Hand detection, segmentation and tracking from egocentric vision

by

Van-Tien Pham

Submitted to the School of Information Technology and Communication
in partial fulfillment of the requirements for the degree of

Master of Science in Information System and Communication

at the

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

October 2020

Author .....
School of Information Technology and Communication
October 10, 2020

Certified by .....
Thi-Thanh-Hai Tran
Associate Professor
Thesis Supervisor

Accepted by .....
Chairman
Chairman, Department Committee on Graduate Theses

Hand detection, segmentation and tracking from egocentric vision

by

Van-Tien Pham

Submitted to the School of Information Technology and Communication
on October 10, 2020, in partial fulfillment of the
requirements for the degree of
Master of Science in Information System and Communication

Abstract

Multiple object tracking is the process of assigning unique and consistent identities to objects throughout a video sequence. A popular approach to multiple object tracking is to use a method called tracking by detection. Tracking by detection is a two-stage procedure: an object detection or segmentation algorithm first detects objects in a given frame, these detected objects are then associated with already tracked objects in a second step by a tracking algorithm. Egocentric vision is an emerging field of computer vision that is characterized by the acquisition of images and video from the first-person perspective. In egocentric view, the two human hands are essential in the execution of actions and characterizing their movements and trajectories are the principal cues to define and recognize actions.

One of the main concerns of this thesis is to develop an automatic tracking by detection algorithm that extracts hands positions and identities in consequence frames from egocentric surveillance video. The proposed framework consists of state-of-the-art detectors from RCNN and YOLO family models combined with the SORT or DeepSORT for object tracking task. The thesis aims to explore how the stand-alone performance of the object detection algorithm correlates with overall performance of a tracking-by-detection system. Finally, the thesis investigates how the use of visual descriptors of DeepSORT in the tracking stage of a tracking-by-detection system effects performance.

Results presented in this thesis suggest that the capacity of the object detection algorithm is highly indicative of the overall performance of the tracking-by detection system. Further, this thesis also shows how the use of visual descriptors in the tracking stage can reduce the number of identity switches and thereby increase performance of the whole system. This thesis also presents a new egocentric hand tracking dataset Micand32 for future researches.

Thesis Supervisor: Thi-Thanh-Hai Tran
Title: Associate Professor

Acknowledgments

First of all, I might want to offer my special thanks to my supervisor, Assoc. Prof. Tran Thi Thanh Hai. I'd really appreciate everything she've guided me all through this thesis.

I would like to thank my colleagues at Viettel High Technology Industries Corporation for supporting me in technical issues. Also, I might want to express gratitude toward Assoc. Prof. Vu Hai and alumni at MICA Institute, Hanoi University of Science and Technology for giving me significant suggestions.

Deep inside my heart, I wish to show my gratefulness to my family for always inspiring and trusting me in every of my steps.

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Chapter 1

Introduction

1.1 Overview of object recognition and tracking from video

1.1.1 Object recognition

Object recognition aims at detecting the presence of an object in image and giving it a label that is the category to which it belongs. Objects of interest can be face, vehicle, hand, people, tree, tumors depending on applications such as face id, intelligent traffic system and autonomous vehicles, human machine interaction, health care, security or bio diversity, etc. Segmentation is fundamental that go further than object detection and classification by give a label to a pixel, not a bounding box. There are two types of segmentation: semantic segmentation and instance segmentation. Semantic segmentation classifies all pixels of an image into meaningful object categories. These categories are "semantically interpretable" and correspond to the classes in the real world. Semantic segmentation gives an unique label to two objects of the same category. This is called dense prediction because it can predict the meaning of each pixel. Instance segmentation otherwise gives every pixel belonging to an object instance a label. Since the past decade, object detection and recognition as well as segmentation has achieved impressive performance thanks to the significant advances of AI and deep learning. Deep learning can learn patterns in visual input in order to predict not only the categories of objects in image but also the ones of pixels. Deep learning architectures used for object detection / recognition / segmentation are convolutional neural networks (CNN) or specific CNN frameworks such as AlexNet, VGG, Inception and ResNet.

1.1.2 Object tracking

Tracking objects in a video stream involves association of moving objects in consecutive video frames. In order to track an object, the target object requires to be firstly detected manually (detection free trackers) or automatically by detection algorithms (detection based tracker). Then tracking algorithm will associate detected objects to existing tracks by putting constraints on distance function between movement and appearance of the object with its previous instances. Besides, giving a complete trajectory of a moving object during time is very important for video analysis. Tracking is more helpful to solve some common challenges (e.g. as lighting changes, motion blur, zoom ratio changes, occlusion when the target is partially or completely hidden by another object in the video for a period of time, poor image quality) from that simple object detection often suffers from.

Almost proposed trackers until now based on Siamese network or Correlation Filter (CF), combined with effective appearance models (CNN, HOG). In challenges on object tracking task, most of the highest performance obtained with CF trackers. Their performances is better than Siam tracker. In term of computational time, Siam tracker's performance is better than CF. Depending on the context and application, tracking could be single object tracking (SOT) and multiple object tracking (MOT). MOT is more challenging than SOT because ID switching is difficult to avoid, especially in crowded videos, the nature and number of objects in each frame are unknown. Therefore, MOT algorithms strongly rely on detection algorithms. Unfortunately, detection algorithms itself are not perfect. A popular object tracking method is to use a method called tracking by detection. It first apply object detection algorithms to detect objects in current frame. These objects are then tracked by associating the objects in the current frame with the objects in the previous frame using a tracking algorithm. Having a reliable object detection method is crucial, because the tracking algorithm depends on the objects detected in each frame. Recently, target detection algorithms based on convolutional neural networks have been able to achieve higher accuracy than traditional target detection methods. The improvement of object de-

tection accuracy promotes the use of one-by-one detection and tracking method for multiple object tracking.

1.2 Context and scope of the thesis

1.2.1 Egocentric vision

Egocentric vision or first-person vision (FPV) is a sub-field of computer vision, which requires the analysis of images and videos captured by a wearable camera, which is usually worn on the head or chest, and naturally approximates the camera wearer vision. Therefore, the visual data captures the part of the scene where the user is focused on performing on-site tasks and provides a valuable perspective to understand the user's activities and their environment in the natural environment. In recent years, the research community has adopted a self-centered perspective to solve computer vision challenges, such as activity recognition [8] and object detection [9] that are traditionally considered to belong to the field of third-person vision. Since then, self-centered vision has been applied to more complex applications, including video summarization [10] and social interaction analysis [11]. It is worth noting that it has also been extended to the field of healthcare [12], in which static camera systems tend to struggle to a greater degree with regard to privacy issues [13]. Ultimately, self-centered vision is associated with the field of augmented reality to enhance human-centered applications that provide help for specific tasks [14]. In the medical field, FPV can help to build applications that aid people with dementia by recording the sensor carrier's daily activities. In rehabilitation therapy, or motor support in the elderly, physicians are often interested in monitoring the patient's recovery progress through movements and daily activities such as arm lifts, movement of the wrist within grasp objects. Currently, monitoring are mainly using the naked eye for observation, automatic monitoring tools are very limited in hospitals. Through the automatic analysis and recognition of activities from the series of images captured by the carrier camera (FPV video), the treating doctor can identify and quantify the

patient's progression for a therapeutic regimen value accordingly. In sports, the use of egocentric cameras is increasingly common. The egocentric systems don't just collect front-view imagery during the journey of speed sports such as cycling and skiing but also supports analysis of accuracy in sports movements such as golf, basketball. One notable feature is that the FPV image sequence does not only contain information about the ego-motion of target itself, but also the interactive movements between the hand and the subject (shadow, golf club) simultaneously. In sports, the recording of movements at critical moments (ball contact point, hand movement direction) plays an important role in determining the athlete's success or failure. In the field of teaching aids, virtual reality, the use of FPV techniques has brought remarkable results. It can be considered the role of the camera carried on the body as a sixth sense (six-sense). On the one hand, FPV assists in detecting the behavior (e.g. hand gestures) of the subject, on the other hand, the camera carries the observer or the affected person. Consequently, the system reacts in accordance with the user's requirements. One particular application is the interaction between an observed object and the person carrying a sensor while visiting a museum. Not only the system can detect the object that the visitor is interested in but also identify the behavior (gesture) that the visitor wants to interact with the system (to support details, or see objects from different perspectives). In educational applications, virtual reality is receiving more and more attention besides medical applications when developing egocentric vision systems.

1.2.2 Background project and motivation

This thesis falls under the scope of the project "Understanding Human Daily Life Activities from Egocentric Vision Using Advanced Technologies of Deep Learning" funded by National Foundation for Science and Technology Development (NAFOSTED). The objective of this project is to analyze activities of human (elderly people or patients) in daily life or during rehabilitation through analyzing two important factors: the role of hand gestures in interacting with objects; and the role of the environment in identifying sensor carrier activity. To achieve these goals, the project focuses

on solving the fundamental problems of egocentric-vision such as: summarizing video from large data sources (up to millions of images per day), effectively exploiting relationships between the sensor carrier and the interactive object, between environment and context. Specifically, the project implements three main tasks: (1) developing advanced deep learning techniques for hand segmentation and activities recognition, (2) exploiting the correlation between different factors such as environment condition, time execution and the way of execution of a certain exercise/activity; (3) deploying proposed techniques to evaluate the rehabilitation of patients after a period of training.

My thesis in the project focuses on detecting, segmenting and tracking human hand in egocentric videos. I first study existing deep learning based methods for object segmentation and tracking then propose a framework that takes consecutive frames as input, segments hand regions frame by frame then track the hands along frames. I then implement the framework which allows to flexibly integrate different CNN architectures for objects detection (i.e. YOLO [5], Faster CNN [3], Mask R-CNN [4]) as well as object tracking (i.e. SORT [15], DeepSORT [7]). To train CNN models, I develop a tool to speed up the tracking annotation using automatic segmentation results. Finally, I evaluate the developed framework on a hand dataset captured in the context of the project.

1.3 Related works

1.3.1 Video object recognition and tracking challenges

Densely Annotated Video Segmentation (DAVIS) challenge [11] is a fairly famous public competition designed for the task of video object segmentation that has been going on for 4 years from 2016 to 2020. It is a benchmark dataset and evaluation methodology for video object segmentation that consists of fifty high quality, full HD video sequences, accompanied by densely annotated, pixel-accurate and per-frame ground truth segmentation. It provides a comprehensive analysis of several state-

of-the-art segmentation approaches using three complementary metrics. These are: semi-supervised challenge, interactive challenge and unsupervised challenge with specific datasets, definitions, rules and evaluation metrics. The popular Visual Object Tracking (VOT) challenges [16] provides the visual tracking community with a precisely defined and repeatable way to compare short-term trackers, and provides a common platform for discussing evaluation and progress in the field of visual tracking. The goal of the challenge is to build a large database of benchmarks and organize seminars or similar activities to promote research on visual tracking. Multiple Object Tracking (MOT) [17] challenge is a well-known benchmark for multi-object tracking that collects a large variety of sequences and provides a framework for the standardized evaluation of multiple object tracking methods. Currently, the benchmark is focused on multiple people tracking, since pedestrians are by far the most studied object in the tracking community. The benchmark includes 2D, 3D and multi-camera challenges. The tracking evaluation is this thesis uses the devkit protocol of the MOT16 which provide several measures, from recall, precision to running time.

1.3.2 Hand gestures recognition related works

One of the first works on the understanding of egocentric activities focused on defining the inner message of the egocentric visual paradigm [8]. They use the extracted visual features to model the relationship between hands, objects, and actions to model activities, and demonstrate the mutual improvement provided by these relationships through bottom-up and top-down models. In [9], the authors have developed a weakly supervised technique that can identify objects by sculpting out objects from large sequences of self-centered activities. In general, this is a difficult task, their algorithm uses domain-specific knowledge from a first-person perspective to make it feasible. The method will automatically segment the active object area, assign some areas to each object, and use semi-supervised learning to disseminate its information. They proved that this method can reliably compare active classes based on the usage patterns of objects.

Movement-based egocentric action recognition is described in [18]. According to

the motion and color-based features and trajectories extracted from the video frames, the interaction points of the hand and the object, the head and the self-movement are declared as actions, but the position of the modeled hand is not paid special attention. [19] describes another multi-modal approach to egocentric activity recognition. The hand segmentation network, the object localization network and the network trained by the motion flow are combined to predict the action.

In [20], an architecture based on two-stream visual segmentation was used to predict the interaction area between hands and objects in a video stream and model them as actions. In [21], the concept of hand-object interaction is further explored, and the object shape-related grasping related to modeling actions is detected. The end-to-end approach also includes [22], where in order to recognize actions, the network is trained on paired frames and optimized for action recognition, object segmentation and inter-frame object interaction, as well as their training targets associated with recurrent networks. This work also assumes that hands and objects are the basis of self-centered actions, but it emphasizes the need to clearly detect the areas and positions of hands and objects to recognize actions. A lot of works have been done in the explicit exploration of opponents and objects and their temporal relationships. Hand detection, segmentation and recognition techniques were developed [23] [24] [25], and the results were used to model behaviors or activities. In [26], hand-based activity recognition from an egocentric perspective is discussed. Before inferring the activity, the EgoHands dataset and the egocentric hand detection and segmentation pipeline were developed. This is one of the manual works, showing the difference between relying on hand detection or segmentation and using manual label for activity classification. In the literature [27], activity recognition based on egocentric hands is considered, in which the distance between the detected hands or the distance between the detected hands and the objects marked as activities is considered as the feature of activity classification. In [28], the egocentric human action recognition is solved by explicitly using the existence and location of the region of interest in the scene without further use of visual features. Their understanding that the human hand is very important in performing actions, and focused on its actions as the main clues

Table 1.1: Problem formulation.

Input	Sequence of frames from an egocentric surveillance video
Output	Trajectories of the tracked hands, includes: • The number of human hands in the current frame • The location of each hand, represented by a bounding box for the detection algorithm, or a set of pixels for the segmentation algorithm • Identity of each hand across the video

to define actions. Finally, in all the works above, the suggestion on the choice of a suitable methods for online tracking applications is unavailable.

1.4 Problem formulation and assumptions

1.4.1 Objective

Table 1.1 defines the input and output of the problem in this thesis. The goal of this thesis is to solve the problem of detecting, segmenting and tracking human hand objects in the video from the first perspective. Along with researching, confirming the theory, and proposing a hand tracking by detection system, I built a module to detect and track human hands in videos from the first perspective. The tracking by detection approach uses the combination of a detector and a tracker as following:

- Detector: Faster RCNN, Mask RCNN, MaskRCNN with region based, YOLOv3, YOLOv4 or Ground-truth
- Tracker: SORT or DeepSORT

The inference results on test dataset are used to analyze the effect of phase detection algorithm selection on hand tracking phase, and also the efficiency of the DeepSORT algorithm compared to the SORT algorithm in the tracking phase.

1.4.2 Contribution

The main contributions of this thesis are three-folds:

- First, a framework for hand detection, segmentation and tracking from egocentric vision pipeline is proposed. This framework contains state-of-the-art detection algorithms in RCNN and YOLO families for object detection and segmentation, and two tracking algorithm SORT and DeepSORT for object tracking. Any other algorithms can be simply integrated into this framework.

- Second, a comparative evaluation of combination of a detector and a tracker is conducted and analyzed on both term of accuracy and performance. Consequently, the recommendation for choosing a suitable method for egocentric hand tracking applications is given.

- Third, Micand32, a new egocentric hand detection, segmentation and tracking datasets is built during this thesis, accompanied with labeling, visualizing and evaluation tools.

A part of this thesis is published in the Proceeding of the 3^{rd} International Conference on Multimedia Analysis and Pattern Recognition [29]. The code developed in this thesis is made available at `https://github.com/pvtien96/Detectron2DeepSortPlus`.

1.4.3 Thesis outline

The thesis is structured into 5 chapters:

1. **Introduction** summarize the overview of object recognition, including object classification, detection, segmentation and tracking in the wild and in the first person vision (FPV). Related works are studied and discussed. The thesis's scope, objective and contribution are then described.

2. **Methodology and Datasets** presents the RCNN and YOLO family models; SORT and DeepSORT algorithms. This chapter analyzes egocentric hand datasets incorporate GTEA family, EgoHand; and furthermore present a new egocentric hand tracking datasets Micand32 alongside with a semi-automatic annotator,the EHTA.

3. **Proposed Framework** introduces the proposed architecture which contains 4 stage: data preparing stage, training stage, inference stage and evaluation stage. This chapter reports in detail techniques in training and testing phase.

4. **Experiments** conserves about the evaluation criteria and then enumerate overall and featured detection, segmentation result following the COCO dataset standard and tracking result of 11 aproaches on Micand32S and Micand32E following the MOT Challenge standard.

5. **Conclusion** terminates the thesis by considering the accomplishment, the drawback and also purpose some potential future works.

Chapter 2

Methodology and Datasets

2.1 Tracking by detection approach

Tracking multiple objects is a task of assigning an unique and consistent identifier to each object in a video series. This section presents an object tracking technique called "tracking by detection". Tracking through detection is a two-stage process: the object detection algorithm first detects the objects present in the frame. These detected objects are then linked to those that have been tracked through a tracking algorithm. Usually, the object detection algorithm and tracking algorithm are completely separate from each other, so they can be analyzed separately. Object detection is a process of detecting specific categories of objects in an image, examples of these categories are things such as pedestrian or car. The purpose of the object detection algorithm is to locate and classify objects belonging to any popular category. Therefore, for each detected object, the object detection algorithm produces an estimate of the object's location, size, and category. The position and size of the detected object are usually represented by a bounding box, which is a rectangular box surrounding the object. The range of the detected object can also be defined by a segmentation mask, which is a pixel-level mask of the object. Due to recent advances in the field of image classification, target detection has made considerable progress. This progress is attributed to a breakthrough in how to use CNN for image classification [30]. The target detection algorithm usually consists of a CNN designed for image classification, and then has an algorithm-specific additional structure around the CNN. CNN is called the backbone of the algorithm, and the algorithm-specific structure is called the meta-architecture. By convention, in this thesis, I identify object detection algorithm through its meta-architecture. CNN-based object detection algorithm can be

divided into two different groups: single-stage and two-stage detectors [31]. The two-stage detector first generates bounding boxes by segmenting the image into regions of interest, and then CNN classifies these regions in the second stage. The single-stage detector is bounding box and class estimates in a single forward pass of the image through the CNN. Traditionally, two-stage detectors have achieved higher accuracy at the expense of speed compared to single-stage detectors. However, the recently introduced loss function Focal loss [32] makes the accuracy of a single-stage detector close to that of a two-stage detector. The trade-off between speed and accuracy is the main design choice, as studied in the paper [33]. The tracking algorithm in the "tracking by detection" framework is responsible for assigning unique identifiers to the tracked objects and establishing object associations between frames. In my work, I focuses on target detection algorithm, and particularly considers two different tracking algorithm: SORT and DeepSORT. SORT stands for simple online and real-time tracking. It is a deliberately simple tracking algorithm that uses a Kalman filter [34] to estimate the future position of an object, and uses the Hungarian method [35] for frame-to-frame correlation. Deep SORT is an extension of SORT that incorporates appearance information when performing object association between frames.

2.2 Object detection and segmentation algorithms

2.2.1 RCNN model family

RCNN

R-CNN is an abbreviation for regions with CNN features, and is an object detection method introduced by Girschick et al in [1]. The system consists of three main components: region proposal, convolution neural network and a set of support vector machines (SVM). Figure 2-1 shows the relation of R-CNN's component. First, the region proposal method divides the image into regions irrelevant to the category. Each image produces approximately 2000 regions. After segmenting the image, each region is deformed into a fixed size to fit the required input size of the CNN. Next, 2000

deformed regions were fed through the CNN, and feature vectors are extracted for each region. Then, the feature vectors are registered by a set of linear SVMs, where each SVM is trained to classify a specific category. Finally, given the class predicted by SVM, regression is used to improve the predicted shape of the bounding box. Given a predicted bounding box coordinate $p = (p_x, p_y, p_w, p_h)$ (center coordinate, width, height) and its corresponding ground truth box coordinates $g = (g_x, g_y, g_w, g_h)$, the regressor is configured to learn scale-invariant transformation between two centers and log-scale transformation between widths and heights. All the transformation functions take p as input.

$$\hat{g}_x = p_w d_x(p) + p_x \tag{2.1}$$

$$\hat{g}_y = p_h d_x(p) + p_y \tag{2.2}$$

$$\hat{g}_w = p_w \exp\left(d_w(p)\right) \tag{2.3}$$

$$\hat{g}_h = p_h \exp\left(d_h(p)\right) \tag{2.4}$$

An obvious benefit of applying such transformation is that all the bounding box correction functions, $d_i(p)$ where $i \in x, y, w, h$, can take any value between $[-\infty, +\infty]$. The targets for them to learn are:

$$t_x = (g_x - p_x)/p_w \tag{2.5}$$

$$t_y = (g_y - p_y)/p_h \tag{2.6}$$

$$t_w = \log\left(g_w/p_w\right) \tag{2.7}$$

$$t_h = \log\left(g_h/p_h\right) \tag{2.8}$$

A standard regression model can solve the problem by minimizing the SSE loss with regularization:

$$\mathcal{L}_{reg} = \sum_{i \in (x,y,w,h)} (t_i - d_i(p))^2 + \lambda ||w||^2 \tag{2.9}$$

The regularization term is critical here and RCNN picked the best λ by cross validation. It is also noteworthy that not all the predicted bounding boxes have corresponding ground truth boxes. For example, if there is no overlap, it does not make sense to run bbox regression. Here, only a predicted box with a nearby ground truth box with at least 0.6 IoU is kept for training the bounding box regression model.

After scoring all regions, non-maximum suppression will be applied to remove prediction bounding boxes that overlap with predictions with higher scores. R-CNN is not limited to any specific segmentation method or specific CNN architecture. In [1], a segmentation method called selective search [36] is used, and the results of using the CNN system structure introduced in [30] and [37] are demonstrated.

The author of R-CNN also shows that supervised pre-training for similar problems is an effective way to initialize CNN weights. In [1], pre-trained CNN is used to classify the data from ILSVRC2013 to obtain the initial weight of CNN [38]. Then, by training it on the Pascal VOC 2012 dataset, fine-tune the CNN to perform object detection, which is an object detection dataset. This is a form of transfer learning, which has proven to be an effective method when adapting CNN to a domain with sparse training data [39].

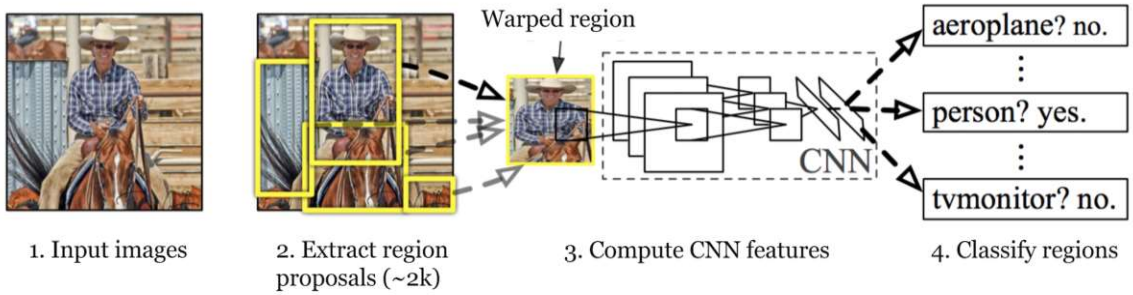

Figure 2-1: Schematic of the R-CNN pipeline [1].

FastRCNN

The main disadvantage of the R-CNN method is its slow speed. This is mainly because each regional proposal is passed through CNN separately, which is very time-consuming. In order to improve the speed, Girshick introduced a target detection

Table 2.1: Some definitions for calculating FastRCNN losses.

Symbol	Explanation
u	True class label, $u \in 0, 1, \ldots, K$; by convention, the catch-all background class has $u = 0$
p	Discrete probability distribution (per RoI) over K + 1 classes: $p = (p_0, \ldots, p_K)$, computed by a softmax over the K + 1 outputs of a fully connected layer.
v	True bounding box $v = (v_x, v_y, v_w, v_h)$.
t^u	Predicted bounding box correction, $t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$.

method called Fast R-CNN in [2]. Fast R-CNN can improve the speed of object detection, mainly by passing the image forward through CNN only once, rather than once for each region performed by R-CNN. As in R-CNN, the region proposal method first divides the image into category-independent regions, creating a region of interest (RoI). Then, the entire image is processed by a CNN, which does a convolutional feature map of the image. Next, for each regional proposal, the RoI pool layer using spatial pyramid pool [40] is applied to the feature map. This will convert each RoI into a fixed-size vector. Then, the feature vectors are processed by fully connected layers, which are divided into two different output layers. One of the output layers is the softmax layer, which estimates probability estimates for object classes. The other layer is the bounding box regressor, which outputs a refined estimate of the bounding box for each object class. Figure 2-2 shows how an image is processed by Fast R-CNN. The model is optimized for a loss combining two tasks (classification +

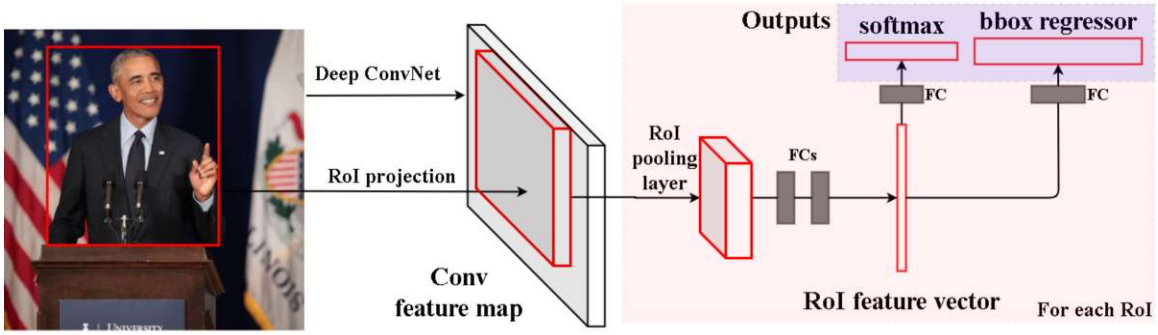

Figure 2-2: The architecture of Fast RCNN [2].

localization): The loss function sums up the cost of classification and bounding box prediction: $\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box}$. For "background" RoI, $\mathcal{L}_{box}$ is ignored by the indicator

function $f[u \geq 1]$, defined as in equation 2.10.

$$f[u >= 1] = \begin{cases} 1 & \text{if } u \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

The overall loss function is:

$$\mathcal{L}(p, u, t^u, v) = \mathcal{L}_{cls}(p, u) + f[u \geq 1]\mathcal{L}_{box}(t^u, v) \tag{2.11}$$

$$\mathcal{L}_{cls}(p, u) = -\log p_u \tag{2.12}$$

$$\mathcal{L}_{box}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{smooth}(t_i^u - v_i) \tag{2.13}$$

The bounding box loss $\mathcal{L}_{box}$ should measure the difference between t_i^u and v_i using a robust loss function. The smooth L1 loss is adopted here and it is claimed to be less sensitive to outliers.

FasterRCNN

Fast R-CNN passing the speed of object detection by passing the image forward through the CNN only once, instead of forward passing through each region of interest in the image. For Fast R-CNN, the bottleneck lies in the image segmentation method usually implemented on the CPU. In order to solve this problem, Ren et al. A method called Faster R-CNN is proposed in [3]. Faster R-CNN eliminates the need for CPU computing by introducing the idea of a Region Proposal Networks (RPNs). The region proposal network to region proposed by sliding a small network on the convolutional feature map. At each location, the small network takes a window of the convolution feature map and converts it into a feature vector. This feature vector is then input into two different fully connected layers, one layer performs bounding box regression, and the other layer is a classification layer that predicts objective scores. The objectivity score is a prediction of the probability that the predicted bounding box contains only one object compared to the background.

For each position, RPN makes several predictions relative to a fixed number of reference frames, which are called anchors. You can anchor as a suggested border for each sliding window position. In [3], anchors are created in 3 ratios and 3 different aspect ratios, and each position provides a total of 9 different anchors. This means that RPN has 9 bounding boxes at each sliding window position, and each anchor point has one bounding box.

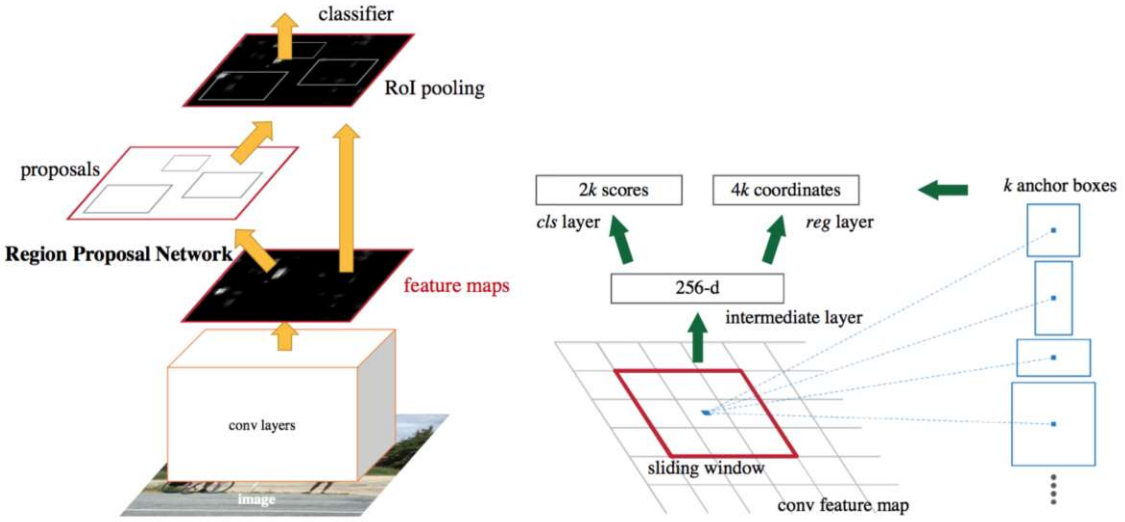

Figure 2-3: An illustration of Faster RCNN model [3].

The regions generated by RPN are then used as region suggestions in fast RCNN. By using RPN, Faster R-CNN eliminates the time-consuming image segmentation required in Fast R-CNN. By using a single CNN for RPN and fast R-CNN, the speed is further increased. This also means that the Faster R-CNN can be trained end-to-end by first training the RPN to propose a region, and then using the region proposal to train the Fast R-CNN.

Faster R-CNN is optimized for a multi-task loss function, similar to fast R-CNN. With notation in Table 2.2.1 the multi-task loss function combines the losses of classification and bounding box regression:

$$\mathcal{L}_{reg} = \mathcal{L}_{cls} + \mathcal{L}_{box} \tag{2.14}$$

Table 2.2: Some definitions for calculating FasterRCNN losses.

Symbol	Explanation
p_i	Predicted probability of anchor i being an object.
p_i^*	Ground truth label (binary) of whether anchor i is an object.
t_i	Predicted four parameterized coordinates.
t_i^*	Ground truth coordinates.
N_{cls}	Normalization term, set to be mini-batch size (∼256) in the paper.
N_{box}	Normalization term, set to the number of anchor locations (∼2400) in the paper.
λ	A balancing parameter, set to be ∼10 in the paper (so that both $\mathcal{L}_{cls}$ and $\mathcal{L}_{box}$ terms are roughly equally weighted).

$$\mathcal{L}(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i \mathcal{L}_{cls}(p_i, p_i^*) + \frac{\lambda}{N_{box}}\sum_i p_i^* \cdot L_1^{smooth}(t_i - t_i^*) \tag{2.15}$$

where $\mathcal{L}_{cls}$ is the log loss function over two classes, as we can easily translate a multi-class classification into a binary classification by predicting a sample being a target object versus not. L_1^{smooth} is the smooth L1 loss.

$$\mathcal{L}_{cls}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*)\log(1 - p_i) \tag{2.16}$$

MaskRCNN

The mask R-CNN is an extension of Faster R-CNN proposed by He et al in [4]. In addition to object detection, Mask R-CNN can also perform object instance segmentation. Segmentation is achieved by adding a third branch to Faster R-CNN, which outputs an object mask for each detected object. In order to improve the segmentation, a method called RoIAlign is introduced to extract more accurate feature maps for each RoI. RoIAlign uses bi-linear interpolation instead of quantizing the feature map to calculate the exact value of the feature map. The author of [4] found that Mask R-CNN has a higher average accuracy than Faster R-CNN in target detection. It turns out that this is partly due to the use of RoIAlign and partly due to the multi-task loss used to train Mask R-CNN. Mask R-CNN has a multi-task loss function, which can simultaneously consider classification, bounding box regression and object segmentation. The multi-task loss function of Mask R-CNN combines the loss

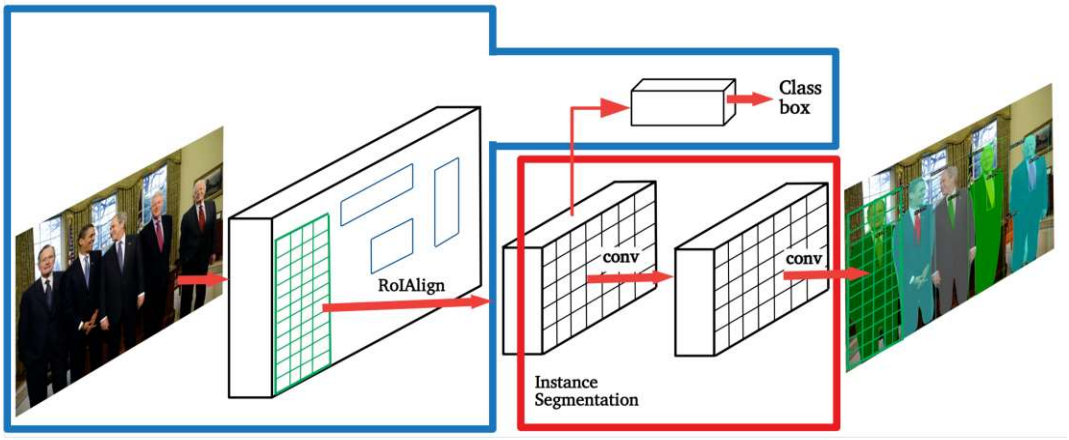

Figure 2-4: The Mask RCNN framework for instance segmentation [4].

of classification, localization and segmentation mask:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{box} + \mathcal{L}_{mask} \tag{2.17}$$

where $\mathcal{L}_{cls}$ and $\mathcal{L}_{box}$ are same as in Faster R-CNN. The mask branch generates a mask of dimension m x m for each RoI and each class; K classes in total. Thus, the total output is of size $K \cdot m^2$. Because the model is trying to learn a mask for each class, there is no competition among classes for generating masks. $\mathcal{L}_{mask}$ is defined as the average binary cross-entropy loss, only including k-th mask if the region is associated with the ground truth class k.

$$\mathcal{L}_{mask} = -\frac{1}{m^2} \sum_{1 \leq i,j \leq m} \left[y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right] \tag{2.18}$$

where y_{ij} is the label of a cell (i, j) in the true mask for the region of size m x m; $\hat{y}_{ij}^k$ is the predicted value of the same cell in the mask learned for ground-truth class k.

2.2.2 YOLO model family

YOLO

In [5], a novel object detection method is introduced, called YOLO, which means "You Only Look Once". Unlike R-CNN and its successors, YOLO does not use any

region proposal method, but uses a single CNN to predict bounding boxes and classes. In YOLO, the input image is first divided into S × S grids. Then, each grid unit is responsible for predicting the bounding box B and the confidence score of each bounding box. The formula for calculating the confidence score is $Pr(Object) * IoU^{gt}_{pred}$, where $Pr(Object)$ is the predicted probability that the box contains an object, and IoU^{gt}_{pred} is the estimated intersection over union (IoU) between the predicted box and the ground truth box. For each grid unit, the probability of object categories C can also be predicted, and these probabilities are conditioned on the unit containing the object. The predicted box and class probabilities are then combined into a single score for each class and box.

Equation 2.19 comes from the introduction by YOLO in [5], which shows how class prediction and box prediction are combined. As shown in the original paper, $Pr(Class_i)$ is used as a simplified representation of $Pr(Class_i|Object)$.

$$Pr(Class_i|Object) * Pr(Object) * IoU^{gt}_{pred} = Pr(Class_i) * IoU^{gt}_{pred} \tag{2.19}$$

This score not only explains the probability that the box contains class i, $Pr(Class_i)$, but also how to estimate the predicted box to fit the ground truth box $Pr(Object) * IoU^{gt}_{pred}$. The base model 2-5 is similar to GoogLeNet with inception module replaced by 1x1 and 3x3 convolution layers. The final prediction of shape S×S×(5B+K) is produced by two fully connected layers over the whole convolution feature map.

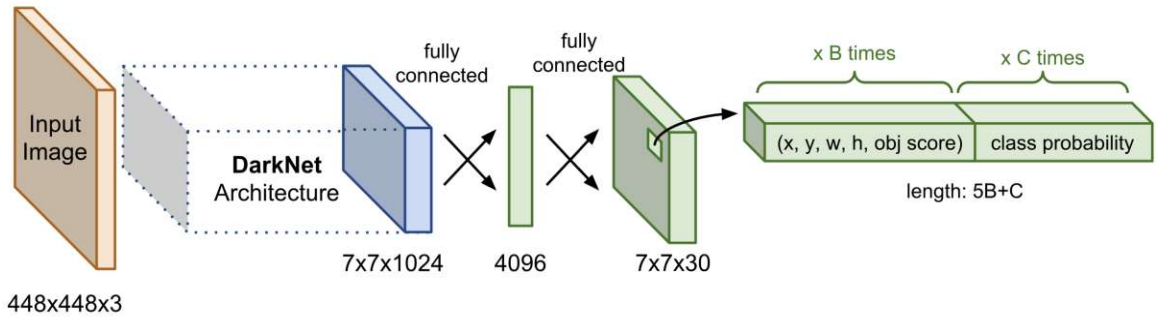

Figure 2-5: The network architecture of YOLO [5].

YOLOv2

In order to improve YOLO, Redmon et al. proposed a method called YOLOv2 in [41]. YOLOv2 is a modified version of YOLO, designed to improve speed and accuracy. Similar to Faster R-CNN, YOLOv2 uses anchor points when predicting the bounding box. For each grid unit, YOLOv2 generates a bounding box by predicting the offset of 5 anchor points. It is now also possible to predict the class for each anchor point instead of each grid unit, and also provide an objective score for each anchor point. As in YOLO, the class is predicted under the condition that the object $Pr(Class_i|Object)$ exists. Objectivity is calculated as the estimated IoU between the predicted box and the estimated ground truth box IoU^{gt}_{pred}. YOLOv2 also uses a new method to determine the anchor point size. YOLOv2 is different from manually selecting anchor points in Faster R-CNN, but uses k-means clustering on the training data to generate anchor points that are more suitable for the data. In order to increase speed, a CNN architecture called Darknet-19 is introduced in YOLOv2. Darknet-19 is able to achieve higher image classification accuracy than both the widely used VGG-16 and the custom network previously used in YOLO. It manages to do this while only using 5.58 * 109 floating point operations per forward pass, compared to 30.69 * 109 operations in VGG-16 and 8.52 * 109 operations in the network previously used in YOLO.

YOLOv3

YOLOv3 includes a further improvement of YOLOv2 proposed by Redmon et al in [42]. Similar to the feature pyramid network (FPN) described in [43], in YOLOv3, the box will be predicted at 3 different scales. This YOLOv3's ability to detect small objects, which was struggling with earlier versions of YOLO. Inspired by the residual network proposed in [44], Darknet-19 was extended to include a residual layer. This new CNN architecture is called Darknet-53 because it has a total of 53 convolutional layers. Compared with Darknet-19, Darknet-53 has higher accuracy but slower speed.

YOLOv4

So far, YOLOv4 is the latest and most advanced iteration [45]. It has the fastest running speed and can be used for optimization of production systems and parallel computing. Some of the new technologies adopted in YOLOv4 are: (1) weighted residual connection, (2) cross-stage-partial connection, (3) cross mini-batch processing, (4) normalization (CmBN), (5) self-confrontation training, (6) Mish-activation, etc. In order to obtain higher precision values, YOLOv4 uses Dense Block, which is a deeper and more complex network. Similarly, the backbone of its function extractor uses CSPDarknet-53, which deploys CSSP connection with Darkenet-53 of the early YOLOv3. In addition to CSPDarknet-53, YOLOv4's architecture also includes SPP add-on modules, PANet path aggregation neck and YOLOv3 anchor-based head. SPP blocks are stacked on CSPDarknet53 to increase the receiving field that can discretize the most significant context features and ensure that its network operation speed will not decrease. Similarly, PANet is used to aggregate parameters from multiple backbone levels, instead of the FPN used in YOLOv3.

2.3 Object tracking algorithms

2.3.1 SORT

SORT is a tracking algorithm introduced by Bewley et al in [15]. SORT is designed to perform multi-object tracking in a tracking-by-detection system. In order to achieve real-time processing, SORT is deliberately kept simple to avoid performing complex and time-consuming tasks. In order to make up for its lack of complexity, SORT uses CNN-based object detectors instead of relying on more accurate object detection. For each new frame, SORT first propagates objects that are already tracked into the current frame. The new positions of these already tracked objects are predicted using a Kalman filter [34] with a linear constant velocity model. Next, an object detection algorithm detects objects present in the current frame. These detected objects are then compared to already tracked objects and a cost-matrix is created. This cost-

matrix is calculated as the IoU between each detection and each of the already tracked objects. Detections are then assigned to already tracked objects using the Hungarian method [35]. When an object is detected in several consecutive frames and does not overlap with any tracked object, a new trajectory is created. To further explain this point, the object model is represented by equation 2.20, where u, v, s, and r represent the horizontal pixel position, vertical pixel position, area, and aspect ratio of the target object, respectively.

$$X = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}] \tag{2.20}$$

As long as the detection is linked to the target object, the detected bounding box is used to inform the target state, and the Kalman filter is used to solve the level and velocity values. This helps to identify the target's identity in consecutive frames and helps tracking.

2.3.2 DeepSORT

DeepSORT is built to reduce the number of identity switches and integrate appearance information into the tracking process proposed in SORT. Similar to SORT, DeepSORT uses Kalman filter to process state estimation. The difference between DeepSORT and SORT is that it utilizes other technologies when assigning detection to the tracked object. The first improvement of DeepSORT is the deep appearance descriptor. To obtain the appearance information of detections and tracks, an appearance descriptor is used to extract features from detection images and track images from previous frames. The appearance descriptor is a CNN trained on large-scale person re-identification dataset. It is able to extract features in a way that features from the same identity are close together and features from different identities are far away from each other in the feature space. The overview of the network architecture of the re-identification model used in DeepSORT framework is shown Table 2.3.2. This thesis customizes the CNN on a custom egocentric hand re-identification dataset, Micand32S. Detail training process is explained in chapter 3, sub-section 3.2.2. Appearance descriptors are computed by forwarding each bounding box through a CNN that

Table 2.3: Overview of the CNN architecture [7]. The final batch and L2 normalization projects onto the unit hyper-sphere.

Name	Patch Size/Stride	Output Size
Conv1	3x3/1	32x128x64
Conv2	3x3x1	32x128x64
Max Pool 3	3x3/2	32x64x32
Residual 4	3x3/1	32x64x32
Residual 5	3x3/1	32x64x32
Residual 6	3x3/2	64x32x16
Residual 8	3x3/2	128x16x8
Residual 9	3x3/1	128x16x8
Dense 10		128
Batch and L_2 normalization		128

has been pretrained on a person re-identification dataset. The appearance descriptor of each new detection is then compared to the appearance descriptors of already tracked objects by calculating the cosine distance between descriptors. Tracked objects and their appearance descriptors are also saved for 30 frames after they are lost so that DeepSORT has the ability to resume tracking identities that have been lost for a number of frames. Using appearance descriptors in this way gives DeepSORT the ability to find a previously tracked object even if it has been occluded for a number of frames. The second improvement of DeepSORT is the data association. With the estimated position of the existing tracks and the appearance descriptor, we can now associate new detection results to the existing tracks in each coming frame. A detection confidence threshold td is used to filter out all the detections with confidence lower than the threshold. New detections have to pass this threshold to be candidates of data association. The Deep SORT algorithm uses a cost matrix to represent the spatial and appearance similarities between each new detections and existing tracks. It is integrated by two distance values. The first distance is shown in equation 2.21 representing the spatial information:

$$d^{(1)}(i,j) = (d_j - y_i)^T S^{-}1_i (d_j - y_i) \tag{2.21}$$

where (y_i, S_i) are the projection of the i-th track in measurement space and d_j is the j-th new detection. This is the Mahalanobis distance [46] between j-th new detection

and estimated position of i-th. The Mahalanobis distance measures how the position of a new detection differs from the positions of already tracked objects in terms of standard deviations from the mean of the tracked objects. This metric allows DeepSORT to avoid assigning a new detection to an already existing track where the frame-to-frame motion would be unreasonable. The second distance is show in equation 2.22 representing the appearance information.

$$d^{(2)}(i,j) = min(1 - r_j^T r_k^{(i)} | r_k^{(i)} \in R_i)) \tag{2.22}$$

where r is the appearance descriptor and Ri are the appearances of the last 100 object associated with the i-th track. Each distance is accompanied with a gate function $b_{i,j}^1$and $b_{i,j}^2$ which are equal to 1 if the distance is smaller than pre-defined threshold and 0 otherwise. The integrated cost matrix is show in equation 2.23:

$$c_{i,j} = \lambda d^{(1)}(i,j) + (1-\lambda) d^{(2)}(i,j) \tag{2.23}$$

With a gate matrix $b_{i,j}$ which equals to 1 only when both spatial and appearance gate function are equal to 1 and otherwise 0, indicating whether (i, j) is a valid match for both spatial and appearance. In each new frame, the new detections are associated with existing tracks using this cost matrix and gate matrix. Track handling is processed as follow: every time a new detection is successfully associated with an existing track, the detection is added to the track and the unassociated age of the track is zero. When new detections fail to associate with existing tracks in frame f, the new detections are initialized as Tentative tracks. The original Deep SORT algorithm checks that the Tentative tracks are associated with new detections in each of the $(f+1), (f+2), ...(f+t_{tentative})$ frames. If successfully associated, the track is updated as Confirmed track. Otherwise, the Tentative track is deleted immediately. As for the existing tracks that fail to associate with new detections in each frame, their unassociated ages will increase by one. If the unassociated age exceeds the max age threshold, the track will also be deleted.

2.4 Egocentric vision datasets

2.4.1 GTEA family datatsets

Georgia Tech Egocentric Activity Datasets (GTEA) [9] contains 7 types of daily activities, each performed by 4 different subjects. The camera is mounted on a cap worn by the subject. GTEA Gaze [47] dataset is collected using Tobii eye-tracking glasses. It consists of 17 sequences, performed by 14 different subjects. GTEA Gaze+ [18] is collected by using SMI eye-tracking glasses at Georgia Tech's AwareHome. This dataset consists of 7 meal-preparation activities, performed by 26 subjects. Subjects perform the activities based on the given recipes. Activities are: American Breakfast, Pizza, Snack, Greek Salad, Pasta Salad, Turkey Sandwich and Cheese Burger. SMI glasses record a HD video of subject's activities at 24 frames per second. They also record subject's gaze at 30 fps. For each activity, ELAN is used to annotate its actions. An activity is a meal-preparation task such as making pizza, and an action is a short temporal segment such as putting sauce on the pizza crust, dicing the green peppers, washing the mushrooms. The authors have completed more than half of the annotations. The current version contains 37 videos with gaze tracking and action annotations. Audio files are also available on request.

Figure 2-6: Upshots from GTEA family datasets.

EGTEA Gaze+ [48] is the largest and most comprehensive dataset for FPV actions and gaze up to date. This dataset coms with HD videos (1920x960), audios, gaze tracking data, frame-level action annotations and pixel-level hand masks at sampled frames. EGTEA Gaze+ is a major expansion of GTEA Gaze+. Specifically, this

new datasets EGTEA Gaze + contains 29 hours (withdrawn status) cooking events from 86 independent sessions on 32 topics of 32 subjects performing 7 different meal preparation tasks. These videos have audio and gaze tracking (30Hz). The authors also provide human annotations for actions (human-object interactions) and hand masks. Action annotations include 10325 fine-grained action examples, such as "cut green peppers" or "pour condiments in a condiment container into a salad". These pixel-level hand annotations include 15,176 hand masks in 13,847 frames from the video of 200 action categories sparsely sampled from all 86 sessions of the datasets. Post-processing EGTEA Gaze+ dataset for the thesis's framework is conducted as follow: The original data in the EGTEA Gaze + dataset is processed into COCO standard in order to be used in the framework. The mask image is converted into binary mask image by thresholding and then applying a contour calculation algorithm of the hand area. At last, the binary mask is converted to the RLE (run length encoding) standard which is usable data in the framework. In the scope of this thesis, the 4 datasets GTEA, GTEA Gaze, GTEA Gaze + and EGTEA Gaze + is regulated into GTEA family datasets.

Figure 2-7: Hand masks after post-processing EGTEA Gaze+.

2.4.2 EgoHands dataset

The EgoHand dataset, which is presented in [6] is a large dataset for hands in complex egocentric interactions. In order to create as realistic data sets as possible while still being able to carry out some experimental control, the author collected data from different pairs of four pairs of participants, who faced each other when faced with different activities. The authors chose four activities that encourage interaction and gesture movement: (1) playing cards; (2) playing chess, in order to improve efficiency,

they encourage participants to focus on speed rather than strategy; (3) solve 24 or 48 pieces of jigsaw puzzles; (4) play Jenga, which involves removing pieces from the 3d puzzle until it collapses. The context is also varied by collecting videos in 3 different locations: a table in a meeting room, a patio table in an outdoor courtyard, and a coffee table at home. The dataset is recorded for several days, and there was no restriction on the clothes of the participants, so there were many types, for example, short-sleeved and long-sleeved shirts, etc. Data is systematically collected from four actors who performed all four activities in all three locations, while randomly assigning participants to interact with each other, resulting in $4 \times 4 \times 3 = 48$ unique video combinations. Each participant wore Google glasses, which recorded a 720×1280 video at a frequency of 30 Hz. In post-processing, the videos are synchronized in pairs with each other and each video pair is cut into exactly 90 seconds (2,700 frames). Ground truth is manually annotated from a random subset of 100 frames in each video (approximately one frame per second) using pixel-level manual masks. Each hand pixel has one of the following four tags: the left or right hand of the camera wearer ("my left hand" or "my right hand"), or the left or right hand of the social partner ("your left hand" or "your right hand "). The ground truth was created by six students who were told to mark any hand-shaped pixels they could see, including small hand areas caused by objects being occluded or truncated at frame boundaries. Importantly, compared with EGTEA Gaze+, this dataset defines "hand" as stopping on the wrist, and this job also includes the arm extending toward the participant's sleeve. In total, this dataset contains approximately 130,000 frames of video, of which 4,800 frames have a pixel-level ground truth consisting of 15,053 hands. The partner's hands appear in the vast majority of frames (left and right 95.2% and 94.0%, respectively), while the wearer's hand appears less (left and right 53.3% and 71.1%, respectively). This may be because one's own hand appears more often outside the camera's field of view, but the right hand appears more frequently because people tend to align their attention with the dominant hand (and all participants are right-handed). Figure 2-8 shows a sample frame with basic facts. This dataset is released on the public web with ground truth accessible through a Matlab API we provided

by the authors. Post-processing EgoHands for the thesis's framework is conducted as

Figure 2-8: Visualizations of EgoHand dataset [6].

follow: The original annotations in EgoHands datasets consists of 4 categories: my left, my right, your left, your right. In the scope of this thesis, I just focus the hands appearing in egocentric video, therefore I convert all 4 categories into 1 category, "hand". From the ".mat" format data, I dumped into ".json" format and feed to the framework.

2.4.3 Micand32 dataset

As introduced in subsection 1.2.2, the NAFOSTED's project is carried out with the aim of using advanced deep learning techniques to evaluate the healing and surgical research of the patient with a sensor through detection segmentation, tracking, gesture recognition, and correlation of the patient's hand with objects. The members in the project prepare the script and perform data collection at Hanoi Medical Hospital. The experiment was carried out by 10 volunteers, patients aged 20 to 60 years, 5 men and 5 women under the supervision, explanation and help of doctors and instructors. The implementation site consists of 3 areas: the desk area, the sink area and the closet area. Patients will wear 5 sensors including: 1 camera at the head, 1 camera in the shoulder, 1 sensor of kinetics (sensors that will collect and store information about the carrier's position, acceleration) in the left hand, 1 kinematic sensor in the right hand, 1 kinetic sensor in the leg. The patient performs 21 agreed and pre-defined actions, such as taking stairs, practicing hands with objects, brushing hair, opening

Figure 2-9: Randomly selected actions 5 6 7 8, from left to right respectively.

cabinets, etc. The average number of executions per action is 3. The recorded videos have very high resolution compared to the previous related datasets, is 1920x1440, frequency 30fps. Average time to perform an action is 10 seconds. Thus, in terms of images, the data set consists of 10 people x 2 cameras x 21 actions x 3 times = 1260 videos, so there will be 1260 videos x 10 seconds x 30 fps = 378000 frames.

The Micand32 dataset is a subset of the NAFOSTED's project dataset. In the context of the problem of detecting, partitioning and tracking human hands, this thesis is concerned with only 4/21 types of actions most relevant to the hand: (5) practice with ball, (6) practice with water bottles, (7) practice with wooden blocks, (8) practice with cylinders. Figure 2-9 visualizes random samples of 4 action types extracted from MICAND32. Micand32 consists of 32 sequences of 1920x1440 resolution, is divided into 2 parts: Micand32Standard includes 26 sequences and Micand32Enhanced includes 6 sequences. In particular, with Micand32Standard, each sequence contains from 100 to 300 frames, in these frames are mainly the journey of one hand doing a whole activity, this corresponds to the type of short-term tracking. This episode has lengths of each sequences that are equivalent to the MOT Challenge standards. With Micand32Enhanced, each sequence contains from 500 to 1700 frames. Most of the frames contain more than 2 hands due to the intervention of the instructor's hand while the patient is performing the action. At the same time, the patient performed multiple repetitions in each sequence, corresponding to a long-term object tracking type. This episode is very close to reality, it is quite challenging and highly applicable. The ground-truth detection and partition section of this data set was labeled by 8 students using the VIA tool [49]. VGG Image Annotator is a simple and standalone manual annotation software for images, audio and video. VIA runs in a web browser

and does not require any installation or settings. The complete VIA software fits a single independent HTML page, which is less than 400 KB in size and can be run as an offline application in most modern web browsers. Table 2.4.3, label statistical visualization and correlogram 2-10 provides detailed statistics on Micand32: 1 category, 11k images, 18k instances, 2 instances/images. The hands highly present at the edges of frame, the size of the hands are various.

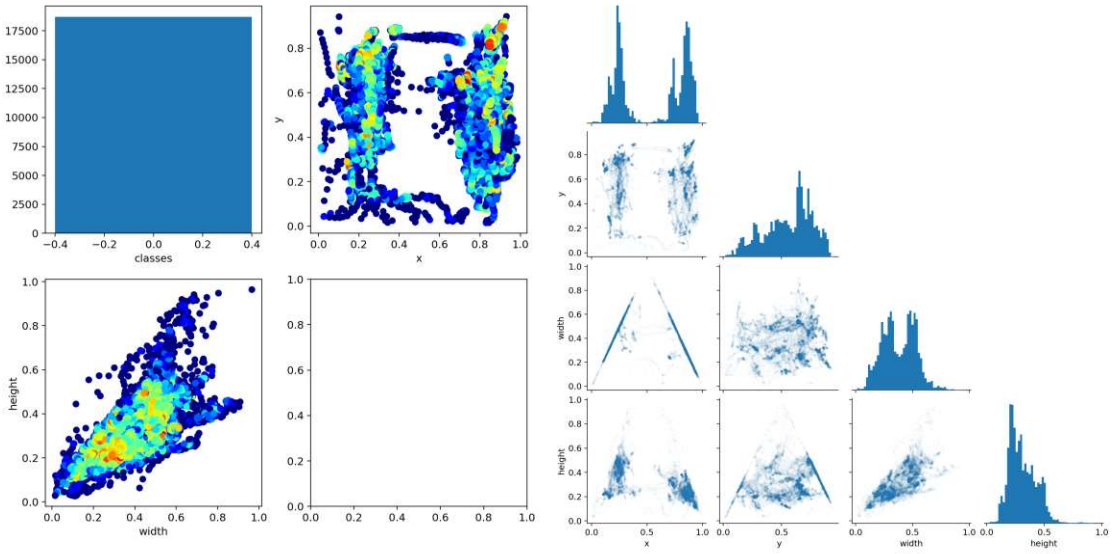

Figure 2-10: Micand32 dataset. Left: labels statistical visualization. Right: labels correlogram.

In order to evaluate the effectiveness of the tracking algorithm, there is currently no standard popular tool to support the id tag of the object in the video frame. To the best of my knowledge, currently there is no egocentric dataset that involves detailed standard for hand tracking, according to [50]. This thesis refers to the MOT Challenge tracking data standard and the annotator has done the tracking tagging through a self-developed tool called EHTA. Egocentric Hand Tracking Annotator was especially designed to model semi-automatic annotation pipelines to speed up the annotation process. Such a semi-automatic can be achieved by using AI generated annotation proposals that are presented to an annotator inside the annotation tool. Also, EHTA integrates a part of the VIA labeling tool [49]. Figure 2-11 performs trajectory of the patient's hands opening water bottle, in which the groundtruth is visualized by

Table 2.4: Detailed enumeration of the Micand32 dataset.

No	Name	Action Type	Numbers of frames	Hand/ Frame
1	GH010354_7_13210_16534	7	120	1
2	GH010354_7_16605_17602_1	7	130	1
3	GH010354_7_16605_17602_2	7	120	1
4	GH010354_8_26079_29031	8	191	1
5	GH010358_7_2490_3390_1	7	160	1
6	GH010358_7_2490_3390_2	7	160	1
7	GH010358_7_352_2464_1	7	110	1
8	GH010358_7_352_2464_2	7	135	1
9	GH010374_6_4944_6241_1	6	140	1
10	GH010374_6_4944_6241_1	6	100	1
11	GH010382_5_5725_7093_1	5	100	1
12	GH010382_5_5725_7093_2	5	130	1
13	GH010382_5_955_4771_1	5	115	1
14	GH010382_5_955_4771_2	5	115	1
15	GH010382_6_18190_20215_1	6	151	1
16	GH010382_6_18190_20215_2	6	150	1
17	GH010382_6_20592_21726_1	6	90	1
18	GH010382_6_20592_21726_2	6	125	1
19	GH010382_8_16207_17479_1	8	100	1
20	GH010382_8_16207_17479_2	8	100	1
21	GH010383_5_462_968_1	5	100	1
22	GH010383_5_462_968_2	5	120	1
23	GH010383_8_3221_3956_1	8	100	1
24	GH010383_8_3221_3956_1	8	100	1
25	GH010383_8_4544_5192_1	8	100	1
26	GH010383_8_4544_5192_2	8	100	1
	MICAND32Standard	5, 6, 7, 8	**3162**	1
27	GH010354_5_17718_19366	5	1684	1
28	GH010373_5_1284_2724	5	1440	5
29	GH010358_6_10208_11900	6	1594	4
30	GH010373_6_3150_4744	6	1692	6
31	GH010358_7_2490_3390	7	900	4
32	GH010358_8_8000_8547	8	547	3
	MICAND32Enhanced	5, 6, 7, 8	**7857**	4
	MICAND32	5, 6, 7, 8	**11019**	2

EHTA.

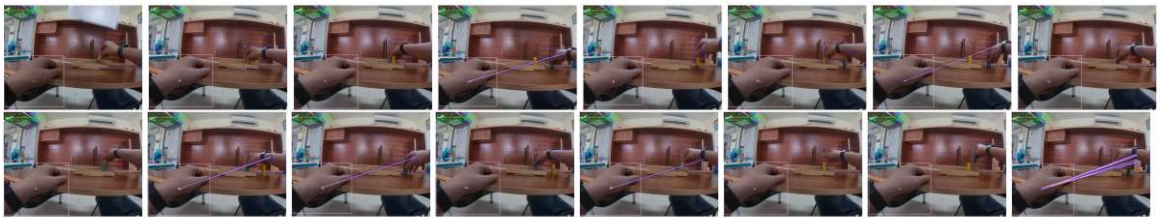

Figure 2-11: Visualization of groundtruth tracklets of patient's hands practicing with cylinders. Frames extracted from GH010358_8_8000_8547, ordered from left to right, up to down: 1, 31, 61, 91, 121, 151, 181, 211, 241, 271, 301, 311, 361, 391, 421, 451.

Figure 2-12 shows the working flowchart of EHTA. Initially, annotator selected 1 model from model zoo that was pre-trained on COCO, GTEA family and EgoHands datasets, for example, this thesis used FasterRCNNR50FPN3x. Input is a set of frames in 1 sequence. For each frame, reference the model on the frame to predict the position of the hands and get the detection of bounding box and confidence score. Next, use the VIA tool to display the prediction in frames, observe and make comments to write code. From observation and experience with the dataset, writing code to automatically editing the labels for the hand's position and identification. Then shows the automatically assigned label portion. If the label is not correct, manually correct the bounding box position, add or remove the bounding box, or correct the incorrect ID. Finally, save ground-truth according to MOT Challenge standards for training, testing and evaluation in txt format. At the same time, save ground-truth video for debugging. Annotation time for 1 image with manually and semi-automatically by EHTA is respectively 5 seconds and 1 second. By using EHTA, the subjectivity of human perception and loss of time are reduced as well as consistency and coordination among different individuals' annotations are accomplished.

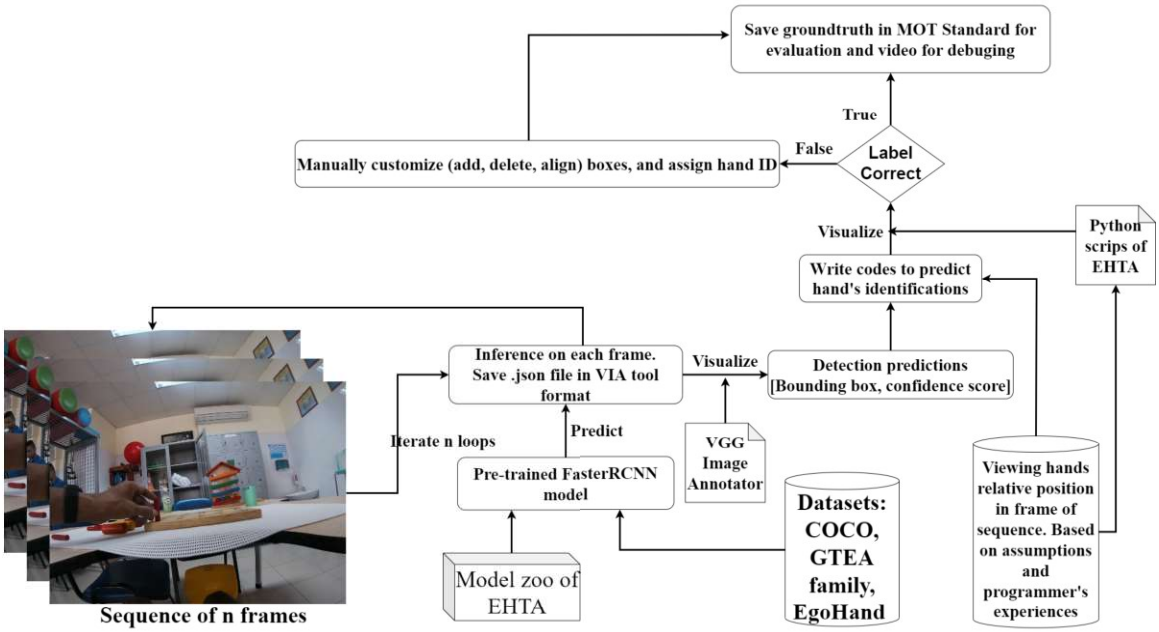

Figure 2-12: The workflow of EHTA.

Chapter 3

Proposed Framework

3.1 Proposed framework: tracking by detection

The framework proposed in this thesis is illustrated in Fig. 3-1. It consists of 4 stages as following:

- Data preparation stage: The framework starts with the data preparation stage. GTEA and EgoHands datasets are pre-processed in order to keep only suitable and related ground-truth samples. The annotators use the semi-automatic EHTA tool to create a new dataset Micand32 (Micand32S and Micand32E) as described in chapter 2, sub-section 2.4.3.

- Training stage: Three datasets Micand32S, GTEA family and EgoHands will be used to train hand detection and segmentation models from egocentric vision models including family of RCNN models and family of YOLO models. At the same time, a deep appearance descriptor is trained on distinctive hand groups ground-truth in Micand32S. While the original DeepSORT's descriptor is made within people tracking context, the descriptor trained in my thesis is made well suited for deep metric learning in egocentric hand context. Detail training techniques are reported in section 3.2.

- Inference stage: Following the training stage is the inference stage in which all videos in Micand32 datasets will be evaluated. For each sequence of n frames, each frame in turn is fed respectively into the detection and segmentation phase to get hand's bounding boxes, masks and their confidence scores. This phase requires the user to select detection or segmentation algorithms from pre-trained D2D model zoo. Detected objects in each frame then input into the tracking

phase. This phase also requires the user to choose a tracking algorithm (SORT or DeepSORT). The tracking phase generates trajectories of egocentric hands along the video. Section 3.3 explains in detail the inference procedure.

- Evaluation stage: Finally, detection bounding boxes and segmentation masks are evaluated by comparing with Micand32's ground-truth. This result is measured according to COCO evaluation protocol. The hand's tracklets are evaluated and reported by MOT Challenge protocol. Detail of evaluation criteria is described in section 4.1.

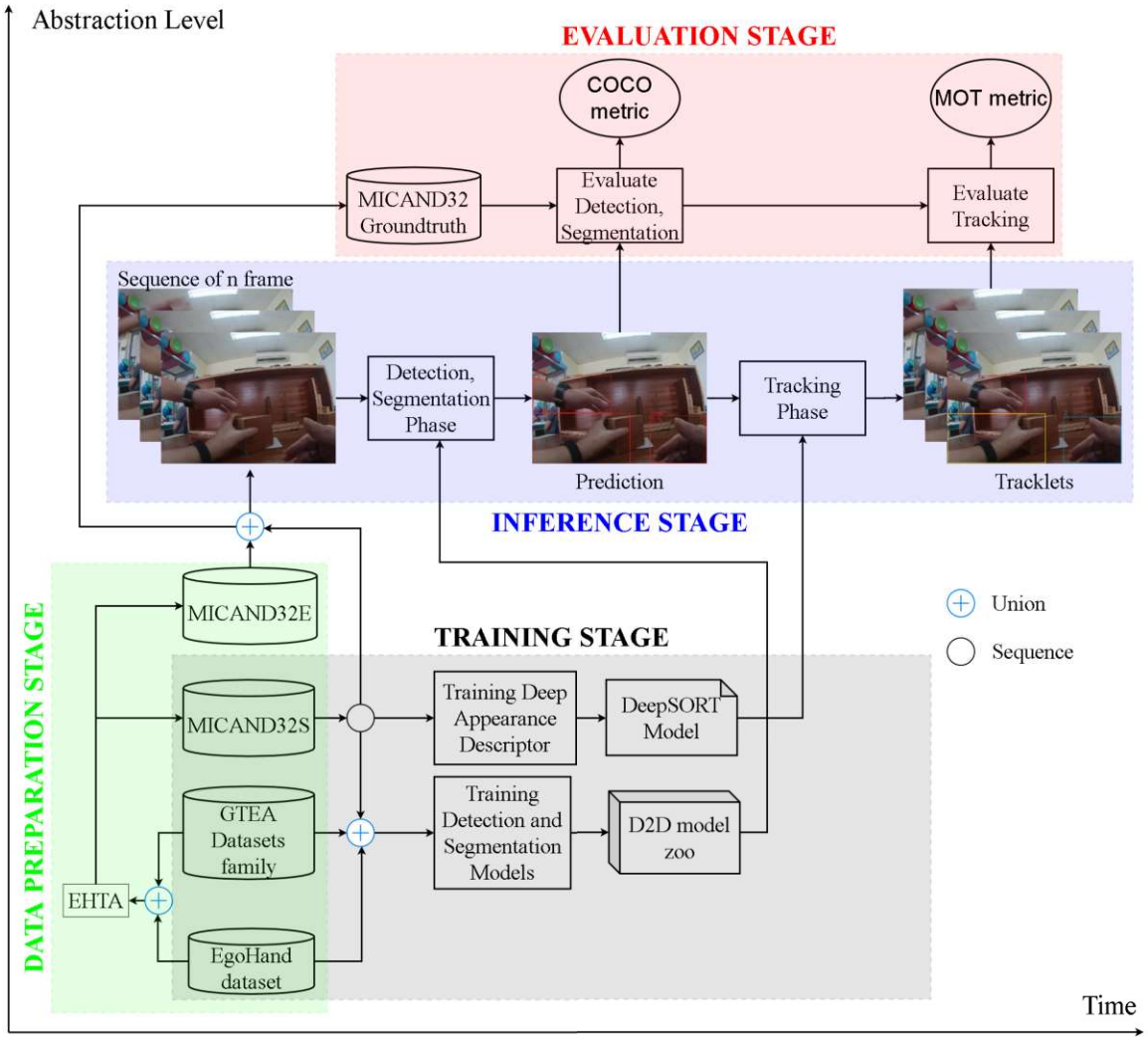

Figure 3-1: Overview of the proposed framework: D2D. The x-axis represents the time flow of 4 stages. The y-axis shows the degree of abstraction levels of stages.

3.2 Training stage

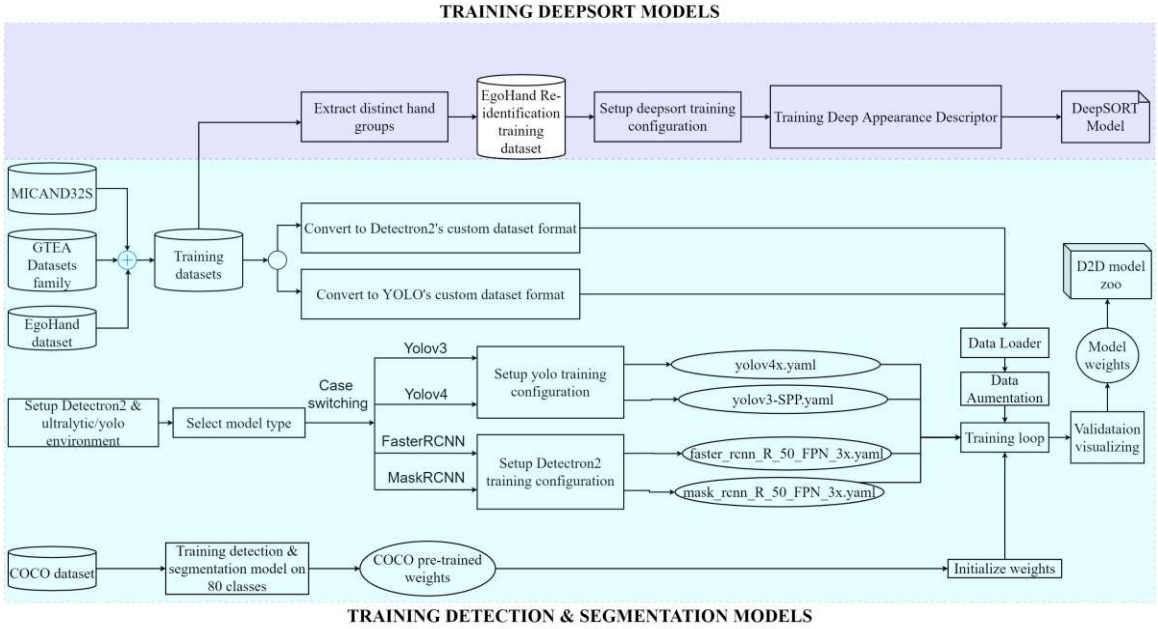

Figure 3-2: Workflow of the training stage.

The training stage, as shown in Fig. 3-2 consists of parts:

- Training detection and segmentation models
- Training DeepSORT model

Input of both parts is combination of 3 datasets: Miand32S, GTEA family and EgoHands dataset. The first part will generate a D2D model zoo which consists of 4 types of models, 2 from yolo family and 2 from the RCNN family. The second part trains a CNN for deep appearance descriptor using cosine metric learning [51] in DeepSORT. Detailed implementation is explained as follows.

3.2.1 Training detection and segmentation models

Initially I set up a programming environment which consists of 2 frameworks: Detectron2 [52] for RCNN model family and Ultralytics [53] for YOLO model family.

Detectron2 is a complete rewritten version of Detectron that originates from maskrcnn-benchmark. The platform is now implemented and powered by PyTorch

deep learning framework. Through a new modular design, Detectron2 is flexible and scalable, and can provide fast training on a single or multiple GPU servers. Detectron2 includes high-quality implementations of state-of-the-art object detection algorithms, including DensePose, panoptic feature pyramid networks, and numerous variants of the pioneering Mask R-CNN model family also developed by FAIR. Its scalable design makes it easy to implement cutting-edge research projects without having to spend the entire code base. The requirements for installing Detectron2 includes: Linux with Python 3.6+, Pytorch 1.4+ and torchvision that matches the PyTorch installation, OpenCV needed by demo and visualization. In this thesis, I build Detectron2 from source. The compilers gcc and g++ version 5+ are required, and ninja is recommended for faster build. After having these prerequisites, I clone the Detectron2 repositories and install it via pip from the local clone.

The Ultralytics is in the repository [53]. The requirements for installing Ultralytics are Python 3.8 or later with all dependencies including Cython, matplotlib, numpy, OpenCV, pillow, PyYAML, scipy, tensorboard, tqdm, pycocotools, scikit-learn, seaborn, coremltools and onnx. I build Ultralytics from source cloned from their github repository and install via pip. After installing programming environments, I have selected the model type to train. There are 4 main models integrated in this thesis's framework, the RCNN family consists of FasterRCNN and MaskR-CNN, while the YOLO family consists of Yolov3 and Yolov4. Depending on model selection, D2D switches to appropriate branches to set up training configurations. For RCNN family, Detectron2 supports different backbone network architectures such as ResNET {50, 101, 152 }, FPN, VGG16, etc.

In this thesis, I choose the standard configs, FasterRCNN_R_50_FPN_3x and MaskRCNN_R_50_FPN_3x with backbone Resnet 50 layers, feature pyramid network 3x architecture. The config file is saved in a ".yaml" format file. Table A represents a typical configuration for FasterRCNN_R_50_FPN_3x. For the YOLO family, Ultralytics supports different model checkpoints type, and in this thesis, I choose the standard YOLOv3-SPP and YOLOv4x. Training configuration is also saved in a ".yaml" format file. Table A represents a typical configuration for Yolov4x.

It should be noted that in this thesis the number of classes for training is 1 class "hand". The initial weights are taken from the pre-trained model on COCO dataset with 80 classes. In my thesis I merge GTEA datasets family, EgoHands dataset and Micand32S datasets into 1 training dataset. To let detectron2 know how to obtain a custom dataset, I implement a function that returns the items in training datasets. The standard representation for a dataset is a combination of multiple dictionaries, each dictionary containing information about one image. The dictionary may have the following fields as in A.

Ultralytics also requires exporting custom dataset labels to YOLO format, with one "*.txt" file per image. The "*.txt" file specifications are: (1) one row per object; (2) each row is "class x_center y_center width height" format; (3) box coordinates must be in normalized xywh format (from 0 - 1). If boxes are in pixels, divide x_center and width by image width, and y_center and height by image height; (4) class numbers are zero-indexed. Each image's label file should be locatable by simply replacing "/images/*.jpg" with "/labels/*.txt" in its pathname.

Data loader is the component that provides data to models. A dataloader usually takes raw information from datasets, and process them into a format needed by the model in training loop. Augmentation is an important part of training. Detectron2's data augmentation system aims at addressing the following goals: (1) allow augmenting multiple data types together (e.g., images together with their bounding boxes and masks); (2) allow applying a sequence of statically-declared augmentation. For Ultralytics, a Mosaic Dataloader is used for training. The data augmentations used are horizontal and vertical flip, resize, Gauss noise, random brightness contract, crop, median blur, color distortion, gamma, converting color space; some of them are illustrated in Figure 3-3. With the benefits of transfer learning techniques discussed in [39], in this thesis, I start from the pre-trained models on 80 classes in COCO datasets and fine-tune it for the particular "hand" class. Training loop runs with defined configurations and augmented data. Table 4.3 reports that training time for Yolov3, Yolov4, FasterRCNN and MaskRCNN in this thesis are respectively 36, 40, 46 and 50 hours. The visualization of training steps is available via Tensorboard during

Figure 3-3: Pictorial of data augmentations in training batch.

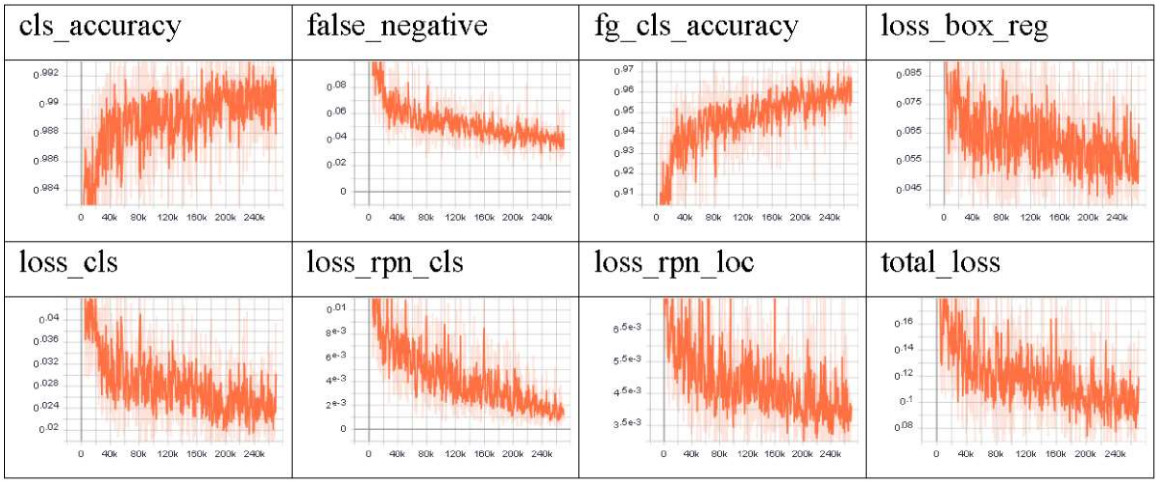

Figure 3-4: FasterRCNN_R_50_FPN_3x losses visualization during training time.

training time as in 3-4. After training, the framework generates the D2D model zoo which contains 4 model with corresponding weights save in ".h" data format and will be used in the inference stage.

3.2.2 Training deep appearance descriptor for DeepSORT

The original DeepSORT is used for person tracking task – a familiar issue in which a given query image is used to look at a huge gallery of images that have been gathered at distinct moments, lighting environments, actions performing that may contain the same individual. To customize it on an egocentric hand tracking problem, it firstly requires an egocentric hand re-identification dataset to train the appropriate appearance descriptor.

To the best of my knowledge, there are very few public egocentric hand re-identification datasets available [50]. From the analysis of 3 datasets described in chapter 2, section 2.4, I build a new egocentric hand re-identification dataset from the identity information annotations. Images of hands are extracted and cropped from these 3 datasets using the bounding box information and also resized to the same size (100, 100) as recommended in [51]. As a result, there are in total 40 identities (26 subject from GTEA + 4 subject from EgoHands + 10 subject from Micand32S = 40 subjects) and 29324 images in this egocentric hand re-identification dataset. The sample of the images used in this dataset is shown in 3-5. Having obtained the ego-

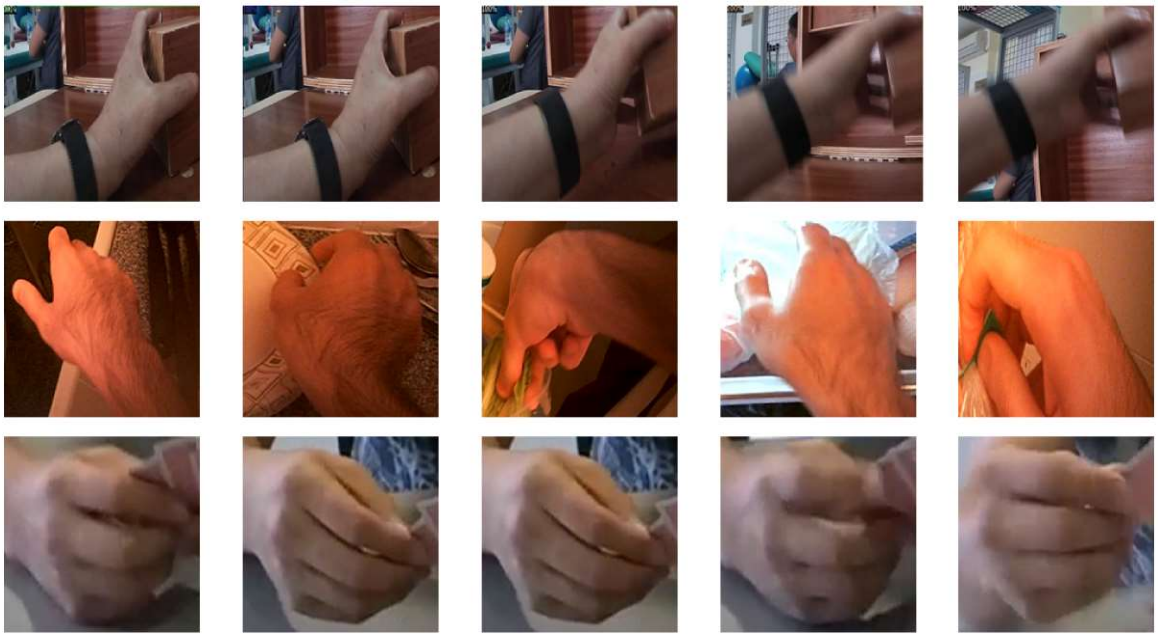

Figure 3-5: Images from the self-generated egocentric hand re-identification dataset. Images in the same row have the same identity.

centric hand re-identification dataset, I train CNN to get the appearance descriptor. The training and testing datasets were splitted into 80% and 20% respectively and both were structured by placing all images of one identity into a specific numbered folder for both train and test sets. I set up the training configuration as in Fig. 3-6. The programming environment in which the training process performed is python 3 with PyTorch framework, numpy, scipy, sklearn, pillow, vizer and edict package. The training chart obtained is shown in 3-6. It tooks 40 hours to train this CNN.

Field	Value
Batch size	64
No. of epochs	40
Learning rate	0.1
Momentum	0.9
Weight decay	5e-4
Learning decay	0.1 after every 10 epochs

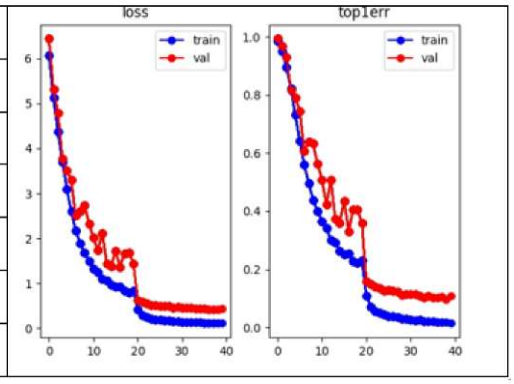

Figure 3-6: Left: training configuration of DeepSORT's appearance descriptor. Right: curve of total loss and top1-error during training loop.

3.3 Inference stage

In the inference stage 3-7, each of 32 videos from the Micand32 dataset will be tested in turn. Initially each frame is fed into the detection phase in chronological order. Next, users select a detection or segmentation model from the D2D model zoo which was already trained in the training stage. In order to analyze the impact of detection phase on tracking phase, the detection ground-truth of Micand32 is also available for users to select in this step. As a consequence, 6 options are available: Yolov3_spp, Yolov4x, FasterRCNNR50FPN3x, MaskRCNNR50FPN3x, MaskRCNNR50FPN3x with region based and detection ground-truth.

Depending on the chosen model type, the D2D framework will set up inference configurations. For example, the default confidence threshold in this thesis is 50%. Appropriate pre-trained weights will be loaded from the D2D model zoo in ".h" format and the inference process will begin. For detection models which include yolo model family and FasterRCNNR50FPN3x, each prediction on image is represented in the following format: $pred^i = [x_0^i, y_0^i, x_1^i, y_1^i, c^i]$ where $pred^i$ is the i-th prediction with $(x_0^i, y_0^i), (x_1^i, y_1^i)$, is respectively the top-left and bottom-right vertex coordinate of the i-th bounding box and ci is the current confident score of this bounding box. For MaskRCNNR50FPN3x, each prediction will consist one more field: $pred^i = [x_0^i, y_0^i, x_1^i, y_1^i, pol^i, c^i]$ where the pol^i is the polygon of the mask in RLE for-

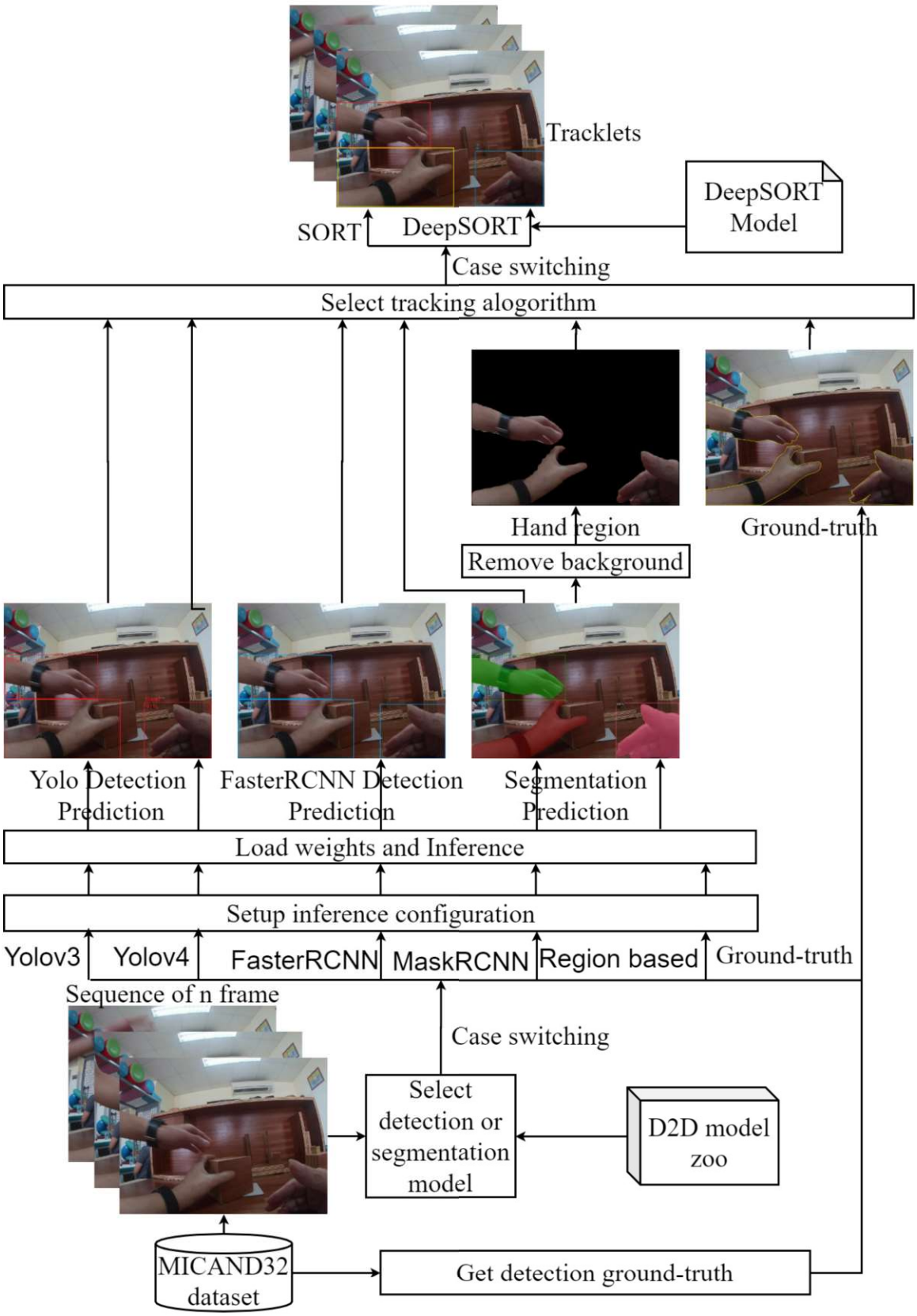

Figure 3-7: Workflow of inference stage.

mat. For MaskRCNNR50FPN3x with region-based option, after obtaining the mask predictions, the framework will execute background subtraction in order to keep only the hand region. This is intended to make the deep appearance descriptor in DeepSORT focus more on the hand region.

In the following step, users will select the tracking algorithm: SORT or DeepSORT. For SORT, detection will be processed in Kalman filter and then associated by the Hungarian algorithm as explained in chapter 2, section 2.3.2. For DeepSORT, the deep appearance descriptor will be loaded from the pre-trained CNN in the training stage. Tracklets for the whole sequences will be compared with the tracking ground-truth in the evaluation stage in order to analyze the effectiveness of the algorithms.

3.4 Evaluation stage

Detail evaluation criteria and measurement metrics is reported in chapter 4, section 4.1. The evaluation stage includes 2 parts: (1) evaluate detection and segmentation results; (2) evaluate tracking results. For detection and segmentation result, this thesis adopts the evaluation API from Detectron2 repository and the py-cocotools package. For tracking results, performance is measured according to the framework presented in [17]. The authors provide evaluation scripts for the official development kit of MOT Challenge. MOT16 was chosen since it is a compilation of other many metrics developed in an attempt to standardize evaluation of multiple object tracking. This devkit requires Python 3.7+, Matlab R2020a, matlab python engine, pandas and pytz. Tracking results will be saved in simple comma-separated value (CSV) files. Each line represents one object instance and contains 9 values as shown in Table 3.1. An example of such an annotation file is:

- 1, 1, 1672, 763, 248, 245, 1, 1 ,1
- 1, 2, 1253, 426, 156, 200, 1, 1 ,1
- 2, 1, 1668, 774, 252, 234, 1, 1, 1

Table 3.1: Data format for both the inference result and ground-truth annotation of tracking.

Position	Name	Description
1	Frame number	Indicate at which frame the object is present
2	Identity number	Each hand trajectory is identified by a unique ID
3	Bounding box left	Coordinate of the top-left corner of the pedestrian bounding box
4	Bounding box top	Coordinate of the top-left corner of the pedestrian bounding box
5	Bounding box width	Width in pixels of the hand bounding box
6	Bounding box height	Height in pixels of the hand bounding box
7	Confidence score	It acts as a flag whether the entry is to be considered (1) or ignored (0)
8	Class	Indicates the type of object annotated. In this thesis, always (1)
9	Visibility	Visibility ratio, a number between 0 and 1 that says how much of the hand is visible. Can be due to occlusion and due to image border cropping.

In this case, there are 2 hand in the first frame of the sequence, with identity tags 1, 2 and 1 hand in the second frame with identity tags 1.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Experiments

4.1 Evaluation criteria

4.1.1 Object detection evaluation metrics

Average Precision is a standard and popular metric in measuring the performance of the object detection and segmentation algorithms. The Precision (Prcn) and Recall (Rcll) are defined as follow:

$$Prcn = \frac{TP}{TP + FP} \tag{4.1}$$

$$Rcll = \frac{TP}{TP + FN} \tag{4.2}$$

where, TP, FP, and FN are number of True Positive, False Positive, and False Negative, respectively. A predicted bounding box is considered a TP if its intersection over union (IoU), or Jaccard Index, with a ground truth box is larger than 0.5. Equation 4.3 shows how the IoU between a predicted box P and a ground truth box G is calculated.

$$IoU(P, G) = \frac{|P \cap G|}{|P \cup G|} \tag{4.3}$$

F1 score is a metric that combines recall and precision into a single score by calculating the harmonic mean of precision and recall:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

Latest research papers tend to give results in the COCO dataset format. In COCO mAP, a 101-point interpolated AP definition is used in the calculation. For COCO, AP is the average over multiple IoU. AP@[. 5: .95] corresponds to the average AP for

IoU from 0.5 to 0.95 with a step size of 0.05. The following are some other metrics collected for the COCO dataset:

4.1.2 Object tracking evaluation metrics

Performance is measured according to the framework presented in MOT16 [17] and in the same manner as performance is measure in the MOTChallange. The authors of [17] provide publicly available code for evaluation, this code is used to calculate the different performance metrics. MOT16 was chosen since it is a compilation of other many metrics developed in an attempt to standardize evaluation of multiple object tracking. MOT16 contains a wide array of metrics for evaluation of multiple object tracking, some which are quite similar to each other. Hence, metrics that were considered to similar to other have not been included in this thesis.

- Identification Recall (IDR) and Identification Precision (IDP): IDR and IDP are similar to the metrics Recall and Precision for object detection. The metrics will however differ since objects are considered tracked only if they can be assigned an identity, which will not be the case for all detected objects. Another difference is that inconsistencies in identity assignments will lower the IDTP score. For each ground truth identity, the predicted identity most similar to it is found. Any other identity assigned to the ground truth identity is then considered a mismatch (IDFP) and will be counted as a false positive instead of a true positive.

$$IDR = \frac{IDTP}{IDTP + IDFP} \tag{4.5}$$

$$IDP = \frac{IDTP}{IDTP + IDFN} \tag{4.6}$$

  where, IDTP is the sum of TP in detection and the number of correctly labeled objects in the tracking; IDFP/IDFN is the sum of FP/FN in detection and the number of correctly predicted objects for positive class in detection but incorrectly labeled in tracking.

- IDF1-score(IDF1): Similar to F1 score for object detection, IDF1 combines both

IDR and IDP into a single score to facilitate comparisons of different trackers. The higher IDF1 is, the better tracker is.

$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{4.7}$$

- Mostly Tracked (MT): The number of ground truth identities that are tracked for 80% or more of their existence.

- Partly Tracked (PT): The number of ground truth identities that are tracked between 20% and 80% of their existence.

- Mostly Lost (ML): The number of ground truth identities that are tracked for less than 20% of their existence.

- Identity Switches (IDs): The number of identity switches. An identity switch is counted every time an already tracked ground truth identity is assigned a new tracking identity.

- Track Fragmentations (FM): The number of track fragmentations. A track fragmentation is counted every time a tracked ground truth identity is lost and then found again in a later frame.

- Multiple Object Tracking Accuracy (MOTA): MOTA combines false negatives, false positives and identity switches into a single score in order to express overall performance with a single value. This is the most important metric for object tracking evaluation and it is defined as:

$$MOTA = 1 - \frac{\sum_t (IDFN_t + IDFP_t + IDS_t)}{\sum_t GT_t} \tag{4.8}$$

where, t is the index of frame, GT is the number of observed objects in the real-world. It is worth to note that MOTA would be a negative value if there are many errors in the tracking process and the number of these errors is larger than that of observed objects.

- Multiple Object Tracking Precision(MOTP): MOTP measures how well correctly predicted bounding boxes TP fit their respective ground truth boxes GT. This is done by calculating the average overlap between true positives and their corresponding ground truth object.

$$MOTP = \frac{\sum_t d_{t,i}}{\sum_t c_t} \tag{4.9}$$

where, c_t denotes the number of matches found in the frame t and $d_{t,i}$ is the sum of distances between all true positive and their corresponding ground truth i. This metric indicate the ability of the tracking in estimating precise object positions.

4.2 Experimental results

The following sub-sections will present result for different object detection and tracking algorithms tested in this thesis. All experiments are conducted on 32 videos of Micand32 datasets. The average inference speed is measured on a Workstation super-micro with Intel® Xeon®, CPU E5-2620 v2@2.1GHz, 6 cores, 12 threads, RAM 12GB, GPU GTX 1080.

4.2.1 Egocentric hand detection and segmentation result

This subsection presents results for the different object detection algorithms consider in this thesis. The algorithms to be evaluated are: Yolov3_spp, Yolov4x, FasterRCNN_R_50_FPN_3x, MaskRCNN_R_50_FPN_3x. Detection evaluation is show in Table 4.2.1, Table 4.2 and Table 4.3.

4.2.2 Egocentric hand tracking result

Tracking results for SORT and DeepSORT with different object detection strategy are presented in this section. As described in section 3.4, the tests are performed

Table 4.1: Object detection and segmentation Average Precision following the COCO standard.

Algorithm	AP	AP50	AP75	AP^{small}	AP^{medium}	AP^{large}
Yolov3	89.2	92.4	92.1	1.1	66.4	54.1
Yolov4x	93.1	95.6	94.6	3.2	72.5	42.9
FasterRCNN	**96.2**	97.9	**97.9**	0.9	**75.8**	6.3
MaskRCNN	92.1	**98.9**	97.9	0.0	32.4	**92.2**

Table 4.2: Object detection and segmentation Average Recall following the COCO standard.

Algorithm	$AR^{max=1}$	$AR^{max} = 10$	$AR^{max} = 100$	AR^{small}	AR^{medium}	AR^{large}
Yolov3	6.5	53.6	76.4	3.2	32.5	75.9
Yolov4x	8.7	65.8	89.7	7.1	40.1	82.7
FasterRCNN	**9.6**	76.8	**97.6**	10.0	**77.8**	**97.6**
MaskRCNN	9.2	**73.9**	94.6	0.0	50.8	94.7

Table 4.3: Object detection and segmentation average training and inference time, speed and machine requirements.

Algorithm	Average time		Speed (Hz)	Memory requirement (MB)	
	Training (hour)	Inference (millisecond)		RAM	GPU
Yolov3	**36**	45	22	**1989**	2260
Yolov4x	40	**38**	**25**	2152	**2076**
FasterRCNN	46	84	12	2351	3297
MaskRCNN	50	**215**	**5**	**2461**	**3674**

in a tracking by detection system where tracking and detection algorithms are completely separated. Therefore, different results for combinations of tracking and detection algorithms are reported. Further, ground-truth detection are also test with SORT and DeepSORT. This is done so as to give an insight how much of the error is due to the object detection algorithm and how much of it is due to the tracking algorithm. Tracking performance with ground-truth detection should only have errors introduced by the tracking algorithms and can thereby give an upper limit for how much better the tracking by detection system can become by changing object detection algorithm. Thus, this frameworks consists of 6 detction and segmentation algorithms, they are: Yolov3_SPP, Yolov4x, Faster_RCNN_R_50_FPN_3x, Mask_RCNN_R50_FPN_3x, Mask_RCNN_R50_FPN_3x with region based and detection ground-truth. Their combination with 2 tracking algorithms, SORT and DeepSORT, we obtain a total of 12 options for the pipeline. However, the combination

Table 4.4: Notation of the 11 approaches mentioned and used in the experiments.

No	Detector	Tracker	Notation
1	Yolov3	SORT	Y3S
2	Yolov4		Y4S
3	FasterRCNN		FS
4	MaskRCNN		MS
5	Groundtruth		GS
6	Yolov3	DeepSORT	Y3DS
7	Yolov4		Y4DS
8	FasterRCNN		FS
9	MaskRCNN		MS
10	MaskRCNN+Region based		RDS
11	Groundtruth		GDS

of MaskRCNN with region based and SORT scheme is identical to the combination of Mask and SORT scheme because in theory, the SORT algorithm does not use visual information fields. To conclude, we have 11 approaches conventionally defined as in the Table 4.2.2 for brief presentation.

In this thesis, I conduct the experiments to evaluate the performance of these approaches on the custom dataset Micand32 which contains 2 sub-datasets: Micand32S and Micand32E. The properties of this dataset is reported in chapter 2, sub-section 2.4.3.

Short-term tracking on a standard dataset: Micand32S

The Micand32S sub-datasets contains 26 sequences containing in total 3162 frames as per the MOT16 Challenge rules and recommendations, each containing from 100 to 300 images. These recordings are chosen so every video has an excursion span of precisely one patient's activity, for instance opening the cap of the water bottle, getting the circle, playing with the ball, turning left of right, etc. Simultaneously, every video generally contains just 1 or 2 active hands. This sub-dataset can be characterized into short-term tracking datasets. The desire set for the framework proposed in this theory is to recognize, segment, and track this entire fundamental excursion of the hand. Since the quantity of recordings is up to 26 recordings, in this part I will list the overall evaluation for each approach on the whole 26 recordings, which is a weighted average of the quantity of frames per video. Besides, I likewise

Table 4.5: Short-term tracking **overall** result on Micand32S following the MOT16 evaluation protocol.

Method	IDF1 (%)↑	IDP (%)↑	IDR (%)↑	Rcll ↑	Prcn ↑	GT ↑	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	**MOTA** (%)↑	MOTP ↓
Y3S	97.0	99.6	94.6	94.8	99.8	35	33	**1**	1	6	207	1	1	94.6	0.087
Y4S	98.6	99.4	97.7	97.8	99.4	35	**34**	**1**	**0**	23	89	**0**	5	97.2	0.108
FS	97.1	99.7	94.6	94.8	99.9	35	33	**1**	1	4	206	1	1	94.7	0.087
MS	98.8	99.7	98.0	98.2	99.9	35	34	**1**	**0**	3	73	1	1	98.1	0.072
GS	**99.9**	99.9	**99.9**	**99.9**	**100.0**	35	**34**	**1**	**0**	1	4	**0**	**0**	**99.9**	0.089
Y3DS	97.1	99.6	94.7	94.9	99.8	35	33	**1**	1	7	203	1	2	94.7	0.088
Y4DS	98.8	99.6	97.9	98.1	99.8	35	**34**	**1**	**0**	7	74	1	2	97.9	0.106
FDS	98.2	99.9	96.5	96.5	**100.0**	35	33	**1**	1	1	140	**0**	**0**	96.5	0.077
MDS	98.2	99.9	96.5	96.5	**100.0**	35	33	**1**	1	1	140	**0**	**0**	96.5	0.076
RDS	99.3	99.3	99.3	99.3	99.4	35	**34**	**1**	**0**	25	27	**0**	6	98.7	0.089
GDS	**99.9**	**100**	99.8	99.8	**100.0**	35	**34**	**1**	**0**	**0**	**0**	**0**	1	99.8	**0.062**

measure a case of 26 recordings running on 1 approach to see the impact of the sort of activity on the presentation of the framework. Note that the hand in the 26 recordings is utilized to train the deep appearance descriptor for DeepSORT as referenced in chapter 3, sub-section 3.2.2. Table 4.5 reports short-term tracking overall result on Micand32S following the MOT16 evaluation protocol. The GDS presents an almost perfect performance on Micand32S, detailed metrics of this method is revealed in Figure B-22. Of all ground-truth-free approaches, the Y4DS approach reaches a remarkable result shown in Figure B-7.

Long-term tracking on a realistic dataset: Micand32E

In this part, I will validate the effectiveness of the 11 approaches on the Micand32E dataset. As referenced in chapter 2, sub-section 2.4.3, the Micand32E contains 6 sequences, the sequence length of each is in range of 1000 frames and 2000 frames, and total the volume of this sub-dataset is 7857 frames. Also, this sub-dataset fully contains 4 types of patient's action: (5) practice with a ball, (6) practice with water bottles, (7) practice with wooden blocks, (8) practice with cylinders. This sub-dataset is very realistic, interesting and quite challenging because it is specifically selected with careful observations by developers in the NAFOSTED's project who are experienced in computer vision problems. Table 4.6 presents the overall tracking results referenced on Micand32E of the 11 approaches. To analyze the complexity and the difficulty of the four types of action, I analyzed the results of the tracking algorithm

Table 4.6: Long-term tracking **overrall** results on Micand32E following the MOT16 protocol.

Method	IDF1 (%)↑	IDP (%)↑	IDR (%)↑	Rcll ↑	Prcn ↑	GT ↑	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	**MOTA** (%)↑	MOTP ↓
Y3S	51.4	59.4	45.2	75.3	99.4	24	7	8	9	68	3630	123	174	74.1	0.133
Y4S	56.7	60.7	53.0	86.4	99.4	24	9	**11**	4	81	1996	134	159	85.0	0.127
FS	74.5	73.9	74.8	97.9	97.1	24	17	7	**0**	426	306	115	91	94.2	0.082
MS	74.5	73.9	74.8	97.9	97.2	24	17	7	**0**	420	304	114	90	94.3	0.082
GS	89.1	89.3	88.7	98.5	99.6	24	21	3	**0**	62	220	91	50	97.5	0.059
Y3DS	58.7	66.0	52.6	78.4	98.7	24	9	7	8	149	3176	123	202	76.6	0.151
Y4DS	65.0	68.1	61.9	89.3	98.5	24	11	9	4	194	1581	122	192	87.1	0.142
FDS	79.4	79.0	79.5	98.1	97.8	24	17	7	**0**	320	282	117	75	95.1	0.060
MDS	83.5	83.5	83.3	98.1	98.7	24	18	5	1	184	275	95	61	96.2	0.054
RDS	**89.6**	**89.5**	**89.5**	98.2	98.6	24	16	8	**0**	212	258	79	74	96.3	0.072
GDS	88.5	88.5	88.1	**99.1**	**99.9**	24	**23**	1	**0**	**12**	**135**	82	**43**	**98.4**	**0.052**

Table 4.7: GDS **detail** results on Micand32E.

Method	IDF1 (%)↑	IDP (%)↑	IDR (%)↑	Rcll ↑	Prcn ↑	GT ↑	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	**MOTA** (%)↑	MOTP ↓
GH010354_5_17718_19366	64.1	64.8	63.5	97.7	99.6	3	3	0	**0**	7	41	5	7	97.0	0.096
GH010373_5_1284_2724	89.1	89.1	88.7	98.8	99.9	**6**	**6**	0	**0**	4	40	19	14	98.1	0.051
GH010358_6_10208_11900	81.4	81.3	81.1	99.3	**100.0**	4	4	0	**0**	**0**	24	8	13	99.0	**0.038**
GH010373_6_3150_4744	**99.7**	**99.8**	**99.5**	**99.8**	**100.0**	4	4	0	**0**	**0**	8	**1**	1	**99.7**	0.048
GH010358_7_2490_3390	97.6	97.4	97.2	99.0	99.9	4	3	**1**	**0**	1	19	9	8	98.5	0.050
GH010358_8_8000_8547	97.2	97.3	97.0	99.7	**100.0**	3	3	0	**0**	**0**	**3**	40	**0**	96.1	0.048
Overall	88.5	88.5	88.1	99.1	99.9	24	23	1	0	12	135	82	43	98.4	0.052

that achieves highest MOTA, the GDS, across these four action types. This result is shown in the Table 4.7. Especially, the approach proposed by the author of this thesis performs an impressive result as in the Table 4.8.

4.3 Discussions

4.3.1 Object detection: tradeoff between accuracy and speed

Concerning egocentric hand detection performance, from Table 4.2.1, it's evident that, on the Micand32 dataset, two-stage detectors, represented by the RCNN family

Table 4.8: RDS **detail** results on Micand32E.

Method	IDF1 (%)↑	IDP (%)↑	IDR (%)↑	Rcll ↑	Prcn ↑	GT ↑	MT ↑	PT ↑	ML ↓	FP ↓	FN ↓	IDs ↓	FM ↓	**MOTA** (%)↑	MOTP ↓
GH010354_5_17718_19366	63.1	63.1	63.1	95.4	95.5	3	1	2	0	79	80	0	10	90.7	0.112
GH010373_5_1284_2724	79.4	79.2	79.2	97.5	98.2	6	4	2	0	60	82	18	27	95.2	0.072
GH010358_6_10208_11900	97.0	96.4	97.2	98.6	98.3	4	3	1	0	9	32	9	12	97.4	0.063
GH010373_6_3150_4744	99.6	99.8	99.5	99.7	100.0	4	3	1	0	1	10	1	2	99.6	0.069
GH010358_7_2490_3390	97.5	97.4	96.9	98.3	99.5	4	3	1	0	9	32	9	12	97.4	0.063
GH010358_8_8000_8547	96.8	96.8	96.8	99.5	99.5	3	2	1	0	5	6	40	1	95.4	0.071
Overall	89.6	89.5	89.5	89.5	98.2	24	16	8	0	212	258	79	74	96.3	0.072

models generally outperform the one-state detector, the YOLO family models. Concerning the average precision, the AP50 obtained by the MaskRCNN and FastrCNN are 98,9% and 97,9% respectively. This points that MaskRCNN performs an impressive segmentation result, while the FastRCNN additionally presents an awesome detection quality where the AP is pretty high, 96.2%. Yolov4 performs a remarkable performance with 95,6% AP50 and 72,5% AP^{medium}. Yolov3 adapts well in predicting large hands, it obtains 54,1% AP^{large}, but is still worse than the MaskRCNN which acquires 92,2% AP^{large}.

In term of average recall, again, the FastRCNN performs the best performance with 97,6% for both AR^{large} and $AR^{max} = 100$, this means that this models can detect almost large hands in the datasets.

Metrics in Table 4.3 is measured in order to evaluate the machine memory requirements and the processing rate of detection phase when inference models. As theory researched in chapter 2, the one-stage detectors represented by Yolo family models completely outperforms the two-stage detectors. Yolov4 performs a real-time speed, 25 frames per second on a normal machine which requires only 2Gb RAM and 2GB GPU memory in an inference stage. Processing rate is slightly reduced in case of Yolov3spp with 22 frames per second. The MaskRCNN takes up to 215 milliseconds computing time for each images, and thus the speed is only 5 frames per second and it's not adaptable for real-time processing application, and also, it requires a huge memory, 3,5GB GPU and 2,4GB CPU. From this investigation result, three recommendations on the choice of egocentric hand detection and segmentation can be provided. First, the Yolov4x is suggested for the application that requires real-time processing and machine resources are restricted. Second, in case the accuracy is essential, the MaskRCNN should be employed. Third, the FastRCNN seems to get the balance trade-off between precision and speed in remaining cases.

4.3.2 The superiority of DeepSORT over SORT

To compare the effectiveness of the 2 tracking method, overall tracking evaluation on Micand32E reported in Table 4.6 is very essential. On the same detection algo-

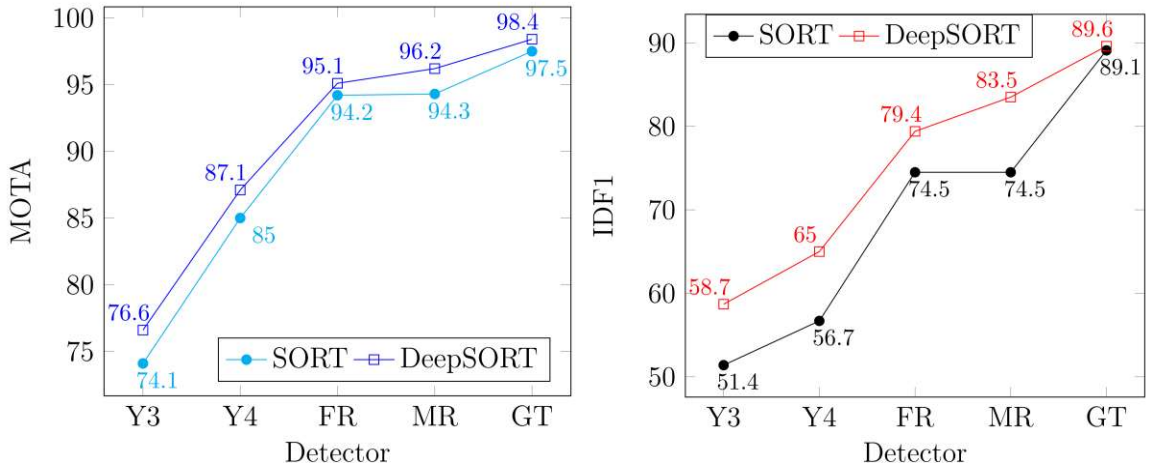

Figure 4-1: Overall MOTA and IDF1 metric on Micand32E.

rithm, DeepSORT generates a higher MOTA in almost cases, for example: Y3DS 76,6% while Y3S 74,1%, MDS 96,2% while MS 94,3%. Figure 4.3.2 and Figure 4.3.2 clearly demonstrates the dominance of DeepSORT over SORT. This result is expected because the DeepSORT tracking inherits the benefits of deep appearance descriptor trained on the egocentric hand re-identification dataset generated by this thesis which is mentioned in the background theory. Also, the identity switches IDs has significantly decreased between SORT and DeepSORT, 91 for GS and only 79 for GDS.

4.3.3 Impact of detection method over tracking result

In the sub-section 4.2.1, the accuracy of the Yolo family models and RCNN family models are reported and we insists that in term of precision, the MaskRCNN and FastRCNN completely outperforms the Yolos. The impact of choosing detection tactic on the tracking performance is also shown clearly in Figure 4 1 and Figure 4 2. On the same tracking method, SORT, MOTA results of Yolov3, Yolov4, FasterRCNN, MaskRCNN and detection groundtruth are respectively 74,1%, 85%, 94,2%, 94,3% and 97,5%. Also it's remarkable that the number of FN decreases, 3176 for Y3DS, 1581 for Y4DS, 282 for FDS, 184 for MDS and 82 for GDS. After researching theory and evaluating experiments, it's quit intuitive but reasonable to conclude that the

better detection method is, the higher tracking quality is.

4.3.4 Complexity of 4 types of patients's actions

The diversity and complexity of 4 type of patient's actions are analyzed via metrics in Table 4.7, the highest results achieved in this experiment. In term of MOTA metric, the GDS obtains best performance on the action type 6, with 2 sequences: GH010373_6_3150_4744 99,7% and GH010358_6_10208_11900 99%. On the contrary, the MOTA for GH010358_8_8000_5447 is 96,1% with GDS. On the other hand, the sequence's volume also has impacts on the tracking qualities. All videos in Micand32S are short-term and the results of all the 11 approaches is almost perfect on this datasets. But in Micand32E, the sequences is long and then many weaknesses of tracking algorithms appear, it's visible via the metrics like IDP and IDR. However, on both datasets, almost approaches can still track hands well, this is shown by metric GT and MT in Table 4.6.

4.3.5 Challenging cases

According to my personal coding experience through this thesis, multiple object tracking is quit difficult to debug errors. The evaluation process is not reversible, when the results are available, the metrics can be calculated, but by just looking at the metrics, it's hard to know where the errors occur, we can only visualize the result through visible video to find the bug. In this way, I find some challenging cases in some sequences that make almost of 11 approaches yield tracking errors. Some typical cases are enumerated as follow:

- When hands are occluded by other obstacles. Trackers lost information about these hands due to the detector naturally can't find those hands. In figure 4-2, in the first, from frame 200^{th} to 250^{th}, SORT are tracing the right hand with ID 1. In the second row, from frame 256^{th} to frame 261^{th}, an experimental equipment - a "toy house", has significantly obscured the right hand part. This makes FasterRCNN can't recognize that hand (false positive), and therefore,

SORT lost that hand's trajectory. Until frame 267^{th}, the hand goes out of the equipment and can be seen clearly. In the third row, FasterRCNN can predict again that right hand, but the shape of the hand seems to be too different for the first row, and of course, SORT initialize a new ID 7 for the hand. A remarkable point in this case is that, with the same condition, the left hand isn't overlapped by the "toy house", so both FaterRCNN and SORT worked perfectly, it keeps the ID 4 for the left hand through all the time.

- When a hand moves rapidly so that the cameras can't catch, this make the motion blur phenomenon occur. As a result, detectors fail to predict the existence of the hand. For example, in figure 4-3, in the frame 119^{th}, the trackers are tracing the ID 1, but in the following frames from 120^{th} to 142^{th} Yolov3 can't predict the hand's existence (false positive), until the frame 143^{th} it find that hand again but at this time the tracker can't retrieve the hand's identity, it initialize a new ID 2 for the hand.

- When the shape of the hand changes heavily. Intuitively, the deep appearance descriptor doesn't work in this case, it simply can't recognize that these different shapes belong to a same id. Consequently, a new ID is initialized and then ID switches occur. In figure 4-4, DeepSORT works stability in frames $1582^{th}, 1592^{th}, 1602^{th}, 1612^{th}, 1622^{th}$ with the hand ID 5. But in frame 1630^{th}, the shape of the right hand changes heavily, DeepSORT initialize a new ID 7 for that hand. Figure 4-2 also proves this: the shape of the right hand in the first row and the third row is too different, and certainly, SORT assign 2 identities for just 1 hand.

- The ambiguous definition of the "hand" causes difficulty. Detection model is training on both GTEA family ("hand" includes wrist, arm) and EgoHand ("hand" just includes hand) so the during inference stage, the detector may be quite tough. In figure 4-5, FasterRCNN and SORT keep the left hand awesomely through 283 frames for 1123^{th} to 1406^{th} with ID 22, presented in the first row. Suddenly, in frame 1407^{th}, FasterRCNN predicts this left hand including the

wrist. This makes SORT assign a new ID 24 for that hand as show in the second row. This issue is reported and will be discussed in the future work in the section 5.2.

- When the hands disappear from the camera's view and return after a long time, this makes the tracker "forget" that instance.

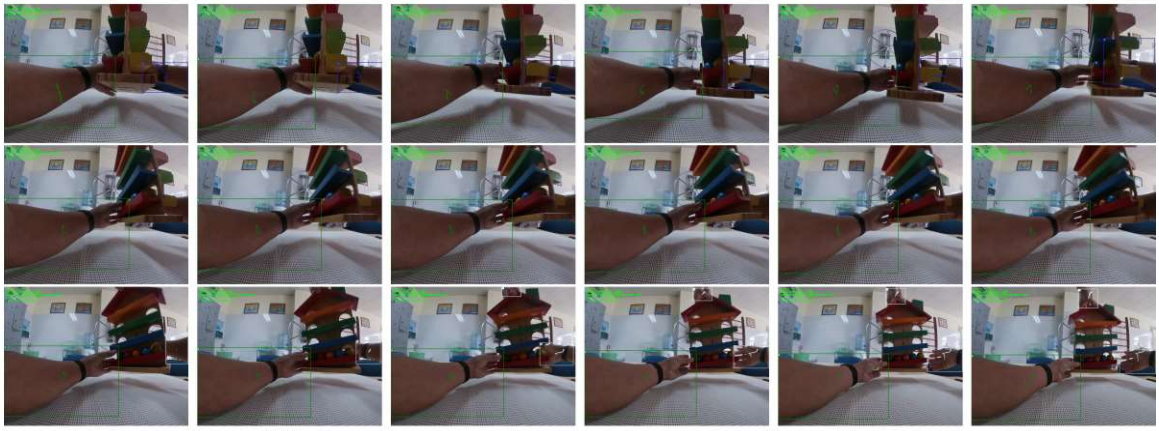

Figure 4-2: Illustration of hand occluded by an obstacle. Pay attention to the patient's right hand. Frames extracted from GH010373_5_1284_2724 using Faster-RCNN+SORT, ordered from left to right, up to down: (200, 210, 220, 230, 240, 250), (256, 257, 258, 259, 260, 261), (267, 270, 273, 276, 279, 282).

Figure 4-3: Motion blur phenomenon due to hand's rapid movement. Frames extracted from GH010354_5_17718_19366 using Yolov3+SORT, ordered from left to right: 119, 125, 130, 131, 137 and 143.

Figure 4-4: Shape changing illustration. Frames extracted from GH010358_6_10208_11900 using MaskRCNN+DeepSORT, ordered from left to right: 1582, 1592, 1602, 1612, 1622 and 1630.

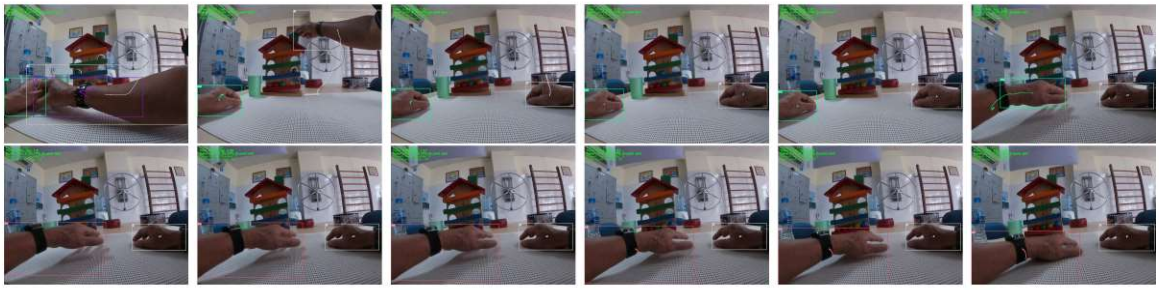

Figure 4-5: The unclear "hand" definition illustration. Pay attention to the patient's left hand. Frames extracted from GH010373_5_1284_2724 using Faster-RCNN+SORT, ordered from left to right, up to down: (1123, 1173, 1223, 1273, 1323, 1406), (1407, 1408, 1409, 1410, 1411, 1412).

Chapter 5

Conclusion

5.1 Conclusion

5.1.1 Accomplishment

In this thesis, I attempted to tackle the issue of detection, segmentation and tracking human hand from egocentric vision. Alongside investigating, affirming the hypothesis, and proposing a hand following by discovery framework, I built a module to detect and track human hands in videos from the first perspective. Chapter 1 summarized the general overview of object recognition, including object classification, detection, segmentation and tracking in the wild and in the FPV. Related works are studied and discussed, associated project is also reported, and from that, the main problem is defined. Chapter 2 covers the background theory and methodology of the RCNN family and YOLO family, the SORT and DeepSORT algorithms.This chapter present and analyze egocentric hand datasets incorporate GTEA family, EgoHand; and furthermore present a new egocentric hand tracking datasets Micand32 alongside with a semi-automatic annotator EHTA. Chapter 3 reported the proposed pipeline which contains 4 stage: data preparing stage, training stage, inference stage and evaluation stage. This chapter reports in detail techniques in the training and testing phase. Chapter 4 describes about the evaluation criteria and then enumerates overall and featured detection, segmentation result following the COCO dataset standard and tracking result of 11 approaches on Micand32S and Micand32E following the MOT Challenge standard. Chapter 5 terminates the thesis by considering the accomplishment, the drawback and also purpose some potential future works.

5.1.2 Drawback

Due to time constraints and limited human resource, the team hasn't completed labeling the whole datasets from Nafosted's project. The Micand32 dataset is completely new in terms of egocentric hand tracking and relatively good compared to other datasets in terms of sample size and labeling quality. However, due to many factors, this set is not general enough to train neural networks to function properly.

The number of experiments are relatively large, so I haven't deeply analyzed the results. Algorithms work stably under laboratory conditions. In the wild, algorithms still have some problems to be solved, typically cases such as excessive arm resistance, long-term disappearance of an instance, blur effects at hand, the patient moves with a sudden acceleration, the abnormal movements of the patient's hands.

Semi-automatic labeling tool needs users to know the basic code and observe to draw experience for each video to be assigned. The tool also works offline and has not integrated the online and real-time model update feature.

In terms of academics, the algorithms used in the thesis have not had new or breakthrough idea due to my shortcomings of expertise.

5.2 Future works

In this proposition, the meaning of the "hand" is ambiguous between datasets utilized. For GTEA family datasets, the "hand" incorporates the arm, wrist and hand while the "hand" in EgoHands datasets just contains only hand area and the "hand" in Micand32 isn't characterized obviously. This influences the nature of the egocentric hand re-identification dataset. Consequently, the deep appearance descriptor can't recognize the hand's highlights and doesn't function admirably. This re-identification dataset will be checked and upgraded in later works.

The EHTA framework currently works offline. An online annotation tool can update machine learning models right after the label adjustment of annotator. This will help the model predict better in the next frame as shown in Figure 5-1. The

EHTA pipeline is being developed to be more adaptable, flexible and productive.

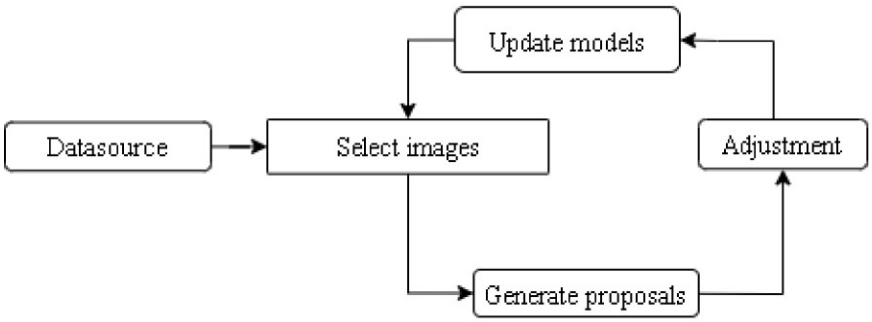

Figure 5-1: Schematic illustration of an online annotation pipeline.

The remainder of the datasets in NAFOSTED's project will be fully annotated. Algorithms then will be tested on this dataset. Likewise, other detection algorithms like SSD [54], Retina [55], CenterNet [56] and EfficientDet [57] are being integrated and tested.

As a relative complicated and integrated computer vision mission like multiple object tracking, the tracking by detection technique referenced in this thesis is still suffering from issues such as a large number of false positive tracks. False positive tracks caused by unreliable detection results are badly affecting the performance of trackers. To reduce the effect of unreliable detection on tracking, in the future works, I propose to incorporate a low confidence track filtering extension into DeepSORT. Tracks with low average detection confidence in their initial several frames will be deleted. This strategy is promising and it has been conducted on a vehicle tracking problem [58] similar to this thesis.

The accelerometer sensors worn by patients generate real-time kinetics information [59]. This acceleration information is extremely useful and can be fed to Kalman filter to predict the motion model better. I propose to match the sensor stream from the accelerometer with the camera to detect instances of the hand's physical movement. This method has a wide range of potential applications in the cyber-physical systems domain such as identification, localization, tracking and action recognition, for example, Leap Motion [60] is a well-known and successful commercial product designed for hand tracking in virtual reality.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Tables

Key	Value
BASE:	"../Base-RCNN-FPN.yaml"
MODEL:	
WEIGHTS:	"detectron2://ImageNetPretrained/MSRA/R-50.pkl"
MASK_ON	False
RESNETS:	
DEPTH:	50
OUT_FEATURES:	["res2", "res3", "res4", "res5"]
FPN:	
IN_FEATURES:	["res2", "res3", "res4", "res5"]
SOLVER:	(210000, 250000)
STEPS:	
MAX_ITER:	270000
ROI_HEADS:	"StandardROIHeads"
NAME:	
IN_FEATURES:	["p2", "p3", "p4", "p5"]

Table A.1: FasterRCNN_R_50_FPN3x training configuration file. Detail information field is explained at the Detectron2's application programming interface (API) documentation. The main difference of FasterRCNN and MaskRCNN in term of configuration is MaskRCNN's option MASK_ON value.

Key	Value
nc:	1 # number of classes
depth_multiple:	0.33 # model depth multiple
width_multiple:	0.50 # layer channel multiple
anchors:	- [10,13, 16,30, 33,23] # P3/8 - [30,61, 62,45, 59,119] # P4/16 - [116,90, 156,198, 373,326] # P5/32
backbone:	# [from, number, module, args] [[-1, 1, Focus, [64, 3]], # 0-P1/2 [-1, 1, Conv, [128, 3, 2]], # 1-P2/4 [-1, 3, BottleneckCSP, [128]], [-1, 1, Conv, [256, 3, 2]], # 3-P3/8 [-1, 9, BottleneckCSP, [256]], [-1, 1, Conv, [512, 3, 2]], # 5-P4/16 [-1, 9, BottleneckCSP, [512]], [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32 [-1, 1, SPP, [1024, [5, 9, 13]]], [-1, 3, BottleneckCSP, [1024, False]], # 9]
head:	[[-1, 1, Conv, [512, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 6], 1, Concat, [1]], # cat backbone P4 [-1, 3, BottleneckCSP, [512, False]], # 13 [-1, 1, Conv, [256, 1, 1]], [-1, 1, nn.Upsample, [None, 2, 'nearest']], [[-1, 4], 1, Concat, [1]], # cat backbone P3 [-1, 3, BottleneckCSP, [256, False]], # 17 (P3/8-small) [-1, 1, Conv, [256, 3, 2]], [[-1, 14], 1, Concat, [1]], # cat head P4 [-1, 3, BottleneckCSP, [512, False]], # 20 (P4/16-medium) [-1, 1, Conv, [512, 3, 2]], [[-1, 10], 1, Concat, [1]], # cat head P5 [-1, 3, BottleneckCSP, [1024, False]], # 23 (P5/32-large) [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)]

Table A.2: YOLOv4x training configuration file. Detail information field is explained at Ultralytic API.

Field	Meaning
file_name	the full path to the image file. Rotation or flipping may be applied if the image has EXIF metadata.
height, width	integer. The shape of the image.
image_id	(str or int): a unique id that identifies this image. Required by many evaluators to identify the images, but a dataset may use it for different purposes.
annotations	(list[dict]): Required by instance detection, segmentation or keypoint detection tasks. Each dict corresponds to annotations of one instance in this image.
bbox	(list[float], required): list of 4 numbers representing the bounding box of the instance.
bbox_mode	(int, required): the format of bbox. It must be a member of structures.BoxMode. Currently supports: BoxMode.XYXY_ABS, BoxMode.XYWH_ABS.
category_id	(int, required): an integer in the range [0, num_categories-1] representing the category label. The value num_categories is reserved to represent the "background" category, if applicable.
segmentation	(list[list[float]] or dict): the segmentation mask of the instance.

Table A.3: Detectron2's custom dataset format.

Appendix B

Figures

In this appendix, experiment results shortened in chapter 4 are reported in detail.

B.1 Short-term tracking results on Micand32S

B.2 Long-term tracking results on Micand32E

B.3 Other

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.061	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.077	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.091	0	0	0
GH010354_7_16605_17602_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.154	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.101	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.089	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.089	0	0	0
GH010358_7_352_2464_2	66.0%	100.0%	49.3%	49.3%	100.0%	2	1	0	1	0	137	0	0	49.3%	0.105	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.055	0	0	0
GH010358_7_352_2464_1	94.4%	96.2%	92.7%	95.5%	99.2%	3	3	0	0	2	11	1	1	94.3%	0.076	0	1	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.083	0	0	0
GH010383_5_462_968_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.088	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.147	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.080	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.115	0	0	0
GH010354_8_26079_29031	99.2%	99.5%	98.9%	98.9%	99.5%	1	1	0	0	1	2	0	0	98.4%	0.078	0	0	0
GH010382_6_20592_21726_1	97.8%	99.4%	96.2%	96.2%	99.4%	3	2	1	0	1	7	0	0	95.7%	0.080	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.153	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.078	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.054	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.073	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.098	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.093	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.128	0	0	0
GH010354_7_13210_16534	98.3%	98.3%	98.3%	98.3%	98.3%	1	1	0	0	2	2	0	0	96.7%	0.083	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.044	0	0	0
OVERALL	97.0%	99.6%	94.6%	94.8%	99.8%	35	33	1	1	6	207	1	1	94.6%	0.087	1	1	1

10:51:13 INFO - Completed

Figure B-1: Y3S detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.062	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.082	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.080	0	0	0
GH010354_7_16605_17602_2	84.0%	84.7%	83.3%	83.3%	84.7%	1	1	0	0	18	20	0	5	68.3%	0.339	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.093	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.095	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.088	0	0	0
GH010358_7_352_2464_2	99.3%	100.0%	98.5%	98.5%	100.0%	2	2	0	0	0	4	0	0	98.5%	0.117	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.097	0	0	0
GH010358_7_352_2464_1	98.1%	99.6%	96.7%	96.7%	99.6%	3	3	0	0	1	8	0	0	96.3%	0.085	0	0	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.122	0	0	0
GH010383_5_462_968_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.097	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.159	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.126	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.120	0	0	0
GH010354_8_26079_29031	99.2%	99.5%	98.9%	98.9%	99.5%	1	1	0	0	1	2	0	0	98.4%	0.085	0	0	0
GH010382_6_20592_21726_1	97.3%	98.4%	96.2%	96.2%	98.4%	3	2	1	0	3	7	0	0	94.6%	0.084	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.148	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.159	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.100	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.076	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.112	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.169	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.149	0	0	0
GH010354_7_13210_16534	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.077	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.049	0	0	0
OVERALL	98.6%	99.4%	97.7%	97.8%	99.4%	35	34	1	0	23	89	0	5	97.2%	0.108	1	0	1

12:03:34 INFO - Completed

Figure B-2: Y4S detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.061	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.077	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.089	0	0	0
GH010354_7_16605_17602_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.156	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.100	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.092	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.088	0	0	0
GH010358_7_352_2464_2	66.0%	100.0%	49.3%	49.3%	100.0%	2	1	0	1	0	137	0	0	49.3%	0.106	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.048	0	0	0
GH010358_7_352_2464_1	94.4%	96.2%	92.7%	95.5%	99.2%	3	3	0	0	2	11	1	1	94.3%	0.081	0	1	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.085	0	0	0
GH010383_5_462_968_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.095	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.151	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.081	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.116	0	0	0
GH010354_8_26079_29031	99.2%	99.5%	98.9%	98.9%	99.5%	1	1	0	0	1	2	0	0	98.4%	0.082	0	0	0
GH010382_6_20592_21726_1	98.1%	99.4%	96.8%	96.8%	99.4%	3	2	1	0	1	6	0	0	96.2%	0.076	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.157	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.078	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.051	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.073	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.093	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.092	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.130	0	0	0
GH010354_7_13210_16534	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.080	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.044	0	0	0
OVERALL	97.1%	99.7%	94.6%	94.8%	99.9%	35	33	1	1	4	206	1	1	94.7%	0.087	1	1	1

10:46:26 INFO - Completed

Figure B-3: FS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.044	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.065	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.071	0	0	0
GH010354_7_16605_17602_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.140	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.082	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.076	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.075	0	0	0
GH010358_7_352_2464_2	99.3%	100.0%	98.5%	98.5%	100.0%	2	2	0	0	0	4	0	0	98.5%	0.065	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.036	0	0	0
GH010358_7_352_2464_1	94.6%	96.2%	93.1%	95.9%	99.2%	3	3	0	0	2	10	1	1	94.7%	0.067	0	1	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.066	0	0	0
GH010383_5_462_968_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.084	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.137	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.069	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.114	0	0	0
GH010354_8_26079_29031	99.5%	100.0%	98.9%	98.9%	100.0%	1	1	0	0	0	2	0	0	98.9%	0.064	0	0	0
GH010382_6_20592_21726_1	97.8%	99.4%	96.2%	96.2%	99.4%	3	2	1	0	1	7	0	0	95.7%	0.058	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.144	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.062	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.031	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.052	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.083	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.075	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.117	0	0	0
GH010354_7_13210_16534	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.071	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.026	0	0	0
OVERALL	98.8%	99.7%	98.0%	98.2%	99.9%	35	34	1	0	3	73	1	1	98.1%	0.072	1	1	1

Figure B-4: MS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.048	0	0	0
GH010358_7_2490_3390_1	99.7%	99.7%	99.7%	100.0%	100.0%	3	3	0	0	0	0	0	0	100.0%	0.063	1	0	1
GH010382_8_16207_17479_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.038	0	0	0
GH010354_7_16605_17602_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.284	0	0	0
GH010382_5_5725_7093_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.052	0	0	0
GH010374_6_4944_6241_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.071	0	0	0
GH010374_6_4944_6241_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.050	0	0	0
GH010358_7_352_2464_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.102	0	0	0
GH010382_5_955_4771_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.096	0	0	0
GH010358_7_352_2464_1	99.6%	100.0%	99.2%	99.2%	100.0%	3	3	0	0	0	2	0	0	99.2%	0.056	0	0	0
GH010382_8_16207_17479_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.089	0	0	0
GH010383_5_462_968_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.104	0	0	0
GH010383_8_4544_5192_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.160	0	0	0
GH010358_7_2490_3390_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.108	0	0	0
GH010383_5_462_968_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.100	0	0	0
GH010354_8_26079_29031	99.7%	99.5%	100.0%	100.0%	99.5%	1	1	0	0	1	0	0	0	99.5%	0.053	0	0	0
GH010382_6_20592_21726_1	99.5%	100.0%	98.9%	98.9%	100.0%	3	2	1	0	0	2	0	0	98.9%	0.057	0	0	0
GH010383_8_4544_5192_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.137	0	0	0
GH010382_5_5725_7093_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.156	0	0	0
GH010382_6_18190_20215_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.098	0	0	0
GH010382_6_20592_21726_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.072	0	0	0
GH010383_8_3221_3956_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.155	0	0	0
GH010383_8_3221_3956_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.154	0	0	0
GH010354_7_16605_17602_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.082	0	0	0
GH010354_7_13210_16534	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.033	0	0	0
GH010382_6_18190_20215_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.032	0	0	0
OVERALL	99.9%	99.9%	99.9%	99.9%	100.0%	35	34	1	0	1	4	0	0	99.9%	0.089	1	0	1

Figure B-5: GS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.060	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.078	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.093	0	0	0
GH010354_7_16605_17602_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.154	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.102	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.089	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.091	0	0	0
GH010358_7_352_2464_2	67.3%	100.0%	50.7%	50.7%	100.0%	2	1	0	1	0	133	0	1	50.7%	0.106	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.054	0	0	0
GH010358_7_352_2464_1	94.4%	96.2%	92.7%	95.5%	99.2%	3	3	0	0	2	11	1	1	94.3%	0.076	0	1	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.082	0	0	0
GH010383_5_462_968_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.089	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.146	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.080	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.116	0	0	0
GH010354_8_26079_29031	99.2%	99.5%	98.9%	98.9%	99.5%	1	1	0	0	1	2	0	0	98.4%	0.079	0	0	0
GH010382_6_20592_21726_1	97.8%	99.4%	96.2%	96.2%	99.4%	3	2	1	0	1	7	0	0	95.7%	0.080	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.153	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.079	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.054	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.073	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.098	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.093	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.129	0	0	0
GH010354_7_13210_16534	97.9%	97.5%	98.3%	98.3%	97.5%	1	1	0	0	3	2	0	0	95.8%	0.084	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.044	0	0	0
OVERALL	97.1%	99.6%	94.7%	94.9%	99.8%	35	33	1	1	7	203	1	2	94.7%	0.088	1	1	1

Figure B-6: Y3DS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.053	0	0	0
GH010358_7_2490_3390_1	99.1%	99.7%	98.4%	98.8%	100.0%	3	3	0	0	0	4	0	0	98.8%	0.070	1	0	1
GH010382_8_16207_17479_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.079	0	0	0
GH010354_7_16605_17602_2	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.322	0	0	0
GH010382_5_5725_7093_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.088	0	0	0
GH010374_6_4944_6241_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.091	0	0	0
GH010374_6_4944_6241_1	99.3%	100.0%	98.6%	98.6%	100.0%	1	1	0	0	0	2	0	0	98.6%	0.078	0	0	0
GH010358_7_352_2464_2	99.3%	100.0%	98.5%	98.5%	100.0%	2	2	0	0	0	4	0	0	98.5%	0.112	0	0	0
GH010382_5_955_4771_2	99.1%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.109	0	0	0
GH010358_7_352_2464_1	94.6%	96.2%	93.1%	95.9%	99.2%	3	3	0	0	2	10	1	1	94.7%	0.071	0	1	0
GH010382_8_16207_17479_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.130	0	0	0
GH010383_5_462_968_2	98.3%	99.2%	97.5%	97.5%	99.2%	1	1	0	0	1	3	0	1	96.7%	0.128	0	0	0
GH010383_8_4544_5192_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.167	0	0	0
GH010358_7_2490_3390_2	99.4%	100.0%	98.8%	98.8%	100.0%	2	2	0	0	0	4	0	0	98.8%	0.133	0	0	0
GH010383_5_462_968_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.118	0	0	0
GH010354_8_26079_29031	99.2%	99.5%	98.9%	98.9%	99.5%	1	1	0	0	1	2	0	0	98.4%	0.071	0	0	0
GH010382_6_20592_21726_1	97.3%	98.4%	96.2%	96.2%	98.4%	3	2	1	0	3	7	0	0	94.6%	0.070	0	0	0
GH010383_8_4544_5192_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.146	0	0	0
GH010382_5_5725_7093_2	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.188	0	0	0
GH010382_6_18190_20215_2	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.104	0	0	0
GH010382_6_20592_21726_2	99.2%	100.0%	98.4%	98.4%	100.0%	2	2	0	0	0	4	0	0	98.4%	0.085	0	0	0
GH010383_8_3221_3956_1	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.091	0	0	0
GH010383_8_3221_3956_2	99.0%	100.0%	98.0%	98.0%	100.0%	1	1	0	0	0	2	0	0	98.0%	0.153	0	0	0
GH010354_7_16605_17602_1	99.2%	100.0%	98.5%	98.5%	100.0%	1	1	0	0	0	2	0	0	98.5%	0.124	0	0	0
GH010354_7_13210_16534	99.2%	100.0%	98.3%	98.3%	100.0%	1	1	0	0	0	2	0	0	98.3%	0.073	0	0	0
GH010382_6_18190_20215_1	99.3%	100.0%	98.7%	98.7%	100.0%	1	1	0	0	0	2	0	0	98.7%	0.034	0	0	0
OVERALL	98.8%	99.6%	97.9%	98.1%	99.8%	35	34	1	0	7	74	1	2	97.9%	0.106	1	1	1

11:36:26 INFO - Completed

Figure B-7: Y4DS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.054	0	0	0
GH010358_7_2490_3390_1	99.7%	99.7%	99.7%	100.0%	100.0%	3	3	0	0	0	0	0	0	100.0%	0.077	1	0	1
GH010382_8_16207_17479_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.042	0	0	0
GH010354_7_16605_17602_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.137	0	0	0
GH010382_5_5725_7093_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.065	0	0	0
GH010374_6_4944_6241_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.057	0	0	0
GH010374_6_4944_6241_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.062	0	0	0
GH010358_7_352_2464_2	67.3%	100.0%	50.7%	50.7%	100.0%	2	1	0	1	0	133	0	0	50.7%	0.126	0	0	0
GH010382_5_955_4771_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.044	0	0	0
GH010358_7_352_2464_1	99.4%	100.0%	98.8%	98.8%	100.0%	3	3	0	0	0	3	0	0	98.8%	0.062	0	0	0
GH010382_8_16207_17479_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.054	0	0	0
GH010383_5_462_968_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.072	0	0	0
GH010383_8_4544_5192_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.143	0	0	0
GH010358_7_2490_3390_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.091	0	0	0
GH010383_5_462_968_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.099	0	0	0
GH010354_8_26079_29031	99.7%	99.5%	100.0%	100.0%	99.5%	1	1	0	0	1	0	0	0	99.5%	0.058	0	0	0
GH010382_6_20592_21726_1	98.9%	100.0%	97.8%	97.8%	100.0%	3	2	1	0	0	4	0	0	97.8%	0.064	0	0	0
GH010383_8_4544_5192_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.141	0	0	0
GH010382_5_5725_7093_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.057	0	0	0
GH010382_6_18190_20215_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.051	0	0	0
GH010382_6_20592_21726_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.053	0	0	0
GH010383_8_3221_3956_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.166	0	0	0
GH010383_8_3221_3956_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.103	0	0	0
GH010354_7_16605_17602_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.104	0	0	0
GH010354_7_13210_16534	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.041	0	0	0
GH010382_6_18190_20215_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.039	0	0	0
OVERALL	98.2%	99.9%	96.5%	96.5%	100.0%	35	33	1	1	1	140	0	0	96.5%	0.077	1	0	1

10:49:22 INFO - Completed

Figure B-8: FDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.059	0	0	0
GH010358_7_2490_3390_1	99.7%	99.7%	99.7%	100.0%	100.0%	3	3	0	0	0	0	0	0	100.0%	0.075	1	0	1
GH010382_8_16207_17479_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.047	0	0	0
GH010354_7_16605_17602_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.145	0	0	0
GH010382_5_5725_7093_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.061	0	0	0
GH010374_6_4944_6241_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.064	0	0	0
GH010374_6_4944_6241_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.061	0	0	0
GH010358_7_352_2464_2	67.0%	100.0%	50.4%	50.4%	100.0%	2	1	0	1	0	134	0	0	50.4%	0.118	0	0	0
GH010382_5_955_4771_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.038	0	0	0
GH010358_7_352_2464_1	99.4%	100.0%	98.8%	98.8%	100.0%	3	3	0	0	0	3	0	0	98.8%	0.064	0	0	0
GH010382_8_16207_17479_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.058	0	0	0
GH010383_5_462_968_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.080	0	0	0
GH010383_8_4544_5192_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.145	0	0	0
GH010358_7_2490_3390_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.088	0	0	0
GH010383_5_462_968_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.100	0	0	0
GH010354_8_26079_29031	99.7%	99.5%	100.0%	100.0%	99.5%	1	1	0	0	1	0	0	0	99.5%	0.059	0	0	0
GH010382_6_20592_21726_1	99.2%	100.0%	98.4%	98.4%	100.0%	3	2	1	0	0	3	0	0	98.4%	0.060	0	0	0
GH010383_8_4544_5192_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.140	0	0	0
GH010382_5_5725_7093_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.054	0	0	0
GH010382_6_18190_20215_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.044	0	0	0
GH010382_6_20592_21726_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.053	0	0	0
GH010383_8_3221_3956_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.166	0	0	0
GH010383_8_3221_3956_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.100	0	0	0
GH010354_7_16605_17602_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.093	0	0	0
GH010354_7_13210_16534	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.039	0	0	0
GH010382_6_18190_20215_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.037	0	0	0
OVERALL	98.2%	99.9%	96.5%	96.5%	100.0%	35	33	1	1	1	140	0	0	96.5%	0.076	1	0	1

10:43:31 INFO - Completed

Figure B-9: MDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.057	0	0	0
GH010358_7_2490_3390_1	99.7%	99.7%	99.7%	100.0%	100.0%	3	3	0	0	0	0	0	0	100.0%	0.064	1	0	1
GH010382_8_16207_17479_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.038	0	0	0
GH010354_7_16605_17602_2	80.8%	80.8%	80.8%	80.8%	80.8%	1	1	0	0	23	23	0	6	61.7%	0.270	0	0	0
GH010382_5_5725_7093_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.058	0	0	0
GH010374_6_4944_6241_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.074	0	0	0
GH010374_6_4944_6241_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.057	0	0	0
GH010358_7_352_2464_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.102	0	0	0
GH010382_5_955_4771_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.081	0	0	0
GH010358_7_352_2464_1	99.6%	100.0%	99.2%	99.2%	100.0%	3	3	0	0	0	2	0	0	99.2%	0.061	0	0	0
GH010382_8_16207_17479_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.090	0	0	0
GH010383_5_462_968_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.079	0	0	0
GH010383_8_4544_5192_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.162	0	0	0
GH010358_7_2490_3390_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.097	0	0	0
GH010383_5_462_968_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.111	0	0	0
GH010354_8_26079_29031	99.7%	99.5%	100.0%	100.0%	99.5%	1	1	0	0	1	0	0	0	99.5%	0.066	0	0	0
GH010382_6_20592_21726_1	99.2%	99.5%	98.9%	98.9%	99.5%	3	2	1	0	1	2	0	0	98.4%	0.065	0	0	0
GH010383_8_4544_5192_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.145	0	0	0
GH010382_5_5725_7093_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.127	0	0	0
GH010382_6_18190_20215_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.090	0	0	0
GH010382_6_20592_21726_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.060	0	0	0
GH010383_8_3221_3956_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.173	0	0	0
GH010383_8_3221_3956_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.171	0	0	0
GH010354_7_16605_17602_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.101	0	0	0
GH010354_7_13210_16534	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.040	0	0	0
GH010382_6_18190_20215_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.044	0	0	0
OVERALL	99.3%	99.3%	99.3%	99.3%	99.4%	35	34	1	0	25	27	0	6	98.7%	0.089	1	0	1

12:02:38 INFO - Completed

Figure B-10: RDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010382_5_955_4771_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.042	0	0	0
GH010358_7_2490_3390_1	99.7%	99.7%	99.7%	100.0%	100.0%	3	3	0	0	0	0	0	0	100.0%	0.069	1	0	1
GH010382_8_16207_17479_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.030	0	0	0
GH010354_7_16605_17602_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.126	0	0	0
GH010382_5_5725_7093_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.046	0	0	0
GH010374_6_4944_6241_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.042	0	0	0
GH010374_6_4944_6241_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.045	0	0	0
GH010358_7_352_2464_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.069	0	0	0
GH010382_5_955_4771_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.025	0	0	0
GH010358_7_352_2464_1	99.6%	100.0%	99.2%	99.2%	100.0%	3	3	0	0	0	2	0	0	99.2%	0.050	0	0	0
GH010382_8_16207_17479_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.042	0	0	0
GH010383_5_462_968_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.070	0	0	0
GH010383_8_4544_5192_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.133	0	0	0
GH010358_7_2490_3390_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.082	0	0	0
GH010383_5_462_968_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.097	0	0	0
GH010354_8_26079_29031	99.5%	100.0%	98.9%	98.9%	100.0%	1	1	0	0	0	2	0	1	98.9%	0.050	0	0	0
GH010382_6_20592_21726_1	98.9%	100.0%	97.8%	97.8%	100.0%	3	2	1	0	0	4	0	0	97.8%	0.042	0	0	0
GH010383_8_4544_5192_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.135	0	0	0
GH010382_5_5725_7093_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.045	0	0	0
GH010382_6_18190_20215_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.023	0	0	0
GH010382_6_20592_21726_2	100.0%	100.0%	100.0%	100.0%	100.0%	2	2	0	0	0	0	0	0	100.0%	0.034	0	0	0
GH010383_8_3221_3956_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.152	0	0	0
GH010383_8_3221_3956_2	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.093	0	0	0
GH010354_7_16605_17602_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.087	0	0	0
GH010354_7_13210_16534	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.032	0	0	0
GH010382_6_18190_20215_1	100.0%	100.0%	100.0%	100.0%	100.0%	1	1	0	0	0	0	0	0	100.0%	0.020	0	0	0
OVERALL	99.9%	100.0%	99.8%	99.8%	100.0%	35	34	1	0	0	8	0	1	99.8%	0.062	1	0	1

Figure B-11: GDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	63.6%	69.1%	58.9%	85.2%	100.0%	3	1	2	0	0	164	36	26	82.0%	0.157	24	11	0
GH010354_5_17718_19366	40.8%	43.5%	38.4%	84.7%	95.9%	3	1	0	2	64	269	11	19	80.4%	0.178	0	11	0
GH010358_7_2490_3390	89.9%	92.2%	87.2%	93.7%	99.9%	4	2	1	1	2	121	6	20	93.3%	0.073	0	6	0
GH010358_6_10208_11900	48.1%	50.1%	46.0%	91.2%	99.9%	4	2	1	1	2	297	17	33	90.6%	0.121	1	15	0
GH010373_6_3150_4744	56.1%	74.8%	44.9%	60.0%	100.0%	4	1	1	2	0	1296	21	25	59.4%	0.145	0	21	0
GH010373_5_1284_2724	24.8%	34.4%	19.3%	55.4%	100.0%	6	0	3	3	0	1483	32	51	54.4%	0.153	1	32	1
OVERALL	51.4%	59.4%	45.2%	75.3%	99.4%	24	7	8	9	68	3630	123	174	74.1%	0.133	26	96	1

10:23:38 INFO - Completed

Figure B-12: Y3S detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.9%	97.3%	96.6%	99.3%	100.0%	3	2	1	0	0	8	40	0	95.7%	0.142	40	0	0
GH010354_5_17718_19366	49.8%	51.7%	48.1%	89.3%	96.0%	3	1	1	1	66	188	5	12	85.2%	0.185	0	5	0
GH010358_7_2490_3390	91.0%	91.8%	89.5%	96.8%	100.0%	4	2	2	0	0	62	4	19	96.6%	0.078	0	4	0
GH010358_6_10208_11900	56.1%	57.2%	54.8%	95.2%	99.8%	4	3	1	0	7	162	11	26	94.6%	0.120	1	9	0
GH010373_6_3150_4744	54.2%	58.3%	50.6%	86.8%	100.0%	4	1	2	1	1	428	25	36	86.0%	0.130	0	25	0
GH010373_5_1284_2724	24.5%	30.6%	20.3%	65.5%	99.7%	6	0	4	2	7	1148	49	66	63.8%	0.128	2	47	1
OVERALL	56.7%	60.7%	53.0%	86.4%	99.4%	24	9	11	4	81	1996	134	159	85.0%	0.127	43	90	1

10:27:11 INFO - Completed

Figure B-13: Y4S detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.4%	96.7%	96.2%	99.4%	99.8%	3	3	0	0	2	7	41	1	95.5%	0.089	40	1	0
GH010354_5_17718_19366	43.0%	42.6%	43.3%	95.6%	94.2%	3	1	2	0	104	78	8	10	89.2%	0.132	3	5	0
GH010358_7_2490_3390	91.2%	90.9%	91.0%	97.5%	98.0%	4	2	2	0	39	49	14	14	94.7%	0.074	2	6	0
GH010358_6_10208_11900	95.7%	94.5%	96.5%	98.8%	97.2%	4	3	1	0	95	42	12	21	95.6%	0.068	4	6	0
GH010373_6_3150_4744	69.0%	69.1%	68.9%	99.5%	99.7%	4	3	1	0	10	17	6	3	99.0%	0.074	1	4	0
GH010373_5_1284_2724	58.0%	57.1%	58.5%	96.6%	94.8%	6	5	1	0	176	113	34	42	90.3%	0.080	9	9	0
OVERALL	74.5%	73.9%	74.8%	97.9%	97.1%	24	17	7	0	426	306	115	91	94.2%	0.082	59	31	0

10:39:03 INFO - Completed

Figure B-14: FS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.4%	96.7%	96.2%	99.4%	99.8%	3	3	0	0	2	7	41	1	95.5%	0.089	40	1	0
GH010354_5_17718_19366	43.0%	42.6%	43.3%	95.6%	94.2%	3	1	2	0	104	78	8	10	89.2%	0.132	3	5	0
GH010358_7_2490_3390	91.3%	90.9%	91.0%	97.6%	98.1%	4	2	2	0	36	47	14	13	95.0%	0.074	3	5	0
GH010358_6_10208_11900	95.7%	94.5%	96.5%	98.8%	97.2%	4	3	1	0	94	42	11	21	95.6%	0.068	3	6	0
GH010373_6_3150_4744	69.0%	69.1%	68.9%	99.5%	99.7%	4	3	1	0	10	17	6	3	99.0%	0.074	1	4	0
GH010373_5_1284_2724	58.0%	57.1%	58.5%	96.6%	94.9%	6	5	1	0	174	113	34	42	90.3%	0.080	9	9	0
OVERALL	74.5%	73.9%	74.8%	97.9%	97.2%	24	17	7	0	420	304	114	90	94.3%	0.082	59	30	0

10:34:55 INFO - Completed

Figure B-15: MS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	97.0%	97.3%	96.8%	99.5%	100.0%	3	3	0	0	0	6	40	0	95.9%	0.062	40	0	0
GH010354_5_17718_19366	64.2%	65.7%	62.8%	95.0%	99.3%	3	2	1	0	11	87	6	11	94.1%	0.115	1	4	0
GH010358_7_2490_3390	97.0%	96.7%	96.7%	98.2%	98.9%	4	3	1	0	21	34	9	8	96.7%	0.060	0	2	0
GH010358_6_10208_11900	78.2%	78.1%	78.0%	99.2%	99.8%	4	4	0	0	6	26	14	13	98.6%	0.046	3	6	0
GH010373_6_3150_4744	99.5%	99.7%	99.4%	99.5%	99.8%	4	3	1	0	5	15	2	2	99.3%	0.046	0	2	0
GH010373_5_1284_2724	95.8%	95.7%	95.3%	98.4%	99.4%	6	6	0	0	19	52	20	16	97.3%	0.055	6	4	0
OVERALL	89.1%	89.3%	88.7%	98.5%	99.6%	24	21	3	0	62	220	91	50	97.5%	0.059	50	18	0

10:43:19 INFO - Completed

Figure B-16: GS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	92.9%	97.0%	89.2%	91.9%	99.9%	3	2	1	0	1	90	38	12	88.4%	0.167	36	0	0
GH010354_5_17718_19366	53.4%	56.0%	51.0%	86.3%	94.7%	3	1	0	2	84	240	8	24	81.1%	0.214	1	4	1
GH010358_7_2490_3390	97.4%	99.1%	95.1%	94.7%	99.4%	4	2	2	0	11	102	6	18	93.8%	0.089	0	3	0
GH010358_6_10208_11900	62.9%	64.7%	60.9%	93.1%	99.5%	4	3	0	1	16	231	21	42	92.0%	0.136	4	9	1
GH010373_6_3150_4744	24.9%	32.2%	20.3%	62.4%	99.3%	4	1	1	2	14	1218	25	29	61.2%	0.159	6	15	0
GH010373_5_1284_2724	49.1%	63.7%	39.7%	61.0%	98.9%	6	0	3	3	23	1295	25	77	59.6%	0.164	5	8	2
OVERALL	58.7%	66.0%	52.6%	78.4%	98.7%	24	9	7	8	149	3176	123	202	76.6%	0.151	52	39	4

10:25:28 INFO - Completed

Figure B-17: Y3DS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.0%	96.5%	95.5%	98.2%	99.2%	3	2	1	0	9	20	40	5	93.8%	0.159	40	0	0
GH010354_5_17718_19366	35.7%	36.6%	34.8%	89.9%	94.4%	3	1	1	1	93	177	16	16	83.7%	0.210	2	5	1
GH010358_7_2490_3390	97.2%	97.7%	96.0%	97.2%	99.6%	4	3	1	0	7	55	2	16	96.7%	0.094	0	2	0
GH010358_6_10208_11900	79.0%	79.9%	77.7%	96.2%	99.4%	4	2	2	0	20	129	8	30	95.3%	0.133	1	4	0
GH010373_6_3150_4744	48.8%	50.9%	46.9%	91.7%	99.6%	4	2	1	1	12	268	16	41	90.9%	0.141	2	8	0
GH010373_5_1284_2724	50.5%	59.1%	43.8%	72.0%	97.8%	6	1	3	2	53	932	40	84	69.2%	0.140	5	10	1
OVERALL	65.0%	68.1%	61.9%	89.3%	98.5%	24	11	9	4	194	1581	122	192	87.1%	0.142	50	29	2

10:29:06 INFO - Completed

Figure B-18: Y4DS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.8%	96.9%	96.8%	99.5%	99.6%	3	3	0	0	4	6	40	0	95.5%	0.062	40	0	0
GH010354_5_17718_19366	42.3%	42.4%	42.1%	94.8%	95.5%	3	1	2	0	79	92	12	13	89.6%	0.117	3	8	1
GH010358_7_2490_3390	96.5%	96.3%	96.0%	97.8%	98.7%	4	2	2	0	24	43	13	8	95.9%	0.057	1	4	0
GH010358_6_10208_11900	77.7%	77.0%	78.1%	98.8%	97.9%	4	3	1	0	72	39	16	17	96.2%	0.047	2	7	0
GH010373_6_3150_4744	99.2%	99.4%	99.0%	99.4%	99.7%	4	3	1	0	9	19	3	4	99.0%	0.047	0	3	0
GH010373_5_1284_2724	65.6%	64.7%	66.1%	97.5%	96.1%	6	5	1	0	132	83	33	33	92.5%	0.057	8	8	0
OVERALL	79.4%	79.0%	79.5%	98.1%	97.8%	24	17	7	0	320	282	117	75	95.1%	0.060	54	30	1

10:33:09 INFO - Completed

Figure B-19: FDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.9%	96.9%	97.0%	99.7%	99.6%	3	3	0	0	5	3	40	0	95.7%	0.048	40	0	0
GH010354_5_17718_19366	64.1%	65.1%	63.2%	95.1%	97.9%	3	2	1	0	36	86	6	9	92.7%	0.100	1	5	0
GH010358_7_2490_3390	96.9%	97.0%	96.2%	98.3%	99.8%	4	3	0	1	4	33	15	9	97.3%	0.048	1	6	0
GH010358_6_10208_11900	78.3%	77.5%	78.7%	98.9%	97.9%	4	3	1	0	72	38	10	16	96.4%	0.041	1	5	0
GH010373_6_3150_4744	99.4%	99.4%	99.4%	99.6%	99.6%	4	3	1	0	13	13	1	2	99.2%	0.049	0	1	0
GH010373_5_1284_2724	71.1%	71.2%	70.6%	96.9%	98.4%	6	4	2	0	54	102	23	25	94.6%	0.052	6	9	0
OVERALL	83.5%	83.5%	83.3%	98.1%	98.7%	24	18	5	1	184	275	95	61	96.2%	0.054	49	26	0

Figure B-20: MDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	96.8%	96.8%	96.8%	99.5%	99.5%	3	2	1	0	5	6	40	1	95.4%	0.071	40	0	0
GH010354_5_17718_19366	63.1%	63.1%	63.1%	95.4%	95.5%	3	1	2	0	79	80	4	10	90.7%	0.112	1	4	1
GH010358_7_2490_3390	97.5%	97.4%	96.9%	98.3%	99.5%	4	3	1	0	9	32	9	12	97.4%	0.063	0	4	0
GH010358_6_10208_11900	97.0%	96.4%	97.2%	98.6%	98.3%	4	3	1	0	58	48	7	22	96.6%	0.058	1	5	0
GH010373_6_3150_4744	99.6%	99.8%	99.5%	99.7%	100.0%	4	3	1	0	1	10	1	2	99.6%	0.069	0	1	0
GH010373_5_1284_2724	79.4%	79.2%	79.2%	97.5%	98.2%	6	4	2	0	60	82	18	27	95.2%	0.072	4	7	0
OVERALL	89.6%	89.5%	89.5%	98.2%	98.6%	24	16	8	0	212	258	79	74	96.3%	0.072	46	21	1

10:36:59 INFO - Completed

Figure B-21: RDS detail results on Micand32E.

	IDF1	IDP	IDR	Rcll	Prcn	GT	MT	PT	ML	FP	FN	IDs	FM	MOTA	MOTP	IDt	IDa	IDm
GH010358_8_8000_8547	97.2%	97.3%	97.0%	99.7%	100.0%	3	3	0	0	0	3	40	0	96.1%	0.048	40	0	0
GH010354_5_17718_19366	64.1%	64.8%	63.5%	97.7%	99.6%	3	3	0	0	7	41	5	7	97.0%	0.096	0	5	0
GH010358_7_2490_3390	97.6%	97.4%	97.2%	99.0%	99.9%	4	3	1	0	1	19	9	8	98.5%	0.050	0	2	0
GH010358_6_10208_11900	81.4%	81.3%	81.1%	99.3%	100.0%	4	4	0	0	0	24	8	13	99.0%	0.038	1	3	0
GH010373_6_3150_4744	99.7%	99.8%	99.5%	99.8%	100.0%	4	4	0	0	0	8	1	1	99.7%	0.048	0	1	0
GH010373_5_1284_2724	89.1%	89.1%	88.7%	98.8%	99.9%	6	6	0	0	4	40	19	14	98.1%	0.051	6	4	0
OVERALL	88.5%	88.5%	88.1%	99.1%	99.9%	24	23	1	0	12	135	82	43	98.4%	0.052	47	15	0

10:41:24 INFO - Completed

Figure B-22: GDS detail results on Micand32E.

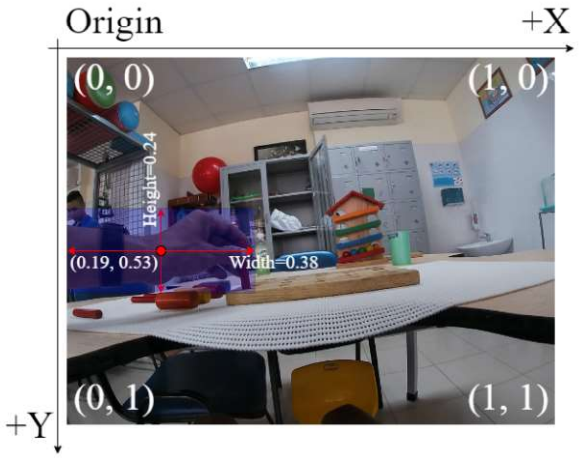

Figure B-23: Illustration of YOLO's data format.

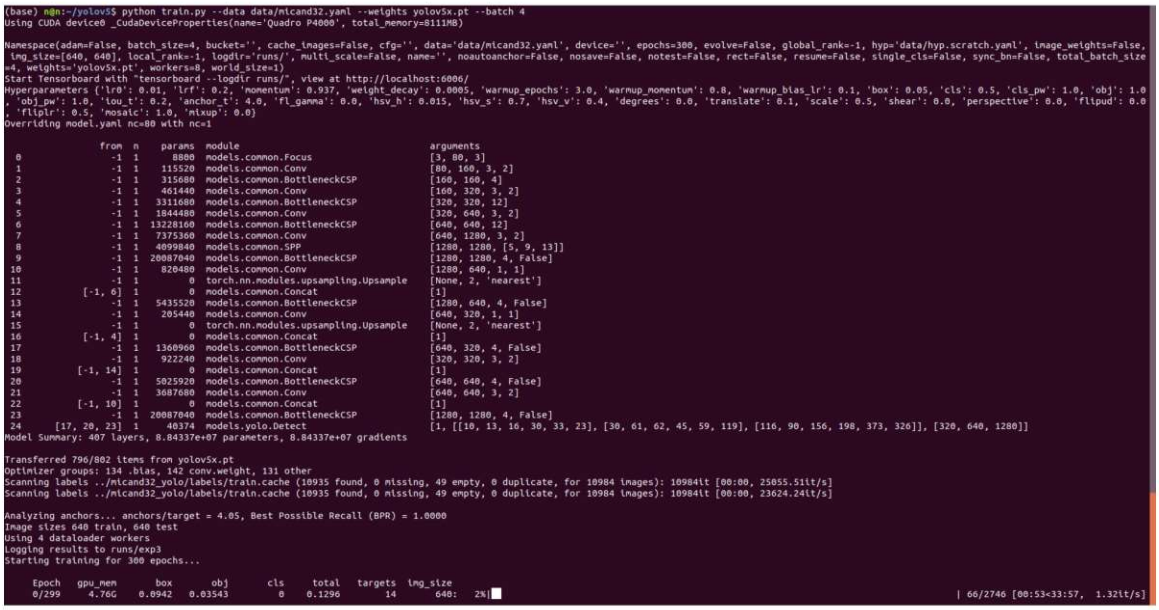

Figure B-24: Illustration of YOLO's training process.

Bibliography

[1] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[2] R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.

[3] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[4] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[5] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[6] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, (USA), p. 1949–1957, IEEE Computer Society, 2015.

[7] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017.

[8] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, (USA), p. 407–414, IEEE Computer Society, 2011.

[9] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *CVPR 2011*, pp. 3281–3288, 2011.

[10] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1346–1353, 2012.

[11] R. Yonetani, K. M. Kitani, and Y. Sato, "Recognizing micro-actions and reactions from paired egocentric videos," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2629–2638, 2016.

[12] A. Doherty, S. Hodges, A. King, A. Smeaton, E. Berry, C. Moulin, S. Lindley, P. Kelly, and C. Foster, "Wearable cameras in health: The state of the art and future possibilities," *American journal of preventive medicine*, vol. 44, pp. 320–3, 03 2013.

[13] D. Townsend, F. Knoefel, and R. Goubran, "Privacy versus autonomy: A tradeoff model for smart home monitoring technologies," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4749–4752, 2011.

[14] T. Leelasawassuk, D. Damen, and W. Mayol-Cuevas, "Automated capture and delivery of assistive task guidance with an eyewear computer: The glaciar system," in *Proceedings of the 8th Augmented Human International Conference*, AH '17, (New York, NY, USA), Association for Computing Machinery, 2017.

[15] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," *CoRR*, vol. abs/1602.00763, 2016.

[16] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Čehovin, "A novel performance evaluation methodology for single-target trackers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 2137–2155, Nov 2016.

[17] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016.

[18] Y. Li, Zhefan Ye, and J. M. Rehg, "Delving into egocentric actions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 287–295, 2015.

[19] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1894–1903, 2016.

[20] G. Bertasius, H. S. Park, S. X. Yu, and J. Shi, "First person action-object detection with egonet," *CoRR*, vol. abs/1603.04908, 2016.

[21] M. Cai, K. M. Kitani, and Y. Sato, "Understanding hand-object manipulation with grasp types and object attributes.," in *Robotics: Science and Systems*, vol. 3, Ann Arbor, Michigan;, 2016.

[22] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," *CoRR*, vol. abs/1806.06157, 2018.

[23] C. Li and K. M. Kitani, "Pixel-level hand detection in ego-centric videos," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3570–3577, 2013.

[24] A. Betancourt, M. M. Lopez, C. S. Regazzoni, and M. Rauterberg, "A sequential classifier for hand detection in the framework of egocentric vision," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 600–605, 2014.

[25] A. Betancourt, P. Morerio, E. Barakova, L. Marcenaro, M. Rauterberg, and C. Regazzoni, "Left/right hand segmentation in egocentric videos," *Comput. Vis. Image Underst.*, vol. 154, p. 73–81, Jan. 2017.

[26] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1949–1957, 2015.

[27] T. Nguyen, J.-C. Nebel, and F. Flórez-Revuelta, *Recognition of Activities of Daily Living from Egocentric Videos Using Hands Detected by a Deep Convolutional Network*, pp. 390–398. 06 2018.

[28] G. Kapidis, R. Poppe, E. Van Dam, L. Noldus, and R. Veltkamp, "Egocentric hand track and object-based human action recognition," in *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 922–929, 2019.

[29] V. T. Pham, T. L. Le, T. H. Tran, and T. P. Nguyen, "Hand detection and segmentation using multimodal information from kinect," in *2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6, 2020.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, p. 84–90, May 2017.

[31] K. S. Chahal and K. Dey, "A survey of modern object detection literature using deep learning," *CoRR*, vol. abs/1808.07256, 2018.

[32] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.

[33] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1611.10012, 2016.

[34] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *2011 International Conference on Computer Vision*, pp. 1879–1886, 2011.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[39] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *CoRR*, vol. abs/1808.01974, 2018.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014.

[41] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016.

[42] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[43] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[45] A. Bochkovskiy, C. Wang, and H. Liao, "Yolov4: Optimal speed and accuracy of object detection. arxiv 2020," *arXiv preprint arXiv:2004.10934*, pp. 1–17, 2020.

[46] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics, Springer, 2009.

[47] A. Fathi, Y. Li, and J. Rehg, "Learning to recognize daily actions using gaze," vol. 7572, pp. 314–327, 10 2012.

[48] Y. Li, M. Liu, and J. M. Rehg, "In the eye of the beholder: Gaze and actions in first person video," *arXiv preprint arXiv:2006.00626*, 2020.

[49] A. Dutta and A. Zisserman, "The via annotation software for images, audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, (New York, NY, USA), p. 2276–2279, Association for Computing Machinery, 2019.

[50] A. Bandini and J. Zariffa, "Analysis of the hands in egocentric vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[51] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," *CoRR*, vol. abs/1812.00442, 2018.

[52] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[53] G. Jocher, A. Stoken, and J. Borovec, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements." `https://github.com/ultralytics/yolov5`, 2020.

[54] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[55] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017.

[56] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6568–6577, 2019.

[57] M. Tan, R. Pang, and Q. Le, "Efficientdet: Scalable and efficient object detection. arxiv 2019," *arXiv preprint arXiv:1911.09070*.

[58] X. Hou, Y. Wang, and L. Chau, "Vehicle tracking using deep sort with low confidence track filtering," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2019.

[59] O. Shigeta, S. Kagami, and K. Hashimoto, "Identifying a moving object with an accelerometer in a camera view," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3872–3877, 2008.

[60] V. Kiselev, M. Khlamov, and K. Chuvilin, "Hand gesture recognition with multiple leap motion devices," in *2019 24th Conference of Open Innovations Association (FRUCT)*, pp. 163–169, 2019.