UNIVERSITÀ DEGLI STUDI DI CATANIA
Dipartimento di Matematica e Informatica
Corso di Laurea Triennale in Informatica

Dario Lazzara

Understanding Egocentric Videos in Surgery Domain

Relazione Progetto Finale

Relatore: Prof. Giovanni Maria Farinella
Correlatore: Prof. Francesco Ragusa

Anno Accademico 2024 - 2025

Understanding Egocentric Videos in Surgery Domain

Dario Lazzara

26/09/2025

”Propizio è essere perseveranti nelle avversità.”

Dall’esagramma 36, ”L’oscuramento della luce”, I Ching

Abstract

This thesis aims to study and apply **machine learning techniques** in the **medical field**, with a particular focus on a still underexplored area: the analysis of **egocentric videos** of surgical procedures. Compared to **minimally invasive surgery (MIS)**, for which there is extensive literature and consolidated datasets, including synthetic ones, **open surgery** presents unique challenges in both *data collection* (and related *annotation*) and *model training*. The purpose of this research is to evaluate the potential for **knowledge transfer** from pre-trained models and the effectiveness of specific *learning strategies*.

The presented study relied on the **EgoSurgery** [15] [14] [10] dataset, which consists of approximately **15 hours** of *egocentric videos* enriched with various annotations: *surgical phases, bounding boxes for hands and instruments*, and *gaze data* recorded through an *eye-tracking system*. These annotations were further complemented by the *segmentation* of **15 surgical instruments** and **4 hand categories**, distinguished by ownership and laterality, obtained using the **Segment Anything (SAM)** [36] model developed by *Meta*. Building such a dataset posed significant challenges, due both to the strong **visual variability** typical of surgical scenes and to the need to safeguard the **privacy** of the subjects involved. The performance of several pre-trained models, with and without optimization, was tested, including **HOS** [11], originally developed for **EPIC-KITCHENS** [9], as well as models trained on generic datasets such as *COCO* [29]. In particular, **HTC** [4], **Mask R-CNN** [21], **QueryInst** [13], and **PointRend** [24] were considered, with **ResNet-50**[20] serving as the baseline backbone.

These models were evaluated for their ability to recognize and generalize across the **19 categories** included in the dataset, both in the more general distinction between *tools* and *hands*, and in relation to the potential improvements achievable through *fine-tuning*. The overall results indicate that direct transfer from one domain to another, without *fine-tuning*, produces limited performance due to the profound differences between domains. Conversely, applying targeted *fine-tuning* yielded only marginal improvements over the

baseline established by the **EgoSurgery** [10] team, thereby confirming the difficulty of the task.

These findings highlight two key aspects: the **limited transferability** across highly distinct visual domains, and the consequent need to design **domain-specific models** and **adaptation strategies** tailored to this type of scenario.

Contents

Introduction

In recent years, **computer vision**, a branch of *artificial intelligence (AI)*, has made significant progress in several fields. Among the most interesting is **egocentric vision** (or *first-person vision*), a subfield that analyzes videos captured by head-mounted cameras from the point of view of the person performing the action (POV). This means not observing events *"from the outside"*, as typically happens with traditional datasets, but putting yourself in the shoes of the person carrying out the action. This perspective is particularly compelling. It captures information about not only objects and the environment, but also about how people interact with them. It is as if the model were not only required to *'recognize a spoon on the table'* from the observer's perspective, but also to understand how it is held, how it is used and what sequence of actions it belongs to, for example, when preparing a recipe. It is not surprising that in recent years this has become a central topic of research, ranging from **personal assistant systems** using wearable devices to more advanced applications such as **human-to-robot learning** [9]. Concretely, we can imagine a *kitchen assistant* that guides the user step by step through the preparation of a recipe, suggesting subsequent actions in real time. Or consider an *occupational safety system* capable of monitoring delicate procedures and intervening in the event of an error or accident. The advantage in these cases lies not only in the ability to recognize an object, but in doing so **from a first-person perspective**, rather than relying on external instructions [9].

However, looking at the research landscape, most of the data sets and models developed so far have focused on *daily scenarios* (such as the well-known research from the **EPIC-KITCHENS** [9] team) or on industrial environments. In reality, the potential application fields are much broader, and one in particular is particularly promising: the **medical domain**. Consider for a moment what it would mean to have such a system in the operating room. A surgeon could receive real-time support in recognizing instruments or correctly executing a sequence of steps. Similarly, residents could benefit from a virtual tutor who can provide feedback during training, while nursing

staff could be assisted in following complex protocols. Not less importantly, these systems could ensure compliance with safety and hygiene standards, minimizing the risk of human error [16] .

However, as expected, this scenario is also among the most challenging. Videos of open surgeries, for example, are much more complex than those filmed in kitchens or laboratories. Lighting conditions vary greatly: The operating room is usually well lit, but only in specific areas, leaving many details in shadow. The hands of surgeons and assistants move quickly and frequently, making it difficult to apply automatic segmentation methods. Moreover, many instruments are similar in shape and color: scissors and needle holders, for example, are easily confused; gauze changes appearance depending on how it is folded or soaked [14]; retractors and forceps can vary significantly from one procedure to another. To make matters even more difficult, the diversity of surgical procedures and environments further increases visual complexity, making it extremely challenging to directly transfer knowledge acquired from models trained in datasets such as **HOS** [11].

The situation is further complicated by the scarcity of data. Recording - and especially annotating - medical videos, such as surgical footage, is neither straightforward nor cost-effective. In addition to this, ethical and privacy concerns must be addressed. All of these factors make it difficult to obtain datasets as large and diverse as those available in other fields. And this is where one of the central aspects of this thesis comes into play: **domain shift**. The basic idea is straightforward: When available data are limited and, as in this case, it is not cost-effective to train a model from scratch, the question arises whether it is possible to exploit pre-trained weights from other contexts (*for example, kitchens, daily activities, or industrial environments*) and transfer this knowledge to the *medical domain*, with or without *fine tuning*. The key issue, therefore, is to what extent a model trained in a *general domain* can be adapted to a model as diverse and complex as surgery.

A crucial starting point for our experiments will be the weights provided by the **EPIC-KITCHENS (VISOR-HOS)** [11] team and, inevitably, more general models such as those trained on **MS-COCO** [29]. To this end, we will rely on the studies conducted by the **EgoSurgery** [10] team, which made available a dataset of approximately **15 hours** of annotated videos. Based on this resource, we will analyze the performance of models such as **Mask R-CNN** [21] and **QueryInst** [13], as well as more advanced segmentation architectures such as **PointRend** [24] and **Hybrid Task Cascade (HTC)** [4]. The general objective is not only to assess the feasibility of *domain transfer* but also to identify which design choices, depending on the task, are most effective.

The structure of this thesis reflects this approach: It begins with a pre-

sentation of the *state of the art* and the main works that have laid the foundations, both in the medical field and in more general contexts of *egocentric vision*. Then it describes the data sets used and the models examined, followed by the *methodology* and *experimental setup*. Finally, the *results* obtained will be discussed, along with potential future directions.

Chapter 1

Related Work

As mentioned, this chapter presents some of the key research in the field. We begin with the **EPIC-KITCHENS** [9] dataset, a benchmark in the home and kitchen domain for object and interaction recognition. We then focus on the research conducted by the **EgoSurgery** [10] team, whose dataset is the subject of this thesis, analyzing the results obtained in different areas and approaches: *Surgical Tool Detection [16], EgoSurgery-Tool [14], EgoSurgery-Phase [15]*, and *EgoSurgery-HTS* [10].

1.1 EPIC-KITCHENS

1.1.1 Origins

The **EPIC-KITCHENS** [9] project was launched in 2018 with the aim of creating a large dataset of *egocentric* (first-person) videos, addressing the significant shortage of similar resources. Previously, existing datasets were often limited in size or collected in monotonous and artificial settings.

EPIC-KITCHENS [9], however, stands out for its scale and diversity: it contains more than **55 hours of recordings**, more than **11 million frames**, and involves **32 participants** from different cities (*Bristol, Catania, Seattle*, and *Toronto*). This heterogeneity introduces cultural and contextual variation: tools, ingredients, and preparation methods differ from country to country, as do the kitchens themselves, with different lighting, colors, and furnishings.

1.1.2 Data Collection

Data collection focused on recording **daily unscripted cooking activities** using a head-mounted **GoPro**, providing an *egocentric* and realistic view of

object interactions. Participants were asked to cook alone and to avoid any elements that could reveal their identity (*for example, mirrors*). The videos, which included both **visual and audio data**, were captured in **Full HD at 60 fps**, ensuring the high quality required for subsequent annotation phases.

1.1.3 Annotations

Two types of annotation are central. The first relates to **action segments**, defined as time intervals in a video during which a single action is performed (*for example, from time t1 to time t2, the action cut takes place*).

The annotation process unfolded in several stages. First, participants narrated their actions after recording, using concise descriptions such as *"cutting an onion"* (*verb + noun*) (Figure 1.1). These narrations were considered the most reliable source for labeling. The transcripts, produced in multiple languages, were then translated using services such as **Amazon Mechanical Turk (AMT)** [8] and automatically aligned with *YouTube* tools. Finally, a refinement step was carried out, applying constraints such as a minimum duration of **0.5 seconds** for each action and avoiding overlaps. In total, approximately **40,000 action segments** were annotated, with an average length of **3.7 seconds**.

The second type of annotation concerns the **bounding boxes of active objects**, which identify the items the user interacts with (or is about to interact with). This task was also performed via **AMT** [8], with three annotators per bounding box to ensure quality control, and subsequently linked to the action segments. Overall, around **455,000 annotated bounding boxes** were produced, covering **125 verb classes** and **331 noun classes** [1].

1.1.4 Benchmarks and Proposed Challenges

The authors introduced three main evaluation tasks, thus providing the scientific community with a common benchmark.

The first is **Object Detection**, namely the recognition of active objects within the scene. To address this problem, the **Faster R-CNN** [37] network was employed with a **ResNet-101** [20] backbone pre-trained on **MS-COCO** [29]. Training was performed with an initial learning rate of *0.0003*, reduced by a factor of **10** after **90,000** iterations, for a total of **120,000** iterations, using a minibatch of **four** across eight *NVIDIA P100 GPUs*. Performance was assessed using the standard **mAP (mean Average Precision)** metric

[1] (*For further details on quality control and on the definition of verb and noun classes, please refer to the authors' original article.*)

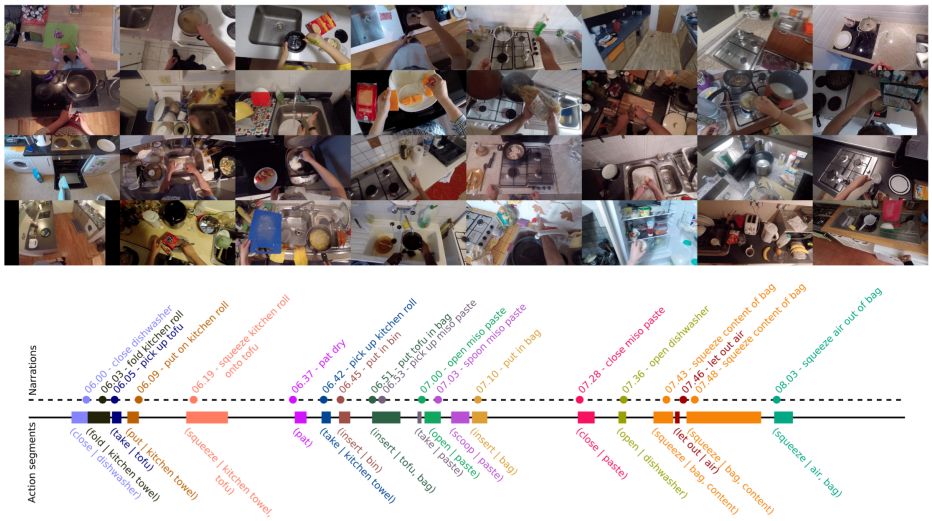


Figure 1.1: Frames from the Epic-Kitchens dataset with participant narrations (Adapted from: **Scaling Egocentric Vision: The EPIC-KITCHENS Dataset** [9]).

[29]. The results reveal several nuances: for instance, certain classes (*such as spoons and knives*) proved difficult to detect despite their frequency, whereas the model achieved comparable performance on both visible and non-visible kitchens (*unseen during training*), indicating good generalization ability.

The second task is **Action Recognition**, which relates to classifying actions within a specific time interval, such as *[t1, t2]*. A **Temporal Segment Network (TSN)** [46], implemented in *PyTorch* [35] with an **Inception** [42] architecture pre-trained on **ImageNet** [39], was adopted. Two separate streams were trained: a *spatial stream* to capture visual information ($LR = 0.01$) and a *temporal stream* to analyze motion dynamics ($LR = 0.001$). The results show that correctly associating *verbs* with *nouns* remains a significant challenge, underscoring the intrinsic complexity of the task.

Finally, **Action Anticipation**, the most demanding of the three tasks, aims to predict the next action before it occurs. This requires not only recognizing the ongoing action but also understanding the broader context and possible intentions of the user. The *egocentric perspective* provides a unique advantage here, since the hands and their interactions with objects serve as primary cues to anticipate subsequent actions. The results highlight both the considerable potential of the data set and the fact that existing methods are still far from solving these challenges with high accuracy.

1.2 VISOR

1.2.1 Origins

Following the success of **EPIC-KITCHENS** [9], in 2021 the same authors decided to further enrich the annotations of the dataset by introducing *pixel-level segmentation*. The goal was to precisely segment hands and active objects, tracking their movements frame by frame. This work took approximately **22 months** and grew out of the desire to study not only the presence of objects in the scene, but also their evolution over time.

For example, consider an onion: From the moment it is removed from the refrigerator until it is cut, it undergoes various transformations in shape, texture, and sometimes even color. With this type of annotation, it also becomes possible to study **hand-object relationships**, that is, to understand when and how contact occurs.

1.2.2 Annotation Process

Manually annotating each individual frame would have been particularly time-consuming and costly. For this reason, the team developed a **semi-automatic annotation pipeline** that combined *human intervention* with **AI-assisted tools**. The videos were first divided into subsequences of *three actions each*. By analyzing previously collected **action narratives**, it was possible to identify the **active objects** to be segmented, for a total of **257 object classes** and **13 hand classes**. The most time-consuming step remained **segmentation**. To accelerate this phase, the team used **TORAS** (TORonto Annotation Suit) [19], an assisted annotation tool that automatically interpolated intermediate frame masks from manually annotated masks, thus *drastically reducing the workload*. This process produced approximately **272,000 manual masks** and **9.9 million interpolated masks** (Figure 1.2).

Finally, the annotators performed a **quality control and refinement phase** to ensure the reliability of the dataset. The outcome is a **rich and detailed resource** that significantly expands the usability of *EPIC-KITCHENS* [9].

1.2.3 Benchmarks

This research also defined some benchmarks for several challenges.

The first, called **Video Object Segmentation (VOS)**, focuses on tracking active objects throughout a sequence. One of the main difficulties lies

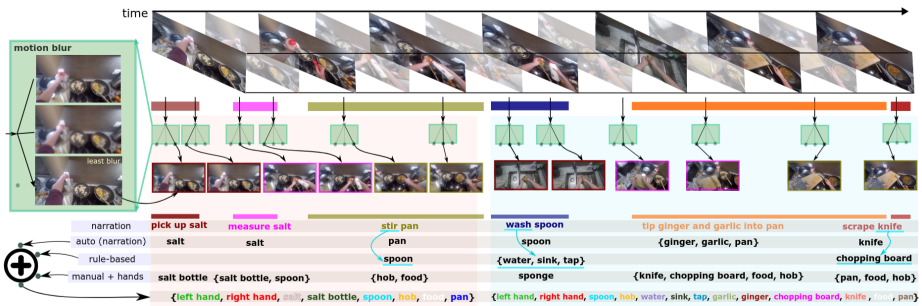


Figure 1.2: VISOR annotations pipeline (Adapted from: **EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations** [11]).

in *occlusions*, i.e., moments when an object temporarily disappears behind hands or other objects in the scene and then reappears. An *STM* [34] model was used as a basis, initially pre-trained on *COCO* [29] and then refined with a *ResNet-50* [20] backbone. Without this second phase, the results were rather poor, since some objects are not included in *COCO* [29]. In general, the presence of very small objects, occlusions, and drastic transformations produced modest results.

The second benchmark, **Hand-Object Segmentation (HOS)**, aimed to analyze the relationship between hands and objects by segmenting both components. Metrics such as *COCO Mask AP* [29] and models such as *PointRend* [24] (*Detectron2* [31]) were adopted with a *ResNet-50 FPN* [27] backbone, augmented with task-specific auxiliary heads. The results showed that hands could be segmented with high accuracy, while objects proved more challenging, suffering from the same limitations already observed in **VOS** (Table 1.1).

Finally, the third benchmark, **WDTCF** (Where Did This Come From), aimed to determine the origin of an object based on the temporal sequence. This meant understanding, for example, whether a glass came from a shelf or whether milk was taken from the refrigerator. This is a particularly complex task, characterized by several biases: For instance, many objects originate from the same container, and segmentation becomes difficult when dealing with very small objects or liquids. For these reasons, the results were limited, but they nevertheless underscored both the importance and the difficulty of the challenge.

	Hand-Contact				Hand-Active Object	
	Hand	Hand, Side	Hand, Contact	Object	Hand	Active Object
Val Mask AP	90.9	87.1	73.5	30.5	91.1	24.1
Test Mask AP	95.4	92.4	78.7	33.7	95.6	25.7

Table 1.1: Results for the HOS task. (Adapted from: **EPIC-KITCHENS VISOR Benchmark VIdeo Segmentations and Object Relations** [11]

1.3 EgoSurgery

We now turn our attention to a context no longer domestic or representative of everyday life, but rather its opposite: the surgical setting. Behind this project is the **EgoSurgery** [16] team, among the first to work on an egocentric dataset in the operating room. The dataset, which will be analyzed in more detail in the following chapters, contains approximately **15,000 frames** recorded from the surgeon's *POV* cameras during operations. This is a valuable contribution considering the scarcity of resources available in this field, due both to the high costs of video acquisition and to sensitive privacy concerns. To make it publicly accessible, anonymization procedures were applied, such as removing faces and excluding frames containing patient-identifying information.

The authors have published several studies, progressively enriching and refining their work over the years by introducing new annotations and proposing increasingly specific models. In the following, we review the main contributions in chronological order: starting with **Surgical Instrument Detection in Open Surgery Videos** [16], moving on to **EgoSurgery-Phase** [15] and **EgoSurgery-Tool** [14], and culminating with the most recent work, **EgoSurgery-HTS** [10], published in 2024.

1.3.1 Surgical Instrument Detection in Open Surgery Videos

The first study by the **EgoSurgery** [16] group coincided with the release of the dataset. This initial version contained approximately **19,000 images** annotated with **67,000 bounding boxes** corresponding to **31 different instruments**. The release was nontrivial, given the challenges mentioned above.

To establish a benchmark, the authors selected well-known models such as *Faster R-CNN* [37] and *RetinaNet* [28], trained with *ResNet* [20] (50, 101, and *ResNeXt101* [20]) initialized with weights pre-trained on *MS-COCO*

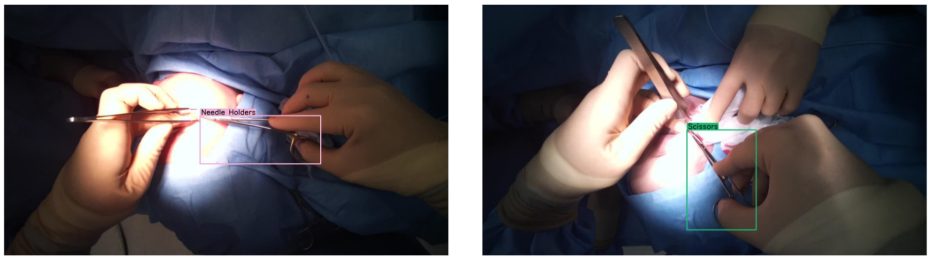


Figure 1.3: Examples of bounding box and similarly shaped tools (Adapted from: **Surgical Tool Detection in Open Surgery Videos** [16]).

[29]. The *Focal Loss* [28] technique was employed to address class imbalance. The results showed the best performance for *Faster R-CNN* [37] with *ResNeXt101*, reaching an average AP of approximately **29.7**. However, the dataset posed several challenges (Figure 1.3): imbalance in tool frequency (some very common, such as forceps, others almost absent), difficulties related to deformable objects such as gauze, and nonhomogeneous data subdivision (*split-video-based*).

1.3.2 EgoSurgery-Phase

The second study focused on **surgical phase recognition**, a crucial component for *automated analysis* with potential applications in **real-time clinical support**. To address the scarcity of public datasets in this area, the **EgoSurgery-Phase** [15] dataset was introduced, as an expansion of the previously described EgoSurgery resource (1.3.1).

The dataset was meticulously annotated under the supervision of medical experts, defining **nine distinct surgical phases**: *disinfection, planning, anesthesia, incision, dissection, hemostasis, irrigation, closure*, and *dressing*. According to the **EgoSurgery-Phase** [15] team, this is the **first and only dataset of its kind**, comprising **15 hours of egocentric video** recorded with a head-mounted camera equipped with an **eye tracker**.

In this research, the authors proposed an innovative model, **GGMAE** (*Gaze-Guided Masked Autoencoder*), which leverages **eye-tracking information** to guide training. The base architecture is a **VideoMAE** [44] with a *ViT-Small backbone* [26]. Unlike the *random masking strategies* typically used in *MAEs* [22], **GGMAE incorporates the surgeon's gaze as a semantic bias**: regions where the gaze is focused—often critical for identifying surgical phases—are more likely to be masked, encouraging the model to learn richer and more context-aware representations.

The results demonstrated that including gaze information significantly

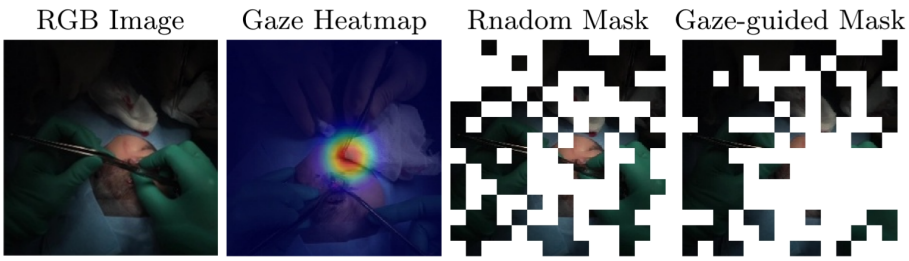


Figure 1.4: Difference between Random Mask and Guided-Gaze mask (Adapted from: **EgoSurgery-Phase: A Dataset of Surgical Phase Recognition from Egocentric Open Surgery Videos** [15]).

enhanced phase recognition. **GGMAE** outperformed both previous *state-of-the-art recognition methods* (**+6.4% Jaccard index**) and other *MAE-based approaches* (**+3.1% Jaccard index**) on the **EgoSurgery-Phase** [15] dataset, confirming the **effectiveness of gaze-guided strategies** for more accurate surgical phase analysis.

1.3.3 EgoSurgery-Tool

In the **third contribution**(EgoSurgery-Tool [14]), the dataset was further enriched with **detailed bounding-box annotations**, covering not only **15 tool categories** but also **4 hand categories** (*surgeon and assistants*), distributed across **15 videos** for a total of approximately **15,000 images**.

To assess the effectiveness of these new annotations, the authors tested **nine widely used detection models**, all employing backbones pre-trained on **MS-COCO** [29]. Among them, **VarifocalNet** [49] emerged as the most effective, achieving an **AP of 45% for tool recognition** and **59% for hand detection**, demonstrating a strong ability to manage **high object density** and **small-scale features**.

Nonetheless, several challenges persist: **non-uniform illumination**, **visual similarity between tools**, and **frequent occlusions**, all of which often lead to **misclassifications**.

1.3.4 EgoSurgery-HTS

The latest milestone was the introduction of **EgoSurgery-HTS** [10], a dataset released in *March 2024*. This work represents a key advancement, addressing one of the major gaps in previous datasets: the absence of *pixel-level annotations* for instruments and hands.

The **EgoSurgery-HTS** [10] annotation process was inspired by methodologies such as **SAMRS** (*Segment Anything Model for Remote Sensing*) [45], which leverage the basic segmentation capabilities of **SAM** (*Segment Anything Model*) [25] to generate initial masks. However, given the complexity and context sensitivity of these automatically generated annotations, a rigorous **manual review and correction phase** was carried out. This *hybrid approach* ensures both **high-quality labels** and **accurate results**.

The research focused on **three main tasks**:

- **Instrument Instance Segmentation** – This task aims to precisely identify and delineate each individual instrument present in the operating room. Distinguishing between **14 types of surgical instruments**, such as *forceps, scalpels, needles*, and *retractors*, is essential for monitoring their correct use and positioning.

- **Hand Instance Segmentation** – Crucial for understanding the surgeon's actions, this task identifies and segments hands, distinguishing between the *left and right hands* of the lead surgeon and those of the assistants.

- **Hand–Instrument Segmentation** – Arguably the most challenging task, it captures the interactions between hands and instruments. Beyond simple segmentation, it establishes relationships, such as whether an instrument is held in the *left or right hand*, or if both hands are required. This interaction analysis is fundamental for interpreting surgical intent.

To evaluate the effectiveness of the new dataset, **four models** were tested: **Mask R-CNN** [21], **QueryInst** [13], **Mask2Former** [6], and **SOLOv2** [47] . All were configured with a **ResNet-50** [20] backbone and pre-trained on **MS-COCO** [29] to ensure a robust and comparable baseline.

The benchmark results showed no single model outperformed the others across all tasks. For instance, **QueryInst** [13] achieved good results in *bounding box detection* of tools, while **Mask2Former** [6] excelled in *pixel-level segmentation*. **Mask R-CNN** [21] performed best in terms of *mAP* for both bounding boxes and masks. This variability suggests that the choice of model depends heavily on the specific application objective.

Despite these advances, critical challenges identified in earlier studies persist. Segmenting tools with similar shapes remains difficult, and the frequency of occurrence of certain instruments in the dataset strongly affects predictive accuracy. Moreover, variations in *lighting, contrast, shapes*, and *complex textures* continue to present significant obstacles.

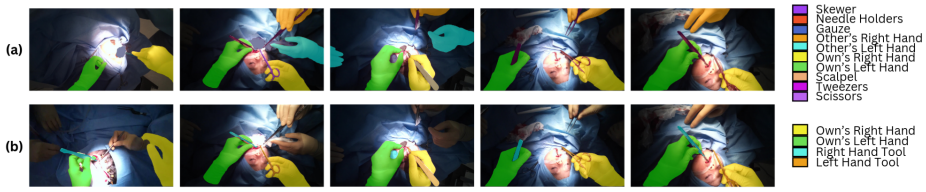


Figure 1.5: Example of segmentation annotations for hands and tools (Adapted from: **EgoSurgery-HTS: A Dataset for Egocentric Hand-Tool Segmentation in Open Surgery Videos** [10]

In conclusion, **EgoSurgery-HTS** [10] represents one of the first datasets with sufficiently detailed annotations to be genuinely useful for future applications and research in this field.

Chapter 2

Dataset and Benchmarks

After introducing the various studies on the application of computer vision to *egocentric videos*, it is now time to present in detail the structure of the dataset used and the benchmarks that served as the basis for my experiments. However, before doing so, it is important to understand what differentiates the dataset I use from that of competitors. Only then can we dive into the description of the **EgoSurgery-HTS** [10] dataset and conclude with the benchmarks reported by the authors.

2.1 Types of Surgical Datasets

The **EgoSurgery** [10] dataset was chosen because it represents one of the very few available resources in the field of egocentric surgical videos for **OP (Open Surgery)**, as opposed to **MIS (Minimally Invasive Surgery)**. This distinction is crucial. **MIS** videos are relatively easy to obtain: they do not directly reveal the patient's identity and are recorded with endoscopic cameras, which are already an integral part of the physician's diagnostic and review process. For this reason, research based on **MIS** datasets is far more mature, with several large-scale public datasets already being used in numerous studies. The main challenges are mainly related to *occlusions* caused by internal tissue.

A recent example of such datasets is **AutoLaparo** [48], which consists of **21 Full HD** videos at **25 fps** of laparoscopic hysterectomy procedures, annotated for segmentation and workflow recognition tasks. Among the most comprehensive is **Cholec80** [38]: its name refers to **80** videos, recorded at **25 fps** and in *Full HD*, for a total of approximately **370,000 frames**. In this case, the application context is laparoscopic cholecystectomy (see Table 2.1).

Dataset	Year	Data Type	Annotation Type
AutoLaparo [48]	2022	21 videos	Segmentation, workflow, motion
Cholec80 [38]	2016-2017	80 videos	Detailed phase and instrument annotations
CholecSeg8k [23]	2020	8,080 images	Pixel-wise semantic segmentation

Table 2.1: Overview of representative **MIS** surgical datasets, including year of release, type of data, and available annotations.

By contrast, the **EgoSurgery** [10] dataset belongs to the **OS** context and has very different characteristics. The videos are recorded from an *egocentric* perspective using a camera mounted on the surgeon's head. This choice is deliberate, as it provides a unique perspective on how humans interact with the environment, sometimes enriched by *eyetracking* data (see **EgoSurgery-Phase** 1.3.2). However, obtaining such footage immediately presents clear challenges.

The **OS** environment is much more crowded: in addition to the patient, several people participate in the operation, and multiple instruments appear simultaneously in the same frame (Figure 2.1). *Occlusions* are frequent, especially due to the hands of surgeons and assistants. Some instruments exhibit strong visual similarity or deformability (, *e.g. sutures or gauze, which change appearance depending on their use*). In addition, instrument categories are often unbalanced, and variable lighting conditions reduce the visual quality of the recordings.

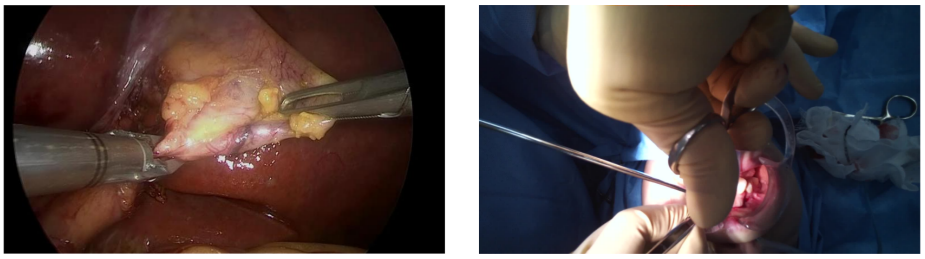


Figure 2.1: A comparison of the visual characteristics of MIS and OS frames (Adapted from: **Surgical Tool Detection in Open Surgery Videos** [16]

Another difficulty lies in the need for reliable expert annotations. For example, in **EgoSurgery-Phase** [15], the surgical phases were directly annotated by specialists. Furthermore, anonymization of personally identifiable information (*PII*) is required, as already discussed.

To mitigate these difficulties, synthetic datasets have also been created. Among the best known is that of **Basiev**[1], with approximately **11,000 frames** generated by simulations of intestinal surgeries. However, it is not

Dataset	Real Env?	BBox Annotated?	Number of Annotated Frames	Number of Annotated Instances	Avg. Instances per Frame	Number of Surgical Tool Categories
Shimuzu [40]	✓	✓	2300	-	-	2
Basiev [1]		✓	11,500	-	-	4
Goldbraikh [17]		✓	1124	-	-	3
AVOS dataset [18]	✓	✓	3348	2843	0.85	3
EgoSurgery-HTS [10]	✓	✓	19,496	149,161	7.65	14

Table 2.2: An overview of some OS surgical datasets

directly comparable to **EgoSurgery** [10] because it does not simulate an *egocentric* view, but rather employs a multi-camera system with two very different angles. In general, Table 2.2 provides a brief comparison with the OS datasets used in recent research.

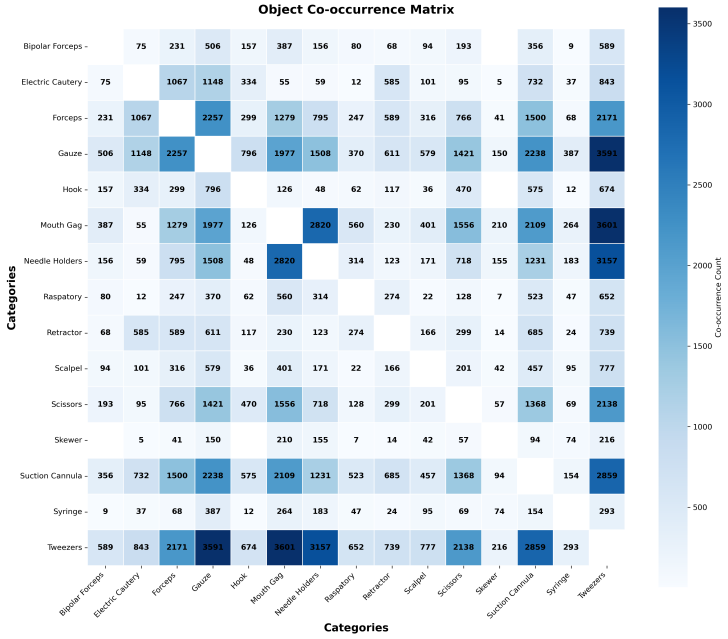
2.2 The EgoSurgery-HTS Dataset

For this research, we focus on the **EgoSurgery-HTS** [10] dataset, which consists of **15 videos**. This number, lower than the original **21**, is due to an inaccuracy by the authors: the additional annotations introduced over the years were not consistently extended to all videos, as happened, for instance, in *EgoSurgery-Phase* [15]. The **15 selected videos** represent a subset of distinct surgical procedures (**10** in total), performed by **8 different surgeons**, for an overall duration of approximately **15 hours**. The recordings were made in *Full HD* at **25 fps** and subsequently downsampled to **1 fps** to reduce redundancy caused by overly similar frames.

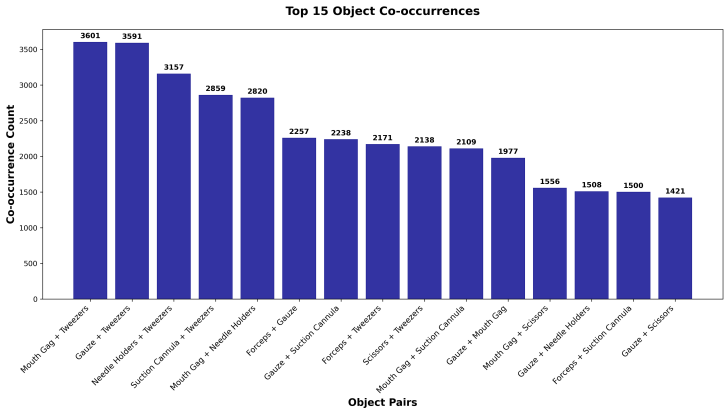
The annotations cover **19 categories**: **4** related to hands (*own left hand, own right hand, other's left hand, other's right hand*) and **15 surgical instruments**, including *bipolar forceps, electric cauterizers, needle holders, retractors, scalpels, scissors, gauze, suction cannulas*, and *syringes* (see Table 2.3). The distribution across these categories is markedly *imbalanced*. Instruments such as *Skewer, Syringe*, or *Bipolar Forceps* appear in very few frames in both training and testing, giving the model limited examples to learn their characteristics and reducing performance in identification and segmentation.

Some instruments occur only in specific procedures (*e.g. mouth gag*) or in combination with others (*e.g., Retractor and Syringe*)(Figure 2.2), further increasing distributional imbalance. For this reason, the authors reduced the original **31 categories** to **15**, discarding those appearing fewer than **10 times** or absent from the validation and/or test sets.

The dataset includes both **boxes** and **segments**. Bounding-boxes were generated with Microsoft's open-source *Virtual Object Tagging Tool (VoTT)*. Segmentations were produced using *SAMRS* [45], which leverages the *SAM*



(a) Co-occurrence matrix of surgical tools



(b) Ranking of the most frequent object co-occurrences

Figure 2.2: Co-occurrence analysis of surgical instruments in the EgoSurgery [10] dataset. (a) shows the complete co-occurrence matrix, capturing the distributional imbalance among tools. (b) presents the ranking of the most frequent pairings, illustrating how certain instruments systematically appear together during open surgery procedures.

Category	Train	Val	Test	Total
Own Hands Left	8,704	1,505	3,834	14,043
Own Hands Right	8,447	1,467	3,670	13,584
Other Hands Left	6,542	1,079	3,412	11,033
Tweezers	6,467	950	2,595	10,012
Other Hands Right	4,033	867	2,760	7,660
Gauze	4,596	455	1,644	6,695
Forceps	2,534	154	3,375	6,063
Mouth Gag	3,807	990	1,188	5,985
Needle Holders	3,031	512	1,286	4,829
Suction Cannula	3,134	509	768	4,411
Scissors	1,780	391	565	2,736
Retractor	2,079	0	325	2,404
Electric Cautery	1,404	101	162	1,667
Hook	1,045	147	157	1,349
Scalpel	739	168	159	1,066
Raspatory	654	76	84	814
Bipolar Forceps	446	55	195	696
Syringe	344	96	141	581
Skewer	212	103	29	344
Total	**59,998**	**9,825**	**27,539**	**97,362**

Table 2.3: Category distribution of the EgoSurgery [10] dataset across training, validation, and test splits. The imbalance among classes is evident, with some tools (e.g., Skewer, Syringe, Bipolar Forceps) appearing only rarely.

[25] model to derive masks from bounding boxes of hands and tools. To identify tools interacting with hands, a *intersection-over-union (IoU)* criterion was applied. However, this approach has limitations: unlike datasets such as **EPIC-KITCHENS** [11], where interactions were manually annotated, here the automated procedure may result in *false positives*, especially due to occlusions and the atypical positions assumed by surgeons' hands in an *egocentric view*.

Finally, the dataset was split by **video** rather than by individual frames to ensure robustness and generalizability (Figure 2.3). A random frame-based split could have led to *overfitting*, as the model would learn specific configurations rather than general patterns. Splitting by video remains the standard practice for this type of task.

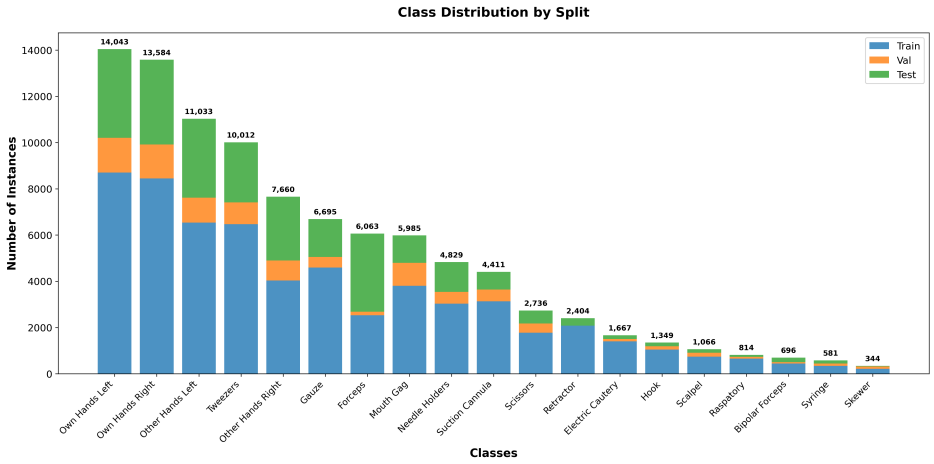


Figure 2.3: Distribution of the number of samples per surgical tool category

2.3 Benchmarks and Results

To establish a solid basis for comparison and assess the effectiveness of detection and segmentation models, we analyzed the results obtained on the **EgoSurgery-Tool** [14] and **EgoSurgery-HTS** [10] benchmarks. These studies provide valuable information on current performance levels and the potential for adaptation through fine-tuning.

EgoSurgery-Tool: Tool and Hand Detection

The first study, based on **EgoSurgery-Tool** [14], evaluated nine of the most widely used object detectors: **Faster R-CNN** [37], **RetinaNet** [28], **DINO** [3], **DDQ** [50], **VarifocalNet** [49], **Cascade R-CNN** [2], **Sparse R-CNN** [41], **CenterNet** [12], and **Deformable-DETR** [51]. Implemented with the *MMDetection* [5] framework and fine-tuned using weights pre-trained on **MS-COCO** [29], the models were evaluated with **COCO's AP** metrics.

As reported in Table 2.4, **VarifocalNet** [49] achieved the best overall results for tool detection, with an **AP** of **45.8%** and an **AP50** of **63.3%**. For hand detection, it reached an **AP** of **59.4%** and an **AP50** of **82.1%**. This performance can be attributed to its ability to capture small-scale features and handle severe occlusions, thanks to innovative techniques such as the *IoU-aware Classification Score (IACS)* and *Varifocal Loss* [49], derived from *Focal Loss* [28]. Notably, **DINO** [3] obtained the best **AP50** for hand detection (**80.2%**).

Methods	Surgical tool detection			Hand detection		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Faster R-CNN [37]	37.7	55.8	43.3	55.3	80.4	62.3
RetinaNet [28]	36.2	53.0	39.8	57.1	81.9	62.9
Cascade R-CNN [2]	38.8	55.7	44.6	55.5	80.7	61.4
CenterNet [12]	42.4	60.2	46.8	56.6	78.5	63.3
Sparse R-CNN [41]	37.0	55.1	41.8	55.4	78.7	60.9
VarifocalNet [49]	45.8	63.3	51.1	**59.4**	**82.1**	**65.3**
Deformable-DETR [51]	30.9	46.3	34.0	54.1	78.6	59.2
DDQ [50]	43.2	59.1	48.7	58.3	73.5	60.8
DINO [3]	39.7	56.7	43.5	58.8	80.2	65.6

Table 2.4: Performance comparison for surgical tool and hand detection.

EgoSurgery-HTS: Segmentation of Tools, Hands, and Hand–Tool Interactions

The second study, more directly related to our segmentation task, explored the performance of four models on **EgoSurgery-HTS** [10]: **Mask R-CNN** [21], **QueryInst** [13], **Mask2Former** [6], and **SOLOv2** [47] (the latter limited to segmentation). All models were configured with a **ResNet-50** [20] backbone.

The results, shown in Table 2.5, indicate that no single model consistently outperforms the others in all tasks.

- For tool segmentation, **Mask2Former** [6] achieved the highest **mAPmask** (**40.9%**), outperforming the other models in per-pixel accuracy.
- For hand segmentation, **Mask R-CNN** [21] excelled, with a **mAPbox** of **63.8%** and a **mAPmask** of **61.9%**. Its strength likely lies in its parallel segmentation process, which generates predictions for each region of interest while preserving exact pixel alignment through *RoIAlign*.
- For hand–tool segmentation, **Mask2Former** [6] obtained the best **mAPmask** (**56.6%**), while **QueryInst** [13] achieved the best **mAPbox** (**55.2%**).

The authors' conclusions across both studies emphasize that, despite notable progress, current performance remains insufficient for practical deployment. The main challenges persist: **class imbalance**, **visual ambiguities**

		Tool		Hand		Hand-Tool	
Methods	Backbone	mAP^{box}	mAP^{mask}	mAP^{box}	mAP^{mask}	mAP^{box}	mAP^{mask}
Mask R-CNN	ResNet-50	36.7	29.1	**63.8**	**61.9**	45.3	44.5
QueryInst	ResNet-50	**47.3**	36.7	54.0	50.7	**55.2**	49.4
Mask2Former	ResNet-50	39.2	**40.9**	50.2	52.5	54.7	**56.6**
SOLOv2	ResNet-50	-	37.0	-	53.8	-	50.7

Table 2.5: Segmentation performance comparison across different tasks

between instruments with similar shapes and textures, and, above all, **occlusions**. These factors make it difficult to reliably distinguish tools and hands in dense and complex surgical scenes, underlining the need for further research aimed at improving robustness and accuracy in this sensitive domain.

Chapter 3

Method

In this chapter, we examine the more technical aspects of the thesis, analyzing in detail the tools and models employed. Since the aim is to provide a comprehensive comparison of the results obtained with different approaches, without delving into excessive technicalities, we will focus primarily on the **PointRend** [24] model. This model, already applied in the context of **EPIC-KITCHENS** [11], has never been used in the studies conducted by the **EgoSurgery** [10] team, making its analysis particularly interesting since it opens the door to exploring new solutions. However, before addressing the details of **PointRend** [24], however, it is necessary to first consider **SAM (Segment Anything)** [25], which played a fundamental role in our work. SAM [25] enabled us to address one of the main gaps in the dataset, namely the absence of clear and uniform segmentation of tools and hands.

3.1 Segment Anything Model (SAM)

The **Segment Anything Model (SAM)** [25], developed by *Meta*, represents a significant step forward in the field of image segmentation, with continuous updates that steadily improve its capabilities. In our work, we relied on **SAM 2.0** [36] (although newer versions, such as **2.1**, promise greater efficiency in video segmentation).

As its name suggests, SAM's distinctive feature is its ability to segment *'anything'* within an image, generating precise masks. This capability derives from its training on a vast and diverse dataset, which grants it remarkable generalization power, enabling it to recognize and segment even previously unseen objects. As illustrated in **Figure 3.1** of the original paper, SAM is based on three interconnected components: a *promptable segmentation task*, a model (**SAM**) designed for annotation and *zero-shot transfer*, and a *data*

engine used to collect **SA-1B**, a dataset containing more than **one billion masks**.

The strength of SAM lies in its **flexibility**. It can be guided by different types of prompts: a single point, a bounding box, or corrections to partial masks. In our study, we specifically leverage its ability to segment objects within predefined bounding boxes, such as those provided by the **EgoSurgery** [10] dataset, to create masks for the **19 available categories**. An additional advantage of **SAM 2.0** [36] is its capability for **video segmentation**, which makes it possible to track objects consistently and in real time in multiple frames. Despite its high performance, SAM [25] is also relatively efficient and can run on consumer-grade GPUs, such as the **RTX 3070** used in our experiments.

Simplified Operation in Three Phases

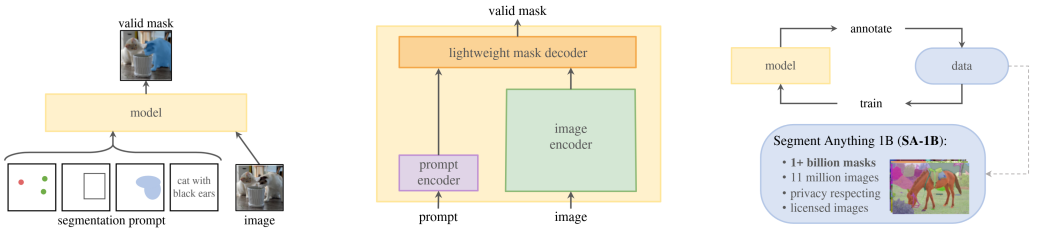
The operation of SAM [25] can be broken down into three main phases, as shown in *Figure 3.1* of the original article:

1. **Image Pre-Processing** – The image is processed by an *Image Encoder* (typically a *Vision Transformer, ViT* [26]). This component analyzes the image as a whole, extracting features and compressing them into an *embedding*, a compact semantic representation. This step is generally the most computationally intensive.

2. **User Input (Prompt)** – At this stage, the user provides a *prompt*. In our case, the bounding boxes were used as references for segmentation. A *Prompt Encoder* converts the box's spatial coordinates (x, y, width, height) into a vector embedding, integrating positional information and making it compatible with the image embedding.

3. **Mask Decoding** – In the final phase, the *Mask Decoder* (a bidirectional transformer) combines the embeddings of the image and the prompt to generate the segmentation mask. By exploiting features at multiple resolutions, it captures fine details and prevents blurry or imprecise masks. The model can also produce multiple candidate masks, each scored (e.g., with IoU), allowing the user to select the most accurate one.

The process can be iteratively refined through a **prompt prediction-refinement cycle**, where additional positive or negative clicks guide the model towards a more accurate segmentation. Although Meta's tool provides excellent results, it is not free of limitations. Without additional refinements,



(a) Promptable segmentation.

(b) Segment Anything Model.

(c) Data engine and dataset.

Figure 3.1: The three interconnected components of the Segment Anything Model (SAM)[25], as introduced by Meta. (Adapted from: **Segment Anything** [25])

the generated masks may contain noise or imprecisions. In our study, we assume these imperfections to be part of the working conditions.

3.2 PointRend

A cornerstone of many modern segmentation approaches, as demonstrated by its application in complex contexts such as the research conducted by **EPIC-Kitchens** [11], **PointRend** [24] represents a significant innovation in the field of segmentation. The primary goal of this model is to overcome the qualitative limitations of existing methods, such as **Mask R-CNN** [21], which, despite being one of the most popular models and widely used as a baseline in numerous benchmarks, often produces less defined object outlines.

To fully understand how PointRend [24] works, it is essential to draw parallels with models like Mask R-CNN [21], highlighting the key differences and the underlying intuition of the research team that developed it.

Phase 1: Feature Extraction

In the initial phase, each image, regardless of the specific model used, is processed by a **convolutional neural network (CNN)** [33]. The goal of this preliminary step is to leverage the leading architectures (such as **ResNet** [42], **ResNeXt** [42], etc.) to extract **feature maps**. These maps are typically smaller than the original image (a common example is **1/16** of the initial size)(Figure 3.2).

Feature maps are crucial because they contain semantic information about the image, essentially answering the question *'What is depicted?'*. The choice to operate on features rather than individual pixels (which can exceed **2 mil-**

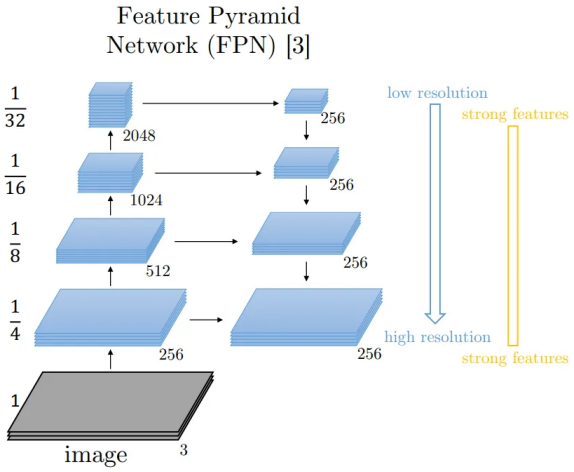


Figure 3.2: The Feature Pyramid Network (FPN) architecture, which constructs a feature pyramid from a single-resolution input image. FPN is commonly used as a backbone in modern segmentation models, including Mask R-CNN [21] and PointRend [24], to extract multi-scale feature maps. (Adapted from: **CloudFactory** [7]).

lion) is dictated by computational efficiency and robustness. However, it is important to note that extracting feature maps leads to spatially coarse semantic information. Imagine an image of an animal: after applying the extraction of features, each *'cell'* (or pixel on the feature map) no longer corresponds to a single initial pixel. This provides a general idea of the presence of the animal in the image, but, as expected, finer details such as edges or nuances in the fur are lost.

Phase 2: Proposal Generation and Alignment

At this point, we enter the second phase. Having access to the feature maps produced by the backbone, a **Region Proposal Network (RPN)** [37] acts on them, proposing candidate bounding boxes that might contain relevant objects.[1] A classifier then assigns a class to each bounding box (e.g., *"object"* or *"animal"*). For each detected bounding box, **RoIAlign** [21] is used to extract the features contained in the box and normalize them to a fixed-size grid (irrespective of the actual size of the object). This produces features

[1]The RPN is a fully convolutional network that applies a small convolutional window (e.g., 3x3) in a sliding-window fashion, generating anchors associated with the scores of two separate heads: one for classification (object/background) and one for bounding-box regression.

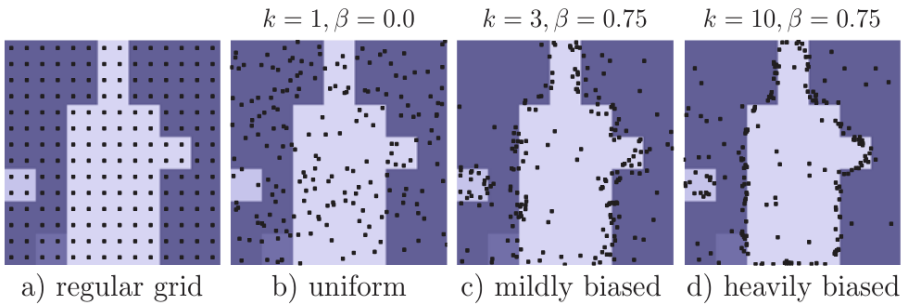


Figure 3.3: The PointRend Coarse-to-Fine Refinement Strategy. The model focuses its computational effort only on a small set of "uncertain points" which are typically located near object boundaries. (Adapted from: **PointRend: Image Segmentation as Rendering** [24]).

that represent the object itself.

Phase 3: Mask Generation

Up to this point, the process remains virtually unchanged between **Mask R-CNN** [21] and **PointRend** [24], which can rely on the same backbone and the same RPN. Substantial differences emerge in the **mask generation phase**.

- **In Mask R-CNN** [21], for each object (RoI), the features obtained with RoIAlign are passed to a small CNN, the *'mask head'* This network produces a low resolution binary map, typically **28x28**. The output is then simply upsampled (via bilinear interpolation) to match the resolution of the original image. This process is purely geometric and adds no detail, often resulting in imprecise and overly smooth edges.

- **PointRend** [24], on the other hand, is not limited to a low-resolution mask. Its key idea is to treat segmentation as a **rendering problem**, similar to computer graphics. Instead of predicting an entire pixel grid, PointRend makes predictions only on a limited number of *'uncertain'* (difficult) points, located primarily at the edges of the object (Figure 3.3). This approach is applied iteratively, progressively refining the mask in a **coarse-to-fine process** .

At these specific points, PointRend [24] builds a unique feature representation by combining two types of information: 1) **Fine-grained features**,

interpolated directly from CNN backbone feature maps at higher resolution. 2) **Coarse-prediction features**, derived from the initial low resolution mask (such as that produced by Mask R-CNN [21]).

This point feature vector is then fed to a small **multilayer perceptron (MLP)** [43], called the *point head*, which predicts the label (e.g., *"background"* or *"object"*) for that point. This iterative process allows PointRend to generate high-resolution masks (e.g. **224x224**) with significantly sharper edges than those of Mask R-CNN [21], while requiring fewer computational operations. Efficiency comes from the fact that computations are not wasted on homogeneous regions of the object.

Chapter 4

Experimental Setup and Results

After analyzing the structure of the dataset and examining some of the tools employed, we can now move on to the final part of this research. Here, I will present the two frameworks used, the modifications introduced to the official dataset **JSON** files to enable testing, and how the concept of *contact*—not included in the original annotations—was defined. Finally, I will discuss the quantitative and qualitative results obtained, both with and without fine-tuning.

4.1 Experimental Setup

Framework and Development Environment

In this study, two different frameworks were used. This choice was motivated by two main reasons: ensuring the **reproducibility** of experiments and following the methodology of the research cited above.

The development environment was containerized using **Docker** [30] with **CUDA**[32] support, in order to parallelize and accelerate training on multiple GPUs.

MMDetection [5], one of the most widely used *open-source* frameworks for object detection based on **PyTorch** [35], was adopted for bounding box analysis and qualitative segmentation evaluation. It is widely used in the literature and is also included in the work of the **EgoSurgery** [10] team, which facilitates both result comparison and training with parameters as close as possible to the reference ones. MMDetection [5] already includes numerous configurations, including those used here: **Mask R-CNN** [21],

QueryInst [13], **PointRend** [24], and **HTC** [4], all with **ResNet-50**[20] as the backbone. This minimizes discrepancies with related research. Moreover, the framework enables straightforward modification of head or loss functions, allowing exploration of alternative approaches.

Detectron2 [31] was instead used for the *hand contact* task. This choice was again motivated by the intention to reproduce existing studies -such as those conducted on **EPIC-KITCHENS** [11]—and adapt them to the specific context of **EgoSurgery** [10].

SAM

As mentioned above (3.1), Meta's **SAM** (*Segment Anything Model*) [25] tool was used to address the lack of complete segmentations in the original dataset. The decision was therefore made to generate them directly using this tool, while accepting a certain amount of noise due to the absence of manual refinement of the masks.

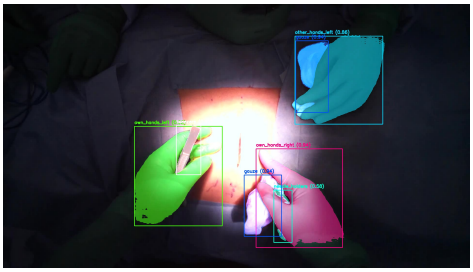
SAM [36] was also introduced to support an alternative technique for defining *contact*, based on the pixel-wise expansion of the segmented masks rather than the traditional overlap of the bounding boxes. This aspect will be explored in more detail in the next section. An example of the segmentations produced (including the noise generated) is shown in **Figure 4.1**.

Interaction and JSON

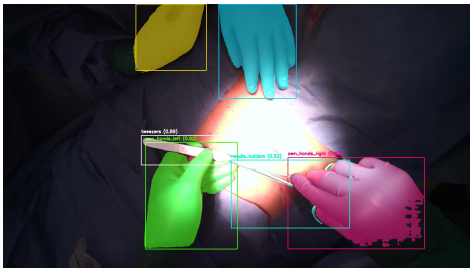
The dataset was already provided in **COCO** [29] format but required some modifications to integrate the new information. In addition to the segmentations generated with SAM [36], a specific attribute was introduced to indicate contact between the hand and the tool.

When observing the dataset frames, it becomes evident that surgeons interact with tools of varying shapes and textures. In complex scenes with strong occlusions, relying solely on IoU-based bounding boxes is not optimal. This approach evaluates the overlap between hand and object based on annotated boxes, but it often proves inaccurate, especially for thin instruments or tools partially covered by the hands. In such cases, there is a high risk of producing an excessive number of false negatives, which negatively impacts both the training and testing phases.

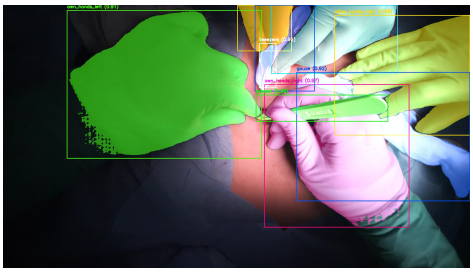
To address this problem, a **segmentation-based technique** was adopted. The idea is to apply a *binary dilation* to the segmented masks, creating a buffer around the object's edges. In our case, a **five-iteration dilation** was used, expanding the mask pixels outward. Contact detection is then performed using a simple **bitwise AND**: if the result is greater than zero, a



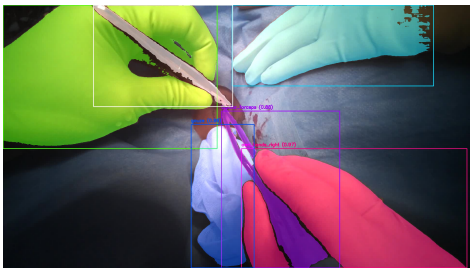
(a) Frame 04_1_1490_sam2.jpg



(b) Frame 04_2_0121_sam2.jpg



(c) Frame 09_1_0541_sam2.jpg



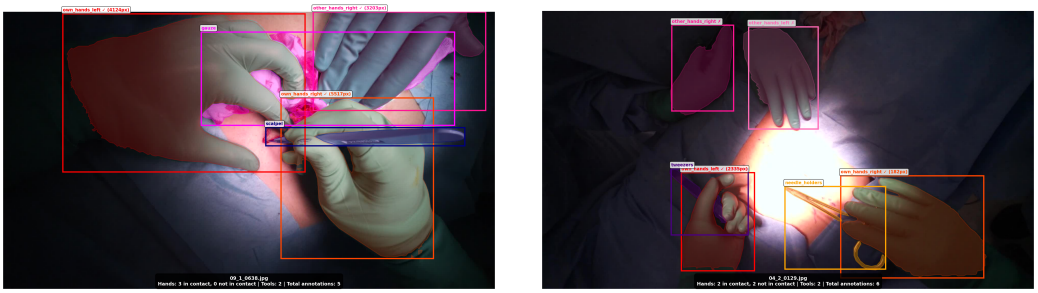
(d) Frame 11_1_0993_sam2.jpg

Figure 4.1: Examples of frames from the EgoSurgery [10] dataset segmented with SAM2 [36]. The segmentations highlight both the ability of the tool to capture surgical instruments and hands, as well as the presence of noise due to the absence of manual refinement.

contact is recorded.

This approach is more robust in the presence of annotation imperfections or in cases involving thin or unusually shaped instruments. Some examples are shown in **Figure 4.2**.

It should be noted, however, that this technique still carries a certain margin of error. For instance, consider the *Mouth Gag*, which remains fixed in the patient's mouth: improper handling may lead the model to erroneously interpret it as being in constant interaction with the surgeon's hand.



(a) Frame 07_09_1_0638.png **(b)** Frame 06_04_2_0129.png

Figure 4.2: Two random frames extracted from the EgoSurgery [10] dataset after applying the contact detection algorithm between hands and objects.

Parameters and Final Configuration

Finally, to ensure the reproducibility of the experiments, I specified that the training parameters (**batch size, learning rate, momentum**, etc.) correspond to the default values in the **MMDetection** [5] and **Detectron2** [31] configurations.

A final note concerns the dataset subdivision: since the EgoSurgery articles do not precisely describe the split used, I reconstructed the criterion based on the available files, obtaining the subdivision shown in **Table 4.1**.

4.2 Results

4.2.1 Quantitative Results

After introducing the models and frameworks used, we can now move on to analyzing the quantitative results. The initial experiments were conducted exclusively with **Detectron2** [31], leveraging the implementation previously described for **EPIC-KITCHENS** [9]. To adapt the model to the surgical

Split	Video IDs	Number of Videos	Number of Frames
Train	01, 02, 03, 06, 08, 11, 12, 13, 14, 15	10	9,627
Val	09, 10	2	1,515
Test	04, 05, 07	3	4,255
Total	**15 videos**	**15**	**15,397**

Table 4.1: Subdivision of the EgoSurgery [10] dataset by video. The table reports the **video IDs**, the corresponding **number of videos**, and the total **number of frames** assigned to each subset (*train*, *validation*, and *test*).

Table 4.2: Evaluation of domain transfer from EPIC-KITCHENS [9] pre-trained weights to the EgoSurgery [10] dataset, reported in terms of $\mathbf{mAP^{box}}$. Results are shown both *without fine-tuning* and *with fine-tuning*, highlighting the impact of targeted adaptation on hands, objects, and hand–object interactions (HOI).

Class	Without Fine-tuning	With Fine-tuning
Hand	14.27	61.34
Object	2.38	22.66
HOI	3,59	32,11

context, it was necessary to manually modify and adjust part of the code, particularly the **VISOR-HOS** [11] search module, to make it as compatible and functional as possible with the **EgoSurgery** [10] dataset.

To evaluate the effectiveness of the generalization of the model both with and without fine tuning, the pre-trained weights of **VISOR-HOS** [11] were combined with **PointRend** [24] and different head variants, resulting in the so-called *HOS* approach. The results, reported in **Table 4.2**, show that the model struggles to adapt to a complex and distant context such as the medical domain when used without further adjustments. However, significant improvements can be observed with fine-tuning, with increases ranging from **25% to 45%**, confirming the need for a targeted adaptation process.

The following **Tables 4.3 and 4.4** focus on experiments conducted with **MMDetection** [5]. In **Table 4.3**, fine-tuning was performed starting from the pre-trained **VISOR-HOS** [11] weights, while in **Table 4.4**, training began with the generic **MS-COCO** [29] weights. In this case, I chose to omit the segmentation results, as they are not entirely reliable: the masks were obtained with **SAM** [36], without any manual refinement, and therefore contain too much noise to be considered a robust basis for comparison.

When analyzing the related task of **Hand-Object Interaction (HOI)**,

the obtained results (**32.11**, see **Table 4.2**) provide further insights. In particular, the task exhibits a remarkably high *Recall* of approximately **80%**, successfully detecting a large number of interactions, and a reasonable *mAP@0.5* of around **29%**. Nevertheless, the major limitation lies in the very low *Precision*, which remains close to **10%**, highlighting the difficulty of filtering out false positives despite the promising recall values. Surely, with a more refined and accurate segmentation process, it would be possible to develop an HOS approach that performs substantially better than the one tested here.

Table 4.3: Model performance on the EgoSurgery [10] dataset after fine-tuning from **VISOR-HOS** [11] pre-trained weights. Reported metrics include $\mathbf{mAP_{box}}$ for validation data, as well as separate evaluation on surgical *tools* and *hands* in the test set.

Model	Val	Test Tool	Test Hand
	mAP_{box}	mAP_{box}	mAP_{box}
Mask R-CNN	57,5	38,5	63,7
PointRend	57,9	39,2	65,2
HTC	60,8	**40,3**	**67,1**

Table 4.4: Model performance on the EgoSurgery [10] dataset after fine-tuning from **MS-COCO** [29] pre-trained weights. Metrics are reported as $\mathbf{mAP_{box}}$ for validation data, test tools, and test hands.

Model	Val	Test Tool	Test Hand
	mAP_{box}	mAP_{box}	mAP_{box}
QueryInst	63,0	42,3	63,9
Mask R-CNN	59,0	**50,3**	64,7
PointRend	63,6	43,9	61,6
HTC	62,5	41,3	**67,3**

The results indicate that the model delivering the best overall performance across the various tasks is **HTC** [4], especially when initialized with **MS-COCO** [29] weights. The differences compared to the values reported in the **EgoSurgery** [10] studies are minimal—only a few percentage points—making it difficult to identify a clearly superior approach. Nevertheless, it is evident that more accurate and refined segmentation annotations could lead to improved performance, especially considering that models such as **HTC** [4]

and **PointRend** [24] are specifically designed to excel in segmentation tasks while also performing well in bounding box detection.

Additionally, I analyzed the distribution of results across the **19** individual categories, as shown in **Table 4.5**[1] .Several noteworthy aspects emerge that deserve further attention. For instance, regarding hands, the model shows a reasonable ability to distinguish the surgeon's and assistant's left hands (with values close to **50%**), but performs much worse in recognizing the assistant's right hand, with accuracy dropping to around **20%**. Conversely, the surgeon's right hand is identified more frequently, reaching the highest values.

As for tools, **Tweezers** and **Scalpels** are among the most reliably recognized, with averages around **50/60%**. In the case of tweezers, this is primarily due to their widespread use: they appear in nearly all procedures, with a large number of annotations that facilitate learning despite their similarity to other instruments. Scalpels, on the other hand, although less represented in the dataset, benefit from having a rigid and distinctive shape (unlike, for example, gauze or sutures) and a consistent appearance. These characteristics reduce the risk of confusion with other tools, contributing to the results observed.

A particularly interesting case is the **Mouth Gag**. Despite achieving one of the highest detection scores ($\mathbf{mAP}_{\text{box}} \approx \mathbf{71\%}$), its segmentation performance remains comparatively low. This discrepancy can be explained by several factors. Although the category is relatively well represented in the dataset (**5,985** annotated instances – Table 2.3), the Mouth Gag is almost always employed in a *fixed position*, placed inside the patient's mouth to keep it open. As a result, the detector can easily learn its approximate location and predict bounding boxes with high accuracy, even in challenging conditions. However, the object is often partially or fully occluded by the surgeon's hands or by other instruments during the procedure, making its contours *ambiguous or invisible* in many frames. These factors together help explain why the model excels at localizing the Mouth Gag in the scene but struggles to produce consistent and accurate segmentation masks.

4.2.2 Qualitative Results

Regarding the qualitative results, I selected a set of sample images from the test inferences. Specifically, I chose to present those generated by the

[1] In Table 4.5, the mask results are reported not for comparison with related work, since they were generated automatically and not manually refined. They are included solely to justify the choice of the model used for the qualitative analysis in the following section (4.2.2).

Table 4.5: Multiclass test results on the EgoSurgery [10] dataset after fine-tuning with **MS-COCO** [29] pre-trained weights. Metrics reported include $\mathbf{mAP_{box}}$ and $\mathbf{mAP_{segm}}$ for each category.

Class	Test mAP^{box}		Test mAP^{segm}	
	PointRend	HTC	PointRend	HTC
Own Hands Left	**55,1**	53,7	**58,5**	43,5
Own Hands Right	59,4	**61,0**	**63,9**	55,1
Other Hands Left	**47,6**	42,5	**54,8**	43,0
Other Hands Right	23,0	**24,5**	**25,6**	25,0
Bipolar Forceps	**44,8**	37,7	**35,0**	11,2
Electric Cautery	**40,9**	22,5	**44,5**	14,4
Forceps	8,5	**10,6**	**6,0**	4,3
Gauze	**12,3**	10,5	**18,1**	12,6
Hook	27,3	**34,5**	**26,2**	1,8
Mouth Gag	**71,7**	70,4	**34,7**	29,6
Needle Holders	28,6	**33,9**	**23,4**	12,2
Raspatory	52,5	**53,1**	**47,3**	25,7
Retractor	4,5	**5,4**	**5,5**	4,1
Scalpel	**54,6**	49,7	**53,5**	35,4
Scissors	24,8	**27,9**	**23,6**	12,7
Skewer	44,1	**56,8**	**37,9**	6,6
Suction Cannula	**45,5**	30,1	**46,1**	9,3
Syringe	**25,6**	18,5	**33,0**	21,9
Tweezers	59,2	**56,3**	**59,2**	26,9

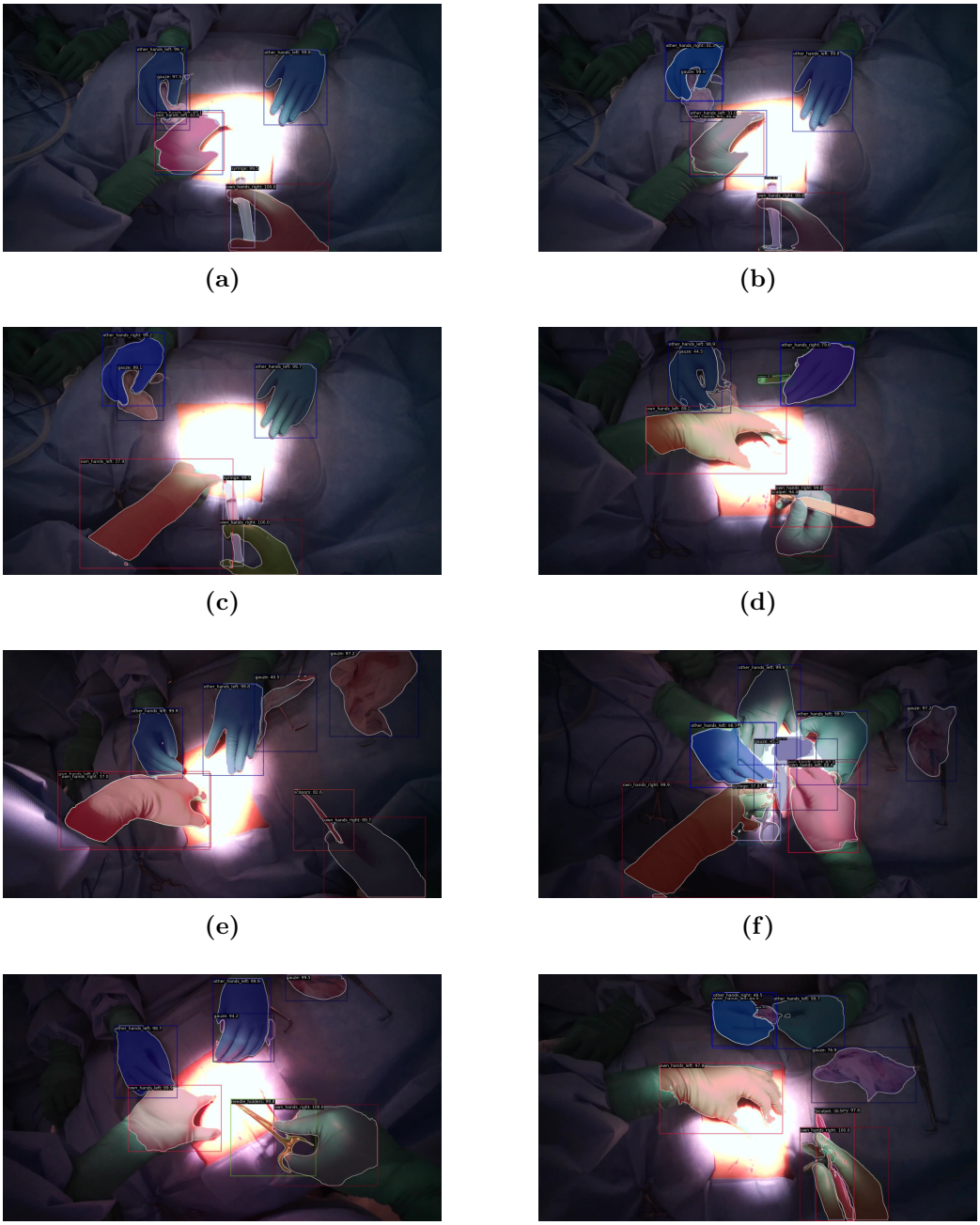
best-performing model, **PointRend** [24]. The qualitative analysis focuses on the examination of the predicted masks: this allows us not only to assess the model's strengths, but also to identify its weaknesses, highlighting the typical challenges encountered in a complex domain such as surgery. [2]

From the qualitative inspection, we can clearly observe the issues described above. The model does not intrinsically struggle to detect the presence of hands or instruments, which are often recognized *relatively easily* within the scene. The real difficulty emerges when it comes to **classifying them correctly**. For example, the model frequently confuses the different

[2]The visual examples presented in this section were intentionally drawn from pre-operative or preparatory stages. This choice was made to ensure clarity in illustrating the model's performance while also preventing the inclusion of intra-operative content that could be considered sensitive or potentially unsettling to the reader.

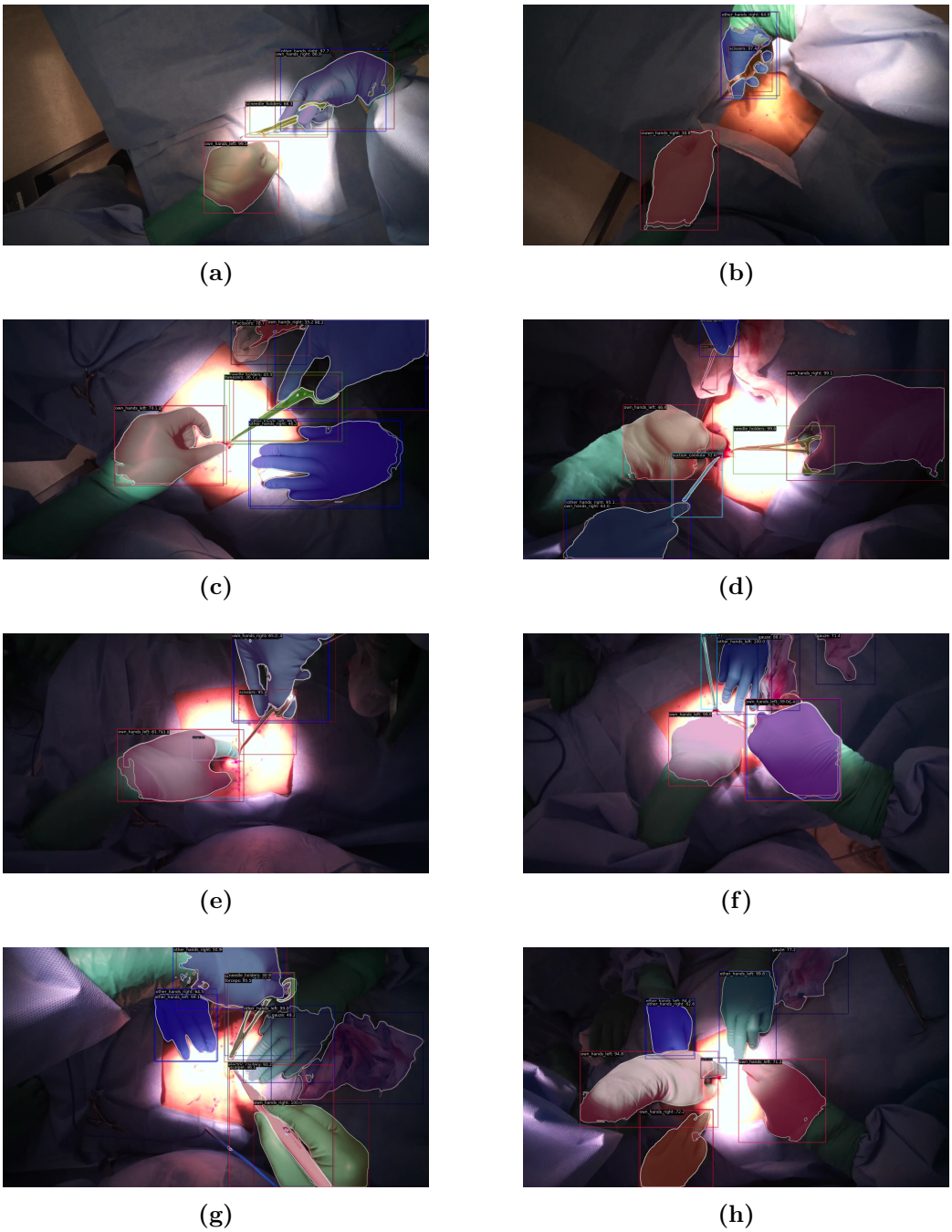
hands, a limitation that can be traced back to three main factors. First, there are very few *egocentric datasets* that contain examples of hands other than those of the *camera wearer*, making it difficult for the model to generalize effectively to the **operating room** scenario. Second, surgeons often assume very specific *hand postures* to hold different instruments, which may involve tilting, rotating, or partially occluding fingers, thereby increasing the **ambiguity** of the visual cues. Third, the presence of **assistants** further complicates the task: their hands frequently overlap with or mimic the position of the surgeon's, leading the model to misclassify them as *own hands*, a trend that clearly emerges in the qualitative results reported in **Figure 4.4**.

Moreover, the surgical environment itself adds a further layer of complexity . During an operation, the visual field becomes crowded with elements: multiple tools, gauzes of varying shapes scattered across the scene, and the simultaneous presence of more than the expected four hands - sometimes even six. These conditions generate **strong occlusions and overlaps** that make the classification of each class particularly challenging. As a result, while the model demonstrates a promising ability to *detect and localize objects*, its capacity to assign the correct label to each instance is significantly hindered by the **intrinsic visual and contextual difficulties** of **open surgery**.



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)**

Figure 4.3: Qualitative examples of model inferences on the EgoSurgery [10] dataset. Frames (a), (e), and (g) illustrate how the assistant's hands are frequently misclassified as "left." In frame (h), a strong occlusion can be observed, with multiple overlapping elements and the presence of six distinct hands, highlighting one of the major challenges encountered in this domain.



(a) **(b)** **(c)** **(d)** **(e)** **(f)** **(g)** **(h)**

Figure 4.4: Qualitative examples focusing on the challenges of hand recognition in the EgoSurgery [10] dataset. Frames (d), (g), and (h) illustrate cases where the surgeon's hands overlap with those of the assistants, often creating the illusion that they belong to the "own" class. The remaining frames show peculiar hand positions, with fingers or thumbs partially or fully occluded, further complicating the recognition process and leading to frequent misclassifications.

Chapter 5

Conclusion

This research has explored an area still underinvestigated: **egocentric analysis** of **open surgery**. Compared to minimally invasive surgery, this context presents *unique challenges* that make it extremely complex but at the same time fascinating to study. The availability of datasets such as **EgoSurgery-HTS** [10], enriched with dense and detailed annotations collected from a first-person perspective (through head-mounted cameras on the surgeon), has represented a valuable starting point. In addition to these, works such as **EPIC-KITCHENS** [9] have provided a useful comparison point, especially to evaluate the potential of **domain transfer** between everyday scenarios and the surgical environment.

The results obtained through the use of different frameworks and models highlight the intrinsic difficulties of **open surgery**: the scarcity of available datasets, the **unbalanced distribution of categories**, the high similarity in shape and texture among many instruments, the **heavy occlusions** caused by hands or other tools, and finally the wide variations in illumination and environmental context. All these factors contribute to limiting the accuracy of the model, keeping the goal of a reliable clinical application *still distant*. However, analysis has also shown that the introduction of an accurate **segmentation** can open up interesting perspectives, particularly for a more precise recognition of **contact** and **side**. Cutting-edge models such as **PointRend** [24] and **HTC** [4] have demonstrated the potential to deliver promising results, especially when tasks require greater attention to *spatial detail*.

Looking ahead, it is clear that to make these techniques truly useful in clinical practice, three key aspects must be addressed: the availability of larger and more balanced datasets, the development of robust strategies to manage **minority classes**, and the testing of model approaches capable of tackling the **specific complexities** of the surgical domain. As already

emphasized by the **EgoSurgery team**, only by increasing the amount of data available and directly addressing the critical issues that emerged will it be possible to take a concrete step toward the integration of these systems into real operational settings.

Ultimately, this study does not provide a definitive solution, but instead lays the foundation and opens the reflections for future directions. It shows that **domain transfer**, despite its limitations, can be a viable path and that advanced segmentation models offer valuable tools to address a complex and challenging problem such as **open surgery**.

Bibliography

[1] Kristina Basiev et al. "Open surgery tool classification and hand utilization using a multi-camera system". eng. In: *International Journal of Computer Assisted Radiology and Surgery* 17.8 (Aug. 2022), pp. 1497–1505. ISSN: 1861-6429. DOI: 10.1007/s11548-022-02691-3.

[2] Zhaowei Cai and Nuno Vasconcelos. *Cascade R-CNN: Delving into High Quality Object Detection*. arXiv:1712.00726 [cs]. Dec. 2017. DOI: 10.48550/arXiv.1712.00726. URL: http://arxiv.org/abs/1712.00726.

[3] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. arXiv:2104.14294 [cs]. May 2021. DOI: 10.48550/arXiv.2104.14294. URL: http://arxiv.org/abs/2104.14294.

[4] Kai Chen et al. *Hybrid Task Cascade for Instance Segmentation*. arXiv:1901.07518 [cs]. Apr. 2019. DOI: 10.48550/arXiv.1901.07518. URL: http://arxiv.org/abs/1901.07518.

[5] Kai Chen et al. *MMDetection: Open MMLab Detection Toolbox and Benchmark*. arXiv:1906.07155 [cs]. June 2019. DOI: 10.48550/arXiv.1906.07155. URL: http://arxiv.org/abs/1906.07155.

[6] Bowen Cheng et al. *Masked-attention Mask Transformer for Universal Image Segmentation*. arXiv:2112.01527 [cs]. June 2022. DOI: 10.48550/arXiv.2112.01527. URL: http://arxiv.org/abs/2112.01527.

[7] *CloudFactory*. en. URL: https://wiki.cloudfactory.com/docs/mp-wiki/model-architectures/fpn.

[8] Kevin Crowston. "Amazon Mechanical Turk: A Research Tool for Organizations and Information Systems Scholars". en. In: *Shaping the Future of ICT Research. Methods and Approaches*. Ed. by Anol Bhattacherjee and Brian Fitzgerald. Vol. 389. Series Title: IFIP Advances in Information and Communication Technology. Berlin, Heidelberg:

Springer Berlin Heidelberg, 2012, pp. 210–221. ISBN: 978-3-642-35141-9 978-3-642-35142-6. DOI: 10.1007/978-3-642-35142-6_14. URL: http://link.springer.com/10.1007/978-3-642-35142-6_14.

[9] Dima Damen et al. *The EPIC-KITCHENS Dataset: Collection, Challenges and Baselines*. arXiv:2005.00343 [cs]. Apr. 2020. DOI: 10.48550/arXiv.2005.00343. URL: http://arxiv.org/abs/2005.00343.

[10] Nathan Darjana et al. *EgoSurgery-HTS: A Dataset for Egocentric Hand-Tool Segmentation in Open Surgery Videos*. arXiv:2503.18755 [cs]. Mar. 2025. DOI: 10.48550/arXiv.2503.18755. URL: http://arxiv.org/abs/2503.18755.

[11] Ahmad Darkhalil et al. *EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations*. arXiv:2209.13064 [cs]. Sept. 2022. DOI: 10.48550/arXiv.2209.13064. URL: http://arxiv.org/abs/2209.13064.

[12] Kaiwen Duan et al. *CenterNet: Keypoint Triplets for Object Detection*. arXiv:1904.08189 [cs]. Apr. 2019. DOI: 10.48550/arXiv.1904.08189. URL: http://arxiv.org/abs/1904.08189.

[13] Yuxin Fang et al. *Instances as Queries*. arXiv:2105.01928 [cs]. May 2021. DOI: 10.48550/arXiv.2105.01928. URL: http://arxiv.org/abs/2105.01928.

[14] Ryo Fujii, Hideo Saito, and Hiroki Kajita. *EgoSurgery-Tool: A Dataset of Surgical Tool and Hand Detection from Egocentric Open Surgery Videos*. arXiv:2406.03095 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2406.03095. URL: http://arxiv.org/abs/2406.03095.

[15] Ryo Fujii et al. *EgoSurgery-Phase: A Dataset of Surgical Phase Recognition from Egocentric Open Surgery Videos*. arXiv:2405.19644 [cs]. Nov. 2024. DOI: 10.48550/arXiv.2405.19644. URL: http://arxiv.org/abs/2405.19644.

[16] Ryo Fujii et al. "Surgical Tool Detection in Open Surgery Videos". en. In: *Applied Sciences* 12.20 (Jan. 2022). Publisher: Multidisciplinary Digital Publishing Institute, p. 10473. ISSN: 2076-3417. DOI: 10.3390/app122010473. URL: https://www.mdpi.com/2076-3417/12/20/10473.

[17] Adam Goldbraikh et al. *Video-based fully automatic assessment of open surgery suturing skills*. arXiv:2110.13972 [cs]. Jan. 2022. DOI: 10.48550/arXiv.2110.13972. URL: http://arxiv.org/abs/2110.13972.

[18] Emmett D. Goodman et al. *A real-time spatiotemporal AI model analyzes skill in open surgical videos*. arXiv:2112.07219 [cs]. Dec. 2021. DOI: 10.48550/arXiv.2112.07219. URL: http://arxiv.org/abs/2112.07219.

[19] Zhibin Gou et al. *ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving*. arXiv:2309.17452 [cs]. Feb. 2024. DOI: 10.48550/arXiv.2309.17452. URL: http://arxiv.org/abs/2309.17452.

[20] Kaiming He et al. *Deep Residual Learning for Image Recognition*. arXiv:1512.03385 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1512.03385. URL: http://arxiv.org/abs/1512.03385.

[21] Kaiming He et al. *Mask R-CNN*. arXiv:1703.06870 [cs]. Jan. 2018. DOI: 10.48550/arXiv.1703.06870. URL: http://arxiv.org/abs/1703.06870.

[22] Kaiming He et al. *Masked Autoencoders Are Scalable Vision Learners*. arXiv:2111.06377 [cs]. Dec. 2021. DOI: 10.48550/arXiv.2111.06377. URL: http://arxiv.org/abs/2111.06377.

[23] W.-Y. Hong et al. *CholecSeg8k: A Semantic Segmentation Dataset for Laparoscopic Cholecystectomy Based on Cholec80*. arXiv:2012.12453 [cs]. Dec. 2020. DOI: 10.48550/arXiv.2012.12453. URL: http://arxiv.org/abs/2012.12453.

[24] Alexander Kirillov et al. *PointRend: Image Segmentation as Rendering*. arXiv:1912.08193 [cs]. Feb. 2020. DOI: 10.48550/arXiv.1912.08193. URL: http://arxiv.org/abs/1912.08193.

[25] Alexander Kirillov et al. *Segment Anything*. arXiv:2304.02643 [cs]. Apr. 2023. DOI: 10.48550/arXiv.2304.02643. URL: http://arxiv.org/abs/2304.02643.

[26] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. *Vision Transformer for Small-Size Datasets*. arXiv:2112.13492 [cs]. Dec. 2021. DOI: 10.48550/arXiv.2112.13492. URL: http://arxiv.org/abs/2112.13492.

[27] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. arXiv:1612.03144 [cs]. Apr. 2017. DOI: 10.48550/arXiv.1612.03144. URL: http://arxiv.org/abs/1612.03144.

[28] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. arXiv:1708.02002 [cs]. Feb. 2018. DOI: 10.48550/arXiv.1708.02002. URL: http://arxiv.org/abs/1708.02002.

[29] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. arXiv:1405.0312 [cs]. Feb. 2015. DOI: 10.48550/arXiv.1405.0312. URL: http://arxiv.org/abs/1405.0312.

[30] Dirk Merkel. "Docker: lightweight Linux containers for consistent development and deployment". In: *Linux J.* 2014.239 (Mar. 2014), 2:2. ISSN: 1075-3583.

[31] G. M. Merz et al. "Detection, Instance Segmentation, and Classification for Astronomical Surveys with Deep Learning (DeepDISC): Detectron2 Implementation and Demonstration with Hyper Suprime-Cam Data". In: *Monthly Notices of the Royal Astronomical Society* 526.1 (Sept. 2023). arXiv:2307.05826 [astro-ph], pp. 1122–1137. ISSN: 0035-8711, 1365-2966. DOI: 10.1093/mnras/stad2785. URL: http://arxiv.org/abs/2307.05826.

[32] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. *CUDA, release: 10.2.89*. 2020. URL: https://developer.nvidia.com/cuda-toolkit.

[33] Keiron O'Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. arXiv:1511.08458 [cs]. Dec. 2015. DOI: 10.48550/arXiv.1511.08458. URL: http://arxiv.org/abs/1511.08458.

[34] Seoung Wug Oh et al. *Video Object Segmentation using Space-Time Memory Networks*. arXiv:1904.00607 [cs]. Aug. 2019. DOI: 10.48550/arXiv.1904.00607. URL: http://arxiv.org/abs/1904.00607.

[35] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. arXiv:1912.01703 [cs]. Dec. 2019. DOI: 10.48550/arXiv.1912.01703. URL: http://arxiv.org/abs/1912.01703.

[36] Nikhila Ravi et al. *SAM 2: Segment Anything in Images and Videos*. arXiv:2408.00714 [cs]. Oct. 2024. DOI: 10.48550/arXiv.2408.00714. URL: http://arxiv.org/abs/2408.00714.

[37] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv:1506.01497 [cs]. Jan. 2016. DOI: 10.48550/arXiv.1506.01497. URL: http://arxiv.org/abs/1506.01497.

[38] Manuel Sebastián Ríos et al. "Cholec80-CVS: An open dataset with an evaluation of Strasberg's critical view of safety for AI". en. In: *Scientific Data* 10.1 (Apr. 2023). Publisher: Nature Publishing Group, p. 194. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02073-7. URL: https://www.nature.com/articles/s41597-023-02073-7.

[39] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. arXiv:1409.0575 [cs]. Jan. 2015. DOI: 10.48550/arXiv.1409.0575. URL: http://arxiv.org/abs/1409.0575.

[40] Tomohiro Shimizu et al. "Hand Motion-Aware Surgical Tool Localization and Classification from an Egocentric Camera". en. In: *Journal of Imaging* 7.2 (Feb. 2021). Publisher: Multidisciplinary Digital Publishing Institute, p. 15. ISSN: 2313-433X. DOI: 10.3390/jimaging7020015. URL: https://www.mdpi.com/2313-433X/7/2/15.

[41] Peize Sun et al. *Sparse R-CNN: End-to-End Object Detection with Learnable Proposals*. arXiv:2011.12450 [cs]. Apr. 2021. DOI: 10.48550/arXiv.2011.12450. URL: http://arxiv.org/abs/2011.12450.

[42] Christian Szegedy et al. *Going Deeper with Convolutions*. arXiv:1409.4842 [cs]. Sept. 2014. DOI: 10.48550/arXiv.1409.4842. URL: http://arxiv.org/abs/1409.4842.

[43] Ilya Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision*. arXiv:2105.01601 [cs]. June 2021. DOI: 10.48550/arXiv.2105.01601. URL: http://arxiv.org/abs/2105.01601.

[44] Zhan Tong et al. *VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training*. arXiv:2203.12602 [cs]. Oct. 2022. DOI: 10.48550/arXiv.2203.12602. URL: http://arxiv.org/abs/2203.12602.

[45] Di Wang et al. *SAMRS: Scaling-up Remote Sensing Segmentation Dataset with Segment Anything Model*. arXiv:2305.02034 [cs]. Oct. 2023. DOI: 10.48550/arXiv.2305.02034. URL: http://arxiv.org/abs/2305.02034.

[46] Limin Wang et al. *Temporal Segment Networks for Action Recognition in Videos*. arXiv:1705.02953 [cs]. May 2017. DOI: 10.48550/arXiv.1705.02953. URL: http://arxiv.org/abs/1705.02953.

[47] Xinlong Wang et al. *SOLOv2: Dynamic and Fast Instance Segmentation*. arXiv:2003.10152 [cs]. Oct. 2020. DOI: 10.48550/arXiv.2003.10152. URL: http://arxiv.org/abs/2003.10152.

[48] Ziyi Wang et al. *AutoLaparo: A New Dataset of Integrated Multi-tasks for Image-guided Surgical Automation in Laparoscopic Hysterectomy*. arXiv:2208.02049 [cs]. Aug. 2022. DOI: 10.48550/arXiv.2208.02049. URL: http://arxiv.org/abs/2208.02049.

[49] Haoyang Zhang et al. *VarifocalNet: An IoU-aware Dense Object Detector*. arXiv:2008.13367 [cs]. Mar. 2021. DOI: 10.48550/arXiv.2008.13367. URL: http://arxiv.org/abs/2008.13367.

[50] Shilong Zhang et al. *Dense Distinct Query for End-to-End Object Detection*. arXiv:2303.12776 [cs]. July 2023. DOI: 10.48550/arXiv.2303.12776. URL: http://arxiv.org/abs/2303.12776.

[51] Xizhou Zhu et al. *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. arXiv:2010.04159 [cs]. Mar. 2021. DOI: 10.48550/arXiv.2010.04159. URL: http://arxiv.org/abs/2010.04159.