FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Video Recording based Sperm Cell Movement Prediction and Modes of Movement Detection
Student:	Matej Kulháň
Supervisor:	Ing. Miroslav Čepek, Ph.D.
Study program:	Informatics
Branch / specialization:	Artificial Intelligence 2021
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

The University of Twente (UT) and the University of Waterloo (UW) are working on research into sperm movement. This thesis aims to help the UT and UW teams understand and predict movements of sperm cells. The thesis will study the machine learning approaches to predict future movements of individual cells and will explore ways to identify and extract movements or styles determining the direction and speed of travel. The basis for the thesis are video recordings obtained from UT and UW teams. The thesis will explore approaches based on features explicitly extracted from the recorded video as well as pure image and video processing approaches.

Individual steps:

1. Conduct a literature review and review the current state of the art.
2. Use and adapt previous work to process the video recordings.
3. Review and explore techniques for movement prediction and next frame predictions.
4. Experiment with techniques based on past positions only.
5. Experiment with selected techniques incorporating additional information: extracted features, direct video input.
6. Using techniques for model explainability identify important features and parts of video input for future movement prediction.
7. Document the accuracy of individual approaches.

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 12 February 2024 in Prague.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Literature:
[1] Noy, Lioz, et al. "Location Prediction of Sperm Cells Using Long Short-Term Memory Networks." Advanced Intelligent Systems 5.9 (2023): 2300161.
[2] OKUMUŞ, F., KOCAMAZ, F., & ÖZGÜR, M. E. (2021). Using polynomial modeling for calculation of quality parameters in computer assisted sperm analysis. Computer Science, 6(3), 152-165. https://doi.org/10.53070/bbd.999296

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 12 February 2024 in Prague.

Bachelor's thesis

VIDEO RECORDING BASED SPERM CELL MOVEMENT PREDICTION AND MODES OF MOVEMENT DETECTION

Matej Kulháň

Faculty of Information Technology
Department of applied mathematics
Supervisor: Ing. Miroslav Čepek, Ph.D.
May 16, 2024

Contents

Acknowledgments vi

Declaration vii

Abstract viii

List of abbreviations ix

1 Introduction 1

2 Theoretical Background 2
2.1 Current Methods for Sperm Analysis . . . 2
2.2 The collaborative project . . . 3
2.2.1 Video Dataset Overview . . . 3
2.2.2 Tracking System Overview . . . 3
2.3 Deep Learning Fundamentals . . . 4
2.3.1 Introduction to Neural Networks . . . 4
2.3.2 Recurrent Neural Networks . . . 4
2.3.3 Long Short-Term Memory Networks . . . 5
2.3.4 Convolutional Neural Networks . . . 6
2.4 Explainable AI and Interpretability . . . 7
2.4.1 Explanation Methods for Convolutional Neural Networks 7
2.4.2 Grad-CAM . . . 7
2.4.3 HiResCAM . . . 8
2.4.4 Respond-CAM . . . 8
2.5 OpenCV for Image and Video Processing . . . 9
2.5.1 Thresholding and Segmentation . . . 9
2.5.2 Image Rotation . . . 9

3 Practical Implementation and Experiments 11
3.1 Position-Based Path Forecasting . . . 11
3.1.1 Dataset Preparation . . . 11
3.1.2 Model Development . . . 14
3.2 Exploration with non-Position-Based Features . . . 16
3.2.1 Isolating Individual Sperm Cells . . . 16
3.3 Image-Based Features . . . 18
3.3.1 The Direction of Sperm Cells . . . 18

3.3.2 Classification of Sperm Cell Direction 21
3.4 Video-Based Features 23
3.4.1 Objective Definition: Predicting Sperm Cell Rotation . 23

4 Results and Evalution 27
4.1 Path Prediction with LSTM 27
4.2 Predicting Direction from Image Data 30
4.2.1 Explaining the Model's Predictions 32
4.3 Predicting Rotation from Video Data 34

5 Conclusion 36

Medium Attachment 39

List of Figures

2.1 Example of a sperm cell video recording 3
2.2 An unrolled recurrent neural network [7] 4
2.3 A Long Short-Term Memory (LSTM) cell [7] 5
2.4 A typical Convolutional Neural Network architecture [10] . . . 6
2.5 Thresholding applied to an image [15] 9

3.1 Sperm cell path 12
3.2 Outliers in the dataset 14
3.3 Single sperm cell 16
3.4 Single sperm cell after thresholding 18
3.5 Noisy path of a sperm cell 19
3.6 Noisy path of a sperm cell smoothed using a spline curve . . . 20
3.7 Comparison of the direction of the sperm cell before and after smoothing the path 20
3.8 Rotational angle of a sperm cell 23
3.9 Distribution of rotational angles 24
3.10 Distribution of rotational angles after filtering 24

4.1 Comparison of LSTM models with different input sequence lengths 27
4.2 Sample path predictions of the LSTM models 28
4.3 Sample path predictions of the LSTM models 29
4.4 Confusion matrix of the CNN model 30
4.5 Sample predictions of the CNN model 30
4.6 Heatmaps of the input images generated by the CNN model . . 32
4.7 Confusion matrix of the best performing rotation prediction model 34

List of Tables

3.1 Layer-wise details of the model architecture 22
3.2 Model Architecture 26

4.1 Comparison of different input resolutions and sequence lengths for rotation prediction models 34

I would like to thank my supervisor Ing. Miroslav Čepek, Ph.D. for his guidance and support throughout the process of writing this thesis. I would also like to thank my family and friends for their support and encouragement.

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 the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on May 16, 2024

Abstract

This thesis explores the use of machine learning techniques to predict the future path of sperm cells, classify their directional orientation, and predict their rotation from video data. The research is motivated by the need to better understand the behavior and motility of sperm cells, which play a crucial role in biomedical research and human reproduction studies.

Keywords sperm cell movement prediction, machine learning, video dataset, python, long short-term memory, convolutional neural network

Abstrakt

Táto práca skúma využitie techník strojového učenia na predikciu budúcej trasy spermií, klasifikáciu ich smerovej orientácie a predikciu ich rotácie z videí. Výskum je motivovaný potrebou lepšieho porozumenia správania a pohybu spermií, ktoré zohrávajú kľúčovú úlohu v biomedicínskom výskume a štúdiách reprodukcie človeka.

Klíčová slova predikcia pohybu spermií, strojové učenie, video dataset, python, long short-term memory, konvolučná neurónová sieť

List of abbreviations

ANN	Artificial Neural Network
API	Application Programming Interface
CNN	Convolutional Neural Network
Grad-CAM	Gradient-weighted Class Activation Mapping
HiResCAM	High Resolution Class Activation Mapping
LSTM	Long Short-Term Memory
ML	Machine Learning
MSE	Mean Squared Error
Respond-CAM	Respond-weighted Class Activation Mapping
RNN	Recurrent Neural Network
UT	University of Twente
UW	University of Waterloo
XAI	Explainable Artificial Intelligence

Chapter 1
Introduction

The movement patterns of sperm cells are of paramount importance in the fields of reproductive biology, offering insights into the underlying mechanisms of fertility and reproductive health. Understanding the trajectories of sperm cells can provide valuable information about their biological characteristics and has practical implications in assisted reproductive technologies and fertility treatments.

This thesis aims to harness machine learning techniques to predict the future movements of individual sperm cells, aiding in understanding the factors that influence their direction and speed. The video recordings provided by the University of Twente (UT) and the University of Waterloo (UW) serve as the basis for this research, capturing the complex movements of sperm cells under the microscope.

As part of this research, a systematic exploration of deep learning approaches tailored to process the video recordings and predict sperm cell movements is conducted. The experimental approach begins with techniques focusing solely on past positions to forecast the future paths of sperm cells. The next step involves integrating additional information, such as features extracted from the video data and direct video input to explore more creative ways of gaining insights into the way sperm cells move.

The culmination of this research will be a comprehensive evaluation of the accuracy of individual approaches, offering insights and recommendations for future studies and applications in the field of sperm cell movement analysis. Through this thesis, we aim to contribute to the ongoing research efforts in understanding and predicting sperm cell movements, ultimately advancing the field and benefiting fertility treatments and reproductive health assessments.

Chapter 2

Theoretical Background

This section provides an overview of the theoretical background and related work relevant to the research conducted in this thesis. It covers the current methods for sperm analysis, the collaborative project between CTU, UW, and UT, deep learning fundamentals, explainable AI and interpretability, and the use of OpenCV for image and video processing. The theoretical foundation presented here serves as the basis for the subsequent analysis and experimentation conducted in this thesis.

2.1 Current Methods for Sperm Analysis

Recent advancements in computer vision and machine learning have revolutionized the field of sperm analysis, enabling researchers to extract valuable insights from sperm cell movement patterns and morphology with high precision and efficiency. For instance, Sandra Ottl et al. developed a Machine Learning Framework for Automatic Prediction of Human Semen Motility [1]. Pol Fernández-López et al. used machine learning and statistical analysis to predict male fertility based on sperm motility [2].

These studies demonstrate the potential of machine learning techniques in predicting sperm cell movement and understanding the underlying factors. In 2023, a study by Lioz Noy et al. used long short-term memory networks, explained in section 2.3.3 to predict the future location of sperm cells. The study showed promising results, with the LSTM model achieving high accuracy in predicting the location of sperm cells up to over 1.4 seconds in advance. [3]

2.2 The collaborative project

The research presented in this thesis stems from a collaborative effort between CTU and two foreign universities, UW and UT. The aim of this collaboration was to explore and analyze sperm cell movement patterns using machine learning techniques.

2.2.1 Video Dataset Overview

The foundation of this project lies in the dataset of sperm cell recordings provided by UW and UT. The dataset consists of approximately 400 videos of sperm cells captured under a microscope. These videos were recorded using a high-speed camera at a rate of 400 frames per second and are of varying resolutions and lengths. Apart from a few exceptions, each video is exactly 400 frames long, while the resolution of the videos varies between (544, 384) and (1560, 1720) pixels. The dataset serves as the primary source of empirical data for the analysis and experimentation conducted in this thesis.

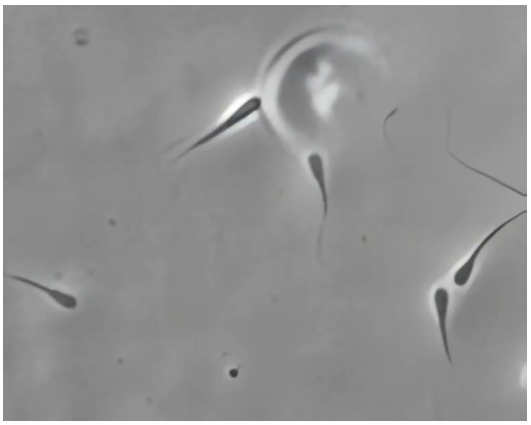


Figure 2.1 Example of a sperm cell video recording

2.2.2 Tracking System Overview

Subsequently, a pivotal contribution came from a student researcher at CTU, Jakub Hořenín, who developed a sophisticated tracking system capable of accurately tracing the trajectories of individual sperm cells within the video recordings. This tracking system lays the groundwork for the subsequent analysis and forecasting tasks undertaken in this thesis. The tracking system is capable of detecting and tracking sperm cells in the video recording frame by frame, providing the necessary data for further analysis and prediction. By leveraging the tracking system, it becomes possible to extract a path each sperm cell takes through the video, enabling the prediction of future movements based on past trajectories. [4]

2.3 Deep Learning Fundamentals

Artificial neural networks (ANNs) offer a powerful computational framework for processing complex data and solving challenging tasks, making them an invaluable tool for various applications in the thesis, where they can enhance prediction accuracy, enable pattern recognition, and facilitate data-driven insights. Their ability to learn from data and adapt their behavior makes them particularly well-suited for analyzing the intricate dynamics of sperm cell movement captured in the dataset provided for this research [5].

2.3.1 Introduction to Neural Networks

ANNs are computational models inspired by the biological structure of the brain. They consist of interconnected nodes organized in layers. Each node, or neuron, processes input data and passes it through activation functions to produce output. Layers are typically organized into an input layer, one or more hidden layers, and an output layer. In feedforward neural networks, information flows in one direction, from input to output, without feedback loops [5].

2.3.2 Recurrent Neural Networks

While feedforward neural networks are effective for many tasks, they may not be suitable for sequential data analysis. For instance, in analyzing sperm cell movement patterns captured in video recordings, temporal dependencies between data points are crucial. Recurrent Neural Networks (RNNs) address this by incorporating feedback loops, allowing information to flow not just forward but also back into the network. This enables RNNs to consider past inputs when processing current data, making them ideal for sequence prediction tasks.

RNNs have been successfully applied to various sequential data types, including text, time series, and images. The architecture of an RNN includes hidden layers with recurrent connections, as depicted in Figure 2.2. These connections enable the network to capture temporal dependencies and learn from sequential data. [6]

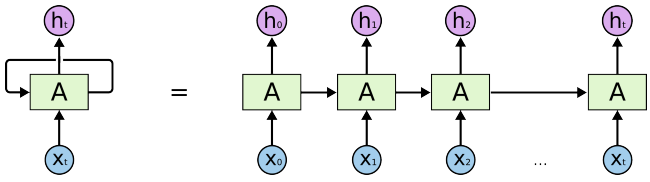


Figure 2.2 An unrolled recurrent neural network [7]

Figure 2.2 illustrates the unrolled architecture of the RNN.

2.3.2.1 The Vanishing Gradient Problem

However, the vanishing gradient problem is a notable issue in RNNs, hindering their effectiveness in learning long-term dependencies. This challenge stems from the difficulty in accurately assessing the significance of distant inputs, exacerbated by the multiplicative nature of information flow within neural networks. As gradients diminish over multiple layers, the ability to learn effectively deteriorates, limiting the suitability of RNNs for tasks requiring the modeling of complex temporal relationships over extended sequences. [8]

2.3.3 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber in 1997, have emerged as a powerful variant of recurrent neural networks (RNNs), promising to overcome the limitations of traditional RNN architectures. Equipped with memory cells capable of storing information for extended periods, LSTM networks excel in capturing long-term dependencies within sequential data. Unlike conventional RNNs, LSTM networks enforce a constant error flow through internal states of specialized units, addressing challenges such as error vanishing and exploding gradients encountered in traditional backpropagation methods. This unique design enables LSTM networks to bridge long time intervals while effectively capturing short-term dependencies, making them highly efficient and accurate in processing sequential data [7].

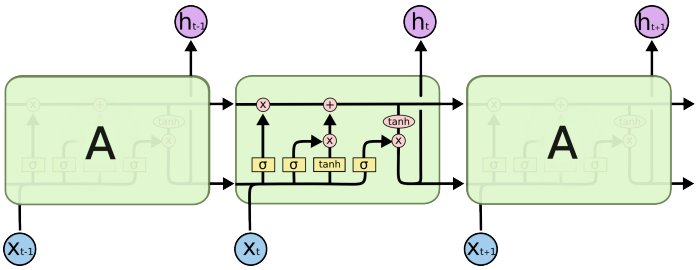


Figure 2.3 A Long Short-Term Memory (LSTM) cell [7]

Figure 2.3 illustrates the architecture of an LSTM cell, comprising three gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information within the cell, allowing it to retain or discard information based on the input data. The input gate controls the flow of new information into the cell, the forget gate manages the retention or removal of existing information, and the output gate determines the output of the cell.

In the context of sperm cell movement prediction, LSTM networks offer a robust framework for capturing the complex temporal dynamics of sperm cell paths, enabling accurate forecasting of future movement patterns. This has

also been demonstrated in the study by Lioz Noy et al., where LSTM networks were used to predict the future location of sperm cells with high accuracy [3].

2.3.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized type of artificial neural network primarily used for pattern recognition tasks in images. Unlike traditional ANNs, CNNs are specifically designed to handle image data by encoding image-specific features into their architecture. This specialization allows CNNs to excel in image-focused tasks while reducing the number of parameters required compared to traditional ANNs. CNNs consist of neurons organized in layers, with each neuron receiving inputs from a local receptive field in the previous layer. This local connectivity pattern, along with weight sharing and pooling operations, enables CNNs to effectively capture spatial hierarchies of features in images. CNNs have become the cornerstone of computer vision applications, achieving remarkable success in tasks such as object detection, image classification, and image segmentation. Despite their effectiveness, CNNs may face challenges with computational complexity when dealing with large-scale image data, necessitating careful design considerations to balance model size and performance. [9]

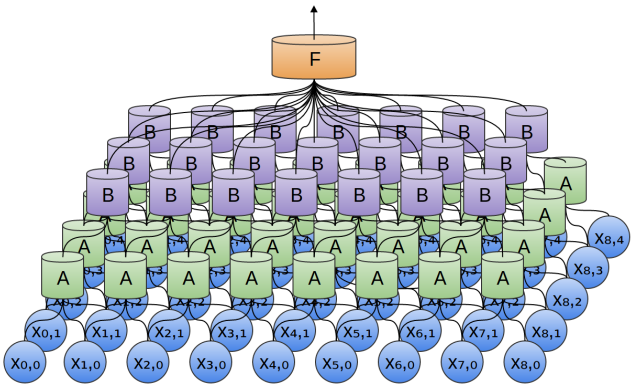


Figure 2.4 A typical Convolutional Neural Network architecture [10]

2.4 Explainable AI and Interpretability

Understanding the decisions made by AI models is crucial not only for ensuring accountability but also for unlocking valuable insights. Explainable AI (XAI) methods provide transparency into the inner workings of these models, shedding light on their decision-making processes. Seeking interpretations of AI models allows us to confirm that they are making rational choices aligned with expectations. Moreover, exploring the explanations can reveal instances where AI systems classify phenomena beyond human comprehension. In such cases, dissecting the explained model can lead to new knowledge and understanding previously inaccessible to humans. Through XAI techniques like Grad-CAM, a journey begins to demystify AI decisions, not just to validate their rationality but also to glean wisdom from their unique perspectives. [11]

2.4.1 Explanation Methods for Convolutional Neural Networks

Explanation methods for CNNs aim to provide human-interpretable insights into the model's internal workings. These methods help answer questions such as "Which parts of the input image are most influential in the model's decision?" and "What features does the model consider when making predictions?" Answering these questions can enhance our understanding of CNNs and potentially identify biases or weaknesses in their decision-making process. [11]

This section explores various explanation techniques for CNNs, discusses the principles behind each method, and highlights their strengths, limitations, and potential applications.

2.4.2 Grad-CAM

One of the most widely used explanation methods for CNNs is Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM generates a heatmap that highlights the regions of an input image that are most relevant to the model's prediction. By computing the gradients of the target class score concerning the feature maps of the last convolutional layer, Grad-CAM identifies the importance of different image regions in influencing the model's decision. This heatmap visualizes the areas of the input image that the model focused on when making its prediction, providing a visual explanation of the model's reasoning.

While Grad-CAM offers valuable insight into the areas of focus for the model, it has some limitations. "Grad-CAM does not directly visualize important locations, and Grad-CAM does not reflect the model's computations, even if Grad-CAM is applied at the last convolutional layer". [12] This limitation can sometimes result in misleading explanations, as the attention maps produced by Grad-CAM tend to expand, potentially highlighting irrelevant features.

Hence, while Grad-CAM excels in creating attention maps, its propensity for misleading interpretations necessitates caution in its application for model understanding and interpretability.[12]

2.4.3 HiResCAM

To address the limitations of Grad-CAM, a new method called High-Resolution Class Activation Mapping (HiResCAM) was developed. HiResCAM enhances the resolution of the attention maps generated by Grad-CAM, providing more detailed and accurate visualizations of the model's focus areas. By incorporating a super-resolution network into the explanation process, HiResCAM produces high-resolution attention maps that better capture the relevant features in the input image. This improved resolution enables more precise localization of the model's attention, enhancing the interpretability and reliability of the explanations provided by HiResCAM. [12]

2.4.4 Respond-CAM

Another novel explanation method, called Respond-weighted Class Activation Mapping (Respond-CAM), offers a unique approach to generating attention maps for CNNs. Respond-CAM leverages the response of the model's neurons to the input image to compute the importance of different regions in the image. By analyzing the response of the neurons in the last convolutional layer, Respond-CAM identifies the regions that elicit the strongest reactions from the model, indicating their significance in the decision-making process. This method provides a different perspective on the model's attention and can offer complementary insights to other explanation techniques like Grad-CAM and HiResCAM. By focusing on the response patterns of the model's neurons, Respond-CAM enhances the interpretability of CNNs and provides valuable information on the features influencing the model's predictions. [13]

2.5 OpenCV for Image and Video Processing

The OpenCV (Open Source Computer Vision) [14, 15] library played a fundamental role in implementing various image processing tasks throughout this research. Its versatile functionalities provided a robust framework for the analysis and manipulation of video recordings capturing sperm cell movements. This section explores the specific applications of OpenCV and outlines its essential contributions to preprocessing, segmentation, and perspective correction tasks.

2.5.1 Thresholding and Segmentation

Thresholding is a fundamental image processing technique used to separate objects from the background in an image. It involves converting an image into a binary format, where pixels are classified as either foreground or background based on their intensity values. In the context of sperm cell movement analysis, thresholding is a crucial step in segmenting sperm cells from the background of the video recordings. By setting a threshold value, pixels with intensity values above the threshold are classified as foreground (sperm cells), while those below the threshold are classified as background. This process effectively isolates sperm cells from the background, enabling further analysis and tracking tasks.

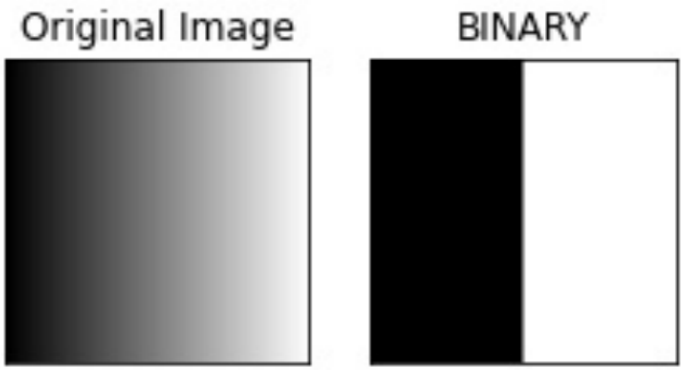


Figure 2.5 Thresholding applied to an image [15]

2.5.2 Image Rotation

Image rotation is a common image processing task used to adjust the orientation of an image. In the context of sperm cell movement analysis, image rotation can be used to reorient video frames to a more suitable perspective for feature extraction. By rotating video frames, it becomes possible to align sperm cells in a consistent direction, making it easier to differentiate between the individual parts of the sperm cell, which can help in tracking a single part of the sperm cell in each frame.

Additionally, flipping and rotating images can be used to augment the dataset, which can help improve the performance of machine learning models by providing more diverse training data. This technique is particularly useful when working with limited datasets, as it can help prevent overfitting and improve the generalization of the model.

Chapter 3

Practical Implementation and Experiments

In this chapter, I outline the development of computational models and experimental setups aimed at gaining insights into sperm cell behavior and morphology. The subsequent sections provide a comprehensive overview of these constructive practical applications, contributing to our understanding of reproductive biology and fertility science.

3.1 Position-Based Path Forecasting

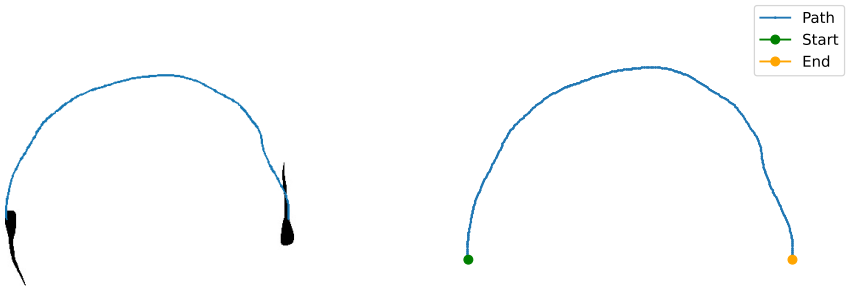
In this section, I delve into the creation of a position-based path forecasting model that uses historical sperm cell positions to predict their future trajectories. Through the application of machine learning and deep learning methods, this model seeks to improve our comprehension of sperm cell behavior and movement dynamics.

I collected the data used in this section by utilizing the sperm cell tracking system introduced in section 2.2.2. The dataset consists of sperm cell paths of varying lengths, with each path represented as a sequence of x, y coordinates denoting the sperm cell's position at each time step. Additional information, such as the frame number and class label of each sperm cell, was also included in the dataset for further analysis.

Figure 3.1 illustrates a sample sperm cell path, showcasing the x, y coordinates representing the sperm cell's movement over time. From now on, paths will be represented as in figure 3.1b, due to the simplicity of the representation.

3.1.1 Dataset Preparation

In order to get the dataset ready for training the model, the paths were preprocessed to ensure the quality and consistence of the data. I applied several



(a) Sperm cell path **(b)** Simplified version of the path

Figure 3.1 Sperm cell path

preprocessing steps to the dataset, including segmentation, choosing the right coordinate system, and filtering out outliers. These steps are crucial for enhancing the model's performance and ensuring the accuracy of the predictions.

3.1.1.1 Splitting the Paths into Sequences

I began the process by breaking down the paths into shorter sequences to facilitate the model's training. Subsequently, I constructed three distinct datasets, each containing paths segmented into sequences of uniform length. The choice of lengths varied across the datasets, with the intention of capturing different levels of granularity in the path dynamics, thus catering to the specific requirements of each dataset. These lengths were strategically selected to address computational constraints while ensuring the effective representation of path dynamics. Specifically, the datasets consisted of sequences categorized into three lengths: short sequences comprised of 30 points, medium sequences comprised of 50 points, and long sequences comprised of 80 points. Each dataset thus provided a tailored input structure for training distinct models, allowing for the exploration of path dynamics at varying levels of detail.

I constructed each dataset by employing a sliding window technique to segment the paths into sequences of the designated length. When creating the dataset, each sequence contained n points, and a path P of length m was split into $m-n+1$ sequences. Since $P_0, P_1, \ldots, P_n$ and $P_1, P_2, \ldots, P_{n+1}$ share $n-1$ points, I augmented the dataset by rotating each sequence by a random angle between 0 and 360 degrees. This approach ensured the diversity of the sequences while maximizing the utilization of the available data. This was done for $n \in \{30, 50, 80\}$.

3.1.1.2 Choosing the Right Coordinate System

When considering path forecasting for sperm cell trajectories, relying solely on absolute positions as features might prove inadequate due to the inherent

uneven distribution of data points within the dataset. To address this issue, I opted to employ a different coordinate system, utilizing relative coordinates to represent the paths.

In this coordinate system, each point is defined by a tuple $(\Delta x, \Delta y)$, where Δx signifies the change in the x-coordinate from the previous point, and Δy denotes the change in the y-coordinate. This representation captures the relative changes in position, offering a more robust and adaptable framework for modeling sperm cell movements. It mitigates the challenges posed by uneven data distributions and enhances my model's capacity to discern underlying patterns and behaviors within the trajectories.

In this coordinate system, each point is defined by a tuple $(\Delta x, \Delta y)$, where Δx signifies the change in the x-coordinate from the previous point, and Δy denotes the change in the y-coordinate. Specifically, Δx_1 and Δy_1 are calculated as:

$$\Delta x_1 = x_1 - x_0 \quad \text{and} \quad \Delta y_1 = y_1 - y_0$$

Therefore, to get the whole path P of length m into the relative coordinate system, the following transformation was applied:

$$P = [(x_0, y_0), (x_1, y_1), \ldots, (x_m, y_m)] \rightarrow P' = [(\Delta x_1, \Delta y_1), \ldots, (\Delta x_m, \Delta y_m)]$$

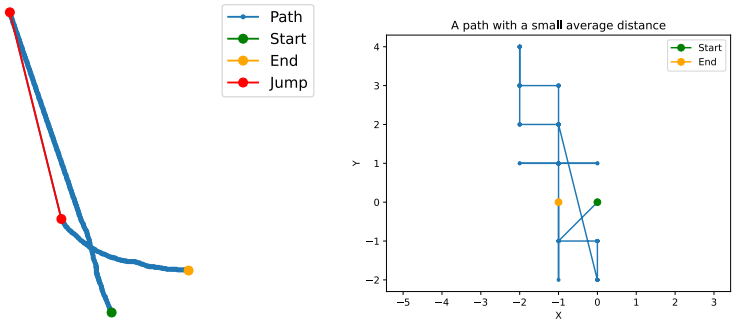
where P' represents a path of length $m-1$ in the relative coordinate system.

3.1.1.3 Filtering Outliers

Due to the noisy background of the videos and the complexity of the task the tracking system faced, the dataset contained paths with unrealistic movements. These problematic paths, characterized by sudden changes and irregular movements, posed challenges to the model's learning process and could potentially hinder the accuracy of the predictions. To address this issue, I implemented a filtering mechanism to identify and remove such outliers from the dataset, ensuring the quality and reliability of the data.

Two sperm cells crossing paths, or a sperm cell going off-screen, could result in a sudden jump in the distance between two points in a single path. By setting a threshold for the maximum distance between two points, these outliers can be identified and removed from the dataset.

On the other hand, paths with very little movement could also represent an error of the tracking system, where the tracked object could be a dead sperm cell, or perhaps even a larger particle from the background. However, these paths cannot be identified just by looking at the smallest distance between two points, as such points occur in nearly every path. To identify these paths, I calculated the average distance between neighboring points for each path. If the average distance was below a certain threshold, the paths were also considered outliers and removed from the dataset.



(a) Path with a sudden jump **(b)** Path with no movement

Figure 3.2 Outliers in the dataset

Figure 3.2 illustrates the outliers present in the dataset. Figure 3.2a clearly shows a path of two sperm cells connected by the big jump between two points. On the other hand, Figure 3.2b depicts a path with very little movement, indicating a potential tracking error or a dead sperm cell. The path is 177 data points long, while the staying almost in the same position, which is a clear indication of an outlier.

3.1.2 Model Development

Once the dataset was prepared, the next step was to develop a model that could learn the patterns and trends in the data and predict the future path of sperm cells. As stated in the theoretical background, the most suitable model for this task is an LSTM-based neural network.

I decided on trying two different approaches on predicting the future path of sperm cells. The first approach was to predict the future position one step at a time, while the second approach was to predict multiple future positions at once. The first approach is more straightforward, and easier to extend to predict any number of future positions. The second approach, on the other hand, is more complex and less flexible, but it could potentially provide better results.

3.1.2.1 Predicting One Future Position at a Time

The first approach involved training an LSTM model to predict the future position of the sperm cell one step at a time.

To create a dataset for this approach, I used the segmented paths in the relative coordinate system. Each sequence was split into an input sequence and a target sequence, with the input sequence containing the first $n - 1$ points and the target sequence containing the last point. The model was trained to predict the target sequence based on the input sequence.

To maintain simplicity and avoid unnecessary complexity, the model employed a straightforward architecture comprising an input layer, an LSTM layer, and an output layer. The input layer accepted a sequence of $n-1$ points, with each point represented by relative coordinates $(\Delta x, \Delta y)$. The LSTM layer, consisting of 64 units, processed the input sequence to capture temporal dependencies. Finally, the output layer predicted the future position of the sperm cell, utilizing relative coordinates $(\Delta x, \Delta y)$ as the output representation.

I trained the model using the mean squared error (MSE) loss function, however, minimizing the MSE on a single future position might not mean that the model is learning the underlying patterns in the data. Therefore, I evaluated the model mainly qualitatively, by visualizing the predicted paths and analyzing the model's performance. After predicting the first future position, the model's prediction was added to the input sequence, and the model predicted the next position based on the updated input sequence. I repeated this process until the desired number of future positions was predicted.

3.1.2.2 Predicting Multiple Future Positions at Once

The second approach involved training an LSTM model to predict k future positions of the sperm cell at once. This approach was more complex and required a different dataset structure and model architecture.

In order to create a dataset for this approach, I needed to increase the base lengths of the sequences mentioned in section 3.1.1.1 by k. This means that the input sequence contained n points, and the target sequence contained k future points.

The model architecture for this approach was more complex, consisting of an input layer, two LSTM layers, with the first LSTM layer having 64 units and the second LSTM layer having 32 units, and between them a dropout layer with a small amount of dropout. Between them, a dropout layer with a small amount of dropout was added to prevent overfitting. The output layer was a dense layer with $2k$ units, followed by a reshape layer that reshaped the output to have k points, each represented by relative coordinates $(\Delta x, \Delta y)$.

3.2 Exploration with non-Position-Based Features

Even though the position-based model is expected to perform well, it is impossible to predict a sudden change in the movement without any additional information. Therefore, it is necessary to explore other features that could be used to improve the prediction accuracy of the model. In this section, I explore the use of image and video-based features to predict the direction and rotation of sperm cells.

3.2.1 Isolating Individual Sperm Cells

To analyze individual sperm cells and extract meaningful features, I preprocessed the dataset to isolate smaller videos containing single sperm cells. This process involved leveraging the paths gathered previously to segment the original videos into smaller clips, each focusing on a single sperm cell's trajectory. These individual sperm cell videos were standardized to a uniform size of 300 by 300 pixels, ensuring consistency and facilitating subsequent analysis. By isolating these standardized videos, I could effectively study the features and characteristics of each cell, faccilitating more granular analysis and insights into sperm behavior. An example frame of such a video is shown in figure 3.3.

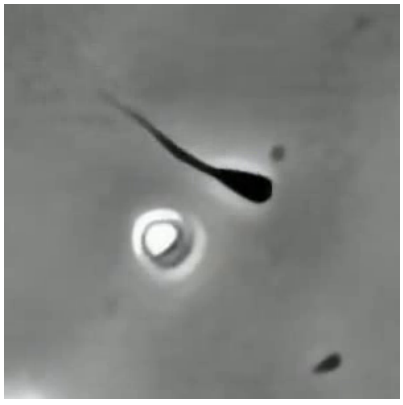


Figure 3.3 Single sperm cell

The original dataset provided by UT and UW contains videos with a notably noisy background, which presents a challenge for extracting meaningful features. This noise includes a multitude of small particles that could potentially be mistaken for sperm cells by the model. Therefore, it is imperative to preprocess the videos to enhance the visibility and clarity of the sperm cells in each frame, ensuring accurate feature extraction and model performance.

3.2.1.1 Segmenting the Single Sperm Cell Videos

To address this issue, I employed segmentation techniques to isolate the sperm cells from the background noise. By applying segmentation algorithms, I effectively distinguished the sperm cells from the surrounding particles, enhancing the quality of the data and facilitating more accurate feature extraction. This preprocessing step was crucial for ensuring the model's ability to focus on the sperm cells themselves.

Thresholding, as discussed in Section 2.5.1, serves as a fundamental image processing technique utilized to distinguish objects from the background in video recordings. By converting the video frames into a binary format based on pixel intensity values, thresholding enables the separation of foreground objects, such as sperm cells, from the background. This critical preprocessing step facilitates subsequent analysis tasks by isolating the sperm cells for further tracking and feature extraction.

However, finding the optimal threshold value consistently across al videos can be challengind due to variations in lighting conditions and background noise. To address this issue, I employed a semi-automated approach, where the threshold value was adjusted based on the size of the contours detected in the frame. By observing the videos and experimenting with different contour sizes, I determined that a minimum contour size of 500 pixels squared was suitable for identifying sperm cells.

By assuming that the sperm cells are the largest objects in the frame, and each frame contains a sperm cell, since the videos are centered around the sperm cells, a threshold value can be determined by finding the smallest possible value that results in a contour of at least 500 pixels squared in every single frame of the video.

In cases where multiple sperm cells were present in a frame, I selected the closest contour to the frame's center as the sperm cell of interest. This decision was based on the assumption that the sperm cell closest to the frame's center was the focus of the video.

In cases where multiple contours exceeded the minimum size threshold, thus indicating the presence of multiple sperm cells, I chose to always select the contour closest to the frame's center as the sperm cell of interest. This decision was based on the assumption that the sperm cell closest to the frame's center was the focus of the video.

Despite efforts to automate the segmentation process, imperfections in the tracking system and data noise sometimes hindered accurate isolation of the sperm cell of interest. Manual inspection of the videos was therefore necessary to ensure data quality. By manually verifying the segmentation results, I created a reliable dataset of 200 videos, each containing a single sperm cell isolated from the background noise. This dataset served as the foundation for subsequent feature extraction and analysis.

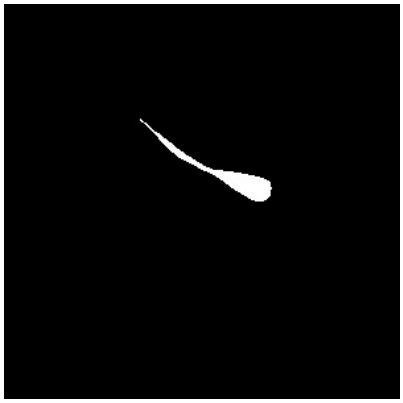


Figure 3.4 Single sperm cell after thresholding

3.3 Image-Based Features

A machine learning model learning to understand the direction of sperm cells based on a single image coudl be a valuable step towards a better understanding of how sperm cells move. In this secion, I explore the use of image-based features to predict the direction of sperm cells.

3.3.1 The Direction of Sperm Cells

Before developing a model that predicts the direction of sperm ells, it is necessary to decide how the direction of a sperm cell should be represented. When trying to calculate the direction at the $-$ time stamp of a path P of length n,

Before developing a model that predicts the direction of sperm cells, it is necessary to decide how the direction os a sperm cell should be represented, and how it should be calculated. In this section I outline the process of selecting the optimal approach for calculating the direction of a sperm cell at timestamp i along a path P of length n.

A logical initial consideration involves utilizing the adjacent points P_i and P_{i-1}, and calculating the angle of the vector connecting these two points with respect to the x-axis. Therefore, the direction of the sperm cell at time i can be calculated as:

$$\text{direction}(P_i) = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)$$

However, this approach has a few limitations. Due to the discrete nature of the data, the angle of the path often changes abruptly from one frame to the next. Additionally, since the tracking system calculates the position of the sperm cell based on the center of mass of the sperm cell, the path of the sperm cell is not always smooth. This can lead to a noisy representation of

the direction of the sperm cell. An example of such path, where the direction of the sperm cell would fluctuate significantly, is shown in figure 3.5.

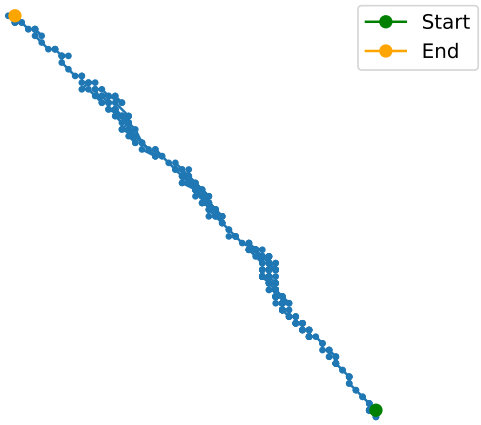


Figure 3.5 Noisy path of a sperm cell

Using a slidiwn window approach to smooth the path of the sperm cell before calculating the direction could be a potential solution to this problem. However, through trial and error, I determined that even with highly noisy paths, similar to the one shown in figure 3.5, the sliding window approach was not sufficient in smoothing the path.

To overcome these challenges, a more robust approach involves smoothing the path of the sperm cell before calculating its direction. This is achieved by fitting a spline curve to the path, a technique implemented using the splprep function from the SciPy library[1]. The splprep function enables the fitting of spline curves to the data points, with the smoothing parameter (s) controlling the level of smoothness in the resulting curve. By smoothing the path, the spline curve provides a clearer representation of the sperm cell's trajectory, facilitating a more accurate calculation of its direction.

[1]SciPy: Scientific Library for Python. Jones, E., Oliphant, T., Peterson, P., et al. (2001). Retrieved from `https://www.scipy.org/`

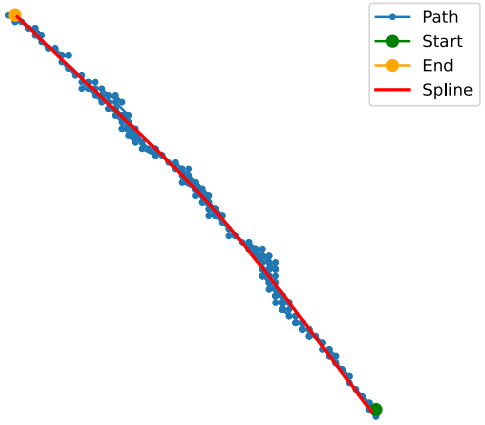
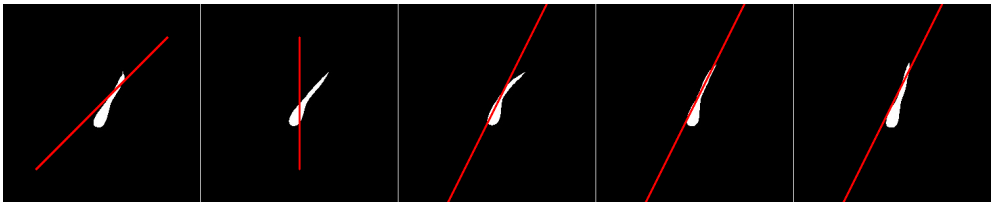
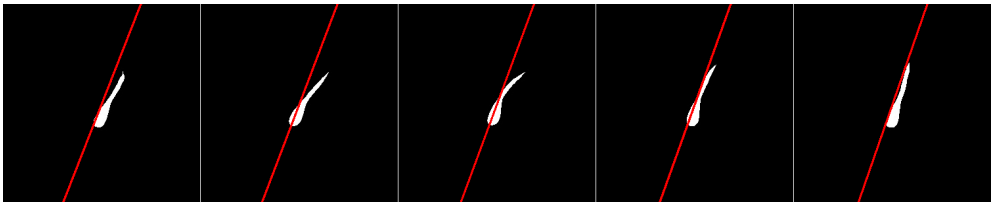


Figure 3.6 Noisy path of a sperm cell smoothed using a spline curve

As seen in figure 3.6, the spline curve provides a more accurate representation of the direction of the sperm cell, enabling the model to learn more effectively. The direction of the sperm cell can then be calculated by taking the angle of the tangent to the spline curve at the current position.



(a) Direction of the sperm cell before smoothing the path



(b) Direction of the sperm cell after smoothing the path

Figure 3.7 Comparison of the direction of the sperm cell before and after smoothing the path

Figure 3.7b illustrates the direction of the sperm cell before and after smoothing the path, highlighting the significant improvement in the direction's consistency and accuracy achieved through the spline curve. By leveraging this approach, I obtained a more reliable representation of the sperm cell's direction, enhancing the quality of the data and facilitating more accurate feature extraction and model training.

3.3.2 Classification of Sperm Cell Direction

Even though the direction of the sperm cell is a continuous variable, thus making predicting the direction of sperm cells a regression task, classification tasks are generally easier to interpret and evaluate. Therefore, I decided to turn this into a classification task.

To make predicting the direction of sperm cells a classification task, the direction angles of the sperm cells will be discretized into 12 classes. The angles range from 0 to 360 degrees, therefore each class will represent a 30-degree interval. The model will be trained to predict the class of the angle of the sperm cell based on the image of the sperm cell.

To make predicting the direction of sperm cells a classification task, I discretized the angles of the sperm cells into 12 classes. The angles range from 0 to 360 degrees, therefore each class represents a 30-degree interval. The model is trained to predict the class of the angle of the sperm cell based on the image of the sperm cell.

3.3.2.1 Creating the Dataset

I created the dataset by extracting the frames from the videos created in Section 3.2.1.1. To reduce the complexity of the task, the frames were resized to 64 by 64 pixels. The angles of the sperm cells were calculated using the method describen in Section 3.3.1. The angles were then discretized into 12 classes, each representing a 30-degree interval.

Since the difference between two consecutive frames is often small, I augmented the dataset by rotating the frames at a random angle between 0 and 360 degrees, and adjusted the angles of the sperm cells accordingly. This approach ensured a more diverse and robust dataset.

I created a separate dataset for testing the model by putting 30 videos aside, and not using them in the training process. This dataset will ensure that the model is evaluated on unseen data and that the evaluation is representative of the model's performance on a real-world scenario.

3.3.2.2 Model Architecture

As stated in section 2.3.4 CNNs are a powerful tool for image classification tasks. In this case, the task is to classify the direction of sperm cells based on images. Therefore, a CNN-based model is the most suitable choice for this task.

Table 3.1 Layer-wise details of the model architecture

Layer (type)	Output Shape
Conv2D	(None, 62, 62, 32)
MaxPooling2D	(None, 31, 31, 32)
BatchNormalization	(None, 31, 31, 32)
Conv2D	(None, 29, 29, 64)
MaxPooling2D	(None, 14, 14, 64)
BatchNormalization	(None, 14, 14, 64)
Conv2D	(None, 12, 12, 128)
MaxPooling2D	(None, 6, 6, 128)
BatchNormalization	(None, 6, 6, 128)
Flatten	(None, 4608)
Dense	(None, 64)
Dense	(None, 12)

Table 3.1 shows the architecture of the model. The model consists of a series of convolutional layers followed by max-pooling and batch normalization layers. The first convolutional layer with 32 filters and a kernel size of (3, 3) extracts basic features from the input images using the ReLU activation function. Max-pooling layers with a pool size of (2, 2) reduce the spatial dimensions of the feature maps, helping in capturing important patterns while reducing computation. Batch normalization layers normalize the activations of the previous convolutional layers, stabilizing the learning process and improving convergence. Successive convolutional layers with 64 and 128 filters further extract higher-level features from the input data. After the final convolutional layer, the feature maps are flattened into a one-dimensional vector and passed through fully connected layers. The dense layers with 64 neurons and ReLU activation function further process the features, followed by a dense layer with 12 neurons and softmax activation, which outputs the probability distribution over the 12 classes, representing the predictions for each input image.

The model is trained using the Adam optimizer and categorical cross-entropy loss function. The results of the task are discussed in Section 4.2.

3.4 Video-Based Features

Predicting the direction of sperm cells based on single images is an important step toward understanding how sperm cells move. However, there is only so much information that can be extracted from a single image. Since sperm cells move dynamically, it makes sense to explore the use of multiple frames to learn more about the movement of sperm cells. In this section, I explore the use of video-based features to predict the direction of sperm cells.

3.4.1 Objective Definition: Predicting Sperm Cell Rotation

Transitioning from image-based to video-based features, the focus shifted towards a more dynamic aspect of sperm cell behavior: predicting rotational movement across sequential frames. This means, that a sequence of n frames is used to predict the change of angle of the sperm cell from the first to the last frame.

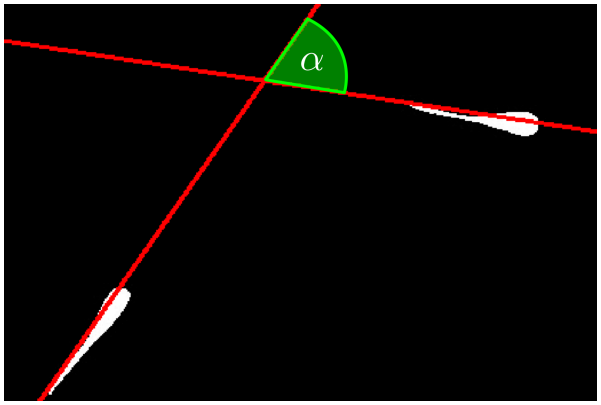


Figure 3.8 Rotational angle of a sperm cell

Figure 3.8 illustrates the rotational angle α of a sperm cell between the first and last frame of a sequence. Note that the positions in the image are only for illustrative purposes, as the angle is not affected by the positions, but only by the directions of the sperm cell.

While the task may seem straightforward at first glance, predicting rotation direction holds intrinsic value. Discerning how sperm cells rotate over time provides valuable insights into their behavior and underlying dynamics. This understanding is crucial for predicting future movements and lays the groundwork for deeper analysis and interpretation of sperm cell behavior.

In essence, while the task of predicting rotation may appear simplistic on the surface, its significance lies in its contribution to the broader goal of comprehending and interpreting the intricate movements of sperm cells.

3.4.1.1 Creating the Dataset

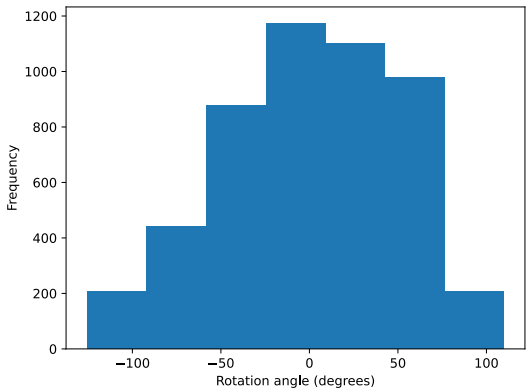


Figure 3.9 Distribution of rotational angles

Figure 3.9 illustrates the distribution of rotational angles across the dataset. The angles range from approximately -120 to 110 degrees, with a notable concentration around 0 degrees. The uneven distribution of angles could pose challenges for the model, hence I set a threshold of -60 to 60 degrees to filter out outliers and ensure the dataset's quality and reliability. The distribution of angles after filtering is shown in figure 3.10.

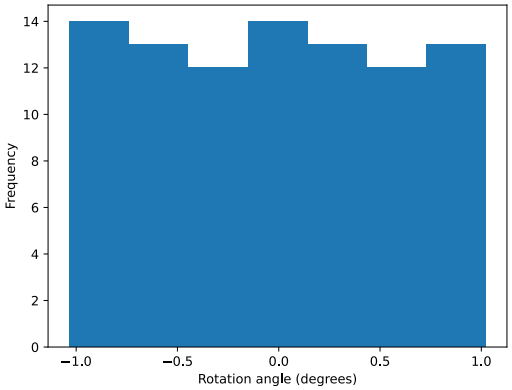


Figure 3.10 Distribution of rotational angles after filtering

The objective of this task is to predict the rotational angle of the sperm cell across a sequence of frames. Similarly to Section 3.3.2, I discretized the angles into 7 classes, each representing a 17-degree interval. I chose the number of classes based on experimentation to ensure a balance between granularity and model complexity. The decision to make the number of classes an odd

number was deliberate, as it allowed for a central class representing the sperm cell moving in a straight line, thus providing a more nuanced classification framework.

The dataset of video sequences was constructed by segmenting the videos captured in Section 3.2.1.1 into sequences of 100 frames. I segmented the videos using a sliding window technique, with each sequence comprising 100 frames, providing a comprehensive temporal representation of the sperm cell's movement. The choice of sequence length was guided by a pragmatic assessment of the data's characteristics. It was essential to strike a balance between informativeness and dataset size. Longer sequences offer richer insights but risk excluding shorter paths that do not span the full 400 frames of the original video. Therefore, by selecting a sequence length of 100 frames, I aimed to capture key movement dynamics while ensuring the dataset remained sufficiently diverse for effective model training.

To ensure the model's robustness and reliability, I augmented the dataset by rotating the sequences by a random angle between 0 and 360 degrees. This was necessary, as the neighboring sequences in the dataset share 99 frames in common, potentially leading to overfitting. By rotating the sequences, I introduced diversity and variation, enhancing the model's capacity to learn the underlying patterns and behaviors within the data.

Since the direction of a sperm cell between two frames changes only slightly, some frames are redundant and do not provide any additional information. To reduce the complexity of the task, I decided to introduce a step size of three different sizes: 5, 10, and 20. This means that the model will be trained to predict the rotation of the sperm cell based on every 5th, 10th, or 20th frame of the sequence. This approach reduces the complexity of the task while still providing valuable insights into the rotational behavior of sperm cells.

Each frame was resized to two different sizes: 64 by 64 pixels and 32 by 32 pixels. The smaller size was chosen to reduce the complexity of the task and to speed up the training process. The larger size was chosen to provide more detailed information to the model.

Therefore, six different datasets were created, each with a different step size and frame size, to explore the impact of these parameters on the model's performance. The comparison of the results achieved using different datasets is discussed in Section 4.3.

3.4.1.2 Model Development

In this subsection, I outline the process of developing the model for predicting sperm cell rotation from video segments. The architecture is constructed using the Keras Sequential API, comprising layers aimed at capturing temporal dependencies and spatial features within the video data.

Layer (type)	Output Shape
TimeDistributed(Conv2D)	(None, 10, 30, 30, 64)
TimeDistributed(MaxPooling2D)	(None, 10, 15, 15, 64)
TimeDistributed(Dropout)	(None, 10, 15, 15, 64)
TimeDistributed(Flatten)	(None, 10, 14400)
LSTM	(None, 10, 256)
Dropout	(None, 10, 256)
LSTM	(None, 128)
Dropout	(None, 128)
Dense	(None, 128)
Dropout	(None, 128)
Dense	(None, 7)

Table 3.2 Model Architecture

The model comprises a sequence of layers tailored for video data analysis. Initially, a TimeDistributed layer wraps a Conv2D layer with 64 filters and a (3, 3) kernel size, to extract to extract foundational spatiotemporal features from the input frames. Subsequently, a TimeDistributed layer followed by MaxPooling2D reduces the spatial dimensions of the feature maps, enhancing computational efficiency. Dropout regularization with a dropout rate of 0.25 is applied to mitigate overfitting. The flattened feature maps are passed through an LSTM layer with 256 units and return sequences enabled, allowing the model to capture temporal dependencies in the video sequences. Another Dropout layer is employed for further regularization. Following this, an LSTM layer with 128 units refines the temporal representations, followed by Dropout regularization. The output is then processed through a Dense layer with 128 neurons and ReLU activation to extract high-level features. Dropout regularization with a dropout rate of 0.5 is applied for regularization. Finally, a Dense layer with 7 neurons and softmax activation outputs the probability distribution across the defined number of bins, representing predictions for each input frame. The results of the task are discussed in Section 4.3.

Chapter 4
Results and Evalution

In this chapter, I outline the results of my research into sperm cell movement. I present the findings of my computational analyses and experimental observations in predicting the future path of sperm cells, classifying their directional orientation, and predicting their rotation from video data. The following sections provide a detailed overview of these findings, contributing to a broader understanding of sperm cell behavior and motility.

4.1 Path Prediction with LSTM

In this section, I present the results of the LSTM-based models trained to predict the future path of sperm cells based on their previous movements. Three different LSTM models were trained and evaluated, each with a different input sequence length.

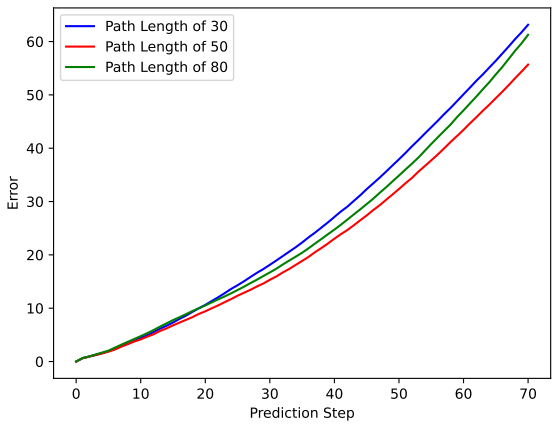


Figure 4.1 Comparison of LSTM models with different input sequence lengths

Figure 4.1 shows the performance of the three LSTM models when predicting the path of the sperm cell up to 70 frames in the future. Each model was evaluated on 100 test sequences, and the average distance between the predicted and actual paths was calculated for each time step. The error at each time step is calculated as the Euclidean distance between the predicted and actual positions, and the average error over all test sequences is shown in the figure.

As shown in the figure, the model with a medium sequence lenght of 50 frames performed the best, which promotes the idea that a balance between the amount of information and the complexity of the model is crucial for accurate predictions. The model with the shortest sequence length of 30 frames performed the worst, indicating that it did not have enough information to make accurate predictions over a longer period of time. The model with the longest sequence length of 80 frames also performed worse than the medium-length model, suggesting that the points further back in time were less relevant for predicting the future path.

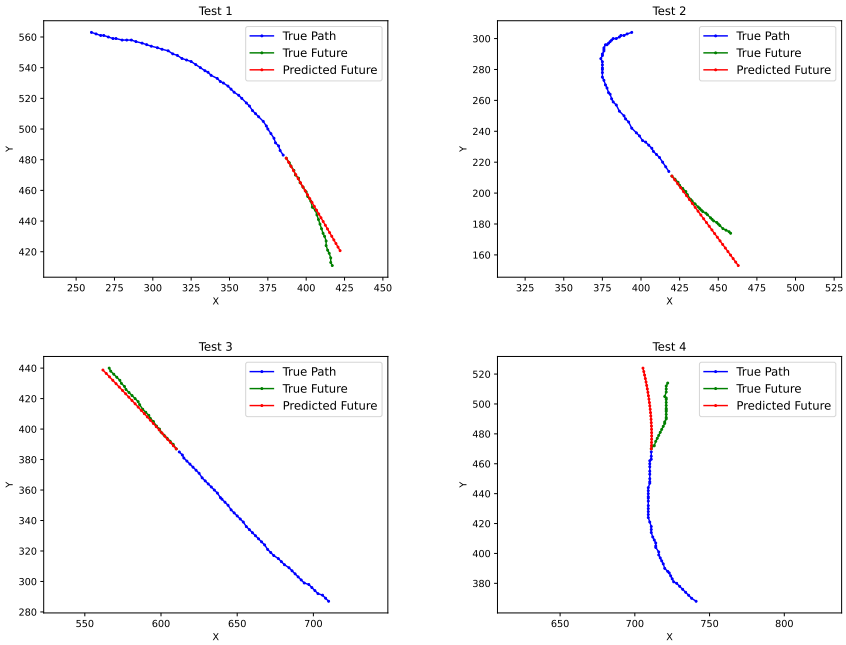


Figure 4.2 Sample path predictions of the LSTM models

Figure 4.2 shows sample path predictions of the LSTM models with different input sequence lengths, predicting the path of the sperm cells 25 frames into the future. The models were able to accurately predict the future path of the sperm cells in most cases, capturing the general direction and movement patterns of the cells. As seen in test path number 4, the models struggled with irregular movement patterns and sudden changes in direction. However, the

models were able to capture the general movement patterns of the sperm cells, indicating that they were able to learn the underlying dynamics of the cells from the input sequences.

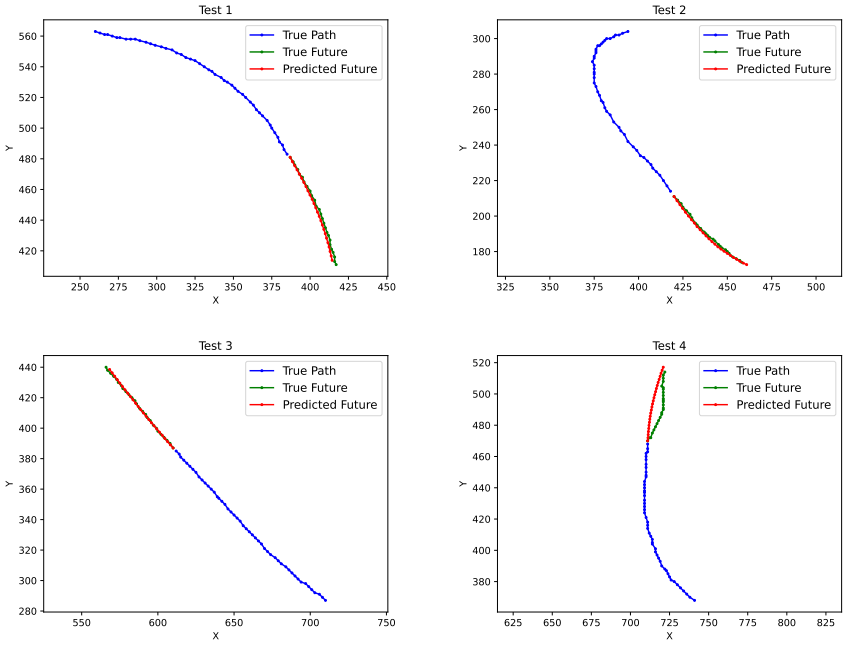


Figure 4.3 Sample path predictions of the LSTM models

On the other hand, Figure 4.3, which shows the predictions of the LSTM model predicting the path 25 frames into future in a single prediction, shows that the model was able to accurately predict the future path even in cases where the movement didn't follow a straight line. The model was able to capture the general movement patterns of the sperm cells, indicating that it was able to learn the underlying dynamics of the cells from the input sequences.

Therefore, predicting the future path of sperm cells all at once seems to be more successful than predicting the path incrementally over time. This is likely due to the fact that the model can learn the general movement patterns of the sperm cells from the input sequences, which allows it to make more accurate predictions over a longer period of time.

4.2 Predicting Direction from Image Data

:

In this section, I present the results of the CNN model trained to classify the directional orientation of sperm cells based on single images that I described in Section 3.3.

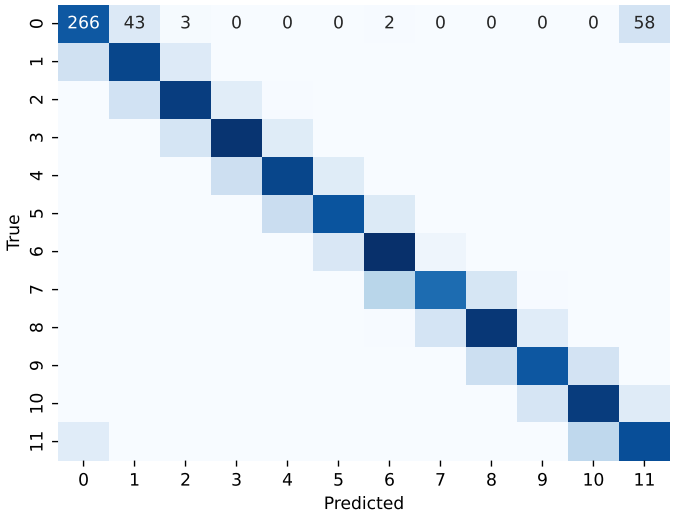


Figure 4.4 Confusion matrix of the CNN model

The confusion matrix in Figure 4.4 shows that the model was able to correctly classify the majority of the images, with most of the errors occuring between neighboring classes.

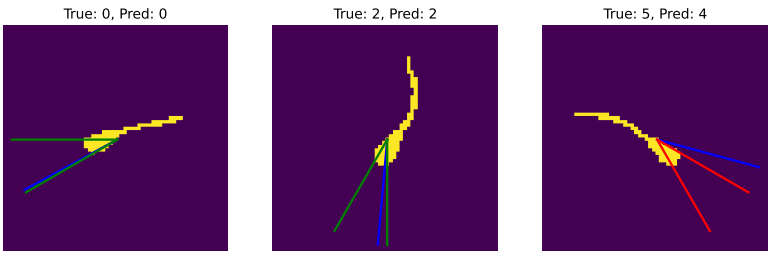


Figure 4.5 Sample predictions of the CNN model

Figure 4.5 shows sample predictions of the CNN model on the test set. The blue line represent the true direction of the sperm cell, while the green,

respectively red lines represent the interval of the predicted class. The lines are green if the predicted class is correct, and red otherwise.

The third image in the figure shows an example of a misclassification, where the model predicted the direction of the sperm cell incorrectly. This is a case where the model meets its limitations, as the single image does not provide enough information to accurately classify the direction of the sperm cell.

4.2.1 Explaining the Model's Predictions

To better understand how the CNN model makes its predictions, I used the algorithms introduced in Section 2.4 to generate heatmaps of the input images. The heatmaps show the regions of the image that the model focuses on when making its predictions, providing insight into the features that are important for classifying the direction of the sperm cell.

I picked 4 images from the test set and generated heatmaps with all three algorithms for each of them. The heatmaps are shown in Figure 4.6.

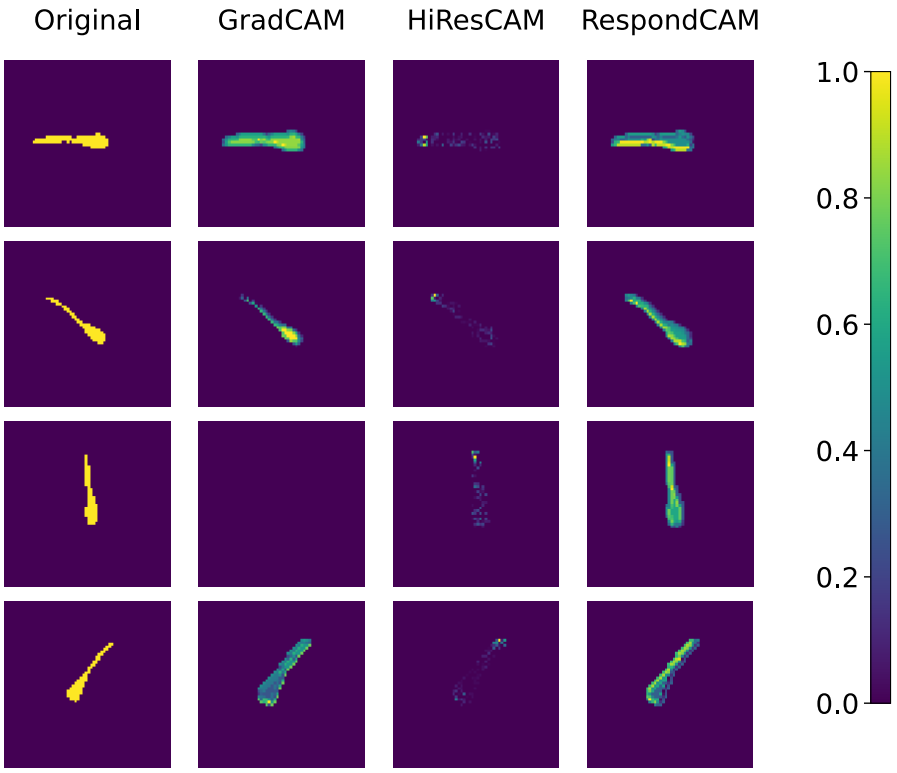


Figure 4.6 Heatmaps of the input images generated by the CNN model

The Grad-CAM algorithm, despite its wide use, exhibited inconsistency in generating heatmaps for the sperm cell direction classification task. In the first two images, the heatmap primarily highlighted the inner part of the sperm cell, suggesting a focus on internal features. However, in the third image, no heatmap was generated at all, indicating that the model did not strongly rely on any specific region of the image for its decision. In the final image, a heatmap was produced, but the focus was on a line along the outer edge of the sperm cell, deviating from the patterns observed in the previous images.

This inconsistency in Grad-CAM's heatmap may be attributed to inherent limitations or challenges within the algorithm itself, as discussed in Section 2.4.

While Grad-CAM remains a popular choice for explainability, these findings suggest caution in its interpretation, particularly in scenarios where consistency and reliability in heatmap generation are crucial.

In contrast, both HiResCAM and RespondCAM provided more consistent heatmaps across all images. HiResCAM consistently highlighted the tail of the sperm cell, indicating its significance in the model's decision-making process. Similarly, RespondCAM consistently focused on the entire sperm cell, with emphasis on a straight line traversing across the cell, possibly indicative of a directional cue utilized by the model.

In contrast, both HiResCAM and RespondCAM provided more consistent heatmaps across all images. HiResCAM consistently highlighted the tail of the sperm cell, indicating its significance in the model's decision-making process. Similarly, RespondCAM consistently focused on the entire sperm cell, with emphasis on a straight line traversing across the cell, possibly indicative of a directional cue utilized by the model.

The most important aspect to note is that both algorithms demonstrate that the model makes its decisions logically and in a manner that aligns with human intuition. This suggests that the model is effectively leveraging meaningful features in the input data to make accurate predictions.

4.3 Predicting Rotation from Video Data

In this section, I present the results of the rotation prediction models introduced in Section 3.4.

Table 4.1 Comparison of different input resolutions and sequence lengths for rotation prediction models

Model	20 frames	10 frames	5 frames
32 pixel images	52.4%	43.2%	30.2%
64 pixel images	42.0%	41.9%	37.2%

Table 4.1 shows how different input video resolutions and amounts of frames in a sequence affect the performance of the rotation prediction models. I evaluated the models on a test set of 30 videos that were split into over 1,500 sequences, each representing 100 frames of the original video. The model with 32 pixel images and 20 frames in a sequence performed the best, achieving an accuracy of 52.4%, while the model with 32 pixel images and 5 frames in a sequence performed the worst, indicating that no size of the input images is superior to the other, but rather a balance between the amount of information and the complexity of the model is crucial for accurate predictions.

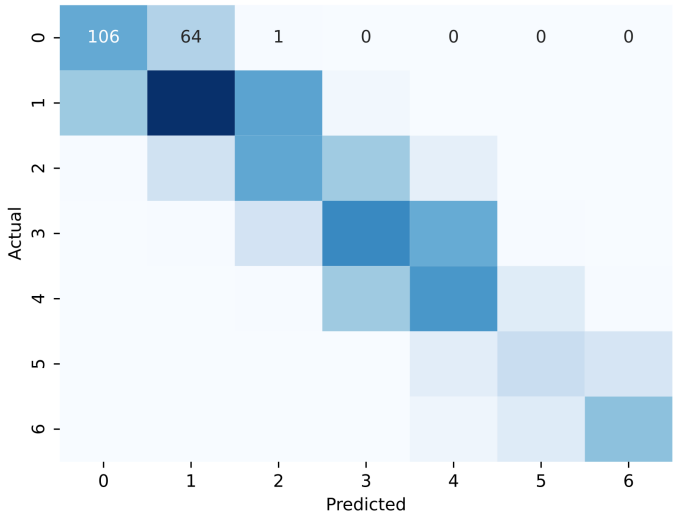


Figure 4.7 Confusion matrix of the best performing rotation prediction model

Even though an accuracy of 52.4% is not very high, the confusion matrix in Figure 4.7 shows that the model was able to learn the general patterns of the sperm cell rotations. The model was either able to correctly predict the rotation of the sperm cell or predict a neighboring class in over 95% of the

cases.

This result is promising, as it indicates that the model was able to learn the underlying dynamics of the sperm cell rotations from the input sequences. The model could be further improved by providing it with a larger, more diverse dataset of sperm cell videos, as well as by fine-tuning the hyperparameters of the model.

Unfortunately, creating a larger dataset was not feasible in the scope of this thesis, as the videos had to be checked manually. However, the results of the rotation prediction models show that it is possible to predict the rotation of sperm cells from the video data, which could be useful to study the behavior of sperm cells in more detail.

Chapter 5
Conclusion

In this thesis, I have explored the use of machine learning techniques to predict the future path of sperm cells, classify their directional orientation, and predict their rotation from video data. I have presented the results of my research, which demonstrate the feasibility of using machine learning models to analyze sperm cell behavior and motility.

I developed an LSTM-based model to predict the future path of sperm cells based on their previous movements. I trained three different LSTM models with different input sequence lengths and evaluated their performance on a test set of 100 sequences. The best model learned to predict the path of the sperm cell up to 50 frames in the future without a significant loss of accuracy.

I also developed a CNN model to classify the directional orientation of sperm cells based on single images. The model achieved an accuracy over 70% on the test set, indicating that the model was able to accurately classify the direction of the sperm cells in most cases. I used explainability algorithms to generate heatmaps of the input images, providing insight into the features that are important for classifying the direction of the sperm cell.

Finally, I developed rotation prediction models to predict the rotation of sperm cells from video data. I evaluated the models on a test set of 30 videos and achieved an accuracy of 52.4%, indicating that the models were able to learn the general patterns of the sperm cell rotations. The models were either able to correctly predict the rotation of the sperm cell or predict a neighboring class in almost all cases, indicating that they were able to learn the underlying dynamics of the sperm cell rotations from the input sequences.

Overall, the results of my research demonstrate the potential of machine learning techniques to analyze sperm cell behavior and motility, opening up new avenues for research in this field. By combining computational analyses with experimental observations, researchers can gain a deeper understanding of sperm cell movement, paving the way for new discoveries and breakthroughs in the field of reproductive biology.

Bibliography

1. OTTL, Sandra; AMIRIPARIAN, Shahin; GERCZUK, Maurice; SCHULLER, Björn W. A Machine Learning Framework for Automatic Prediction of Human Semen Motility. *CoRR*. 2021, vol. abs/2109.08049. Available from arXiv: 2109.08049.
2. FERNÁNDEZ-LÓPEZ, Pol; GARRIGA, Joan; CASAS, Isabel; YESTE, Marc; BARTUMEUS, Frederic. Predicting fertility from sperm motility landscapes. *Communications Biology*. 2022, vol. 5, no. 1, p. 1027. ISSN 2399-3642. Available from DOI: 10.1038/s42003-022-03954-0.
3. NOY, Lioz; AUTHORS], [Other. Location Prediction of Sperm Cells Using Long Short-Term Memory Networks. *Advanced Intelligent Systems*. 2023, vol. 5, no. 9, p. 2300161. Available from DOI: DOI.
4. [N.d.].
5. ABRAHAM, Ajith. Artificial neural networks. *Handbook of measuring system design*. 2005.
6. SCHMIDT, Robin M. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. *arXiv*. 2019, vol. 1912.05911v1. Available from arXiv: 1912.05911v1 [cs.LG]. Version 1.
7. OLAH, Christopher. *Understanding LSTM Networks – colah's blog — colah.github.io* [https://colah.github.io/posts/2015-08-Understanding-LSTMs/]. 2015. [Accessed 25-04-2024].
8. NICHOLSON, Chris. *A Beginner's Guide to LSTMs and Recurrent Neural Networks* [https://skymind.ai/wiki/lstm]. 2019. Accessed: 03 May 2024.
9. O'SHEA, Keiron; NASH, Ryan. An Introduction to Convolutional Neural Networks. *CoRR*. 2015, vol. abs/1511.08458. Available from arXiv: 1511.08458.

10. CHRISTOPHER, Colah. *Conv Nets: A Modular Perspective - colah's blog* [`https://colah.github.io/posts/2014-07-Conv-Nets-Modular/`]. 2014-07. (Accessed on 05/16/2024).

11. XU, Feiyu; USZKOREIT, Hans; DU, Yangzhou; FAN, Wei; ZHAO, Dongyan; ZHU, Jun. Explainable AI: A brief survey on history, research areas, approaches and challenges. In: *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*. Springer, 2019, pp. 563–574.

12. DRAELOS, Rachel Lea; CARIN, Lawrence. *Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks*. 2021. Available from arXiv: `2011.08891 [eess.IV]`.

13. ZHAO, Guannan; ZHOU, Bo; WANG, Kaiwen; JIANG, Rui; XU, Min. Respond-cam: Analyzing deep models for 3d imaging data by visualizations. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I*. Springer, 2018, pp. 485–492.

14. HEINISUO, Olli-Pekka; TEAM, OpenCV. *OpenCV Python packages*. 2016. Available also from: `https://github.com/opencv/opencv-python`.

15. BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.

Medium Attachment

- src
  - path_prediction ..... Source codes of the path prediction
  - direction_prediction ..... Source codes of the direction prediction
  - rotation_prediction ..... Source codes of the rotation prediction
  - thesis ..... LaTeX source code
- text ..... Text of the thesis
  - thesis.pdf ..... PDF text of the thesis