

# HOI\_Bitirme\_19011046\_170119 09

*by Burak Erdilli*

---

**Submission date:** 21-Jan-2024 09:14PM (UTC+0300)

**Submission ID:** 2275042167

**File name:** HOI\_Bitirme\_19011046\_17011909.pdf (1.81M)

**Word count:** 4915

**Character count:** 27731

<sup>6</sup>  
REPUBLIC OF TURKEY  
YILDIZ TECHNICAL UNIVERSITY  
DEPARTMENT OF COMPUTER ENGINEERING



**SEGMENTATION OF SPERM AND NON-SPERM CELLS  
IN FULL SLICE OCULAR IMAGES WITH DEEP  
LEARNING NETWORKS**

19011046 – Burak Erdilli  
17011909 – Yusuf Elhaşaş

**SENIOR PROJECT**

Advisor  
Assist. Prof. Dr. Hamza Osman İLHAN

January, 2024



## **ACKNOWLEDGEMENTS**

---

To everyone who made this project possible, we would like to extend our sincerest gratitude and appreciation. We really appreciate the invaluable guidance and support that our consultant, Asst. Prof. Hamza Osman İLHAN, provided us with during the project. The level of expertise and accuracy of our work has really benefited from his viewpoint, critique, and knowledge.

Burak Erdilli  
Yusuf Elhaş

# 1 TABLE OF CONTENTS

---

<b>LIST OF ABBREVIATIONS</b>	v
<b>LIST OF FIGURES</b>	vi
<b>LIST OF TABLES</b>	vii
<b>ABSTRACT</b>	viii
<b>ÖZET</b>	x
<b>1 INTRODUCTION</b>	1
<b>2 PRE-REVIEW</b>	2
2.1 CNNs for Sperm Analysis . . . . .	2
2.1.1 Performance Metrics . . . . .	2
2.1.2 Framework Inaccessibility . . . . .	2
2.1.3 Metrics Bias . . . . .	3
2.2 Data Issues and Resolutions . . . . .	3
2.2.1 Dataset Scarcity . . . . .	3
2.2.2 Dataset Customization . . . . .	3
2.2.3 Class Definitions . . . . .	3
2.3 YOLOv8 in Sperm Analysis . . . . .	4
2.3.1 Real-time Efficiency . . . . .	4
2.3.2 Sperm Variation Analysis . . . . .	4
<b>3 FEASIBILITY</b>	5
3.1 Technical Feasibility . . . . .	5
3.1.1 Software . . . . .	5
3.1.2 Hardware . . . . .	5
3.2 Time Feasibility . . . . .	6
3.3 Legal Feasibility . . . . .	6
3.4 Economic Feasibility . . . . .	6
<b>4 SYSTEM ANALYSIS AND DESIGN</b>	7

4.1	Software Design . . . . .	7
4.1.1	Library Utilization . . . . .	7
4.1.2	Dataset . . . . .	7
4.1.3	Data Labeling and Preparation . . . . .	8
4.1.4	YOLOV8 Model . . . . .	9
4.2	Input-Output Design . . . . .	9
<b>5</b>	<b>Implementation</b>	<b>17</b>
5.1	Training Dataset . . . . .	17
5.1.1	Training a Model to Detect 1 Class . . . . .	17
5.1.2	Training a Model to Detect 18 Classes . . . . .	19
<b>6</b>	<b>Experimental Results</b>	<b>22</b>
6.1	Testing Results of The Models . . . . .	22
6.1.1	Model Results for 1 Class . . . . .	22
6.1.2	Model Results for 18 Class . . . . .	23
<b>7</b>	<b>Performance Analysis</b>	<b>25</b>
<b>8</b>	<b>Conclusion</b>	<b>26</b>
	<b>References</b>	<b>27</b>
	<b>Curriculum Vitae</b>	<b>28</b>

## LIST OF ABBREVIATIONS

---

YOLOv8	You Only Look Once, version(8)
TFLite	TensorFlow Lite
React-Native	JavaScript framework for building cross-platform nativ mobile apps
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
CLI	Command Line Interface
GPU	Graphics Processing Unit
CPU	Central Processing Unit
OS	Operating System
RAM	Random Access Memory
GB	Gigabyte
TB	Terabyte
MI	Xiaomi Mi

## 1 LIST OF FIGURES

---

Figure 3.1 Gantt Diagram of the Project . . . . .	6
Figure 4.1 Dataset Tree . . . . .	8
Figure 4.2 Selecting the Type of Patient . . . . .	10
Figure 4.3 The List of Existing Patients . . . . .	11
Figure 4.4 Adding a new patient . . . . .	12
Figure 4.5 Choosing a Sample and Sending It to the Server . . . . .	13
Figure 4.6 Receiving the Result From Server by Using 1-Class Model . . . . .	13
Figure 4.7 Receiving the Result From Server by Using 18-Class Model . . . . .	14
Figure 4.8 Patient Folders Listing . . . . .	15
Figure 4.9 Saved Single Class Results Inside a Patient's Folder . . . . .	15
Figure 4.10 Saved 18 Class Results Inside a Patient's Folder . . . . .	16
Figure 5.1 Confusion Matrix . . . . .	18
Figure 5.2 Various Training Metrics . . . . .	18
Figure 5.3 Predictions of the Model for a Validation Set . . . . .	19
Figure 5.4 Confusion Matrix . . . . .	20
Figure 5.5 F1-Confidence Curve . . . . .	20
Figure 5.6 Various Training Metrics . . . . .	21
Figure 5.7 Predictions of the Model for a Validation Set . . . . .	21
Figure 6.1 Histogram of Detections Using the Model Trained for Single Class Detection . . . . .	23
Figure 6.2 Example of Detections Using the Model Trained for Single Class Detection . . . . .	23
Figure 6.3 Histogram of Detections Using the Model Trained for Detecting 18 Classes . . . . .	24
Figure 6.4 Example of Detections Using the Model Trained for Detecting 18 Classes . . . . .	24

## **LIST OF TABLES**

---

Table 2.1 Sperm cell classes used in the study . . . . .	4
Table 3.1 Personal Computers . . . . .	6
Table 7.1 Results of Model 1 (Trained With One Class) and Model 2 (Trained With 18 Classes) After Predicting Same 500 Test Images	25

## ABSTRACT

---

# Segmentation of Sperm and Non-Sperm Cells in Full Slice Ocular Images with Deep Learning Networks

Burak Erdilli  
Yusuf Elhaşş

<sup>1</sup> Department of Computer Engineering  
Senior Project

Advisor: Assist. Prof. Dr. Hamza Osman İLHAN

This study highlights the shortcomings of conventional sperm cell counting and classification techniques in the context of infertility assessment, highlighting their laboriousness and error-proneness. In order to automate and improve the effectiveness, accuracy, and dependability of sperm cell analysis, the research investigates the application of YOLOv8, an advanced deep learning-based object detection model, in order to get over these constraints. The research carefully selects 4955 sperm cell ocular images from a wide range of patient types. The YOLOv8 model is trained at the center of the study, utilizing the large dataset to provide a detailed understanding of sperm cell variances. After training successfully, the TensorFlow Lite (TFLite) format is created to use as the model, and it is integrated into an Android application using React. React-Native integration guarantees a cross-platform solution with a user-friendly interface, improving usability. The Android application allows for real-time sperm cell analysis at the point of care, powered by the trained YOLOv8 model. The combination of cutting-edge technology and user-centered design offers researchers and doctors a useful tool for quick and precise evaluation of sperm cell shape. This comprehensive approach combines YOLOv8, careful dataset selection, TFLite optimization, and React-Native development in an attempt to revolutionize sperm cell identification and classification in infertility diagnosis. The multidisciplinary synergy demonstrates how cutting-edge technology has the potential to transform medical diagnostics, especially in the area of reproductive health, and ultimately enhance patient outcomes and treatment.

**Keywords:** YOLOv8, Sperm Cells, NON-Sperm Cells, TFLite, CNN, CUDA, React Native, Deep Learning Networks

## ÖZET

# Tam Çerçeve (Full Slice) Oküler Görüntüleri İçerisinde Sperm ve Sperm Olmayan Hücrelerin Derin Öğrenme Ağları ile Segmentasyonu

Burak Erdilli  
Yusuf Elhaşaş

Bilgisayar Mühendisliği Bölümü  
Bitirme Projesi

Danışman: Dr. Ögr. Üyesi Hamza Osman İLHAN

Bu çalışma, kısırlık değerlendirmesi bağlamında geleneksel sperm hücresi sayımı ve sınıflandırma tekniklerinin eksikliklerini vurgulayarak, bu yöntemlerin zahmetlerine ve hata eğilimlerine çözüm oluşturmayı hedefler. Sperm hücresi analizinin etkinliğini, doğruluğunu ve güvenilirliğini otomatikleştirmek ve artırmak amacıyla, çalışmadımız bu kısıtlamaları aşmak için gelişmiş bir derin öğrenme tabanlı nesne tespiti modeli olan YOLOv8'in uygulanmasını içermektedir. Araştırmamız geniş bir hasta tipi yelpazesinden 4955 sperm hücresi oküler görüntüsünü içeren bir datasete sahiptir. YOLOv8 modeli başarılı bir şekilde eğitimden sonra uygun formatı oluşturular ve React kullanılarak bir Android uygulamasına entegre edilir. React-Native entegrasyonu, kullanılabilirliği artırarak kullanıcı dostu bir arayüz ve çapraz platform uyumluluğu sağlar. Android uygulaması, eğitilmiş YOLOv8 modeli tarafından desteklenen gerçek zamanlı sperm hücresi analizi yapma imkanı sunar. Bu kapsamlı yaklaşım, YOLOv8, isabetli işaretlenmiş eğitim veri setini, model optimizasyonunu ve React-Native geliştirmesini bir araya getirerek infertilite teşhisinde sperm hücresi tanımlama ve sınıflandırmayı ileri taşımayı amaçlamaktadır.

**Anahtar Kelimeler:** YOLOv8, Sperm Hücreleri, Non-Sperm Hücreler, CNN, CUDA TFLite, REACT NATIVE, Derin Öğrenme Ağları

# 1

## INTRODUCTION

---

Traditional methods of counting and classifying sperm cells are often tedious, time-consuming, and prone to human error. Providing the best treatment possible for infertile patients may be more challenging due to these drawbacks, which might lead to inaccurate and inconsistent diagnoses [1][2]. In order to overcome these constraints, this work explores the potential of YOLOv8, an advanced deep learning-based object identification model. We want to automate and simplify the identification and classification of sperm cells by utilizing the capabilities of this sophisticated model, making it possible for more effective, precise, and trustworthy infertility diagnosis.

Dataset of 4955 sperm cell ocular images forms the basis of our research. This extensive collection of images, which spans a range of patient demographics, offers a detailed representation of the characteristics and look of sperm cells. To ensure the integrity and correctness of the dataset, every picture is meticulously annotated, linking individual sperm cells to their corresponding frames, accurate positional coordinates, and given classifications.

The research uses React-Native to construct a cross-platform solution that works with a variety of mobile devices. The declarative programming approach of React-Native makes it easier to create an interface that is clear and easy to use, which improves the usefulness of the application even more. [3]. Real-time sperm cell identification and classification results are displayed to the user in a seamless manner thanks to the seamless integration with TensorFlow Lite, which facilitates fluid interaction between frontend components, React elements, and the backend deep learning model. The mobile application offers a useful tool for physicians and researchers to effectively and precisely examine sperm cell shape and help improve infertility diagnosis.

## 2

### PRE-REVIEW

---

Since the topic of automated sperm cell categorization and identification utilizing ocular image analysis has garnered a lot of attention, researchers are looking into various machine learning approaches [4]. This chapter presents a comprehensive review of the literature to emphasize the significance of the chosen YOLOv8 technique and to clarify the challenges and variations in approach.

#### 2.1 CNNs for Sperm Analysis

Because convolutional neural networks (CNNs) are so amazing at processing medical images, several research articles recommend using them to automate the classification and identification of sperm cells. However, due to variations in participant selection, picture pre-processing methods, and validation procedures, assessing classification performance across studies continues to be difficult. These differences make it more difficult to provide a common criteria by which to judge how well CNNs perform while analyzing sperm cells [5] [6].

##### 2.1.1 Performance Metrics

Comparing classification performance across research is extremely difficult because to variations in picture pre-processing methods, validation procedures, and participant selection. Variations in these essential elements impede the creation of a uniform criteria for assessing CNNs' effectiveness in sperm cell analysis.

##### 2.1.2 Framework Inaccessibility

Another obstacle is that many research journals do not make their frameworks publicly available, making it difficult to find frameworks and implementation details. Because of this lack of openness, it is challenging to reproduce and confirm existing findings.

### **2.1.3 Metrics Bias**

One concerning tendency in the literature is the fact that over half of the papers under examination had data leaks. This raises doubts about the validity of the reported performance indicators since it might lead to biased reporting due to inadequate validation and model selection processes. The reliability of the proposed CNN-based solutions might be jeopardized by discrepancies in the verification procedures.

## **2.2 Data Issues and Resolutions**

One ongoing difficulty is the lack of high-quality medical databases, particularly when it comes to photos of ocular sperm. Having identified this problem, our work resolves it by creating a carefully selected bespoke dataset.

### **2.2.1 Dataset Scarcity**

It is difficult to obtain high-quality ocular sperm imaging databases since so few researchers make their datasets publicly available. The difficulty is exacerbated by the screening procedures needed to get these databases, which restrict their availability to field researchers.

### **2.2.2 Dataset Customization**

Our study decided to create a bespoke dataset from Istanbul University's infertility clinic as a solution to the dataset challenge. The photos in this dataset, which have a resolution of  $3024 \times 4032$  pixels, include a variety of sperm cell morphological representations. The integrity and ethical compliance of our dataset are guaranteed by the ethical clearances obtained for the purpose of taking images with a Xiaomi Mi 8 smartphone mounted to a microscope at a magnification of  $100\times$ .

### **2.2.3 Class Definitions**

Our dataset includes a wide range of sperm cell classes, with 18 distinct classes offering a thorough foundation for training the YOLOv8 model to discern between sperm cell morphologies that are normal and pathological (as shown in Table 2.1).

**Table 2.1** Sperm cell classes used in the study

	ENGLISH	TURKISH
1	Amorphous Head	Amorf Baş
2	Asymmetric Entry Neck	Asimetrik Giriş Boyun
3	Twisted Tail	Bükük Kuyruk
4	Bent neck	Bükük boyun
5	Narrow Acrosome area	Dar Akrozom alanı
6	Thick Entry Neck	Kalın Giriş Boyun
7	Conical Head	Konik Baş
8	Short Tail	Kısa Kuyruk
9	Curved Tail	Kıvrımlı kuyruk
10	Normal	Normal
11	Pyriform Head	Piriform Baş
12	Long Tail	Uzun Kuyruk
13	Vacuolated	Vakuollü
14	Round Head	Yuvarlak baş
15	Double Tail	Çift Kuyruk
16	Double Head	Çift baş
17	Slim Entry Neck	İnce Giriş Boyun
18	Needle head	İğne baş

### 2.3 YOLOv8 in Sperm Analysis

In contrast to the challenges identified in other studies, the YOLOv8 model is employed in our experiment to analyze sperm cells. YOLOv8, a contemporary deep learning-based object recognition model, offers notable advantages in terms of efficiency, accuracy, and real-time analysis.

#### 2.3.1 Real-time Efficiency

The YOLOv8 model has a reputation for being an effective model for object recognition tasks, which makes it useful for real-time analysis. The incorporation of this model into our mobile application has the potential to revolutionize the accuracy and speed of sperm cell analysis at the point of care.

#### 2.3.2 Sperm Variation Analysis

With precise spatial coordinates and classifications provided, our dataset has been meticulously labeled, providing YOLOv8 with an extensive and informative data source. This gives the model a thorough understanding of the variations in sperm cells, which aids in the distinction between normal and abnormal morphologies.

# 3

## FEASIBILITY

---

The four main components of the project's feasibility evaluation are its technical, temporal, legal, and economic issues. Each of these components determines the project's viability and practicality of implementation.

### 3.1 Technical Feasibility

The software and hardware feasibility of the suggested system were the two main criteria used to comprehensively assess its technological viability. Each of these elements has a significant impact on the project's viability and capacity for execution.

#### 3.1.1 Software

Because of its strong support for deep learning frameworks like TensorFlow and PyTorch ("torch"), Python was chosen as the main programming language for this project. This allowed for the easy integration of YOLOv8, a powerful object identification model. The language's extensive ecosystem, which includes IPython and Matplotlib, facilitates effective development and visualization, and its compatibility with the Ultralytics library provides a smooth implementation. The choice to utilize Python in conjunction with React-Native is a great fit for the project's objective of developing a cross-platform mobile application with an intuitive user experience. To sum up, Python is the best option for our project because of its many libraries, including PyTorch, its versatility, and its robust community.

#### 3.1.2 Hardware

The project involves the use of multiple computers for development. The hardware specifications for each computer are as shown in Table 3.1.

**Table 3.1 Personal Computers**

	COMPUTER 1	COMPUTER 2
<b>RAM</b>	32GB	16GB
<b>STORAGE</b>	8TB	2TB
<b>CPU</b>	AMD Ryzen 5 5600X	11th Gen Intel Core(TM) i5-11300H 3.10GHz
<b>GPU</b>	NVIDIA RTX 3090 24GB	NVIDIA GEFORCE RTX 3050 12GB
<b>OS</b>	Ubuntu	Windows

### 3.2 Time Feasibility

Two computer engineering students are working on a project that will be finished in a semester. The project requires reading relevant literature, analyzing the dataset, selecting an appropriate model, developing a deep learning structure, testing and training the structure, evaluating success metrics, and producing a mobile application.



**Figure 3.1 Gantt Diagram of the Project**

### 3.3 Legal Feasibility

We employed a meticulously built bespoke dataset from Istanbul University's infertility clinic in our study. These 3024 x 4032 pixel photos, captured with a Xiaomi Mi 8 phone attached to a microscope, provide a close-up view of sperm cells. We have authorization to use these images in a way that adheres to moral principles. We ensured that we adhered to all regulations and maintained transparency in our research by using open-source libraries to analyze and interpret the data.

### 3.4 Economic Feasibility

The study datasets were free to use and were available from Istanbul University, as were the programming languages and libraries used in the development process. Since using a server was not necessary, there are no related costs.

# 4

## SYSTEM ANALYSIS AND DESIGN

---

### 4.1 Software Design

#### 4.1.1 Library Utilization

The YOLOv8 object detection paradigm was effectively used for this project by utilizing a number of powerful libraries. Each library made a distinct contribution to enhancing the functionality, efficacy, and outcome of the system.

- Python3
- PyTorch
- Ultralytics
- TensorFlow
- React-Native
- Matplotlib
- NumPy
- OpenCV

Together, these libraries provide a solid and unified software base for the project. Combining the power of PyTorch, TensorFlow, Ultralytics, React-Native, Matplotlib, NumPy, and OpenCV with the adaptability of Python3 was essential to the creation, training, and implementation of the YOLOv8 object identification model.

#### 4.1.2 Dataset

The dataset was obtained from Istanbul University's infertility clinic. including 4966 sperm cell ocular pictures, carefully arranged into three folders (Train, Valid, and

Test). This collection includes eighteen distinct classes, representing both aberrant and normal sperm cell morphologies.

- Train: include subdirectories for labels and photos. The label files identify each sperm's position in the associated picture.
- Valid: Exhibits a similar structure to the training set, with distinct image and label subfolders.
- Test: consists of image subfolders; labels are not provided for this set.



**Figure 4.1** Dataset Tree

#### 4.1.3 Data Labeling and Preparation

Making sure the dataset is properly prepared and categorized is essential to the project's success. Each image of a sperm cell undergoes a rigorous labeling process that assigns a unique spatial position and class to each cell. Through a series of intricate procedures, the YOLOv8 model is guaranteed access to a dataset that is rich in information and helps it discriminate between normal and abnormal sperm cell shapes.

#### **4.1.4 YOLOv8 Model**

The YOLOv8 model's advanced architecture, which incorporates convolutional layers and shortcut connections, allows it to effectively capture the minute details and complex relationships present in ocular images. The model leverages a robust backbone network, often based on the potent CSPDarknet53, to achieve effective feature extraction. This makes it possible for the model to recognize intricate patterns in the sperm cell's morphology. Using a meticulously labeled dataset, YOLOv8 is rigorously trained using PyTorch. To accomplish precise localization and classification, parameters are adjusted using loss functions. Because it can detect and localize sperm cells rapidly and accurately by meticulously separating photos into a grid to predict bounding boxes and class probabilities, YOLOv8 is unusual in that it can distinguish things in real time. The seamless integration of YOLOv8 with Ultralytics' model management platform expedites all stages of the development process, including assessment, prediction, and training.[4] [7].

## **4.2 Input-Output Design**

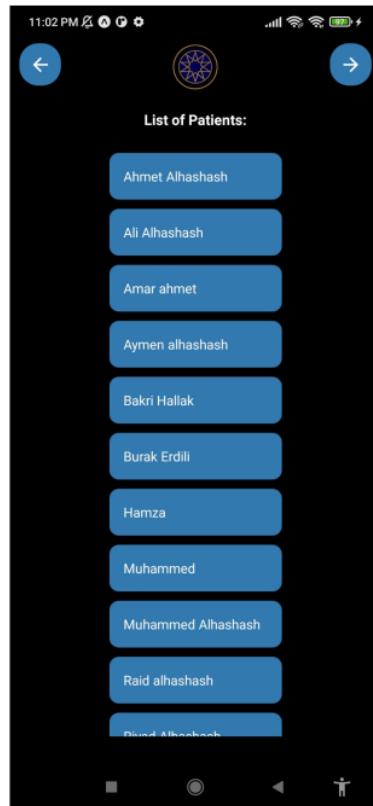
The system's input involves ocular images of sperm cells, which are fed into the YOLOv8 model for analysis. The output of the system includes the identification and classification of sperm cells based on their morphology. The YOLOv8 model provides bounding box coordinates and class probabilities, indicating the location and type of each identified sperm cell. The mobile application is designed smoothly to help specialists analyze sperm cells directly and save and sort the data and results into special files for each patient.

For our project, when we open the mobile application, the home screen will seem like it does in Figure 4.2.



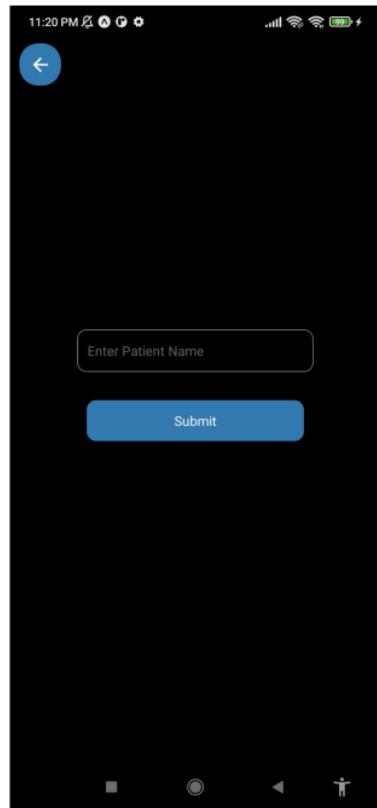
**Figure 4.2** Selecting the Type of Patient

The first step to using the application is to determine whether the patient previously existed in the list of old patients or not. If the patient already exists, the patient's folder can be viewed by clicking on the Old Patient option to show all the folders of previously existing patients, as shown in Figure 4.3.



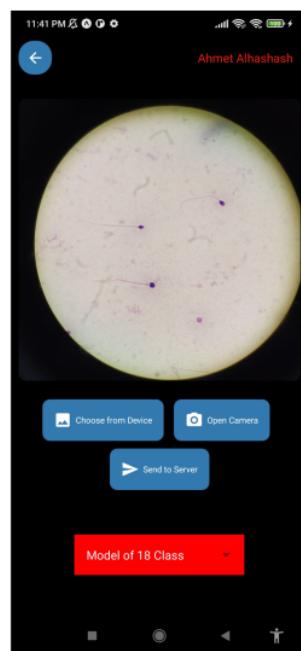
**Figure 4.3** The List of Existing Patients

If the patient is new, a special folder must be created for him to store data and results by clicking on the New Patient option, then entering the patient's name and confirming the creation, as shown in Figure 4.4.

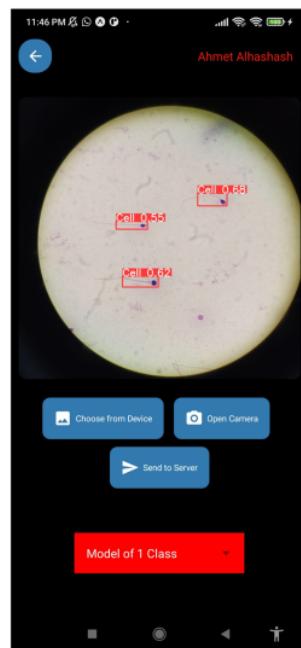


**Figure 4.4** Adding a new patient

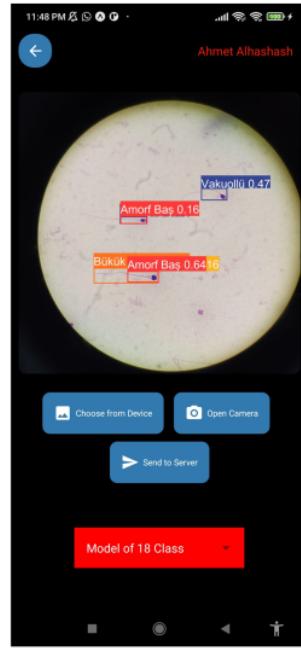
The second stage is collecting and examining patient data. If the patient is new, a page will appear to use the camera or choose an image from the device to send to the server once his folder is created. We can also choose the model (1-class or 18-class) that we want to analyze the image with, as shown in Figure 4.5. Following selection of the image and clicking on the "Send to the Server" button, the original image and its analysis result will be saved in the patient's folder on the server that contains the Yolov8 model. The server will then return the result to the application so that we can see it, as shown in Figures 4.6 and 4.7. However, if the patient is on the list of old patients, after clicking on his personal folder, two options appear: either viewing the patient's data or taking a picture directly using the phone's camera or choosing a picture from the device to send it to the server.



**Figure 4.5** Choosing a Sample and Sending It to the Server

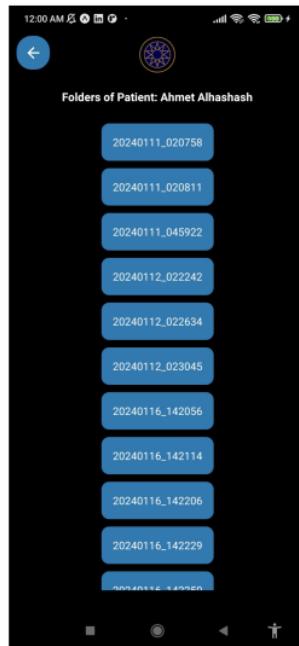


**Figure 4.6** Receiving the Result From Server by Using 1-Class Model

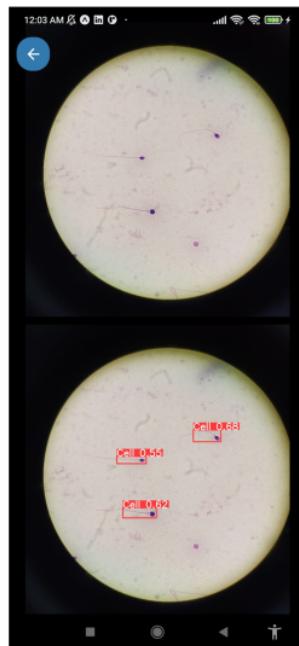


**Figure 4.7** Receiving the Result From Server by Using 18-Class Model

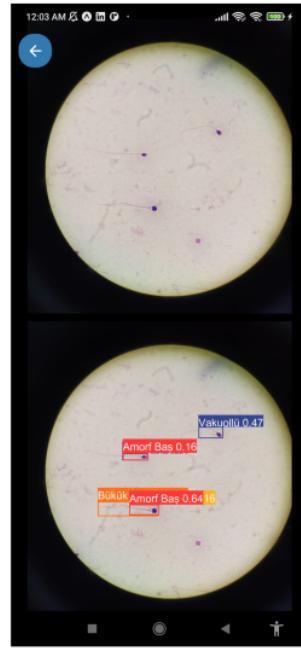
In the final stage, as seen in Figure 4.8, we can click on the patient's folder and then select the option to view all data if we would like to see the patient's data and findings. Subsequently, every folder will manifest, with every folder holding a single image and its outcome, as seen in figures 4.9 and 4.10.



**Figure 4.8** Patient Folders Listing



**Figure 4.9** Saved Single Class Results Inside a Patient's Folder



**Figure 4.10** Saved 18 Class Results Inside a Patient's Folder

# 5

## Implementation

---

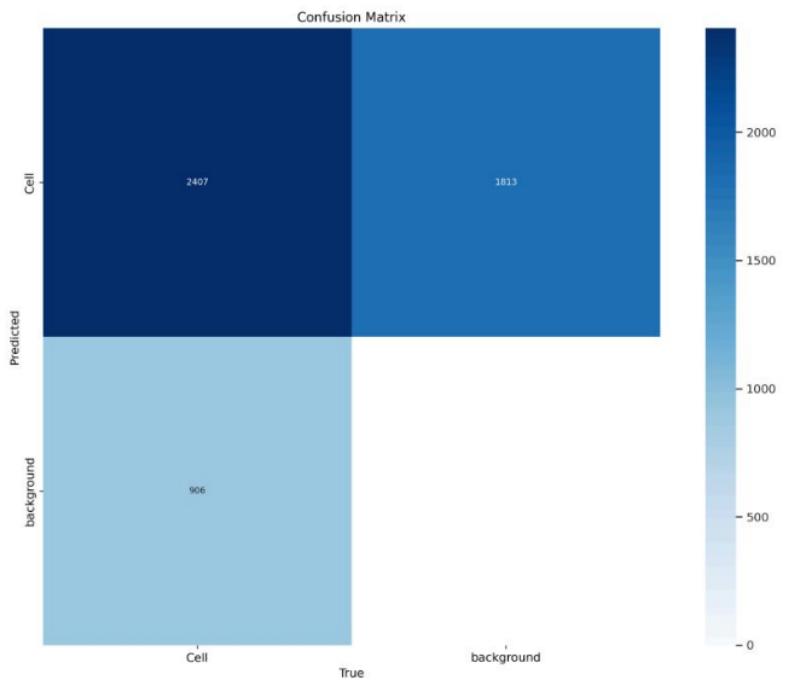
### 5.1 Training Dataset

After preprocessing and organizing the paths of the images and their annotation label text files, we ended up with 3321 files for training 67%, 1044 files for validation 21%, and 590 files for testing the model 12%. This study presents two types of trained models. The first one is trained to treat all 18 different sperm classes as one class and detect the sperm cells as they are. The second one is trained to detect and classify 18 different sperm cell classes.

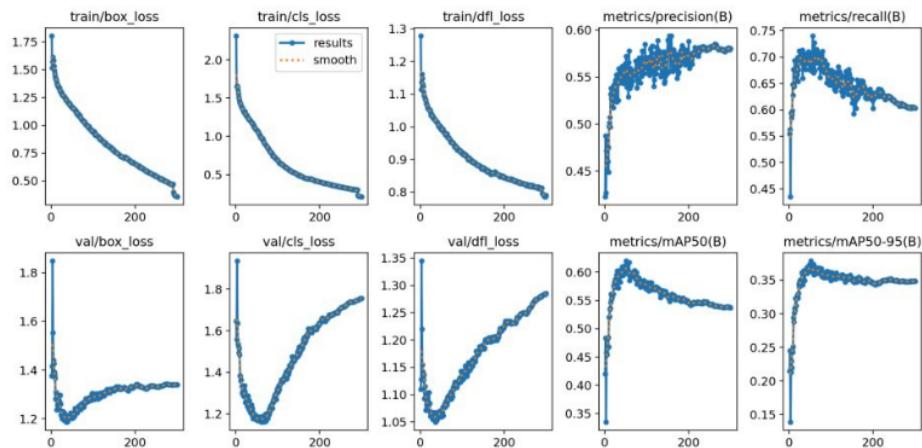
#### 5.1.1 Training a Model to Detect 1 Class

After the training of the model that detects only one class has been completed, which has the parameters: 300 epochs, patience = 0, imgsz = 640, batch = 16. During testing, the performance difference was obvious when the model was trained for a single class, considering all cells as part of that class (cell class) instead of detecting sperm cells in 18 different classes.

Following the initial model's training, the following metrics were obtained:



**Figure 5.1** Confusion Matrix



**Figure 5.2** Various Training Metrics

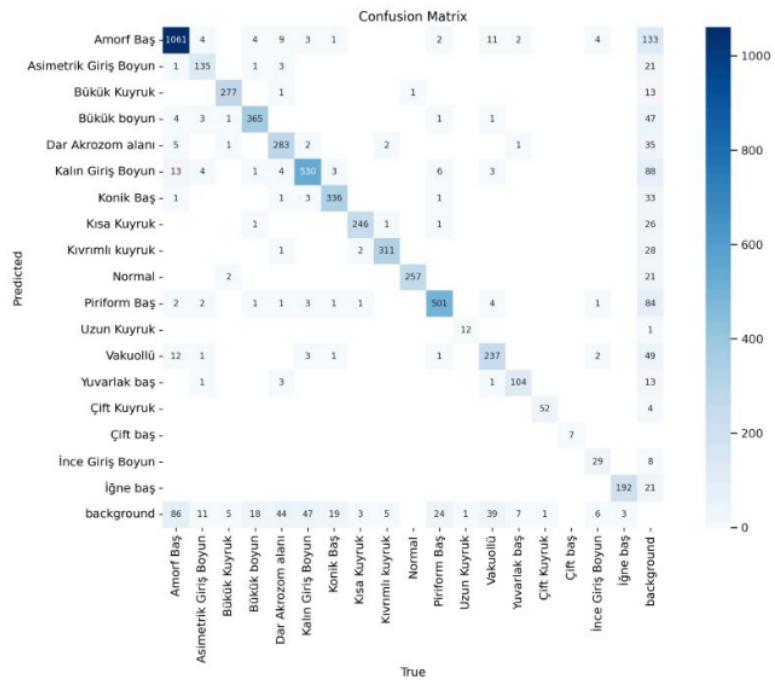


**Figure 5.3** Predictions of the Model for a Validation Set

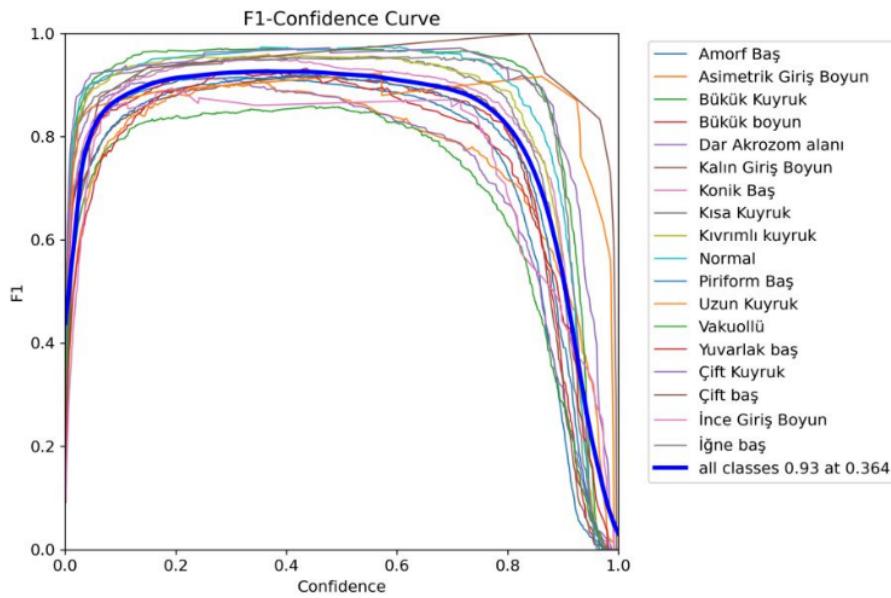
#### 5.1.2 Training a Model to Detect 18 Classes

After completing the training of the model that detects and classifies all 18 classes, the parameters used were 100 epochs, patience = 50, imgsz = 416, and batch = 16. It's worth noting that the initial 100 epochs were performed on a previous model trained with a smaller amount of data, and an additional 100 epochs were conducted using new training images, making a total of 300 epochs for this specific model, but with the same validation data referencing the previous model.

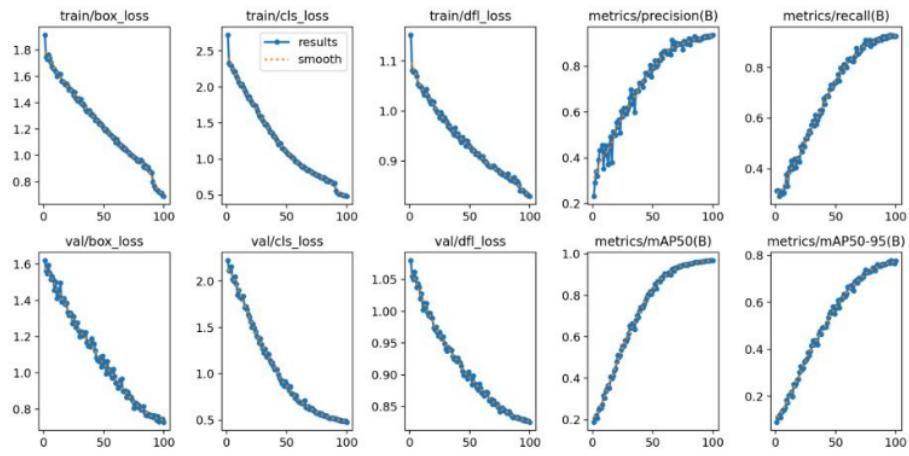
Following the initial model's training, the following metrics were obtained:



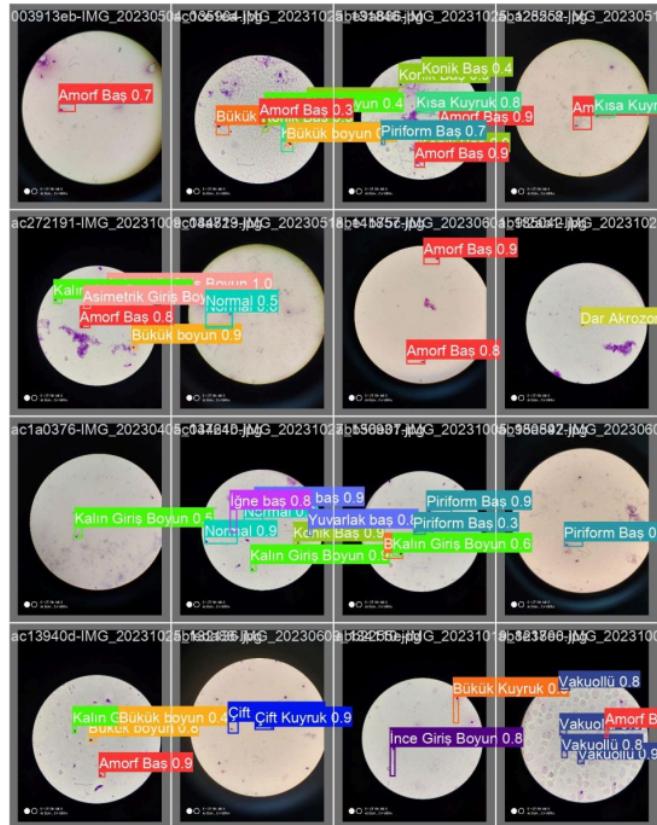
**Figure 5.4 Confusion Matrix**



**Figure 5.5 F1-Confidence Curve**



**Figure 5.6** Various Training Metrics



**Figure 5.7** Predictions of the Model for a Validation Set

# 6

## Experimental Results

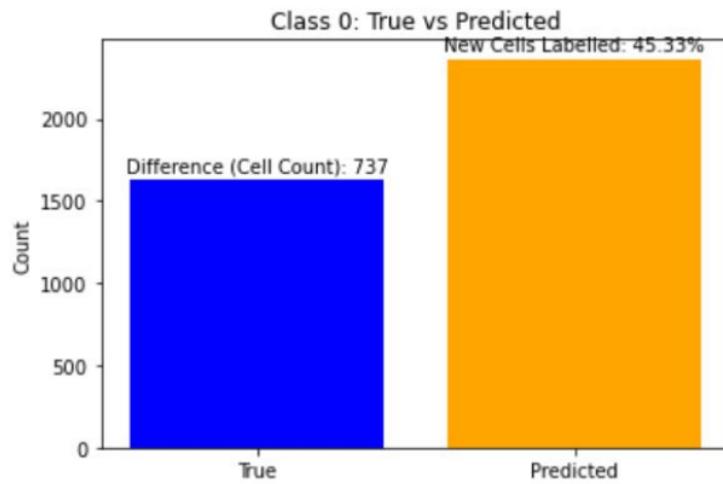
---

### 6.1 Testing Results of The Models

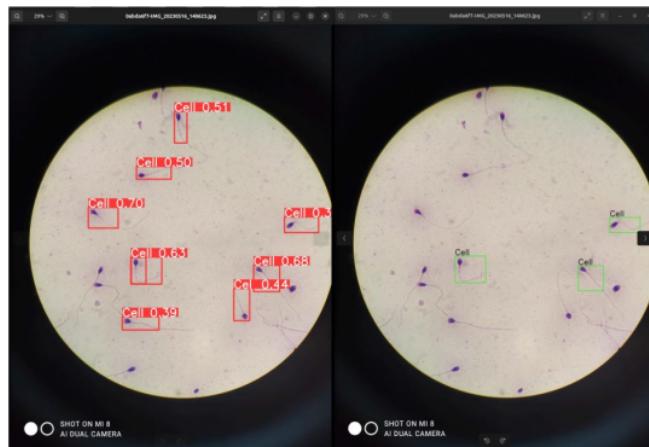
After testing both models with the same set of 500 testing images, we have analyzed and compared the specifications of the two models.

#### 6.1.1 Model Results for 1 Class

Once the single-class detection model has been trained, the values are as follows (parameters: 300 epochs, patience=0, imgsz=640, batch=16): 1044 for validation, 590 for testing, and 3321 for training. In the first testing phase, out of the 312 true labels in the 100 test images, the model considered 13 as background. The success rate of cell detection is 95.8% . Similarly, within the 100 test images, all 456 cell detection made by the model were accurate, and it identified new detection in 46% more marked data. In the second testing phase, among 500 test images, the total difference between true labels and predicted detection is 737. The model failed to detect 234 out of 1626 true-labeled cells, considering them as background. The accuracy rate for detecting true values is 85.6% . According to the validation confusion matrix we had after training the model, this accuracy value is 82.3% . The remaining 503 differences are all related to the detection of unmarked new cells, and it has identified 30% more new cells.



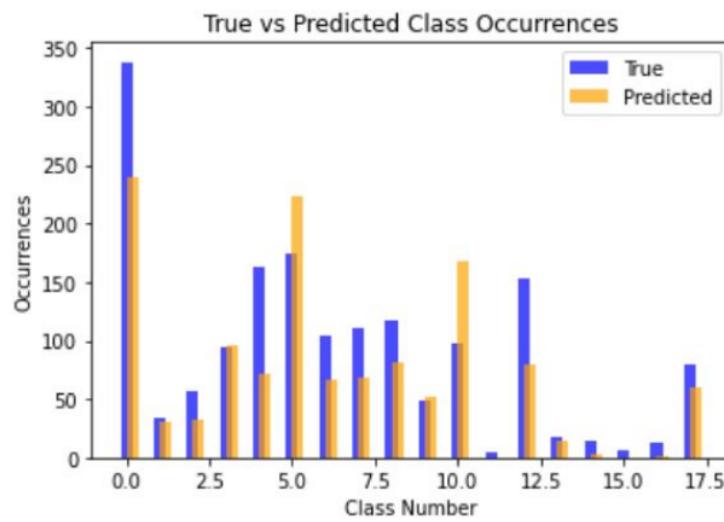
**Figure 6.1** Histogram of Detections Using the Model Trained for Single Class Detection



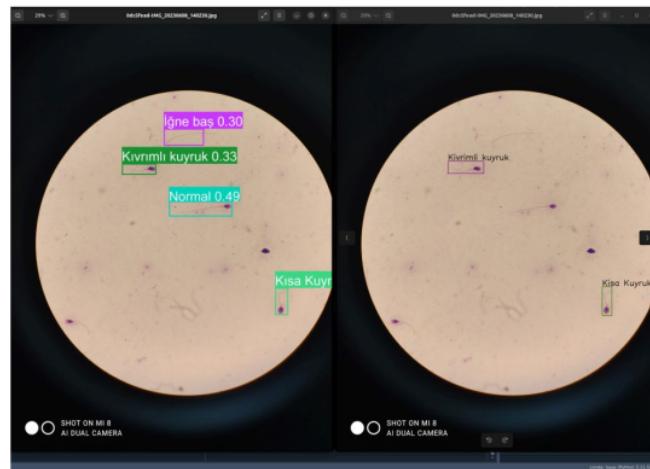
**Figure 6.2** Example of Detections Using the Model Trained for Single Class Detection

### 6.1.2 Model Results for 18 Class

Using the same 500 test pictures, we got 1626 true labeled annotations for the model trained to recognize 18 sperm types. 463 is the difference, or the number of true labeled occurrences that are more than the expected counts for that class. The algorithm trained to recognize 18 kinds of sperm cells is unable to detect 28.42% of the annotations and wrongly classifies them as background, yielding an overall accuracy of 71.53% .



**Figure 6.3** Histogram of Detections Using the Model Trained for Detecting 18 Classes



**Figure 6.4** Example of Detections Using the Model Trained for Detecting 18 Classes

# 7

## Performance Analysis

---

Different levels of success are revealed by the performance analysis of the two trained YOLOv8-based models for the detection and categorization of sperm cells. The single-class detection model had an impressive 85.6% success rate, indicating its high level of efficiency.

However, the model that detects 18 different classes encountered difficulties, yielding an overall accuracy of 71.53%. About 28.42 out of 100 annotations were difficult for the model to identify, and it incorrectly identified them as background. This shows that the model's ability to accurately distinguish between the 18 various types of sperm is limited. The model may need to be further examined and refined in order to enhance its ability to identify various types of sperm cells.

**Table 7.1** Results of Model 1 (Trained With One Class) and Model 2 (Trained With 18 Classes) After Predicting Same 500 Test Images

Model	True Labels	Predicted	True labels could not be found	New accurate annotations	Difference in Count	Overall Accuracy
Model 1	1626	2363	234	503	737	%85.6
Model 2	1626	2211	463	112	585	%71.5

## 8 Conclusion

---

In conclusion, the project has successfully developed and evaluated two YOLOv8-based models for sperm cell detection and classification in ocular images. The single-class detection model demonstrated a high success rate of 85.6%, showcasing its efficiency in identifying sperm cells. However, the 18-class detection model faced challenges, yielding an overall accuracy of 71.53% and struggling to detect a significant portion of annotations. The system's input-output design, coupled with the mobile application for specialists, provides a comprehensive tool for managing and analyzing sperm cell data. The three-stage process for patient data management allows for efficient data gathering, analysis, and result retrieval, catering to both new and old patients. In summary, the initiative advances the field of computer-aided sperm analysis and provides reproductive health specialists with insightful knowledge. Improved accuracy through additional model and system refinement could lead the way for improvements in fertility-related diagnoses and therapies.

## References

---

- [1] R. A. Movahed, E. Mohammadi, and M. Orooji, "Automatic segmentation of sperm's parts in microscopic images of human semen smears using concatenated learning approaches," *Computers in Biology and Medicine*, vol. 109, pp. 242–253, 2019.
- [2] F. Shaker, S. A. Monadjemi, and A. R. Naghsh-Nilchi, "Automatic detection and segmentation of sperm head, acrosome and nucleus in microscopic images of human semen smears," *Computer methods and programs in biomedicine*, vol. 132, pp. 11–20, 2016.
- [3] V. H. Sahin, I. Oztel, and G. Yolcu Oztel, "Human monkeypox classification from skin lesion images with deep pre-trained network using mobile application," *Journal of Medical Systems*, vol. 46, no. 11, p. 79, 2022.
- [4] R. Qureshi, M. G. RAGAB, S. J. ABDULKADER, A. ALQUSHAIIB, E. H. SUMIEA, H. Alhussian, *et al.*, "A comprehensive systematic review of yolo for medical object detection (2018 to 2023)," 2023.
- [5] O. L. Şavkay, E. Cesur, M. E. Yalçın, and V. Tavşanoğlu, "Sperm morphology analysis with cnn based algorithms," in *2014 14th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, IEEE, 2014, pp. 1–2.
- [6] M. Yüzkat, H. O. Ilhan, and N. Aydin, "Multi-model cnn fusion for sperm morphology analysis," *Computers in Biology and Medicine*, vol. 137, p. 104790, 2021.
- [7] P. P. Guarnido Lopez, F. Ramirez-Agudelo, E. Ribeira-Ferreiro, and M. B. Aouda, "301 determining individual dry matter intake of charolais bulls from ingestion time recorded with rgb-d cameras," *Journal of Animal Science*, vol. 101, no. Supplement\_3, pp. 247–248, 2023.

# **Curriculum Vitae**

---

## **FIRST MEMBER**

**Name-Surname:** Burak Erdilli  
**Birthdate and Place of Birth:** 19.09.2000, İstanbul  
**E-mail:** burak.erdilli@std.yildiz.edu.tr  
**Phone:** 0545 274 77 71  
**Practical Training:**

## **SECOND MEMBER**

**Name-Surname:** Yusuf Elhaşas  
**Birthdate and Place of Birth:** 01.01.1997, İstanbul  
**E-mail:** yusuf.elhasas@std.yildiz.edu.tr  
**Phone:** 0531 240 00 68  
**Practical Training:**

## **Project System Informations**

**System and Software:** Windows Operating System , Python  
**Required RAM:** 8GB  
**Required Disk:** 512MB

## ORIGINALITY REPORT

<b>3</b> %	<b>3</b> %	<b>1</b> %	<b>%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

## PRIMARY SOURCES

- |   |   |      |
|---|---|------|
| 1 | <a href="http://www.slideshare.net">www.slideshare.net</a><br>Internet Source                                     | 1 %  |
| 2 | "Front Matter", 2023 8th International Conference on Computer Science and Engineering (UBMK), 2023<br>Publication | <1 % |
| 3 | Krause. "Providers and Configuration", Pro ASP NET Extensibility, 2009<br>Publication                             | <1 % |
| 4 | <a href="http://br.atsit.in">br.atsit.in</a><br>Internet Source   | <1 % |
| 5 | <a href="http://libres.uncg.edu">libres.uncg.edu</a><br>Internet Source   | <1 % |
| 6 | <a href="http://dspace.yildiz.edu.tr">dspace.yildiz.edu.tr</a><br>Internet Source                                 | <1 % |
| 7 | <a href="http://github.com">github.com</a><br>Internet Source   | <1 % |
| 8 | <a href="http://archives.njit.edu">archives.njit.edu</a><br>Internet Source                                       | <1 % |

9

[ntnuopen.ntnu.no](http://ntnuopen.ntnu.no)

Internet Source

<1 %

10

[www.eng.mu.edu](http://www.eng.mu.edu)

Internet Source

<1 %

11

[www.eoc.org.uk](http://www.eoc.org.uk)

Internet Source

<1 %

Exclude quotes

On

Exclude matches

Off

Exclude bibliography

On