FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Advancing Microrobotics for Biomedical Applications through Machine Learning
Student:	Bc. Daniil Pastukhov
Supervisor:	Mgr. Alexander Kovalenko, Ph.D.
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

Czech Technical University (CTU) and Twente University are collaborating to investigate the potential of machine learning (ML) in advancing micro-robotics, particularly in the context of biological microrobots for drug delivery applications.

Objective:

The primary objective of this thesis is to explore the use of ML techniques in understanding and controlling biological microrobots, such as sperm cells and MagnetoSperm, as well as improving the detection and external control of microrobots like micro-screws in the bloodstream.

Research Areas:

Dynamics of Sperm Cells as Microrobots: The thesis will apply ML techniques, including image processing, object detection, object tracking, next-frame prediction, and symbolic regression, to better comprehend the dynamics of sperm cells for potential microrobot applications in targeted drug delivery.

MagnetoSperm Microrobots: The research will extend ML techniques to MagnetoSperm microrobots, sperm cells coated with magnetic nanoparticles controllable via a weak external magnetic field. By analyzing and modeling their behavior, the aim is to develop

---

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 10 May 2023 in Prague.

accurate and efficient control methods for improved biomedical applications.

Detection of Microscrews and Other Microrobots: The thesis will explore the role of ML in detecting and tracking microrobots like micro-screws within the body, which is essential for their safe and effective use in medical procedures.

External Control of Microrobots in the Bloodstream: The research will focus on developing ML algorithms for improved external control of microrobots navigating the complex arterial structures within the bloodstream, enhancing precision and safety.

To successfully complete the diploma thesis the following research steps should be taken:

- Conduct a literature review on the fundamentals of computer vision (landmark detection, object detection, object tracking, image augmentation), machine learning-based motion control, and deep symbolic regression;
- Preprocess the data obtained from scientific partners;
- Train, evaluate, and optimize landmark detection model;
- From the obtained landmarks evaluate microswimmers' dynamics;
- Train, evaluate, and optimize object detection model;
- Train, evaluate, and optimize trajectory prediction model for better navigation in the bloodstream;
- Explore the possibilities of deep symbolic regression on the data to extract governing equations;

Literature:
- Khalil, Islam SM, et al. "MagnetoSperm: A microrobot that navigates using weak magnetic fields." Applied Physics Letters 104.22 (2014): 223701.
- Magdanz, Veronika, et al. "IRONSperm: Sperm-templated soft magnetic microrobots." Science Advances 6.28 (2020): eaba5855.
- Zhang, Kaixuan, et al. "Locomotion of bovine spermatozoa during the transition from individual cells to bundles." Proceedings of the National Academy of Sciences 120.3 (2023): e2211911120.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master's thesis

Advancing Microrobotics for Biomedical Applications through Machine Learning

Bc. Daniil Pastukhov

Department of Department of Applied Mathematics
Supervisor: Mgr. Alexander Kovalenko, Ph.D.

January 11, 2024

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on January 11, 2024 . . . . . .

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Pastukhov, Daniil. *Advancing Microrobotics for Biomedical Applications through Machine Learning*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Abstrakt

Tato práce se zabývá integrací technik strojového učení do mikrorobotiky se zaměřením na biologické mikroroboty využívající jako platformu spermie. Šetření zahrnuje podrobnou analýzu relevantních prací v oblasti mikrorobotiky a strojového učení v biomedicínském kontextu, čímž jsou položeny základy pro mnohostranné zkoumání. Mezi klíčové příspěvky patří kurátorství a anotace datových souborů přizpůsobených pro trénování a vyhodnocování modelů. Byly vyvinuty a zváženy modely detekce objektů pro přesnou identifikaci spermií a jejich hlavic, zatímco model odhadu klíčových bodů byl použit pro detekci klíčových bodů bičíků. Kromě toho byl implementován a vyhodnocen systém sledování objektů pro sledování dynamických pohybů hlaviček spermatických buněk, což zlepšuje pochopení jejich interakcí v dynamickém prostředí. Dále byl vycvičen a vyhodnocen model pro předpovídání trajektorie. Tato studie představuje významný pokrok v integraci strojového učení a mikrorobotiky a nabízí inovativní perspektivy a přístupy, které lze využít v různých biomedicínských a technologických oblastech. Práce přispívá k současnému chápání biologických mikrorobotů a pokládá základy pro budoucí pokrok, odkrývá potenciál pro přesné řídicí mechanismy a rozšiřuje aplikace v různých oblastech.

Klíčová slova mikroroboti, strojové učení, detekce objektů, sledování objektů, predikce trajektorie

Abstract

This thesis explores the integration of machine learning techniques in microrobotics, focusing on biological microrobots utilizing sperm cells as a platform. The investigation includes a detailed analysis of relevant works in microrobotics and machine learning in the biomedical context, laying the groundwork for a multifaceted exploration. Key contributions include curating and annotating datasets tailored for training and evaluating models. Object detection models were developed and considered for precisely identifying sperm cells and their heads, while a keypoint estimation model was employed to detect flagellum keypoints. Additionally, an object-tracking system was implemented and evaluated to track the dynamic movements of sperm cell heads, enhancing the understanding of their interactions in dynamic environments. Further, a trajectory prediction model was trained and evaluated. This study marks a notable advancement in the integration of machine learning and microrobotics, offering innovative perspectives and approaches that can be utilized in various biomedical and technological fields. The work contributes to the current understanding of biological microrobots and lays the foundation for future advancements, unlocking the potential for precise control mechanisms and expanding applications in various fields.

Keywords microrobots, machine learning, object detection, object tracking, trajectory prediction

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The field of microrobotics has the potential to revolutionize medical treatments by enabling precise and targeted drug delivery. The microrobots offer unique advantages due to their natural motility and ability to navigate complex environments. However, controlling and directing these microrobots in a controlled manner remains a challenge.

Machine learning (ML) has emerged as a powerful tool for addressing this challenge. ML algorithms can be used to extract patterns from complex data, enabling the development of intelligent control strategies for biological microrobots. In [1], the authors provide a comprehensive overview of the most relevant works in microrobotics and machine learning in biomedical robotics. Different energy sources, such as magnetic fields, acoustic waves, light, and chemical reactions, can control microrobots. In this thesis, we experiment with self-propulsive and magnetically controlled microrobots, specifically IRON-Sperm [2], which are based on sperm cells. The sperm cells are propelled by their flagellum, a long, whip-like structure attached to the sperm cell's head. Given the data's complexity and the microrobots' nature, ML techniques can significantly contribute to understanding and controlling biological microrobots.

1.2 Problem Statement

Flagella represent whip-like appendages that are widespread across a diverse spectrum of organisms, ranging from single-celled bacteria like *Escherichia coli* and *Salmonella to archaea* and extending to eukaryotic species, including certain algae and protozoans such as *Giardia lamblia*. Notably, flagella serve as distinctive features on male gametes in multicellular organisms, playing a crucial role as facilitators of motility. Often compared to an outboard motor, the flagellum is the primary driving force for an organism's propulsion, con-

tributing significantly to essential activities like nutrient acquisition, predator evasion, and maintaining optimal environmental conditions for survival.

Flagella may function as biological templates in micro-robotics, offering inspiration for creating micro-robots proficient in precise navigation within fluidic environments. This holds significant importance for precision medicine, especially in drug delivery, as it opens avenues for developing targeted treatment approaches that minimize side effects and enhance therapeutic effectiveness. The study of flagellar dynamics can contribute to biophysical models of cellular motility and advance nanotechnologies for applications in both environmental and industrial settings.

Consequently, an in-depth exploration of the dynamics of these systems is crucial not only for enhancing our comprehension of biological systems but also for catalyzing progress across diverse domains, including medicine, technology, and environmental science.

This thesis addresses the challenge of developing a scalable approach for detecting the flagellar dynamics of sperm cells in real time. Our approach involves utilizing a Deep Learning (DL) model to detect cells and flagellum landmarks and dynamically approximate the parameters required to compute wave propagation along flagella.

To enhance the efficiency of our proposed solution, we have implemented an object-tracking system that ensures the continuous monitoring of microrobots and their flagella in a dynamic environment. Object tracking enables the accurate tracking of individual cells over time, providing a comprehensive understanding of their movements and interactions.

Furthermore, we have integrated a trajectory prediction model into our framework to forecast the future positions of microorganisms based on their historical movement patterns. This predictive capability is instrumental in anticipating the trajectory of flagella, allowing for proactive adjustments in imaging and analysis strategies. The trajectory prediction model enhances the real-time nature of our approach, contributing to a more robust and adaptive system for studying flagellar dynamics.

1.3 Contribution

The main contribution of this thesis is as follows:

- Overviewed most relevant works in the field of microrobotics and machine learning in biomedical robotics;
- Prepared and annotated dataset for training and evaluating object detection, keypoint estimation, and trajectory prediction models. The dataset will also be used for further research by the biologists and engineers;
- Trained and evaluated object detection models for detecting sperm cells and sperm cell heads;

- Trained and evaluated keypoint estimation model for detecting flagellum keypoints;
- Trained and evaluated trajectory prediction model for better navigation in the bloodstream;

Chapter 2

Literature Review

This chapter will review the most relevant works in microrobotics and machine learning in biomedical robotics.

2.1 Microrobotics in Biomedical Applications

Microrobots are a promising tool for biomedical applications, such as targeted drug delivery [3, 4], minimally invasive surgery [5], and cell manipulation [6].

Robots can be classified into two main categories: soft [7] and rigid [8]. Soft robots are made of flexible or extendible materials, such as polymers, gels, and elastomers [9, 3, 10]. Rigid robots, on the other hand, are made of rigid materials, such as metals and ceramics [11].

Both soft and rigid robots have advantages and disadvantages for biomedical applications. Soft robots are more flexible and can be used for tasks that require high dexterity, such as cell manipulation or drug delivery. However, they are also more challenging to control, as they are subject to various physical phenomena. Rigid robots, on the other hand, are easier to control, but they are also more challenging to manufacture, and they are less flexible.

Biologically inspired microrobots encompass microrobots with joints, soft segments, and continuous bodies inspired by natural organisms, while biohybrid microrobots include sperm-based microrobots, bacteria-driven microrobots, and algal microrobots.

The potential of microrobots for biomedical applications has been demonstrated in several works.

In [2], the authors have demonstrated the potential of soft bio-microrobots for targeted drug delivery and minimally invasive surgery. They have developed IRONspems, bio-hybrid microrobots made of sperm cells and iron nanoparticles. The sperm cells are used for propulsion, while the iron nanoparticles are used for steering the microrobots using magnetic fields. This allows us to control and guide the microrobots to the target location remotely.

In [1], the authors have comprehensively overviewed the most relevant works in microrobotics and machine learning in biomedical robotics. They explore various types of microrobots, their applications, and the associated challenges. As highlighted in the study above, soft bio-microrobots exhibit remarkable capabilities in targeted drug delivery. These microrobots can navigate with precision, leveraging tactics inspired by microorganisms or cells, and interact with their surroundings, allowing them to deliver therapeutic drugs to specific locations within the body. The versatility of soft bio-microrobots extends to performing biopsies by mimicking the motion and function of natural creatures. This emulation allows them to sample tissues, showcasing their potential in medical diagnostics effectively. Furthermore, soft bio-microrobots contribute to biofilm eradication, addressing a significant concern in various biomedical and industrial settings. Their ability to remove biofilms highlights their potential to maintain hygiene and prevent complications in diverse applications.

Another study describes the potential of microrobots' use in vitro fertilization (IVF) [12]. In a conventional IVF procedure, an ovum undergoes fertilization extracorporeally, and the resultant embryo is subsequently implanted into the uterus. This procedure frequently encounters challenges leading to failure. However, envisioning a scenario where microbots transport the embryo back to the fallopian tube or endometrium could offer a more conducive environment for embryonic development, thereby enhancing implantation rates. The authors believe that using microrobots guided by magnetic fields, proficient in gripping or carrying the embryo, can be a promising solution to the problem of implantation failure.

2.2 Machine Learning in Biomedicine

This section will review the most relevant literature on machine learning in medicine and biomedical engineering.

Machine learning methods are usually divided into three main categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning methods are used for tasks where the data is labeled, e.g., in classification and regression. Such models as linear regression [13], logistic regression [14], support vector machines [15], and random forests [16] are commonly used for supervised learning tasks. The applications of supervised learning in medicine include disease diagnosis [17], drug discovery [18], and medical image analysis [19].

While supervised methods are used for tasks where the data is labeled, unsupervised ones are used for tasks where the data is unlabeled. Popular unsupervised learning methods include k-means clustering [20], principal component analysis (PCA) [21], and autoencoders [22]. In biomedicine, unsupervised learning methods are used for tasks such as synthetic data generation

[23] and dimensionality reduction of the RNA sequences [24].

Reinforcement learning (RL) is a machine learning type involving an agent interacting with an environment and learning to perform a task by maximizing a reward signal.

RL has been successfully applied to various tasks, such as game playing [25], robotics [26], and biomedical research [27]. Recently, RL has been actively used in the field of microrobotics.

The authors of [28] have employed a deep RL-based approach to enable microswimmers to navigate towards specific targets. The AI-powered microswimmer is trained to achieve targeted navigation and to switch between distinct locomotory gaits (steering, transition, and translation). They demonstrate that the AI-recommended strategy remains resilient in the face of flow perturbations and possesses versatility, allowing the swimmer to execute intricate tasks like path tracing without explicit programming. Their findings underscore the extensive potential of AI-powered swimmers for use in unpredictable and complex fluid environments.

In [29], the authors also employ a deep RL approach to control microrobots' locomotion. They have built a physical, biomimetic, and fluidic arena with multidimensional magnetic actuation and deployed a helical agar magnetic robot. The robot was tasked with swimming in a clockwise direction through a fluid-filled lumen in an arena under the control of a nonuniform rotating magnetic field generated by a three-axis array of electromagnetic coils. The goal of the RL algorithm was to learn a policy that would manipulate the shape and magnitude of the magnetic field to guide the robot to the target location. Soft Actor-Critic (SAC) algorithm was used for training the robot with a reward function that penalized the robot for deviating from the desired trajectory (i.e., the angle between the robot's orientation and the desired orientation) and for moving too slowly. After training for 100,000 steps, the resulting model successfully navigated the robot to the target location.

2.3 Machine Learning Concepts Overview

This section will introduce the most relevant concepts in machine learning and discuss the most common approaches used for different tasks. In the next chapter, we provide more details about the methods used in our work.

2.3.1 Object Detection

Object detection is a computer vision task that involves detecting objects in an image or video and determining their location. In biomedical engineering, there are plenty of applications for object detection, such as cell detection [30] and detection of anatomical structures in medical images [31].

There are different approaches that can be divided into two main categories: traditional computer vision methods and deep learning methods.

Traditional methods leverage hand-crafted features and machine learning algorithms, such as support vector machines (SVM) and random forests (RF), to detect objects in images. Often, these methods utilize a sliding window approach, where a fixed-size window is moved across the image, and a classifier is used to determine whether the window contains an object. This approach is computationally expensive, requiring the classifier to be evaluated for every window in the image. Moreover, it could be more accurate and robust to changes in scale and orientation.

Deep learning methods are based on deep neural networks (DNNs) and are much more accurate and robust. In computer vision, convolutional neural networks (CNNs) [32] are the most commonly used types of DNNs. CNNs are based on the concept of convolution. Convolution is a mathematical operation that takes two functions, f and g, and produces a third function, h, that represents how the shape of f is modified by g. In computer vision, the input image is convolved with a set of filters, also known as kernels, to produce a feature map. The filters are learned during the training process, allowing the model to learn the most relevant features for the task. The convolution operation is usually followed by a non-linear activation function, e.g., ReLU, and a pooling operation, which reduces the dimensionality of the feature map. The resulting feature map is fed into a fully connected layer, producing the final output. For object detection, the output is a set of bounding boxes and class probabilities for each object in the image. At the time of writing this thesis, state-of-the-art object detection models include the following:

- **You Only Look Once** (YOLO) [33]: YOLO is a family of object detection models based on DarkNet architecture. The main concept behind YOLO is that it divides the input image into a grid of cells and predicts bounding boxes and class probabilities for each cell. This approach is much faster than the sliding window approach, which was used previously, as it only needs to evaluate the classifier for each cell instead of each window. Moreover, YOLO is more accurate and robust to changes in scale and orientation.
- **Visual Transformer** (ViT) [34]: ViT is a transformer-based model that processes images as a sequence of patches, then serialized into vectors and fed into a transformer encoder, similar to how transformers process tokens in natural language processing. It allows the model to capture local and global features of the image, enabling it to achieve great performance on different tasks.
- **Swin Transformer** [35]: Swin Transformer has been proposed to improve ViT. It introduces two key concepts to address the issues faced by ViT: shifted window attention and patch merging. The attention mechanisms operate over a series of shifted windows of different sizes, allowing the model to attend to different parts of the image at different scales and

better capture spatial information. The hierarchical structure allows the output of the shifted window attention mechanisms at each scale to be passed through a patch merging layer, which builds hierarchical feature maps by merging image patches in deeper layers.

2.3.2 Landmark Detection

Landmark detection, also known as pose estimation, is a crucial task in the realm of computer vision, aiming to identify and precisely locate significant points of interest within an image. As technology continues to advance, applications of landmark detection have become increasingly diverse, ranging from human pose estimation in video analytics to facial landmark localization in facial recognition systems [36, 37].

It is expected to distinguish between two primary approaches: bottom-up and top-down.

Bottom-up Approach involves the initial detection of all keypoints in an image, followed by grouping these points into coherent objects. This method is particularly advantageous in scenarios with varying occlusions and scale changes, as it prioritizes identifying individual keypoints independently of the overall object context. Despite its robustness, the bottom-up approach tends to be computationally expensive, requiring substantial processing power and time.

Top-down Approach follows a different strategy. It begins by detecting objects in an image and identifying ke points associated with each recognized object. This method often proves to be faster than its bottom-up counterpart, as it leverages the knowledge of object presence to guide the subsequent landmark detection. Moreover, the top-down approach can exhibit increased resilience to occlusions and scale changes.

For our task, we have chosen a top-down approach over bottom-up due to its better suitability for handling overlapping keypoints in sperm cell analysis. The top-down method, focusing on object detection before keypoint identification, proves more adept at navigating the challenges posed by overlapping structures. In contrast, the bottom-up approach may struggle with accurately selecting individual keypoints in such scenarios, potentially leading to reduced performance.

2.3.3 Object tracking

Object tracking is a computer vision task that involves identifying and locating an object in a sequence of video frames. Different approaches can be employed for object tracking, which can be broadly categorized into traditional computer vision methods and deep learning-based methods.

Several challenges must be addressed when tracking an object in a video sequence. Firstly, the appearance of an object may change over time due to factors such as lighting variations, occlusions, and deformations. Secondly, the object may enter or exit the frame or be partially or fully occluded by other objects, necessitating the tracker's ability to re-identify the object. Thirdly, the shape of an object may change due to pose variations or deformations. Finally, the object's motion may be non-linear, requiring the tracker to employ non-linear motion estimation techniques.

Motion-based methods utilize the object's movement pattern to predict its location in subsequent frames. These methods typically employ statistical models to capture the object's motion characteristics, such as velocity and acceleration. One prominent example is the Kalman filter, a recursive Bayesian filter that estimates the object's state, including its position, velocity, and acceleration. By incorporating motion constraints and measurements from successive frames, the Kalman filter effectively tracks the object's trajectory, even in the presence of noise and occasional occlusions.

Another popular motion-based method is the particle filter, representing the object's state as a set of weighted particles. Each particle represents a possible state of the object, and the filter dynamically updates the particle weights based on new observations and motion constraints. This probabilistic approach allows the particle filter to handle uncertainty and effectively track objects with non-linear motion patterns.

Feature-based methods rely on extracting distinct visual features from the object's appearance in the first frame and then matching these features to subsequent frames to locate the object. These features can be based on various image properties, such as edges, corners, texture patterns, or edges, corners, texture patterns, or deep learning features extracted from the model's intermediate layers. Once the features are extracted, various techniques can be employed to match these features in subsequent frames and track the object's movement:

- **Correlation filters** [38]: Correlation filters are a powerful and efficient approach for feature-based tracking. They utilize a filter to match the object's appearance in the first frame. As the object moves, the filter is updated to adapt to the changing appearance, enabling the tracker to maintain its location even when partially or fully occluded.
- **Mean-shift algorithms** [39]: Mean-shift algorithms employ a density estimation technique to find the region in the image that most closely resembles the object's appearance in the first frame. This technique effectively tracks objects with significant appearance changes due to illumination variations or object deformations.

- **Histogram of Oriented Gradients (HOG)** [40]: HOG is a feature descriptor that captures an image's distribution of intensity gradients. It is often used with other tracking methods, such as correlation filters or mean-shift algorithms, to improve tracking performance.
- **Deep learning features**: Deep learning features extracted from a pre-trained CNN can be used for tracking. These features are more robust to appearance changes and occlusions than hand-crafted features, enabling the tracker to maintain the object's location even when partially or fully occluded.

The tracker methods also differ by the number of objects they can track: **single-object trackers** and **multi-object trackers**. Single-object trackers are designed to track a single object in a video sequence, while multi-object ones track multiple objects simultaneously.

2.3.4 Trajectory Prediction

Trajectory prediction is a problem that involves predicting the spatial trajectory of an object for a given time interval.

Numerous approaches have been developed for trajectory prediction, each with strengths and limitations. One common approach utilizes recurrent neural networks (RNNs) [41], precisely long short-term memory (LSTM) networks [42], to capture the sequential nature of motion patterns. LSTMs excel at learning long-range dependencies and handling temporal information. However, they suffer from difficulties in parallelizing calculations and may struggle to represent complex interactions between objects.

To address these challenges, CNNs have emerged as a promising alternative for trajectory prediction. CNNs excel at extracting spatial features from sequential data, enabling them to capture local patterns and relationships. This spatial sensitivity proves beneficial for trajectory prediction as it allows models to capture the spatial context of the moving object and its surroundings.

2.3.5 Image Pre-processing

Medical images are often noisy and contain artifacts, which makes it difficult to analyze them. Model training and inference are usually affected by the quality of the input data, so image pre-processing is an essential step in the pipeline.

Pre-processing can be divided into two main categories: image denoising and image enhancement.

Image Denoising usually relies on filtering the image with a low-pass filter, such as a Gaussian filter, median filter, or bilateral filter [43]. These filters

are simple and fast, but they can lead to information loss and blurring of the image, which can affect small details in the data.

Image Enhancement aims to improve the quality of the image. Different approaches can be used for image enhancement, including traditional and deep learning methods. Traditional methods usually involve applying histogram equalization [44], gamma correction, or other transformations to the image. Such methods enhance the image in terms of visual quality, but they do not upscale the image, so they do not improve the quality of the image in terms of information content.

On the other hand, deep learning methods can enhance the image regarding information content, as they can be used to upscale the image. Generative adversarial networks (GANs) [45] are a type of neural network that can be used for image enhancement. One of the most popular GANs for image enhancement is *Real-ESRGAN* [46], a state-of-the-art super-resolution model.

2.3.6 Data Augmentation

Data augmentation is critical in machine learning, especially in biomedical applications. This technique artificially expands the size of a dataset by applying various transformations to the existing data samples [47]. Its primary goal is to enhance the performance of machine learning models. Data augmentation helps against overfitting, which usually leads to poor performance on unseen data. By introducing variations in the training data, data augmentation forces the model to learn more generalizable features.

Additionally, data augmentation exposes the model to a broader range of data variations, making it more robust to noise, distortions, and anomalies that may arise in real-world applications. This robustness is crucial for biomedical data, where factors like imaging artifacts and variations in experimental conditions can influence data.

Chapter 3

Methodology

This chapter presents the methodology used to solve the problem described in Chapter 1. First, we talk about the data we used to train our models. Then, we describe the models we used to solve the problem. Finally, we discuss how we trained our models and the measures we used to evaluate their performance.

3.1 Dataset

Dataset Collection There are two datasets: one for training and validation and one for testing. All data was collected by the University of Waterloo researchers using a Mitutoyo MF-A4020D MF Series 176 Measuring Microscope 176–865–10. The microscope slides were positioned at a distance approximately equal to 12 times the length of a sperm cell, with water serving as the medium.

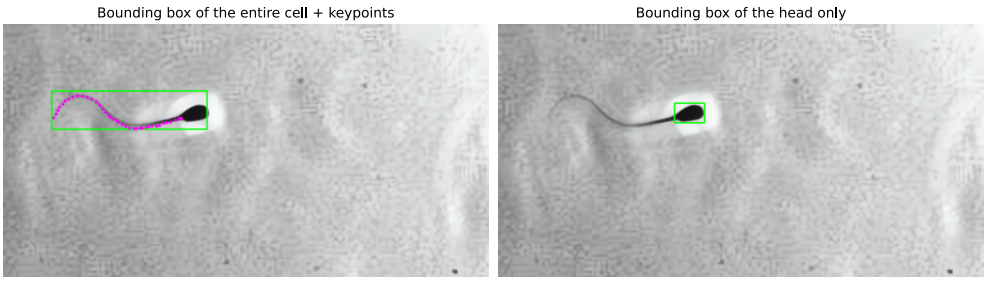


Figure 3.1: On the left, we can see the annotation for the object detection and keypoint estimation tasks. On the right, we can see the annotation for the object tracking and trajectory prediction tasks.

Dataset Content The first dataset is approximately 330MB, comprising 8 video samples featuring sperm cells. Each video consists of an average of

50 frames, with a single sperm cell captured in each frame. The recordings showcase bovine sperm cells observed through a microscope at 40x magnification. We have prepared two annotation types for this dataset: (1) bounding boxes for sperm cells and 40 keypoints for flagellum, and (2) bounding boxes for sperm cell heads. The annotation was done using CVAT[1], an open-source web-based application designed for annotating videos and images to support computer vision algorithms. Before manual annotation, the sperm cells were pre-annotated using Grounding-DINO [48] with Swin Transformer as the backbone, with *sperm cell* as a text prompt, 0.1 as the confidence threshold, and 0.5 as the IoU threshold. The first annotation type is used for training the object detection and keypoint estimation models, while the second type is used for object tracking and trajectory prediction model training. Figure 3.1 compares original and augmented data samples.

The second dataset is a collection of video samples featuring sperm cells and microrobots. There are 371 video samples featuring sperm cells and 5 video samples featuring microrobots. This dataset has no annotation, as its primary purpose is to assess the model's performance on unseen data and validate its strong generalization capabilities. We only use a small subset of this dataset for testing purposes.

We want to note that two phenomena can be observed in the data: (1) sperm cells sometimes form a bundle, and (2) sperm cell heads often appear overexposed. Sperm bundling arises due to a combination of hydrodynamic and adhesive interactions among the cells, often occurring with extended incubation periods [49]. The "glowing" effect of the spermatozoa head is caused by the change in the rotation of the spermatozoa head, resulting from the spermatozoa's flagellum movement. When the spermatozoa head rotates, the light reflected from the head changes, resulting in a "glowing" effect. Both phenomena are challenging to deal with, as they can lead to inaccurate results. Figure 3.2 and Figure 3.3 illustrate bundle formation and the "glowing" effect, respectively.

[1] `https://cvat.ai`

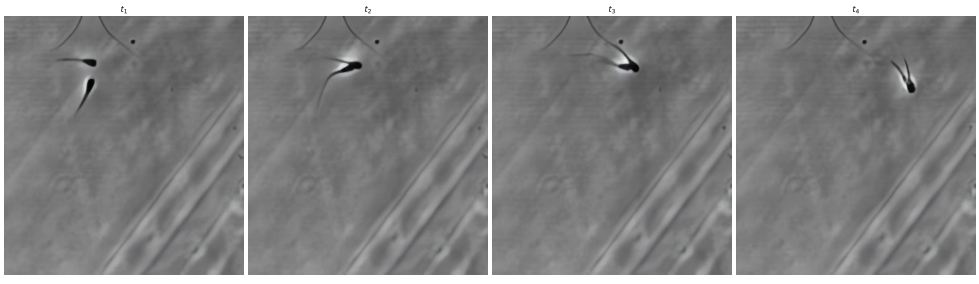


Figure 3.2: Chronological frames from a video sample featuring bundle formation.

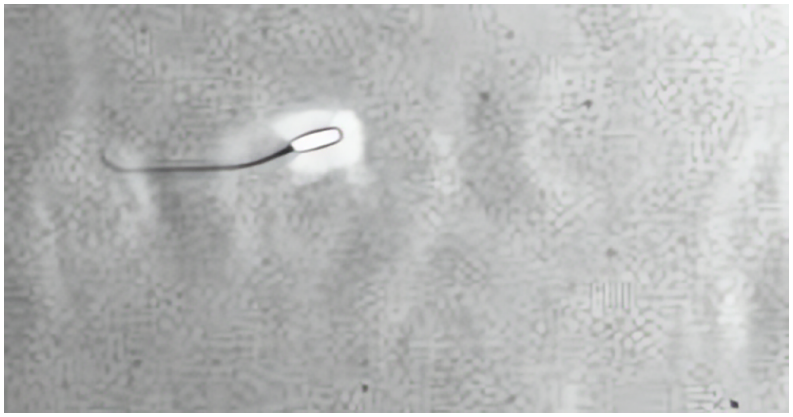


Figure 3.3: "Glowing" effect of the spermatozoa head.

Dataset Pre-processing Our dataset preparation and annotation processes follow a systematic workflow:

1. Each video sample undergoes upscaling and denoising through Real-ESRGAN [46].
2. Sperm cells are automatically annotated using Grounding-DINO [48].
3. Subsequently, bounding boxes are manually refined using the CVAT tool.
4. The finalized dataset is exported in the *CVAT for images 1.1* format and converted to YOLO format using *Datumaro* [2].

Data Splitting The training dataset was partitioned into training (80%) and validation (20%) sets, consisting of 6 and 2 video samples, respectively. The groups of subjects were split into different sets to ensure that the model would be trained on other subjects than it would be tested on. This was done to avoid data leakage and overfitting and ensure the model would generalize well to new data.

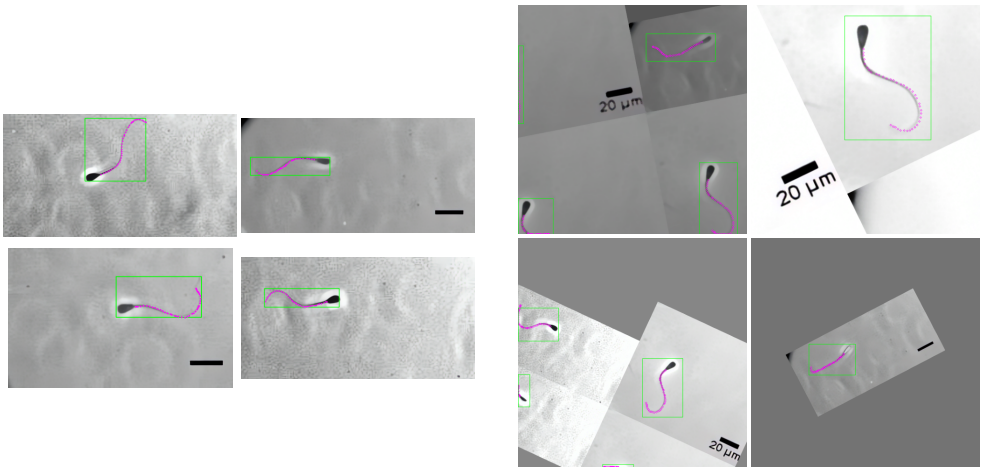
Data Augmentation We applied the following augmentations during YOLO model training:

- Apply mosaic augmentation with a probability of 0.85.
- Apply random rotation from -30 to 30 degrees with a probability of 0.5.
- Apply blur with a probability of 0.01.
- Apply median blur with a probability of 0.01.
- Apply CLAHE with a probability of 0.01.
- Apply random HSV shift (H: ±1.5%, S: ±70%, V: ±40%).
- Apply random horizontal and vertical flips with a probability of 0.5.

Furthermore, for trajectory prediction model training, we have used the following augmentations:

- Apply random 90-degree rotation with a probability of 0.5.

[2] `https://github.com/openvinotoolkit/datumaro`



(a) Ground truth data samples.

(b) Data samples with applied augmentations.

Figure 3.4: Ground truth vs augmented data samples. Bounding boxes and keypoints are visualized. Note: four stitched images in (b) are produced by Mosaic augmentation.

- Apply random horizontal and vertical flips with a probability of 0.5.
- Apply affine transformation (scale 0.5–0.5 and rotation -60–60 degrees) with a probability of 0.5.

No augmentations were applied during the testing processes.
Figure 3.4 shows an example of the applied augmentations.

3.2 Model Training

We have decided to leverage the YOLO model, specifically the YOLOv8 model, for our task detection and keypoint estimation tasks. The model can be trained in a multi-task fashion, which allows training both tasks simultaneously. This is beneficial, as it allows the features to be shared between the tasks and improves the model's performance.

3.2.1 YOLO architecture

YOLOv8 is a state-of-the-art object detection model based on DarkNet [50], Spatial Pyramid Pooling (SPP) [51], and Path Aggregation Network (PAN) [52], focusing on speed and accuracy. It is a single-stage model capable of performing several tasks simultaneously, including object detection, classification, pose estimation, and segmentation. Like other YOLO models, YOLOv8 is a CNN that uses a single neural network to make predictions from full images in one evaluation.

YOLOv8 consists of three main parts: backbone, neck, and head:

1. Backbone is a CSPDarknet network that extracts features from the input image.
2. Neck is an SPP and PAN that combines features from different layers.
3. Head is a CNN used to make predictions.

YOLOv8 works by dividing the input image into a grid of cells. For each cell, the model predicts a set of bounding boxes and the class probabilities for each bounding box. The bounding boxes are represented by coordinates that indicate the object's center and width/height, and the class probabilities indicate the likelihood that the object belongs to each of the predefined classes. Figure 3.5 illustrates the entire architecture of YOLOv8.

We have leveraged the pre-trained YOLOv8-nano model on the COCO

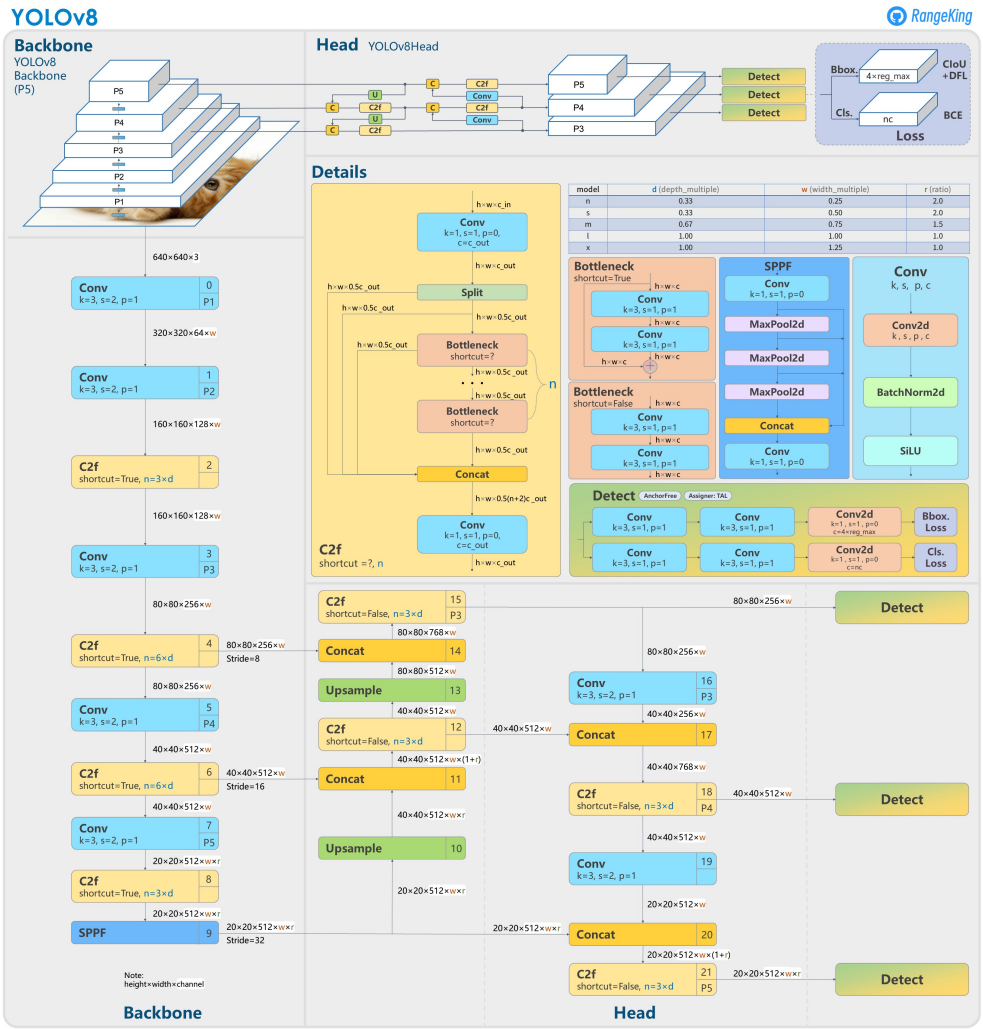


Figure 3.5: YOLOv8 architecture [53].

dataset [54] for our experiments. Transfer learning is highly beneficial in our case. The model has already learned low-level features, such as edges and corners. Therefore, fine-tuning the pre-trained model on our dataset is faster and more efficient than training from scratch.

3.3 Trajectory Prediction Model

We have decided to use a three-layer model for trajectory prediction. The model consists of the 1D convolutional layer with 16 filters, a linear layer with 32 neurons, and a linear layer with $2 *$ output size neurons. The activation function for the first two layers is ReLU, and the activation function for the last layer is HardSigmoid [55] (to ensure that the output is in the range of $[0, 1]$). The model accepts a flattened sequence of normalized coordinates as input. The output is a sequence of normalized coordinates.

3.4 Loss Functions

ML is an optimization problem where we find the parameters of the model that minimize the loss function. Therefore, choosing the loss function is important since it affects the model's performance and convergence speed. This section will describe the loss functions used to train our models for different tasks.

3.4.1 Object detection

Object detection boils down to two main tasks: classification and localization. The total loss combines the classification loss and the localization loss. We can write the total loss as follows:

$$Loss_{total} = \frac{1}{N}\sum_{i=1}^{N}(Loss_{cls}(p_i, y_i) + \lambda_{obj} Loss_{obj}(b_i, \hat{b}_i)), \tag{3.1}$$

where N is the number of objects, p_i is the predicted probability of the correct class, y_i is the ground truth probability of the correct class, b_i is the predicted bounding box, $\hat{b}_i$ is the ground truth bounding box, $Loss_{cls}$ is the classification loss, $Loss_{obj}$ is the localization loss, and λ_{obj} is a hyperparameter that controls the importance of the localization loss.

For classification, we can use a classic cross-entropy loss function:

$$CrossEntropy(p, y) = -\sum_{i=1}^{C} y_i \log(p_i), \tag{3.2}$$

where p is the predicted probability of the correct class, y is the ground truth probability of the correct class, and C is the number of classes.

For localization, we can use a combination of Complete Intersection over Union **(CIoU)** [56] and distribution focal loss (**DFL**) [57] for bounding box regression. Experimentally, CIoU and DFL achieve better results than other loss functions, such as IoU and L2/L1 losses.

The formula for the total loss is as follows:

$$Loss_{obj} = (CIoU(b, \hat{b}) + DFL(p, y)), \tag{3.3}$$

where b is the predicted bounding box, $\hat{b}$ is the ground truth bounding box, p is the predicted probability of the correct class, y is the ground truth probability of the correct class and $CIoU$ and DFL are defined below.

CIoU is defined as follows:

$$CIoU(b, \hat{b}) = 1 - IoU(b, \hat{b}) + \frac{\rho^2(b, \hat{b})}{c^2} + \alpha v, \tag{3.4}$$

where b is the predicted bounding box, $\hat{b}$ is the ground truth bounding box, IoU is the intersection over union, ρ^2 is the Euclidean distance between the center points of the bounding boxes, c is the diagonal of the smallest enclosing box covering both bounding boxes, α is a hyperparameter that controls the importance of the aspect ratio, and v is a hyperparameter that controls the importance of the area ratio.

DFL is used in bounding box regression. The main idea is to predict the distribution of the box offsets instead of directly predicting the coordinates. It is defined as follows:

$$DFL(p, y) = -\alpha_t (1 - p_t)^\gamma \log(p_t), \tag{3.5}$$

where p is the predicted probability of the correct class, y is the ground truth probability of the correct class, α_t is the focal weight, and γ is the focal power.

3.4.2 Keypoint Estimation

Keypoint estimation can be viewed as a regression problem, where we try to predict the coordinates of the keypoints. Therefore, we can use a regression loss function, such as L2 or L1 loss. L2 loss is defined as follows:

$$L_2Loss(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{3.6}$$

where y is the ground truth keypoint, $\hat{y}$ is the predicted keypoint, and n is the number of keypoints.

L1 loss is defined as follows:

$$L_1Loss(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{3.7}$$

where y is the ground truth keypoint, $\hat{y}$ is the predicted keypoint, and n is the number of keypoints.

We have tested both L2 loss and L1 loss, and L1 loss achieved better results. Therefore, we have decided to use L1 loss for our experiments. For trajectory prediction, we have also utilized L1 loss since the nature of the task is similar to keypoint estimation.

3.5 Metrics

In our experiments, we must evaluate the performance of object detection, keypoint estimation, and trajectory prediction models.

3.5.1 Object Detection and Tracking

Intersection over Union (IoU) is a standard metric for evaluating object detection models. It is defined as follows:

$$IoU = \frac{Area(Intersection)}{Area(Union)}, \tag{3.8}$$

where $Area(Intersection)$ is the area of the intersection between the predicted bounding box and the ground truth bounding box, and $Area(Union)$ is the area of the union between the predicted bounding box and the ground truth bounding box.

3.5.2 Keypoint Estimation and Trajectory Prediction

We have decided to use Mean Absolute Percentage Error (MAPE) as a metric for evaluating the keypoint estimation and trajectory prediction models. MAPE is defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}, \tag{3.9}$$

where y is the ground truth keypoint, $\hat{y}$ is the predicted keypoint, and n is the number of keypoints.

3.6 Inference improvements

Small objects are challenging to detect due to the limited receptive field of the network. Due to several max-pooling layers, YOLO models struggle with detecting small objects [58]. To address this issue, we have used the **slicing aided hyper inference** (SAHI) [59] approach when dealing with small objects.

SAHI is a simple yet effective method for improving the detection of small objects. It works by slicing the input image into multiple overlapping patches and performing inference on each patch separately. The final prediction is obtained by combining the predictions from all patches. This approach allows us to increase the receptive field of the network and improve the detection of small objects at the cost of increased inference time.

3.7 ByteTrack

ByteTrack is a robust and efficient motion-based multi-object tracking (MOT) algorithm that utilizes a Kalman filter for tracking individual objects and a Hungarian algorithm for associating detections across frames [60]. Its key strength lies in its ability to track objects with low confidence scores, often discarded by other tracking algorithms. This feature makes ByteTrack well-suited for challenging scenarios involving occlusion, illumination changes, and background clutter. It achieves state-of-the-art performance on benchmark MOT datasets while maintaining real-time tracking speed.

ByteTrack works on top of the object detection model, which provides the bounding boxes for each frame. This method expects $frame_t$ and tracklets from the previous frame $\mathcal{T}_{t-1}$ as the input and outputs a list of tracklets for the current frame $\mathcal{T}_t$. The detailed algorithm is as follows:

1. Detect Objects: Analyze the current frame to identify objects. Save the results as $\mathcal{D}_t$.
2. Separate Detections: Split the detected objects into two groups based on their confidence scores. High-score objects: $\mathcal{D}_t^h$ (scores above a certain threshold). Low-score objects: $\mathcal{D}_t^l$ (scores at or below the threshold).
3. Associate High-Score Objects: Match high-score objects with existing tracklets from the previous frame using motion similarity. Results in pairs of matched objects and tracklets: $(\mathcal{M}_t^h, \mathcal{T}_t^h)$.
4. Associate Low-Score Objects: Associate low-score objects based on their appearance features with high-score tracklets. Results in matched low-score objects, high-score tracklets, and unmatched low-score objects: $(\mathcal{M}_t^l, \mathcal{T}_t^l, \mathcal{N})$.
5. Update Tracklets: Update the existing tracklets with information from both high-score and low-score associations. Create new tracklets for any low-score objects that were not associated. The updated tracklets are stored as $\mathcal{T}_t$.
6. Return: Provide the updated tracklets $\mathcal{T}_t$ as the output of the algorithm.

3.8 Propulsion Estimation

Propulsion estimation is a crucial task for controlling microrobots. Combined with trajectory prediction, a propulsion estimator can be used to predict the

future positions of microrobots.

The naive approach calculates the propulsion based on the distance between the positions in two consecutive frames. However, this approach is not robust to noise and can produce inaccurate results since it heavily depends on the quality of the tracker.

Another approach is to use wave propagation. The flagellum movements form a wave-like pattern, which can be modeled as a harmonic wave. The equation for the harmonic wave is as follows:

$$y(x,t) = A\sin(kx - \omega t + \phi), \tag{3.10}$$

where A is the amplitude, k is the wave number, ω is the angular frequency, and ϕ is the phase constant. The displacement at any given position undergoes harmonic motion. The wave's period corresponds to the period of this harmonic motion. With each period, the wave advances by one wavelength. Consequently, the wave velocity can be expressed as such:

$$v = \frac{\lambda}{T} = \frac{\omega}{k}, \tag{3.11}$$

where v is the wave velocity, λ is the wavelength, T is the period, ω is the angular frequency, and k is the wave number.

The ultimate goal is to estimate the parameters A, k, ω, and ϕ from the given data.

3.9 Keypoints Refinement

The predicted keypoints are not always accurate, especially with curved flagellum. We can employ a refinement model that adjusts the predicted keypoints to address this issue. The refinement model is a simple two-layer autoencoder with an additional layer for each parameter from Equation 3.10. The refiner is trained using a custom physics-informed loss:

$$Loss = \alpha_1 L2Loss(y, \hat{y}) + \alpha_2 FlagellumLoss(\hat{y}, \hat{A}, \hat{t}, \hat{k}, \hat{\omega}, \hat{\phi}), \tag{3.12}$$

where y is the ground truth keypoint, $\hat{y}$ is the predicted keypoint, α_1 is the weight for the L2 loss, α_2 is the weight for the Flagellum loss, $\hat{A}$ is the predicted amplitude, $\hat{t}$ is the predicted time, $\hat{k}$ is the predicted wave number, $\hat{\omega}$ is the predicted angular frequency, $\hat{\phi}$ is the predicted phase constant, and $FlagellumLoss$ is defined as a L2 loss between the predicted keypoints and the keypoints generated using the parameters from Equation 3.10. Flagellum loss enforces the predicted keypoints to form a harmonic wave. We can tweak the weights to control the importance of each loss.

Chapter 4

Experiments and Results

In this chapter, we will present our experiments to understand the dynamics of microrobots better. We have conducted three experiments: flagellum keypoint estimation, sperm cell tracking, and trajectory prediction.

4.1 Flagellum Keypoints Estimation

Flagellum is the primary source of propulsion for sperm cells. It is a long, whip-like structure attached to the sperm cell's head. This experiment aimed to develop an accurate and efficient algorithm for detecting flagellar keypoints in sperm cells, which are crucial for understanding their dynamics and controlling their behavior. Accurate detection of these flagellar keypoints is pivotal in extracting biomechanical parameters, including wave frequency and amplitude. These parameters, in turn, offer valuable insights into the fundamental mechanisms governing the locomotion of sperm cells.

Dataset

For this experiment, we have used a dataset with annotated flagellum keypoints described in Section 3.1. In total, there were 304 images for training and 90 images for testing.

Training and Evaluation

We have used the YOLOv8-nano model trained in a multi-task fashion to detect bounding boxes and flagellum keypoints simultaneously. The model was trained for 1500 epochs with the early stopping criterion of 300 epochs. The optimizer was Adam, with a learning rate 0.001 and weight decay of 0.0005. The batch size was set to 8, and the input images were resized to 512 pixels on the shorter side. The training took approximately 2 hours on a single NVIDIA GeForce RTX 3060Ti GPU.

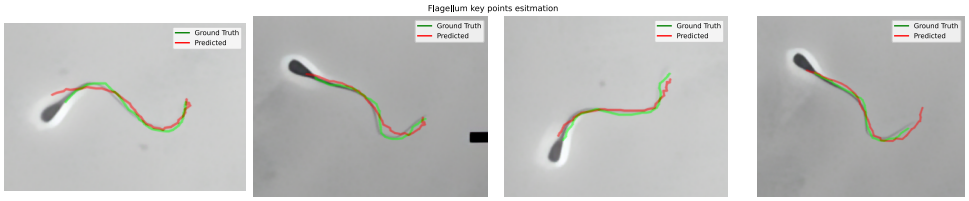


Figure 4.1: Predictions made by the YOLOv8 model on the testing set. The ground truth is shown in green, and the predicted keypoints are shown in red.

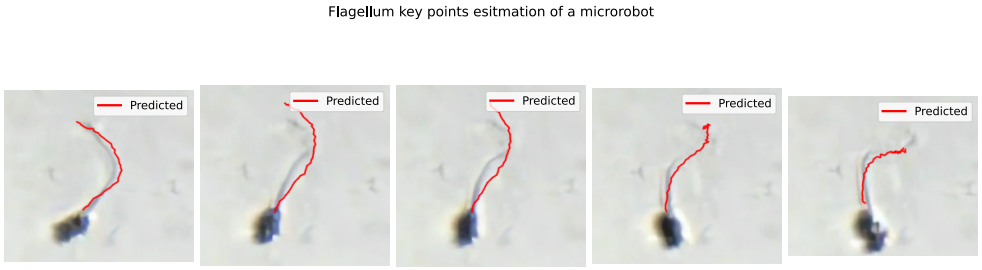


Figure 4.2: Predictions made by the YOLOv8 model on the microrobots dataset.

We conducted a thorough evaluation of the model using both the testing set and the microrobots dataset. In Figure 4.1, we visually compare ground truth and predicted keypoints extracted from the testing set. Additionally, Figure 4.2 showcases the prediction results derived from the microrobots dataset.

Furthermore, we assessed the model's performance using the MAPE metric on a validation set. Figure 4.3 illustrates the MAPE for each specific keypoint index. The average MAPE score across all keypoints was **0.04**.

We have also experimented with a refinement model described in Section 3.9, but we did not achieve the desired results. Thus, we omit the results from this thesis.

4.2 Sperm Cell Tracking

Sperm cells exhibit continuous movement as their flagellum moves in a wave-like pattern. The inherent complexity of the data makes it challenging to precisely predict bounding boxes that encompass both the sperm cell head and flagellum. In response, we employed an object detection model specifically designed to detect sperm cell heads. Subsequently, we employ the ByteTrack algorithm to facilitate tracking these detected sperm cell heads over time.

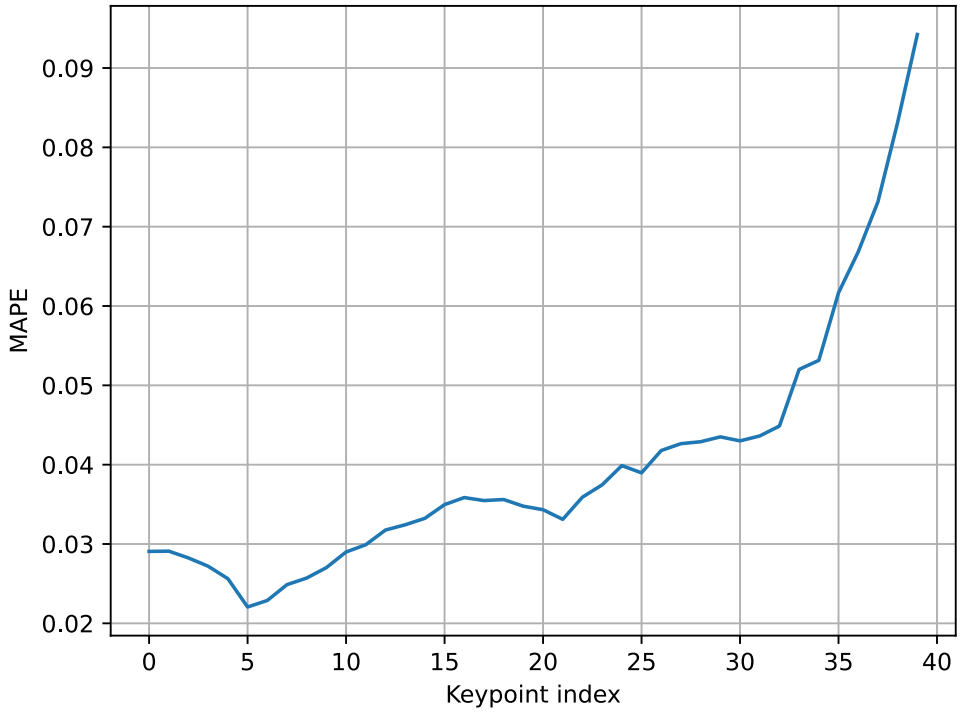


Figure 4.3: MAPE for each keypoint index. Lower is better.

Dataset

This experiment used a dataset with annotated sperm cell heads described in Section 3.1. The content was the same as for the flagellum keypoints estimation experiment, but the annotations differed. The dataset consisted of 304 images for training and 90 images for testing.

Training and Tracker Configuration

The training was performed in the same fashion as for the flagellum keypoints estimation experiment. The following configuration for ByteTrack was used:

- Detection model: YOLOv8-nano;
- High score detections threshold: 0.7;
- Low score detections threshold: 0.5;
- New detections threshold: 0.3;
- IoU matching threshold: 0.9;
- Track buffer: 30 frames;

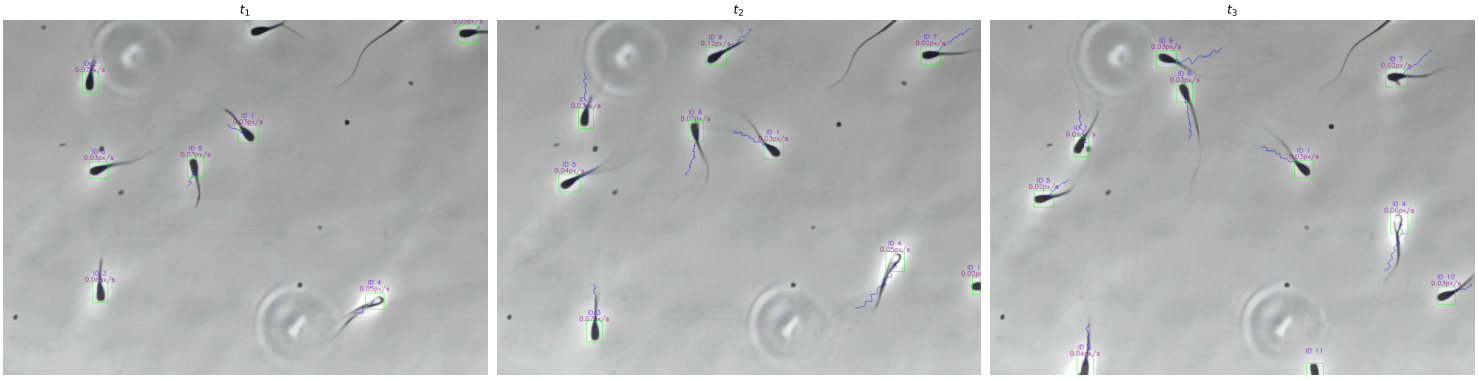


Figure 4.4: Tracking results on the testing dataset.

Evaluation

We thoroughly evaluated the detection model using the validation set, revealing an average IoU score of **0.68** (higher is better). This result is considered satisfactory, considering the dataset's complexities. Notably, we chose not to assess tracking performance, as each frame contains only a single cell, making tracker evaluation irrelevant in this scenario.

Furthermore, the model underwent a visual evaluation on the testing set, without annotations. The results are presented in Figure 4.4, showcasing results across three chronological frames from the testing set.

4.3 Trajectory Prediction

Trajectory prediction is a crucial task for controlling microrobots. In this experiment, we have leveraged the object detection model to track the sperm cell heads, which are used as input for the trajectory prediction model.

The model takes as input the last n positions of the sperm cell and predicts the next m positions. A centroid of the bounding box of the sperm head represents each position. The model architecture consists of one convolutional layer and two fully connected layers. We have experimented with different layers, such as LSTM, RNN, and GRU, but the results were unsatisfactory. The convolutional model achieved the best results, so we have used it for our experiments.

Dataset

We have used a dataset with annotated sperm cell heads with a few preprocessing steps. First, we have calculated the centroids of the bounding boxes of the sperm cell heads. Then, we applied a sliding window technique to extract the input and output sequences. The sliding window size was set to the input size, 10. The stride was set to 1, which means that the window

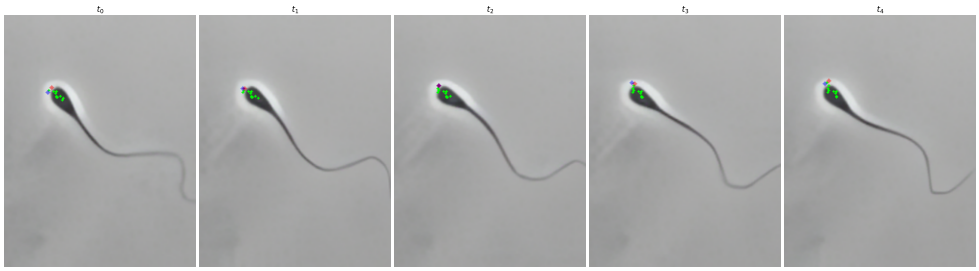


Figure 4.5: Trajectory prediction results on the testing dataset. The red dot represents the ground truth, and the blue dot represents the prediction. Green dots represent the input sequence.

was shifted by one position. The training set yielded 238 samples, while the testing set yielded 68.

Training and Evaluation

We have trained a CNN model for trajectory prediction described in Section 3.3. The model accepts the last ten positions of the spermatozoon head and predicts the next position. We have experimented with RNN-based models, such as LSTM, but the results were unsatisfactory. Therefore, we have chosen a CNN model that achieves better results. The model was trained for 150 epochs with a batch size of 8, AdamW optimizer [61], with a learning rate 0.001 and no weight decay. The input length was set to 10, and the output length was set to 1.

The evaluation was performed on the validation set, which resulted in **0.022** MAPE (lower is better). Figure 4.5 shows the results of the trajectory prediction model on the testing set.

Chapter 5

Discussion

In this chapter, we will discuss the results of the experiments described in the previous chapter. We will also discuss the limitations of the current approach.

5.1 Flagellum Keypoints Estimation

The results of the flagellum keypoints estimation experiment look promising. In Figure 4.1, we can see that the model can detect the flagellum keypoints with high accuracy, following the pattern of the flagellum. The model performs well with out-of-distribution data, such as microrobots. However, due to the noisy data, the model struggles to estimate the keypoints at the end of the flagellum. This task is challenging even for humans, as it is hard to distinguish the end of the flagellum from the background in the presence of artifacts and noise.

Figure 4.3 shows the average error for individual keypoints. The 0th keypoint is located at the sperm cell head, while the last keypoint is located at the end of the flagellum. As we expected, the closer the keypoint is to the sperm cell head, the more accurate the prediction is. The reason for that is that the beginning of the flagellum is more distinguishable from the background than the end of the flagellum.

Furthermore, our research partners from the University of Twente (Netherlands) have developed a method for estimating the propulsion of the sperm cell based on the flagellum keypoints [62]. Therefore, our method can be combined with their method to estimate the propulsion of the sperm cell.

5.2 Sperm Cell Tracking

Figure 4.4 shows the results of the tracking algorithm on the testing set. In general, the tracking algorithm performs well. Sometimes, we can observe that the detector fails to detect "glowing" sperm cells and bundles during their

formation. It is also worth noting that sometimes, the algorithm struggles to track the sperm cells occluded by other sperm cells. Once the sperm cells are separated, the algorithm can track them again.

5.3 Trajectory Prediction

The trajectory prediction model achieved **0.02** MAPE on the validation set. The model was able to predict the trajectory of the spermatozoa with reasonable accuracy, as we can see in Figure 4.5. The cell movement is stochastic, so the model cannot predict the exact trajectory at each time step. However, the model can predict the general direction of the spermatozoon movement.

5.4 Limitations

The main limitation of the current approach is the need for more data. Deep learning highly benefits from the diverse data with many samples. Having only 304 images for training and 90 images for testing puts constraints on model performance. The results could be significantly improved by gathering and annotating more diverse data.

Chapter 6

Conclusion and Future Work

In this thesis, we have explored the potential of machine learning in advancing microrobotics, specifically in the context of biological microrobots based on sperm cells. We have presented a comprehensive overview of the most relevant works in microrobotics and machine learning in biomedical robotics. We have also prepared and annotated datasets for training and evaluating object detection, keypoint estimation, and trajectory prediction models. We have trained and evaluated object detection models for detecting sperm cells and sperm cell heads. We have trained and evaluated a keypoint estimation model for detecting flagellum keypoints. We have evaluated the object-tracking model for tracking sperm cell heads. We have also trained and evaluated a trajectory prediction model for better navigation in the bloodstream.

6.1 Future Work

In the future, we plan to continue improving sperm cell tracking and trajectory prediction. The current results can be improved by gathering more diverse data. We also plan to improve the annotation process using more accurate tracking algorithms. We will further experiment with a refinement network that we have not managed to successfully train.

One future direction is to utilize the keypoint estimation model and trajectory prediction model with reinforcement learning. Current studies do not consider the dynamics of the microrobots, which can be crucial for precise control. By extracting biomechanical parameters, such as wave frequency and amplitude, we can provide insights into the underlying mechanisms governing sperm cell locomotion and use them for reinforcement learning.

The existing work can further be extended by using a closed-loop control system. Closed-loop control is a system that uses environmental feedback to control the robot. In the context of microrobotics, it can control the microrobots' behavior in real-time. For example, if the microrobot is drifting

away from the target, the closed-loop controller can adjust the direction of the microrobot to compensate for the drift.

Bibliography

1. WANG, Zihan; KLINGNER, Anke; MAGDANZ, Veronika; MISRA, Sarthak; KHALIL, Islam S. M. Soft Bio-Microrobots: Toward Biomedical Applications. *Advanced Intelligent Systems*. [N.d.], vol. n/a, no. n/a, p. 2300093. Available from DOI: https://doi.org/10.1002/aisy.202300093.
2. MAGDANZ, Veronika; KHALIL, Islam S. M.; SIMMCHEN, Juliane; FURTADO, Guilherme P.; MOHANTY, Sumit; GEBAUER, Johannes; XU, Haifeng; KLINGNER, Anke; AZIZ, Azaam; MEDINA-SÁNCHEZ, Mariana; SCHMIDT, Oliver G.; MISRA, Sarthak. IRONSperm: Sperm-templated soft magnetic microrobots. *Science Advances*. 2020, vol. 6, no. 28, eaba5855. Available from DOI: 10.1126/sciadv.aba5855.
3. FUSCO, Stefano; ULLRICH, Franziska; POKKI, Juho; CHATZIPIRPIRIDIS, George; ÖZKALE, Berna; SIVARAMAN, Kartik M; ERGENEMAN, Olgaç; PANÉ, Salvador; NELSON, Bradley J. Microrobots: a new era in ocular drug delivery. *Expert Opinion on Drug Delivery*. 2014, vol. 11, no. 11, pp. 1815–1826. Available from DOI: 10.1517/17425247.2014.938633. PMID: 25001411.
4. JANG, Deasung; JEONG, Jinwon; SONG, Hyeonseok; CHUNG, Sang Kug. Targeted drug delivery technology using untethered microrobots: A review. *Journal of Micromechanics and Microengineering*. 2019, vol. 29, no. 5, p. 053002.
5. YORK, Peter A; PEÑA, Rut; KENT, Daniel; WOOD, Robert J. Microrobotic laser steering for minimally invasive surgery. *Science Robotics*. 2021, vol. 6, no. 50, eabd5476.
6. VILLA, Katherine; KREJČOVÁ, Ludmila; NOVOTNỲ, Filip; HEGER, Zbynek; SOFER, Zdeněk; PUMERA, Martin. Cooperative multifunctional self-propelled paramagnetic microrobots with chemical handles for cell manipulation and drug delivery. *Advanced Functional Materials*. 2018, vol. 28, no. 43, p. 1804343.

7. HU, Chengzhi; PANÉ, Salvador; NELSON, Bradley J. Soft micro-and nanorobotics. *Annual Review of Control, Robotics, and Autonomous Systems*. 2018, vol. 1, pp. 53–75.

8. ABDALLAH, Chaouki; DAWSON, Darren M; DORATO, Peter; JAMSHIDI, Mohammad. Survey of robust control for rigid robots. *IEEE Control Systems Magazine*. 1991, vol. 11, no. 2, pp. 24–30.

9. WEIBEL, Douglas B; GARSTECKI, Piotr; RYAN, Declan; DILUZIO, Willow R; MAYER, Michael; SETO, Jennifer E; WHITESIDES, George M. Microoxen: Microorganisms to move microscale loads. *Proceedings of the National Academy of Sciences*. 2005, vol. 102, no. 34, pp. 11963–11967.

10. PALAGI, Stefano; MARK, Andrew G.; REIGH, Shang Yik; MELDE, Kai; QIU, Tian; ZENG, Hao; PARMEGGIANI, Camilla; MARTELLA, Daniele; SANCHEZ-CASTILLO, Alberto; KAPERNAUM, Nadia; GIESSELMANN, Frank; WIERSMA, Diederik S.; LAUGA, Eric; FISCHER, Peer. Structured light enables biomimetic swimming and versatile locomotion of photoresponsive soft microrobots. *Nature Materials*. 2016, vol. 15, no. 6, pp. 647–653. ISSN 1476-4660. Available from DOI: 10.1038/nmat4569.

11. ZHONG, Yong; HU, Luohua; XU, Yinsheng. Recent Advances in Design and Actuation of Continuum Robots for Medical Applications. *Actuators*. 2020, vol. 9, no. 4. ISSN 2076-0825. Available from DOI: 10.3390/act9040142.

12. NAUBER, Richard; GOUDU, Sandhya R.; GOECKENJAN, Maren; BORNHÄUSER, Martin; RIBEIRO, Carla; MEDINA-SÁNCHEZ, Mariana. Medical microrobots in reproductive medicine from the bench to the clinic. *Nature Communications*. 2023, vol. 14, no. 1, p. 728. ISSN 2041-1723. Available from DOI: 10.1038/s41467-023-36215-7.

13. GALTON, Francis. Regression Towards Mediocrity in Hereditary Stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*. 1886, vol. 15, pp. 246–263. ISSN 09595295. Available from DOI: 10.2307/2841583. Full publication date: 1886.

14. COX, David R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1958, vol. 20, no. 2, pp. 215–232.

15. CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. *Machine learning*. 1995, vol. 20, no. 3, pp. 273–297.

16. BREIMAN, Leo. Random forests. *Machine learning*. 2001, vol. 45, pp. 5–32.

17. ESTEVA, Andre; KUPREL, Brett; NOVOA, Roberto A; KO, Justin; SWETTER, Susan M; BLAU, Helen M; THRUN, Sebastian. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. 2017, vol. 542, no. 7639, pp. 115–118.

18. DANA, Dibyendu; GADHIYA, Satishkumar V; ST SURIN, Luce G; LI, David; NAAZ, Farha; ALI, Quaisar; PAKA, Latha; YAMIN, Michael A; NARAYAN, Mahesh; GOLDBERG, Itzhak D; NARAYAN, Prakash. Deep Learning in Drug Discovery and Medicine; Scratching the Surface. *Molecules*. 2018, vol. 23, no. 9.

19. LITJENS, Geert; KOOI, Thijs; BEJNORDI, Babak Ehteshami; SETIO, Arnaud Arindra Adiyoso; CIOMPI, Francesco; GHAFOORIAN, Mohsen; LAAK, Jeroen A W M van der; GINNEKEN, Bram van; SÁNCHEZ, Clara I. A survey on deep learning in medical image analysis. *Med Image Anal*. 2017, vol. 42, pp. 60–88.

20. LLOYD, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*. 1982, vol. 28, no. 2, pp. 129–137. Available from DOI: 10.1109/TIT.1982.1056489.

21. F.R.S., Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 1901, vol. 2, no. 11, pp. 559–572. Available from DOI: 10.1080/14786440109462720.

22. KRAMER, Mark A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*. 1991, vol. 37, no. 2, pp. 233–243. Available from DOI: https://doi.org/10.1002/aic.690370209.

23. CHEN, Richard J; LU, Ming Y; CHEN, Tiffany Y; WILLIAMSON, Drew FK; MAHMOOD, Faisal. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*. 2021, vol. 5, no. 6, pp. 493–497.

24. XIANG, Ruizhi; WANG, Wencan; YANG, Lei; WANG, Shiyuan; XU, Chaohan; CHEN, Xiaowen. A Comparison for Dimensionality Reduction Methods of Single-Cell RNA-seq Data. *Frontiers in Genetics*. 2021, vol. 12. ISSN 1664-8021. Available from DOI: 10.3389/fgene.2021.646936.

25. OPENAI; BERNER, Christopher; BROCKMAN, Greg; CHAN, Brooke; CHEUNG, Vicki; DEBIAK, Przemysław; DENNISON, Christy; FARHI, David; FISCHER, Quirin; HASHME, Shariq; HESSE, Chris; JÓZEFOWICZ, Rafal; GRAY, Scott; OLSSON, Catherine; PACHOCKI, Jakub; PETROV, Michael; OLIVEIRA PINTO, Henrique Pondé de; RAIMAN, Jonathan; SALIMANS, Tim; SCHLATTER, Jeremy; SCHNEIDER, Jonas; SIDOR, Szymon; SUTSKEVER, Ilya; TANG, Jie; WOLSKI, Filip; ZHANG, Su-

san. Dota 2 with Large Scale Deep Reinforcement Learning. 2019. Available from arXiv: 1912.06680.

26. ZHANG, Tengteng; MO, Hongwei. Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems*. 2021, vol. 18, no. 3, p. 17298814211007305. Available from DOI: 10.1177/17298814211007305.

27. MAHMUD, Mufti; KAISER, Mohammed Shamim; HUSSAIN, Amir; VASSANELLI, Stefano. Applications of Deep Learning and Reinforcement Learning to Biological Data. *IEEE Trans Neural Netw Learn Syst*. 2018, vol. 29, no. 6, pp. 2063–2079.

28. ZOU, Zonghao; LIU, Yuexin; YOUNG, Y-N; PAK, On Shun; TSANG, Alan CH. Gait switching and targeted navigation of microswimmers via deep reinforcement learning. *Communications Physics*. 2022, vol. 5, no. 1, p. 158.

29. BEHRENS, Michael R.; RUDER, Warren C. Smart Magnetic Microrobots Learn to Swim with Deep Reinforcement Learning. *Advanced Intelligent Systems*. 2022, vol. 4, no. 10, p. 2200023. Available from DOI: https://doi.org/10.1002/aisy.202200023.

30. XIA, Tiancheng; JIANG, Richard; FU, Yong Qing; JIN, Nanlin. Automated Blood Cell Detection and Counting via Deep Learning for Microfluidic Point-of-Care Medical Devices. *IOP Conference Series: Materials Science and Engineering*. 2019, vol. 646, no. 1, p. 012048. Available from DOI: 10.1088/1757-899X/646/1/012048.

31. ZHANG, Jun; LIU, Mingxia; SHEN, Dinggang. Detecting Anatomical Landmarks From Limited Medical Imaging Data Using Two-Stage Task-Oriented Deep Neural Networks. *IEEE Trans Image Process*. 2017, vol. 26, no. 10, pp. 4753–4764.

32. LECUN, Yann; BENGIO, Yoshua, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*. 1995, vol. 3361, no. 10, p. 1995.

33. REDMON, Joseph; DIVVALA, Santosh; GIRSHICK, Ross; FARHADI, Ali. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. Available from arXiv: 1506.02640 [cs.CV].

34. DOSOVITSKIY, Alexey; BEYER, Lucas; KOLESNIKOV, Alexander; WEISSENBORN, Dirk; ZHAI, Xiaohua; UNTERTHINER, Thomas; DEHGHANI, Mostafa; MINDERER, Matthias; HEIGOLD, Georg; GELLY, Sylvain; USZKOREIT, Jakob; HOULSBY, Neil. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Available from arXiv: 2010.11929 [cs.CV].

35. LIU, Ze; LIN, Yutong; CAO, Yue; HU, Han; WEI, Yixuan; ZHANG, Zheng; LIN, Stephen; GUO, Baining. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. Available from arXiv: 2103.14030 [cs.CV].

36. ANDRILUKA, Mykhaylo; PISHCHULIN, Leonid; GEHLER, Peter; SCHIELE, Bernt. 2d human pose estimation: New benchmark and state of the art analysis. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014, pp. 3686–3693.

37. KORTLI, Yassin; JRIDI, Maher; AL FALOU, Ayman; ATRI, Mohamed. Face recognition systems: A survey. *Sensors*. 2020, vol. 20, no. 2, p. 342.

38. LIU, Shuai; LIU, Dongye; SRIVASTAVA, Gautam; POŁAP, Dawid; WOŹNIAK, Marcin. Overview and methods of correlation filter algorithms in object tracking. *Complex & Intelligent Systems*. 2021, vol. 7, no. 4, pp. 1895–1917. ISSN 2198-6053. Available from DOI: 10.1007/s40747-020-00161-4.

39. COMANICIU, Dorin; RAMESH, Visvanathan; MEER, Peter. Real-time tracking of non-rigid objects using mean shift. In: *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*. IEEE, 2000, vol. 2, pp. 142–149.

40. MALISIEWICZ, Tomasz; GUPTA, Abhinav; EFROS, Alexei A. Ensemble of Exemplar-SVMs for Object Detection and Beyond. In: *ICCV*. 2011.

41. RUMELHART, David E; HINTON, Geoffrey E; WILLIAMS, Ronald J, et al. *Learning internal representations by error propagation*. Institute for Cognitive Science, University of California, San Diego La ..., 1985.

42. HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*. 1997, vol. 9, no. 8, pp. 1735–1780.

43. TOMASI, C.; MANDUCHI, R. Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 1998, pp. 839–846. Available from DOI: 10.1109/ICCV.1998.710815.

44. PIZER, Stephen M.; AMBURN, E. Philip; AUSTIN, John D.; CROMARTIE, Robert; GESELOWITZ, Ari; GREER, Trey; TER HAAR ROMENY, Bart; ZIMMERMAN, John B.; ZUIDERVELD, Karel. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*. 1987, vol. 39, no. 3, pp. 355–368. ISSN 0734-189X. Available from DOI: https://doi.org/10.1016/S0734-189X(87)80186-X.

45. GOODFELLOW, Ian J.; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David; OZAIR, Sherjil; COURVILLE, Aaron; BENGIO, Yoshua. *Generative Adversarial Networks*. 2014. Available from arXiv: 1406.2661 [stat.ML].

46. WANG, Xintao; XIE, Liangbin; DONG, Chao; SHAN, Ying. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In: *International Conference on Computer Vision Workshops (ICCVW)*. 2021.

47. SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on image data augmentation for deep learning. *Journal of big data*. 2019, vol. 6, no. 1, pp. 1–48.

48. LIU, Shilong; ZENG, Zhaoyang; REN, Tianhe; LI, Feng; ZHANG, Hao; YANG, Jie; LI, Chunyuan; YANG, Jianwei; SU, Hang; ZHU, Jun; ZHANG, Lei. *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*. 2023. Available from arXiv: 2303.05499 [cs.CV].

49. MORCILLO I SOLER, Paula; HIDALGO, Carlos; FEKETE, Zoltán; ZALANYI, Laszlo; KHALIL, Islam S. M.; YESTE, Marc; MAGDANZ, Veronika. Bundle formation of sperm: Influence of environmental factors. *Frontiers in Endocrinology*. 2022, vol. 13. ISSN 1664-2392. Available from DOI: 10.3389/fendo.2022.957684.

50. BOCHKOVSKIY, Alexey; WANG, Chien-Yao; LIAO, Hong-Yuan Mark. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. Available from arXiv: 2004.10934 [cs.CV].

51. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*. 2015, vol. 37, no. 9, pp. 1904–1916.

52. LIU, Shu; QI, Lu; QIN, Haifang; SHI, Jianping; JIA, Jiaya. *Path Aggregation Network for Instance Segmentation*. 2018. Available from arXiv: 1803.01534 [cs.CV].

53. RANGEKING. *Brief summary of YOLOv8 model structure* [https://github.com/ultralytics/ultralytics/issues/189]. 2023.

54. LIN, Tsung-Yi; MAIRE, Michael; BELONGIE, Serge; BOURDEV, Lubomir; GIRSHICK, Ross; HAYS, James; PERONA, Pietro; RAMANAN, Deva; ZITNICK, C. Lawrence; DOLLÁR, Piotr. *Microsoft COCO: Common Objects in Context*. 2015. Available from arXiv: 1405.0312 [cs.CV].

55. COURBARIAUX, Matthieu; BENGIO, Yoshua; DAVID, Jean-Pierre. *BinaryConnect: Training Deep Neural Networks with binary weights during propagations*. 2016. Available from arXiv: 1511.00363 [cs.LG].

56. ZHENG, Zhaohui; WANG, Ping; LIU, Wei; LI, Jinze; YE, Rongguang; REN, Dongwei. *Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression*. 2019. Available from arXiv: 1911.08287 [cs.CV].

57. LI, Xiang; WANG, Wenhai; WU, Lijun; CHEN, Shuo; HU, Xiaolin; LI, Jun; TANG, Jinhui; YANG, Jian. *Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection*. 2020. Available from arXiv: 2006.04388 [cs.CV].

58. NGUYEN, Nhat-Duy; DO, Tien; NGO, Thanh Duc; LE, Duy-Dinh. An evaluation of deep learning methods for small object detection. *Journal of electrical and computer engineering*. 2020, vol. 2020, pp. 1–18.

59. AKYON, Fatih Cagatay; ONUR ALTINUC, Sinan; TEMIZEL, Alptekin. Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. In: *2022 IEEE International Conference on Image Processing (ICIP)*. 2022, pp. 966–970. Available from DOI: 10.1109/ICIP46576.2022.9897990.

60. ZHANG, Yifu; SUN, Peize; JIANG, Yi; YU, Dongdong; WENG, Fucheng; YUAN, Zehuan; LUO, Ping; LIU, Wenyu; WANG, Xinggang. *ByteTrack: Multi-Object Tracking by Associating Every Detection Box*. 2022. Available from arXiv: 2110.06864 [cs.CV].

61. LOSHCHILOV, Ilya; HUTTER, Frank. *Decoupled Weight Decay Regularization*. 2019. Available from arXiv: 1711.05101 [cs.LG].

62. WANG, Zihan; KLINGNER, Anke; MAGDANZ, Veronika; HOPPENREIJS, Merijn W.; MISRA, Sarthak; KHALIL, Islam S. M. Flagellar Propulsion of Sperm Cells Against a Time-Periodic Interaction Force. *Advanced Biology*. 2023, vol. 7, no. 1, p. 2200210. Available from DOI: https://doi.org/10.1002/adbi.202200210.

Appendix A

Acronyms

ML Machine Learning

DL Deep Learning

IVF In vitro fertilization

RL Reinforcement Learning

SAC Soft Actor-Critic

DNN Deep Neural Network

CNN Convolutional Neural Network

YOLO You Only Look Once

ViT Visual Transformer

HOG Histogram of Oriented Gradients

RNN Recurrent Neural Network

LSTM Long Short-Term Memory

GAN Generative Adversarial Network

SPP Spatial Pyramid Pooling

PAN Path Aggregation Network

CIoU Complete Intersection over Union

DFL Distribution Focal Loss

MAPE Mean Absolute Percentage Error

SAHI Slicing Aided Hyper Inference

MOT Multi-Object Tracking

Appendix B

Contents of enclosed CD

- `scripts` ..... folder containing scripts
  - `microrobots` ..... main Python package
  - `notebooks` ..... Jupyter notebooks
  - `test_videos` ..... videos for testing
  - `data_40kp` ..... dataset for keypoint estimation
  - `data_head_only` ..... dataset with sperm cell heads annotation
  - `requirements.txt` ..... Python dependencies
  - `latex` ..... the LaTeX source code of the thesis
  - `thesis.pdf` ..... the thesis text in PDF format