it would be more complex to do it.

We have tested three approaches for grasp recognition and we've seen that the video model obtained the best results. The architecture that we have used is not exactly the one that obtains the highest accuracy in the action recognition video dataset benchmarks. Therefore, we could try with other more advanced architectures to see how far we can get. In particular, it would be interesting to use the SlowFast architecture, which differs from Slow by using a Fast video stream, that captures better the temporal information. This way we could see more clearly the potential of capturing the temporal dimension for grasp recognition. Besides, we could explore the possibility of using hand recognition as a first step for training a video model instead of just an image model.

Another thing to consider is to explore different treatments of the temporal dimension. We have used frames that were extracted uniformly between the start and end frame of each grasp, regardless of its real duration. This means that for long grasps we have captured distant frames in time, while for short grasps, the frames are very close temporally. This makes our system rather inconsistent but is the simplest way to deal with sequences that have different duration. There are many possibilities to explore. For instance, we could draw frames with the same temporal resolution from either the beginning, the middle or the end of the action. We could also try different temporal resolutions or number of frames.

We also could further study the usage of other data modalities for grasp recognition. One

possibility could be to calculate the optical flow and use a two-stream architecture. Something more interesting would be using depth data. The first possibility would be using an egocentric RGB-D camera. We also could use what is called a Monocular Depth Estimation Model. It is a model that given one or several RGB images, it estimates the depth channel. Of course, they are not perfect, and there is still a reasonable error margin, but in the late years they have developed considerably.

In a first-person setup for grasping, most part of the relevant things that happen occur in short distances. This is because the elements of our body that are relevant to the action (mainly the arms or hands) will be always inside a fixed short range to the camera, which can be held on the head or the chest. This will vary depending on the height and the length of the arms of the person, but in general, it will not change much.

This depth data could be used in our grasp recognition models by adding an extra depth channel to them, or by adding a new independent branch and then fusing the results from both the RGB and the depth streams. Another possibility would be using depth data for assisting in the process of hand detection. Given that the hands will always be in a fixed range with respect to the camera, it would be possible to crop some undesired parts of the images so that we eliminate pixels that can disturb the hand recognition process. It is unclear the improvement capacity of the system if we applied these ideas, but adding this extra information will probably help even if it is not much.

Last, and probably most importantly, it would be very interesting to study the usage of hand pose estimation and object detection for grasp recognition. Grasp recognition is very related to the hand pose estimation problem, with the particularity that it should be done while grasping an object, which is heavily difficulted because of the hand-object occlusions. Work like [31] and [30] have tried to estimate hand pose while interacting with objects. If we can develop a model that can more or less reliably infer hand position under those conditions, it will be really helpful for grasp recognition. The 3D positions of the hand bones would be very powerful information and could be either used to complement the RGB data or standalone. Moreover, the uncertainty of a particularly bad frame could be overcomed by using a sequence of several frames.

On the other hand, object detection could also be helpful for grasp recognition. The grasps that we do are conditioned in great measure by the characteristics of the object that we grasp. Being able to recognize the dimensions, shape and type of the object could be very beneficial to give some context and restrict the plausible grasp space.

References

[1] Vaudois Regional Hospital. https://www.chuv.ch/fr/chuv-home. Accessed: 2021-03-02.

[2] Physilog 5 motion sensor. https://research.gaitup.com/physilog/. Accessed: 2021-03-01.

[3] Jason Brownlee. A Gentle Introduction to Object Recognition With Deep Learning. https://machinelearningmastery.com/object-recognition-with-deep-learning/. Accessed: 2021-03-12.

[4] Ilija Mihajlovic. Everything You Ever Wanted To Know About Computer Vision. https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-a Accessed: 2021-03-01.

[5] Shivam Sharma. Deep Learning Architectures for Action Recognition. https://towardsdatascience.com/deep-learning-architectures-for-action-recognition-83e5061ddf90. Accessed: 2021-03-12.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. pages 6201–6210, 10 2019.

[7] Taha Anwar. Introduction to Video Classification and Human Activity Recognition. https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/. Accessed: 2021-03-11.

[8] Dima Damen, Hazel Doughty, Giovanni Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020.

[9] Thomas Feix, Javier Romero, Heinz-Bodo Schmiedmayer, Aaron M. Dollar, and Danica Kragic. The GRASP Taxonomy of Human Grasp Types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.

[10] Ian M. Bullock, Raymond R. Ma, and Aaron M. Dollar. A Hand-Centric Classification of Human and Robot Dexterous Manipulation. *IEEE Transactions on Haptics*, 6(2):129–144, 2013.

[11] Artur Saudabayev, Zhanibek Rysbek, Raykhan Khassenova, and Huseyin Atakan Varol. Human grasping database for activities of daily living with depth, color and kinematic data streams. *Scientific Data*, 5(1), 2018.

[12] Ian M. Bullock, Thomas Feix, and Aaron M. Dollar. The Yale human grasping dataset: Grasp, object, and task data in household and machine shop environments. *The International Journal of Robotics Research*, 34(3):251–255, 2014.

[13] The Yale human grasping dataset: Download link. `https://www.eng.yale.edu/grablab/humangrasping/`. Accessed: 2021-04-15.

[14] Human grasping database for activities of daily living with depth, color and kinematic data streams: Download link. `https://figshare.com/collections/_/3858397`. Accessed: 2021-04-12.

[15] Minjie Cai, Kris M. Kitani, and Yoichi Sato. An Ego-Vision System for Hand Grasp Analysis. *IEEE Transactions on Human-Machine Systems*, 47(4):524–535, 2017.

[16] UT Grasp Dataset V2: Download link. `https://drive.google.com/drive/folders/164iMs9bR1QTGd2muZtk6YoH7MscAWngu`. Accessed: 2021-04-26.

[17] Ian M. Bullock, Joshua Z. Zheng, Sara De La Rosa, Charlotte Guertler, and Aaron M. Dollar. Grasp Frequency and Usage in Daily Household and Machine Shop Tasks. *IEEE Transactions on Haptics*, 6(3):296–308, 2013.

[18] Thomas Feix, Ian M. Bullock, and Aaron M. Dollar. Analysis of Human Grasping Behavior: Object Characteristics and Grasp Type. *IEEE Transactions on Haptics*, 7(3):311–323, 2014.

[19] Thomas Feix, Ian M. Bullock, and Aaron M. Dollar. Analysis of Human Grasping Behavior: Correlating Tasks, Objects and Grasps. *IEEE Transactions on Haptics*, 7(4):430–441, 2014.

[20] F. Cini, V. Ortenzi, P. Corke, and M. Controzzi. On the choice of grasp type and location when handing over an object. *Science Robotics*, 4(27), 2019.

[21] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015, December 2014.

[22] Enric Corona, Albert Pumarola, Guillem Alenya, Francesc Moreno-Noguer, and Gregory Rogez. Ganhand: Predicting human grasp affordances in multi-object scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020.

[24] Mo Han, Sezen Yağmur Günay, Gunar Schirner, Taşkın Padır, and Deniz Erdoğmuş. HANDS: a multimodal dataset for modeling toward human grasp intent inference in prosthetic hands. *Intelligent Service Robotics*, 13(1):179–185, 2019.

[25] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. Contactdb: Analyzing and predicting grasp contact via thermal imaging. In *CVPR*, 2019.

[26] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J. Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344, 2020.

[27] Bardia Doosti. Hand pose estimation: A survey. `https://arxiv.org/pdf/1903.01013.pdf`, 2019.

[28] Google Mediapipe: ML solutions. `https://google.github.io/mediapipe/`. Accessed: 2021-08-11.

[29] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *The European Conference on Computer Vision (ECCV)*, August 2020.

[30] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2018.

[31] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3203, 06 2020.

[32] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[33] Ryan J. Visee, Jirapat Likitlersuang, and Jose Zariffa. An Effective and Efficient Method for Detecting Hands in Egocentric Videos for Rehabilitation Applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(3):748–755, 2020.

[34] Minjie Cai, Kris M. Kitani, and Yoichi Sato. A scalable approach for understanding the visual structures of hand grasps. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 1360–1366, 2015.

[35] Minjie Cai, Kris M. Kitani, and Yoichi Sato. Understanding hand-object manipulation with grasp types and object attributes. In *Robotics: Science and Systems*, 2016.

[36] Grégory Rogez, James S. Supancic, and Deva Ramanan. Understanding everyday hands in action from rgb-d images. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[37] Chiho Choi, Yoon Sang Ho, Chen Chin-Ning, and Karthik Ramani. Robust hand pose estimation during the interaction with an unknown object. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[38] Pytorchvideo Model Zoo. `https://pytorchvideo.readthedocs.io/en/latest/model_zoo.html`. Accessed: 2021-04-06.

[39] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. pages 12038–12047, 06 2019.