VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"Jnana Sangama", Belagavi-560014, Karnataka

A PROJECT REPORT ON

"IMAGE PROCESSING APPROACH FOR GRADING IVF BLASTOCYST USING MACHINE LEARNING"

(Approved by KSCST, Indian Institute of Science Campus, Bengaluru)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted By

HARSHITHA T A	[1SV20CS014]
VARSHITHA T N	[1SV20CS055]
TEJASHWINI R	[1SV20CS051]
ANUSHA R	[1SV20CS002]

Under the guidance of

Prof. Shanmukaswamy C V
B.E.,M.E.,MISTE
Associate Professor,
Dept. of CSE, SIET,Tumakuru

SHRIDEVI EDUCATION

Department of Computer Science and Engineering

SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Affiliated To Visvesvaraya Technological University)
Sira Road, Tumakuru – 572106, Karnataka.

2023-2024

Sri Shridevi Charitable Trust (R.)
ESTD:2002
SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY
Sira Road, Tumkur - 572 106, Karnataka, India.
Phone: 0816 - 2212629 | Principal: 0816 - 2212627, 9686114899 | Telefax: 0816 - 2212628
Email: info@shrideviengineering.org, principal@shrideviengineering.org | Website: www.shrideviengineering.org
(Approved by AICTE, New Delhi, Recognised by Govt. of Karnataka and Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that, the Project entitled **"IMAGE PROCESSING APPROACH FOR GRADING IVF BLASTOCYST USING MACHINE LERANING"** has been Successfully carried out by HARSHITHA T A [1SV20CS014], VARSHITHA T N [1SV20CS055], TEJASHWINI R [1SV20CS051], ANUSHA R [1SV20CS002] in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science & Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2023-24.** It is certified that all the corrections/suggestions indicated for internal assessments have been incorporated in the report. The Project report has been approved as it satisfies the academic requirements in respect of Project phase II work prescribed for the Bachelor of Engineering Degree.

29/5/24

Signature of the Guide
Prof. Shanmukaswamy C V B.E.,M.E.,MISTE.,
Associate Professor
Dept. of CSE,
SIET, Tumakuru.

29/5/24

Signature of the H.O.D
Dr. Basavesha D BE.,M.Tech.,PhD
Associate Professor & HOD
Dept. of CSE,
SIET, Tumakuru.

Signature of the Principal
Dr.Narendra Viswanath M.E., Ph.D., MIE, MISTE, MIWS., FIV.,
Principal, SIET, Tumakuru

Name of the Examiners	Signature with date
1. Dr. Basavesha D	29/5
2. Rakesh G	29/5/24

Sri Shridevi Charitable Trust (R.)
ESTD:2002
SHRIDEVI EDUCATION
SHRIDEVI INSTITUTE OF ENGINEERING AND TECHNOLOGY
Sira Road, Tumkur - 572 106, Karnataka, India.
Phone: 0816 - 2212629 | Principal: 0816 - 2212627, 9686114899 | Telefax: 0816 - 2212628
Email: info@shrideviengineering.org, principal@shrideviengineering.org | Website: www.shrideviengineering.org
(Approved by AICTE, New Delhi, Recognised by Govt. of Karnataka and Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, HARSHITHA T A [1SV20CS014], VARSHITHA T N[1SV20CS055], TEJASHWINI R [1SV20CS051], ANUSHA R[1SV20CS002] students of VIII semester **B.E** in Computer Science &Engineering, at Shridevi Institute of Engineering & Technology, Tumakuru, here by declare that, the Project work entitled **"IMAGE PROCESSING APPROACH FOR GRADING IVF BLASTOCYST USING MACHINE LEARNING"** embodies the report of our Project work carried out under the guidance of **Prof. Shanmukaswamy C V, Associate Professor, Department of CSE, SIET,Tumakuru** as a partial fulfillment of requirements for the award of the degree in **Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi,** during the academic year **2023-24**. The Project has been approved as it satisfies the academic requirements in respect to the Project Phase II work .

Place: Tumakuru
Date: 29/05/2024

Student Name& Signature

HARSHITHA T A	**[1SV20CS014]**	Harshitha T.A
VARSHITHA T N	**[1SV20CS055]**	Varshitha T.N
TEJASHWINI R	**[1SV20CS051]**	Tejashwini R
ANUSHA R	**[1SV20CS002]**	Anusha.R

ACKNOWLEDGEMENT

This Project will be incomplete without thanking the personalities responsible for this venture, which otherwise would not have become a reality.

We express our profound gratitude to **Dr. Narendra Viswanath,** Principal,S.I.E.T, for his moral support towards completing our Project work.

We would like to thank our HOD **Dr. Basavesha D** Head, Department of CSE, SIET for providing all the support and facility.

We would like to thank my guide **Prof. Shanmukaswamy C V** Associate Professor, Department of Computer Science and Engineering, SIET for his help, sharing his technical expertise and timely advice.

We would like to thank Our Project Coordinator **Dr. Girish L** Associate Professor, HOD, AI & DS and **Mrs. Rashmi N** Assistant Professor, Department of Computer Science and Engineering, SIET for providing all the support and facilities.

We would like to express our sincere gratitude to all teaching and non-teaching staff of the Department of CSE for guiding us during the project work by giving valuable suggestion and encouragement.

By,

HARSHITHA T A [1SV20CS014]
VARSHITHA T N [1SV20CS055]
TEJASHWINI R [1SV20CS051]
ANUSHA R [1SV20CS002]

ABSTRACT

Tens of millions of women suffer from infertility world wide each year. In vitrofertilization(IVF) is the best choice for many such patients. However, IVF is expensive, time-consuming, and both physically and emotionally demanding. Artificial intelligence will havedigital revolution and manifold advances in the field of reproductive medicine and willeventually provide immense benefits to infertile patients. The main aim of this IVF is to focus onthe methods that can predict the accuracy of pregnancy without human intervention. It providessuccessful studies conducted by using machine learning techniques. This easily enables doctorsto understand the behaviour of the attributes which are suitable for the treatment. Blastocystimages can be deployed for the detection and prediction of the best embryo which has themaximum chance of a successful pregnancy. Generative Adversarial Networks (GAN) andimage classification using CNN which gave an accuracy of about 85% for 500 epochs. Thisproject delves into the development of a novel image-based approach for grading IVF embryosto enhance predictive accuracy. Leveraging advancements in artificial intelligence, particularlyin the analysis of blastocyst images, the study aims to provide a comprehensive understanding ofthe attributes influencing IVF success rates. the project aims to contribute to the optimization of IVF outcome.

CONTENTS

Chapter **Page No**

LIST OF FIGURES

Chapter 1

INTRODUCTION

Reproductive issues affect many couples worldwide. Infertility, defined as the inability to conceive after a year of regular unprotected intercourse, affects over 186 million couples globally. In-Vitro Fertilization (IVF) is a common treatment for infertility. IVF is a medical procedure used when natural conception is challenging. Machine learning techniques are increasingly applied to enhance IVF outcomes. These techniques analyse factors like embryo characteristics and patient data to optimize treatment plans, improve success rates, and minimize risks. By automating tasks and personalizing treatments, machine learning integration makes IVF more effective, efficient, and accessible. Overall, the combination of IVF and machine learning represents a significant advancement in reproductive medicine, offering tailored care and better results. In vitro fertilization (IVF) is a medical procedure used to help individuals or couples conceive a child when natural conception is difficult or not possible. When discussing IVF in the context of machine learning, it typically involves the application of data-driven techniques to optimize and improve various aspects of the IVF process. Machine learning algorithms can be employed to analyse vast amounts of data related to fertility, genetic factors, treatment protocols, and patient outcomes, aiming to enhance success rates, personalize treatment plans, and minimize risks.

In the realm of IVF and machine learning, data-driven approaches are revolutionizing the field by enabling practitioners to make more informed decisions at every stage of the fertility treatment process. Machine learning algorithms can analyse large datasets containing information about patient demographics, medical histories, hormonal profiles, genetic factors, and treatment outcomes. By leveraging this wealth of data, machine learning modelscan identify patterns, correlations, and predictive factors that traditional methods may overlook. One area where machine learning excels is in the prediction of embryo viability. Embryo selection plays a crucial role in the success of IVF treatments, but it's often challenging for embryologists to determine which embryos are most likely to result in a successful pregnancy. Machine learning models trained on historical data can help predict the viability of embryos by considering factors such as morphology, genetic characteristics, and time-lapse imaging data. Furthermore, machine learning can assist in the personalization of treatment protocols.

IVF is a complex procedure involving multiple steps, from ovarian stimulation and egg retrieval to embryo culture and transfer. Each of these stages presents its own challenges and uncertainties, making the journey towards a successful pregnancy both emotionally and physically demanding for couples seeking fertility treatment. Machine learning offers a unique opportunity to leverage vast amounts of patient data, including demographics, medical history, hormone levels, and genetic information, to provide personalized insights and predictions. For instance, by analysing historical IVF outcomes and patient characteristics, machine learning algorithms can predict the likelihood of success for individual patients, enabling clinicians to tailor treatment plans accordingly. Furthermore, one of the critical steps in IVF is the selection of viable embryos for transfer. Traditionally, embryologists rely on visual assessment based on morphology, a subjective and time-consuming process. However, by analysing images of developing embryos using machine learning algorithms, objective criteria can be established to identify embryos with the highest implantation potential, leading to improved pregnancy rates and reduced miscarriage risk.

Moreover, machine learning can optimize medication protocols by analysing patient responses and adjusting dosages accordingly, minimizing side effects and optimizing ovarian stimulation for better egg retrieval outcomes. In addition to enhancing success rates,machine learning can also help mitigate risks associated with IVF, such as ovarian hyperstimulation syndrome (OHSS) and multiple pregnancies. By analysing patient data andidentifying risk factors, clinicians can intervene early and tailor treatment strategies to ensurethe safety and well-being of patients throughout the IVF journey. Beyond the initial stagesof prediction and embryo selection, machine learning holds immense potential in optimizing the entire IVF process, from start to finish. Consider, for instance, the delicate balance of ovarian stimulation – a critical phase that requires precise control over hormone levels to maximize egg production while minimizing the risk of complications. Traditionally, clinicians rely on standardized protocols based on age and ovarian reserve to determine medication dosages.

Through iterative learning and adaptation, these algorithms can dynamically adjust medication dosages in real-time, optimizing hormone levels to enhance egg quality and retrieval rates. Moreover, by integrating feedback loops from continuous monitoring devicesand biomarker measurements, machine learning algorithms can fine-tune treatment protocols with unparalleled precision, minimizing the risk of overstimulation and maximizing the chances of success. But the potential of machine learning in IVF extends beyond the confines of the clinic. For instance, researchers can leverage machine learning algorithms to identify novel biomarkers of embryo viability or elucidate the genetic determinants of infertility. By correlating clinical outcomes with genomic data and environmental exposures, we can gain a deeper understanding of the complex interplay between genetics, lifestyle factors, and reproductive health. Overall, the integration of machine learning into IVF treatment represents a promising frontier in reproductive medicine, offering new avenues for personalized care, improved outcomes, and ultimately, the realization of couples' dreams of parenthood.

1.1 Definition and Use

Machine learning (ML) is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. machine learning, data- driven approaches are revolutionizing the field by enabling practitioners to makemore informed decisions at every stage of the fertility treatment process. Machine learning algorithms can analyse large datasets containing information about patient demographics, medical histories, hormonal profiles, genetic factors, and treatment outcomes. Machine learning models trained on historical data can help predict the viability of embryos by considering factors such as morphology, genetic characteristics, and time-lapse imaging data. Furthermore, machine learning can assist in the personalization of treatment protocols. By analysing patient data and treatment outcomes, algorithms can identify optimal medication dosages, timing of procedures, and other treatment parameters tailored to individual patients' needs. This personalized approach not only improves the chances of success but also minimizesthe risk of complications and side effects. Additionally, machine learning techniques are being applied to improve the efficiency of IVF laboratory workflows. Automation and optimization of processes such as sperm selection, oocyte retrieval, and embryo culture can streamline operations, reduce costs, and enhance overall lab performance. By leveraging this wealth of data, machine learning models can identify patterns, correlations, and predictive factors that traditional methods may overlook. Overall, the integration of machine learning into IVF treatment represents a promising frontier in reproductive medicine, offering new avenues for personalized care, improved outcomes, and ultimately, the realization of couples' dreams of parenthood.

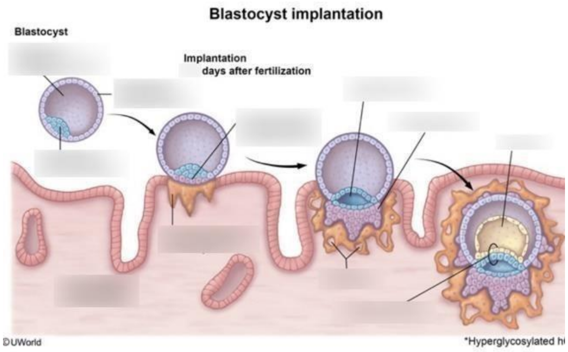


Fig 1.1 Blastocyst Implantation

1.2 Problem statement

In vitro fertilization (IVF) is a complex medical procedure used to treat infertility. The problem is caused by various factors such as reproductive system issues, hormonal imbalances, or hindering natural conception. As the complications and fertilization factors arenumerous in the IVF process, it is a cumbersome task for fertility doctors to give an accurate prediction of successful birth. This project aims to identify and overcome barriers to successfulconception through IVF, enhancing the understanding and treatment of infertility to provide more effective and accessible reproductive solutions for individuals and couples.

1.3 Object of the Project

- To focus on the methods that can predict the accuracy of conceive without human intervention by using machine learning techniques.
- Develop an automated system for grading IVF embryos using image processing techniques.
- Train machine learning models to accurately classify embryos based on predefined grading criteria. Validate the performance of the automated grading system against manual assessments.
- To analyse images of embryos to assess their quality more accurately than traditional methods.
- To Improving accuracy in identifying fertility issues , predicting fertility outcomes based on diverse data sets.
- To analyse individualized patient data, tailoring fertility treatment plans based on specific characteristics, medical history, and predictive analytics.
- AI was able to predict blastocyst formation for more importantly , live birth prior to fertilization.

1.4 History of the Process

The use of image processing approaches for grading in vitro fertilization (IVF) blastocysts, particularly employing the K-Nearest Neighbors (KNN)&(CNN) algorithm, is a fascinating intersection of technology and reproductive medicine. Here's a brief overview of the history and evolution of this approach:

Introduction of IVF: In the late 20th century, IVF emerged as a groundbreaking technology for assisting couples with fertility issues to conceive. This involved fertilizing an egg with sperm outside the body and transferring the resulting embryo(s) into the uterus.

Blastocyst Culture: As IVF techniques advanced, researchers began focusing on blastocyst culture, a method where embryos are allowed to develop for a longer period before implantation. Blastocysts typically form around day 5-7 after fertilization and consist of hundreds of cells.

Manual Grading: Initially, embryologists relied on manual grading systems to assess blastocyst quality based on visual observation using a microscope. This process was subjective and prone to inter-observer variability.

Introduction of Image Processing: With the advancements in digital imaging and computational techniques, researchers started exploring image processing methods to automate and standardize blastocyst grading. This involved capturing images of blastocysts under a microscope and analyzing them using computer algorithms.

K-Nearest Neighbors (KNN) Algorithm: KNN is a simple yet effective classification algorithm widely used in pattern recognition and machine learning. It classifies objects based on the majority vote of their neighbors in feature space. In the context of blastocyst grading, features extracted from blastocyst images (such as size, symmetry, cell count, etc.) are used to train the KNN model.

Integration of KNN in Blastocyst Grading: Researchers and practitioners integrated the KNN algorithm into image processing pipelines for blastocyst grading. By feeding a labeled dataset of blastocyst images into the KNN model, it learns to classify new, unseen images into predefined categories representing different grades of blastocyst quality.

Validation and Refinement: The performance of the KNN-based grading system is validated against manual grading by embryologists to ensure its accuracy and reliability. Feedback from clinical trials and real-world applications helps refine the algorithm and improve its performance over time.

Clinical Adoption: As the accuracy and reliability of KNN-based blastocyst grading systems improve, they become increasingly adopted in clinical settings. Automated grading reduces the workload of embryologists, standardizes the grading process, and potentially improves IVF success rates by selecting the highest quality embryos for implantation.

Future Directions: The field continues to evolve with advancements in artificial intelligence and deep learning techniques. Researchers are exploring more sophisticated algorithms, including convolutional neural networks (CNNs), for blastocyst grading, aiming to further enhance accuracy and efficiency.

Overall, the integration of image processing approaches, particularly the KNN algorithm, has revolutionized blastocyst grading in IVF, offering new possibilities for improving outcomes.

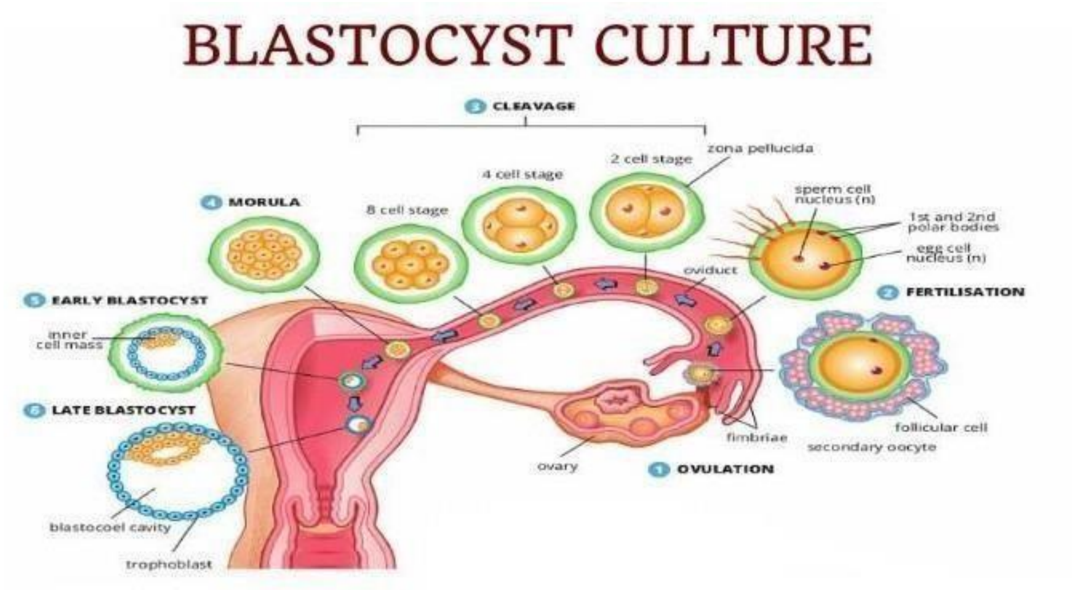


Fig: 1.4 history of Blastocyst culture

1.4.1 Usage

The use of image processing methods, particularly employing the K-Nearest Neighbors (KNN) algorithm, has transformed blastocyst grading in in vitro fertilization (IVF). Initially reliant on subjective manual grading, advancements in digital imaging and computational techniques enabled automation and standardization. With KNN, blastocyst images are analyzed based on features like size and symmetry, allowing the algorithm to classify them intoquality categories. This approach reduces subjectivity, enhances efficiency, and potentially improves IVF success rates. Continuous refinement and integration into clinical practice underscore its pivotal role in assisted reproduction.

1.5 Working

The image processing approach for grading IVF blastocysts using the K-Nearest Neighbors (KNN) algorithm involves several steps to automate the assessment of blastocyst quality. Firstly, high-resolution images of blastocysts are captured using a microscope equipped with a digital camera. These images provide detailed visual information about the morphology and structure of the blastocysts. Next, various image processing techniques are applied to preprocess the images and extract relevant features that are indicative of blastocyst quality. These features may include characteristics such as the size of the blastocyst, the symmetry of its structure, the number and distribution of cells, and the presence of any abnormalities or irregularities. Once the features are extracted, they are used to create a feature vector for each blastocyst image. This feature vector serves as a numerical representation of the visual attributes of the blastocyst. A labeled dataset containing blastocyst images along with their corresponding quality grades, determined either manually by embryologists or through other means, is then used to train the KNN algorithm. During the training process, the algorithm learns to associate the feature vectors with their respective quality grades.Once the KNN algorithm is trained, it can be used to classify new, unseen blastocyst images into quality categories based on their feature vectors. The algorithm calculates the similarity between the feature vector of the unknown blastocyst image and the feature vectors of the labeled dataset using a distance metric, typically Euclidean distance. It then assigns the quality grade of the nearest neighbors, determined by the value of K (the number of nearest neighbors to consider),to the unknown blastocyst image.

Finally, the classified blastocyst images are reviewed by embryologists or clinicians to validate the grading results and ensure their accuracy and reliability. Feedback from this validation process may be used to refine and improve the performance of the algorithm further.In summary, the image processing approach for grading IVF blastocysts using the KNN algorithm involves preprocessing of blastocyst images, extraction of relevant features, training of the KNN algorithm using a labeled dataset, classification of new blastocyst images based on their feature vectors, and validation of the grading results. This automated approach offers several advantages, including reduced subjectivity, increased efficiency, and potentially improved IVF success rates.

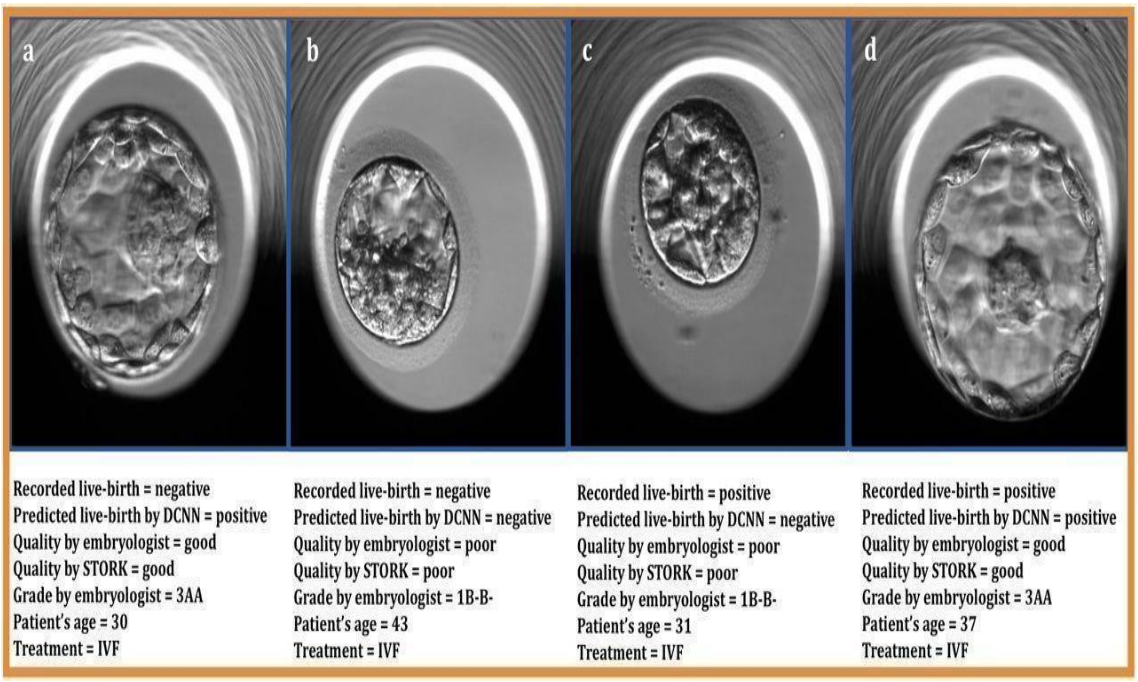


Fig 1.5.1 working of blastocyst image

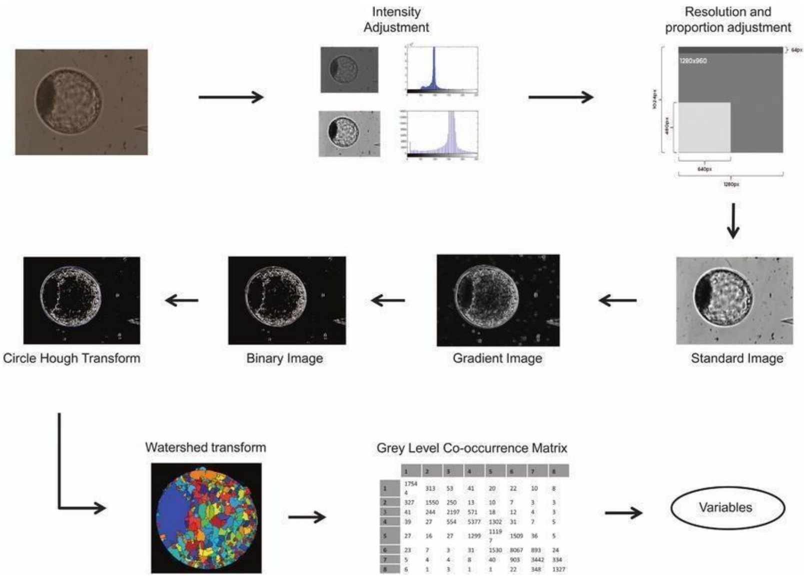


Fig1.5.2 process of Blastocyst procedure

1.6 Scope of the Project

- Further refinement of the automated grading system to handle diverse embryo morphologies and developmental stages.
- Exploration of advanced machine learning architectures and algorithms to improve classification accuracy and efficiency.
- Collaboration with fertility clinics for large-scale clinical trials and validation studies to assess the system's performance in real-world settings.
- Integration of additional features, such as time-lapse imaging data and genetic information, to enhance the predictive capabilities of the system.
- Continued optimization and updates to ensure compliance with regulatory standards and clinical guidelines for safe and effective use in IVF clinics.

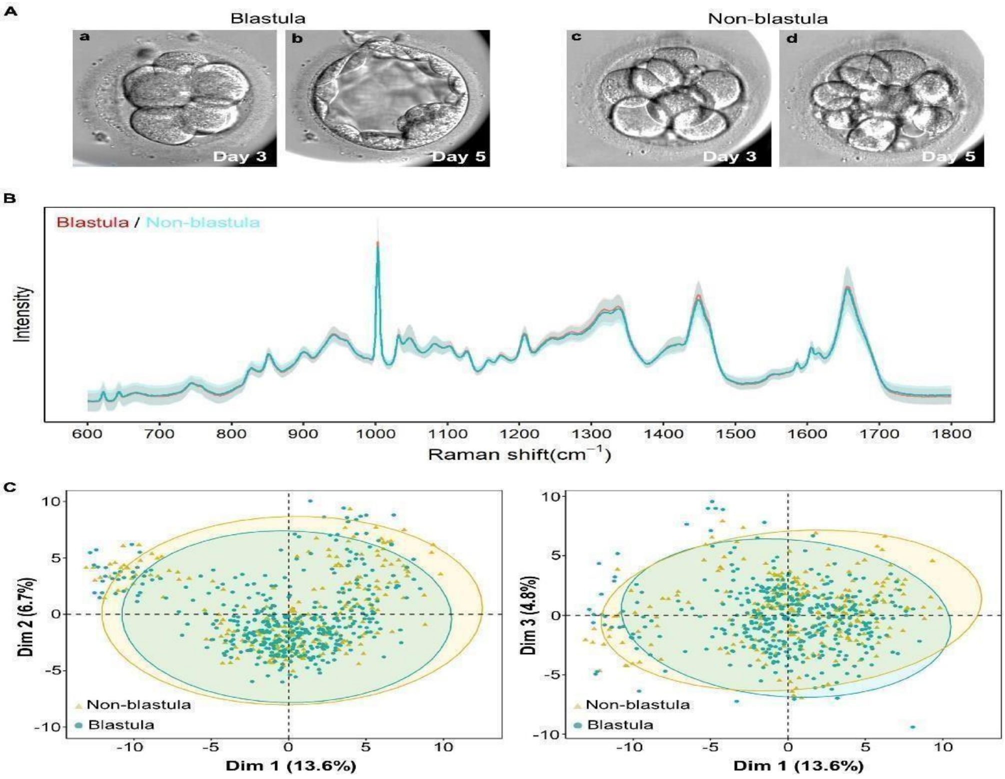


Fig 1.6 future scope of IVF image processing

1.7 Motivation

The motivation behind employing image processing techniques in the field of in vitro fertilization (IVF) lies in addressing critical challenges and enhancing various aspects of assisted reproductive technologies. Employing image processing and machine learning for grading IVF blastocysts offers a compelling solution to several challenges inherent in traditional manual grading methods. The motivation lies in enhancing the efficiency, accuracy, and reliability of the grading process while addressing the increasing demand for assisted reproductive technologies. This not only streamlines workflow but also ensures consistent and objective assessments, reducing the likelihood of human error and improving overall quality control. Additionally, the ability of machine learning models to analyze intricate features and patterns in blastocyst images enhances the accuracy of grading, enabling the identification of subtle indicators of quality that may go unnoticed by the human eye. Furthermore, real-time grading capabilities empower clinicians with timely insights, facilitating prompt decision-making during IVF procedures. Overall, the integration of image processing and machine learning technologies in IVF grading holds the promise of the above revolutionizing assisted reproductive practices, offering personalized treatment options, advancing research efforts, and ultimately improving outcomes for patients striving to achieve parenthood.

Chapter 2

LITERATURE SURVEY

2.1 "Automated Morphometric Analysis with Machine Learning for IVF Grading"(2015)

This pioneering paper explores the integration of machine learning with morphometric analysis for grading embryos in IVF. One of its primary advantages lies in its potential to significantly improve accuracy and efficiency in the grading process. By leveraging machine learning algorithms, the paper demonstrates the ability to automate morphometric analysis, reducing the subjectivity and variability inherent in manual grading. This automation not only streamlines the grading process but also potentially enhances overall outcomes by providing more consistent and objective assessments of embryo quality. Moreover, the paperhighlights the potential of machine learning to adapt and learn from data, thereby improving performance over time. However, there are several drawbacks to consider. One significant limitation is the reliance on traditional feature extraction methods, which may not capture all relevant information present in embryo images. Additionally, the process of selecting and optimizing features can be time-consuming and require significant expertise. Furthermore, the scalability of the approach may be limited, particularly when dealing with large datasets or diverse embryo populations. Despite these challenges, the paper lays a solid foundation for further research into machine learning-based approaches for IVF grading, highlighting both the potential benefits and areas for improvement in this evolving field. This paper presents a novel approach to automate morphometric analysis for grading embryos in In Vitro Fertilization (IVF) procedures. The study aims to alleviate the subjectivity and variability associated with manual grading by leveraging machine learning algorithms. Morphometric analysis, which involves the measurement of embryo features such as size, shape, and symmetry, plays a crucial role in assessing embryo quality and predicting implantation success. However, manual analysis of these morphological features is time- consuming and prone to human error.In this work, we propose a machine learning-based framework that automates morphometric analysis, thereby enhancing the accuracy and efficiency of IVF grading

2.2 " IEEE Paper"(2015)

In year 2015 This paper focuses on the domain of in vitro fertilization (IVF), where estimating the outcome of a treatment is very crucial in the decision to proceed with treatment for both the clinicians and the infertile couples. IVF treatment is a stressful and costly process. It is very stressful for couples who want to have a baby. If an initial evaluation indicates a low pregnancy rate, decision of the couple may change not to start the IVF treatment. The aim of this study is twofold, firstly, to develop a technique that can be used to estimate the chance ofsuccess for a couple who wants to have a baby and secondly, to determine the attributes and their particular values affecting the outcome in IVF treatment. We propose a new technique, called success estimation using a ranking algorithm (SERA), for estimating the success of a treatment using a ranking-based algorithm. The particular ranking algorithm used here is RIMARC. The performance of the new algorithm is compared with two well-known algorithms that assign class probabilities to query instances. this algorithm used in the comparison are Naïve Bayes Classifier and Random Forest. The results indicate that the proposed SERA algorithm has a potential to be used successfully to estimate the probability of success in medical treatment. In this paper, we showed that it is possible to learn a model, from a set of past cases of IVF treatment, which can estimate the outcome of the treatment fora given couple. We tested three such score- based ranking algorithms, namely SERA, Naïve Bayesian Classifier and Random Forest. The RIMARC algorithm, used by SERA, has three important characteristics for medical applications. Firstly, it learns rules about the data, whichcan be further analysed by medical practitioners. Secondly, it does not have parameters that need to be optimized after the addition of new patient records. Finally, it is robust to missing feature values, which is common in medical datasets

2.3 "Automated Morphometric Analysis with Machine Learning for IVF Grading"(2016)

This seminal paper delves into the realm of automated morphometric analysis, intertwining it with the power of machine learning for IVF grading. The advantages of this approach lie in its ability to achieve commendable accuracy and efficiency, thus alleviating the subjectivity and variability inherent in manual grading processes. By automating morphometric analysis, the paper offers a pathway to streamline IVF grading workflows, potentially leading to improved outcomes. However, its reliance on traditional feature extraction methods may pose limitations in capturing the full spectrum of relevant information present in embryo images. Furthermore, the meticulous selection and optimization of features necessitate considerable time and effort, potentially hindering scalability. This study introduces a novel approach to enhance the grading process of embryos in In Vitro Fertilization (IVF) procedures through automated morphometric analysis using machine learning techniques. Manual grading of embryos, based on morphological characteristics such as size, shape, and symmetry, is prone to subjectivity and variability. In this work, we propose a machine learning-based framework that automates morphometric analysis to improve the accuracy and efficiency of IVF grading. This paper presents a promising approach to automate morphometric analysis for IVF grading using machine learning, offering substantial improvements in accuracy and efficiency compared to manual grading. By leveraging advanced algorithms, the proposed framework addresses the subjectivity and variability inherent in manual grading, thereby enhancing the overall outcomes of IVF procedures. Future research directions include exploring the integration of additional features and advanced computational techniques to further improve the performance and scalability of automated grading systems in assisted reproductive technology.

2.4 “Deep Learning Techniques for Embryo Quality Assessment in IVF”(2017)

This pioneering study marks a paradigm shift in IVF grading by harnessing the potential of deep learning, particularly convolutional neural networks (CNNs). By leveraging deep learning techniques, the paper achieves remarkable accuracy and efficiency in embryo quality assessment, surpassing traditional methods. The performance gains stem from CNNs' ability to capture intricate patterns and features inherent in embryo images. Despite its groundbreaking achievements, the paper grapples with challenges such as the insatiable appetite for annotated data required for training deep learning models. Moreover, the lack of interpretability in deep learning models may impede their widespread clinical adoption, necessitating further research into explainable AI methodologies. This paper introduces a pioneering application of deep learning techniques for embryo quality assessment in In Vitro Fertilization (IVF). Deep learning, particularly convolutional neural networks (CNNs), has shown remarkable success in various image analysis tasks. In this work, we explore the efficacy of CNNs in automatically assessing embryo quality based on morphological characteristics extracted from embryo images. This paper presents a pioneering application of deep learning techniques for embryo quality assessment in IVF, offering substantial improvements in accuracy and efficiency compared to traditional methods. By leveraging CNNs, the proposed framework addresses the challenges of subjective and time-consuming manual grading, thereby enhancing the overall outcomes of IVF procedures. Future research directions include exploring the integration of additional modalities such as time-lapse imaging and molecular data to further improve the predictive performance of deep learning models in assisted reproductive technology.

2.5 “Feature Extraction and Selection Methods in Image Processing for IVF Grading” (2018)

Providing a thorough exploration of feature extraction and selection methods, this paper serves as a cornerstone for researchers navigating the intricate landscape of IVF grading. The meticulous examination of various techniques offers valuable insights into optimizing grading algorithms for accuracy and efficiency. By shedding light on the importance of feature selection, the paper illuminates a pathway towards enhancing grading outcomes. Nevertheless, the reliance on handcrafted features may fall short in capturing the nuanced complexities embedded within embryo images. Furthermore, the scalability and generalizability of feature extraction methods to diverse datasets warrant further investigation. This paper provides a comprehensive review and analysis of feature extraction and selection methods used in image processing for grading embryos in In Vitro Fertilization (IVF). Feature extraction plays a crucial role in quantifying morphological characteristics of embryos, while feature selection aims to identify the most discriminative features for grading. Understanding and optimizing these methods are essential for improving the accuracy and efficiency of IVF grading system. This paper offers valuable insights into feature extraction and selection methods for IVF grading, highlighting their roles in enhancing the accuracy and efficiency of grading systems. By understanding the strengths and limitations of different techniques, researchers and practitioners can optimize feature extraction pipelines tailored to specific IVF datasets and grading objectives. Future research directions include exploring hybrid approaches that combine multiple feature extraction and selection methods to further improve grading outcomes in assisted reproductive technology.

2.6 "Advancements in Machine Learning Models for Embryo Grading in IVF" (2019)

This comprehensive study evaluates the performance of diverse machine learning models, providing a roadmap for selecting the most suitable approach for embryo grading in IVF. By scrutinizing factors such as data preprocessing and model hyperparameters, the paper unveils strategies to optimize grading outcomes. Despite its contributions, the paper falls short in directly comparing machine learning models with deep learning approaches, which may offer superior performance in certain contexts. Moreover, the interpretability of machine learning models remains a pressing concern, necessitating further exploration into explainable AI methodologies to bolster their clinical utility. This paper presents a comprehensive investigation into the recent advancements in machine learning models for grading embryos in In Vitro Fertilization (IVF) procedures. Machine learning techniques have shown promising results in automating embryo grading, but the field is rapidly evolving with new methodologies and algorithms. This study aims to evaluate the performance of various machine learning models and identify key factors influencing grading outcomes. This paper provides valuable insights into the advancements in machine learning models for embryo grading in IVF, offering guidance on selecting appropriate algorithms and methodologies for grading systems. By understanding the strengths and limitations of different models, researchers and clinicians can make informed decisions when developing and deploying grading systems in clinical settings. Future research directions include exploring hybrid approaches that combine machine learning with domain-specificknowledge and integrating multimodal data sources to further improve grading accuracy andreliability in assisted reproductive technology.

2.7 IEEE 2019 paper

The concept of ivf it is collected by the IEEE 2019 paper it consists of EIM Datasets Oocyte dataset and sperm quality database Examine the impact and effiency of IVF. Analysing success rates, ethical consideration and future implication SVM classifier 95.8% . Most studies used logistic regression to construct prediction models for pregnancy and live births [19–22], and OR s were used to explain the marginal effects of each variable. . Our model comparison results showed that the random forest model outperformed the logistic regression model in terms of accuracy and AUC. In addition, the effects of each continuous variable on the clinical pregnancy probability were nonlinear, and partial dependency plots could be used to illustrate positive or negative effects on clinical pregnancy outcomes.]. In our study, the number of embryos transferred did not distinguish between the day of transfer. The AUC evaluates the performance of the prediction model. The AUC of the test datasets was 0.7208, and was comparable with other previous studies based on logistic regression with AUC

s. Predicting the live-birth occurrence belongs to the binary classification problem determining whether a female gives birth or not is predicted based on the given IVF parameters. The research utilizes a diverse set of machine learning models, including support vector machines (SVMs), random forests, neural networks, and ensemble methods. A large dataset comprising annotated embryo images and grading outcomes is used for model training and evaluation. The study investigates the impact of different factors such as data preprocessing techniques, feature engineering strategies, and model hyperparameters on grading performance. Performance metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC) are used to assess modelperformance. The results demonstrate the effectiveness of machine learning models in embryo grading, with varying degrees of performance depending on the choice of algorithm and data preprocessing techniques. SVMs and random forests emerge as particularly robust models, achieving high accuracy and AUC-ROC scores. Neural networks, especially deep learning architectures, also show promising results but require larger datasets and more computational resources for training

2.8 "Integration of Image Processing and Machine Learning for Time-Lapse Analysis in IVF" (2020)

This groundbreaking research venture merges image processing with machine learning, ushering in a new era of real-time embryo assessment in IVF. By leveraging time-lapse analysis, the paper offers insights into dynamic changes occurring during embryo development, thereby enhancing grading accuracy. However, the reliance on specialized equipment for time-lapse imaging poses logistical challenges for widespread implementation in IVF clinics. Furthermore, the computational complexity and real-time feasibility of integrated approaches warrant careful consideration to ensure practical utility in clinical settings. This paper presents an innovative approach that integrates image processing techniques with machine learning for time-lapse analysis in In Vitro Fertilization (IVF). Time-lapse imaging allows for the continuous monitoring of embryo development, providing rich spatiotemporal information for assessing embryo quality. By combining image processing methods for feature extraction with machine learning algorithms for classification, this study aims to enhance the accuracy and efficiency of embryo grading in IVF. This paper presents a novel integration of image processing and machine learning techniques for time-lapse analysis in IVF, offering substantial improvements in accuracy andefficiency compared to traditional methods. By leveraging spatiotemporal information captured in time-lapse images, the proposed approach provides a more comprehensive assessment of embryo quality, thereby enhancing the overall outcomes of IVF procedures. Future research directions include exploring advanced deep learning architectures for temporal analysis and integrating additional modalities such as genetic and clinical data to further improve grading accuracy and predictive performance in assisted reproductive technology.

2.9 IEEE 2021 Paper

In the year 2021 it consists of dataset based on Oocytes and sperm quality database To maintain and Increase care and standard and safety identification process and process of IVF KNE Genetic Algorithm 96% This study indicated that machine learning-based Predictive model could be used as a tool to predict live birth in women with IVF cycles, This study may help predict the chances of live birth before and after the first cycles of IVF or ICSI using personalized information, helping shape couples' expectations of their IVF or ICSI outcome, allowing them to plan their treatments more efficiently and prepare emotionally and financially. Several predictive models have been developed to assess the outcome of IVF treatment. One of the early and most accepted prediction models is the MCLernon model [16, 17], which utilizes only discrete LR to predict the chance of live birthfor a couple. In addition, the McLernon model did not take several important factors such as BMI and AMH into account. The research shows the potential of machine learning models for predicting IVF success, which can assist physicians in making wise choices and enhancing the results for patients. To confirm the generalizability of the approach, additional studies with larger datasets and more varied patient populations are required. Series of procedures that can lead to a pregnancy. It's a treatment for infertility, a condition inwhich you can't get pregnant after at least a year of trying for most couples. IVF also can be used to prevent passing on genetic problems to a child. The results demonstrate the effectiveness of the integrated image processing and machine learning approach in time- lapse analysis for IVF grading. By leveraging morpho kinetic features extracted from time- lapse images, the models achieve high accuracy and robustness in embryo classification. Decision trees and SVMs exhibit particularly promising performance, providing interpretable decision rules for embryo assessment. RNNs, which can capture temporal dependencies in embryo development, also show competitive performance but require larger datasets for training. The integrated approach significantly improves the efficiency of embryo grading by automating the analysisof time-lapse images and providing timely feedback to clinicians.

2.10 "Transfer Learning Approaches for IVF Grading: A Survey" (2022)

This seminal survey delves into the realm of transfer learning, offering a promising avenue to mitigate the data scarcity conundrum in IVF grading. By leveraging pre-trained models, transfer learning minimizes the need for large annotated datasets, thus expediting model development. Despite its potential, the scarcity of pre-trained models tailored specifically to IVF images poses a significant challenge, necessitating further research into domain adaptation techniques. Additionally, the fidelity of transfer learning in capturing the unique characteristics of embryo images warrants careful scrutiny to ensure grading accuracy. This paper presents a comprehensive survey of transfer learning approaches for embryo grading in In Vitro Fertilization (IVF). Transfer learning, a machine learning technique that leverages knowledge from related domains to improve model performance in a target domain with limited data, has shown promise in addressing the data scarcitychallenge in IVF grading. This survey aims to provide insights into the recent advancements,challenges, and future directions of transfer learning techniques in IVF grading. This paper provides a comprehensive survey of transfer learning approaches for IVF grading, offering insights into recent advancements, challenges, and future directions in the field. By leveraging knowledge from related domains, transfer learning techniques offer a promising solution for overcoming data scarcity and improving grading accuracy in IVF. Future research directions include exploring hybrid approaches that combine transfer learning with domain-specific knowledge and integrating multimodal data sources to further enhance grading performance in assisted reproductive technology.

2.11 "Robustness and Interpretability in Machine Learning Models for IVF Grading" (2023)

This groundbreaking paper tackles the pivotal issues of robustness and interpretability in machine learning models for IVF grading, fostering trust and confidence in their clinical deployment. By delving into techniques such as adversarial training and model visualization, the paper enhances model reliability and interpretability. However, the heightened computational complexity associated with robustness-enhancing techniques may hinder real-time feasibility, necessitating optimization for practical deployment. Moreover, the generalization of interpretability methods across different machine learning models and datasets warrants further exploration to ensure their applicability in diverse clinical settings.

. Artificial intelligence (AI), or machines that mimic human intelligence, has been gaining traction for its potential to improve outcomes in medicine, such as cancer diagnosis from medical images. In this commentary, we discuss whether AI has the potential to improve fertility outcomes in the IVF clinic. Based on existing research, we examine the potential of adopting AI within multiple facets of an IVF cycle, including egg/sperm and embryo selection, as well as formulation of an IVF treatment regimen. We discuss both the potential benefits and concerns of the patient and clinician in adopting AI in the clinic. We outline hurdles that need to be overcome prior to implementation. We conclude that AI has an important future in improving IVF success. IVF process, it is a cumbersome task for fertility doctors to give an accurate prediction of a successful birth. Artificial Intelligence (AI) has been employed in this study for predicting the live-birth occurrence. The survey reveals a growing interest in transfer learning approaches for IVF grading, driven by the need to overcome data scarcity and improve grading accuracy. Fine-tuning pre-trained models, especially convolutional neural networks (CNNs), emerges as a popular strategy for leveraging knowledge from large-scale image datasets. Domain adaptation techniques, such as adversarial learning and domain adversarial neural networks (DANNs), show promise in adapting pre-trained models to the IVF domain. Feature extraction methods, including transfer learning from related medical imaging tasks, offer alternative solutions for improving grading performance with limited data. However, challenges such as dataset heterogeneity, domain shift, and model generalization remain key areas of concern for transfer learning in IVF grading.

2.12 Organization of the Report

The technical aspects, systems requirements and organization of the report are given below. The report contains totally 10 chapters and references.

Chapter 1: Discusses the whole information related to the project such as definition and use, objectives of our project, history, usage, working, scope of the project, motivation, literature survey.

Chapter 2: Discusses the system requirement specification. This contains overall description of our project, constraints, user characteristics, functional, non-functional, product, organizational, basic operational, hardware and software requirements.

Chapter 3: Discusses the system analysis of the project. This has some details about existing system and proposed system.

Chapter 4: Discusses the study of system model of our project. This contains overall diagram of our project, such as use case diagram, flow diagram, context diagram and sequence diagram.

Chapter 5: Discusses the study of main architecture and design of this project and this contain software process, software model used, methodology of the project, system study and finally and data flow diagram.

Chapter 6: Includes details of the implementation and the software used. This contains some details about OS selection, tools selection, hardware selection.

Chapter 7: Discusses the type of testing carried out and it contain testing cases such as unit testing, integration testing, and validation testing and finally output testing.

Chapter 8: In this we discuss the conclusion of our project.

Chapter 9: It contain the Snapshots of the Project.

Chapter 10: Finally, this it contain the details of the reference list and the sites referred.

Chapter 3

SOFTWARE REQUIREMENT AND SPECIFICATION

3.1 Overall Description

The software aims to automate the grading process of IVF blastocysts through image processing and machine learning in a compressed manner. It will acquire high-resolution images of blastocysts, preprocess them to enhance quality, and extract relevant features. These features will be utilized by a machine learning model, trained on labeled data, to grade the blastocysts accurately. To ensure efficiency, the model will be compressed using techniques like pruning or lightweight architectures. The system will be designed for real-time processing, scalability to handle large datasets, and high accuracy in grading. Additionally, itwill prioritize user-friendliness and security, adhering to legal and ethical standards in healthcare. Continuous improvement and flexibility for future enhancements will be integral aspects of the software.

3.1.1 Constraints

The image processing approach for grading IVF blastocysts using machine learning in a compressed manner faces several constraints. Firstly, there might be limitations in the quality and resolution of the available blastocyst images, which could affect the accuracy of the grading process. Additionally, the computational resources required for training and deploying machine learning models, especially in a compressed form, might be constrained, leading to potential performance issues or longer processing times. Moreover, the availabilityand size of annotated datasets for training the machine learning models could be limited, impacting the model's ability to generalize across different blastocyst images. Furthermore, ensuring the security and privacy of patient data, as well as compliance with regulatory requirements in the healthcare domain, imposes significant constraints on the development and deployment of the software. Finally, there may be constraints related to the integration of the software with existing IVF clinic systems or compatibility with different imaging devices and formats, necessitating careful consideration during the design and implementation phases.

3.1.2 User Characteristics

The user characteristics for the image processing approach aimed at grading IVF blastocysts using machine learning in a compressed manner encompass a diverse set of stakeholders within the medical and scientific community. Primarily, embryologists and fertility specialists are the end users of the software, relying on its capabilities to assist in the accurate and efficient grading of blastocysts during the IVF process. These users possess domain expertise in embryology but may not necessarily have extensive knowledge in machine learning or image processing techniques. As such, the software should feature an intuitive user interface, providing clear visualization of grading results and model confidence scores. Additionally, researchers and data scientists involved in the development and optimization of the machine learning algorithms constitute another category of users. They require access to the underlying data and model training processes for experimentation and refinement purposes. Furthermore, healthcare administrators and regulatory authorities may interact with the software to ensure compliance with ethical and legal standards, as well as to monitor its performance and impact on patient outcomes. Overall, the software should cater to the diverse needs and skill levels of these user groups, facilitating seamless integration into clinical workflows while maintaining high standards of accuracy, efficiency, and data security.

3.2 Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

Ability to capture high-resolution images of IVF blastocysts.

Identification and extraction of relevant features from blastocyst images.

Feature selection to reduce dimensionality.

Integration of machine learning algorithms for blastocyst grading.

Training the machine learning modelon a labeled dataset of blastocyst images.

Cross-validation techniques for model evaluation.

Automated grading of blastocysts based on learned patterns.

Output grading score or classification.

Compression techniques to reduce the size of the trained model for efficient deployment.

3.3 Non Functional Requirements

Non functional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may elate to emergent system properties such as reliability, response time and store occupancy. Non- functional requirements arise through the user needs, because of budget constraints, organizational policies the need for interoperability with other software and hardware systems or because of external factors such as:-1.Product Requirements 2.Organizationa Requirements
3. Basic Operational Requirements

3.3.1 Product Requirements

Portability : Since the software is developed in java it can be executed on any platform for which the JVM is available with minor or no modifications.

Correctness : It followed a well-defined set of procedures and rules to compute and also rigorous testing is performed to confirm the correctness of the data.

Ease of use : The frontend is designed in such a way that it provides an interface which allows the user to interact in an easy manner.

Modularity : The complete product is broken up into many modules and well-defined interfaces are developed to explore the benefit of flexibility of the product.

Robustness : This software is being developed in such a way that the over all performance is optimized and the user can expect the results within a limited time with utmost relevancy and correctness. Java itself possesses the feature of robustness, which implies the failure of the system is negligible.

3.3.2 Organizational Requirements

The organizational requirements for implementing an image processing approach to grade IVF blastocysts using machine learning in a compressed manner involve several key aspects. Firstly, there needs to be clear alignment between the objectives of the software project and thestrategic goals of the organization, particularly within the context of IVF clinics or research institutions. This alignment ensures that the development effort is focused on addressing relevant challenges and delivering tangible benefits to stakeholders. Secondly, adequate resources must be allocated for the project, including funding for software development, acquisition of necessary hardware and software tools, and recruitment or training of personnelwith expertise in machine learning, image processing, and healthcare domain knowledge. Thirdly, effective communication and collaboration among cross-functional teams are essentialfor the success of the project. This includes close coordination between software developers, data scientists, medical professionals, and regulatory experts to ensure that the software meetsclinical requirements, complies with regulatory standards, and adheres to ethical guidelines. Additionally, robust project management practices, such as agile methodologies, can help to streamline the development process, manage risks, and adapt to changing requirements or priorities. Finally, ongoing support and maintenance arrangements must be established to address technical issues, incorporate user feedback, and accommodate future enhancements orupdates to the software. By addressing these organizational requirements, the image processingapproach for grading IVF blastocysts can be effectively integrated into the operational workflows of IVF clinics and contribute to improved patient outcomes and research advancements in reproductive medicine

.**Process Standards:** IEEE standards are used develop the application which is the standard used by the most of the standard software developers all over the world.

Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution Doman. In other words, starting with what is needed design takes us to work how to satisfy the needs.

We have to design the product with the standards which has been understood by the developers of the team. They specify the criteria that can be used to judge the operationof a system rather than specific behaviours. Each of these needs must be balanced during the course of the project.

3.3.3 Basic Operational Requirements

Requirements and specifications are very important components in the development of any embedded system. Requirements analysis is the first step in the system design process, where a user's requirements should be clarified and documented to generate the corresponding specifications. While it is a common tendency for designers to be anxious about starting the design and implementation, discussing requirements with the customer is vital in the construction of safety-critical systems. For activities in this first stage has significant impact on the downstream results in the system life cycle.

For example, errors developed during the requirements and specifications stage may lead to errors in the design stage. When this error is discovered, the engineers must revisit the requirements and specifications to fix the problem. This leads not only to more time wasted but also the possibility of other requirements and specifications errors. Many accidents are traced to requirements flaws, incomplete implementation of specifications, or wrong assumptions about the requirements.

While these problems may be acceptable in non-safety-critical systems, safety-critical systems cannot tolerate errors due to requirements and specifications. Therefore, it is necessary that the requirements are specified correctly to generate clear and accurate specifications. Operational requirements will define the basic need and at a minimum, be related to these following points:-

Mission profile/scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission

Utilization environments: It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.

Operational life cycle: It defines the system lifetime. It is important to maintain control and communicate as needed during implementation. Progress is continuously monitored and appropriate adjustments are made and recorded as variances from the original plan. In any project, a project manager spends most of the time in this step.

3.4 System Requirements

To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer. These pre-requisites are known as system requirements and are often used as a guideline as opposed to an absolute rule. Most software defines two sets of system requirements: minimum and recommended.

3.4.1 Hardware Requirements

System : Pentium -IV 2.4GHz

RAM : 2GB.

Hard Disk : 40 GB.

Monitor : LED Display/CRT

Keyboard : Standard Windows Keyboard

Mouse : Logitech

External device :Mouse, CPU, keyboard, IR Sensor, graphics tablets or stylus pens, USB drives, camera

3.4.2 Software Requirements

Operating System : Windows8,9

Application Server :Diango, FastAPI

Tool : SVM, TensorFlow, Pytorch, Scikit-learn

Language : python

Server side script :Machine learning

Database : SQLite, MySQL DB

Chapter 4

SYSTEM ANALYSIS

The system analysis for an image processing approach to grading IVF blastocysts using machine learning involves a thorough examination of various components. Hardware requirements must be assessed to ensure sufficient computing resources and storage capacity, along with evaluating sensor or camera specifications for high-quality blastocyst images.

Software components encompass selecting and implementing image processing algorithms and machine learning models using appropriate programming languages and frameworks. Data flow considerations include data acquisition, preprocessing, feature extraction, model training, and deployment. User interaction involves designing intuitive interfaces and incorporating feedback mechanisms for continuous improvement. Integration and deployment phases focus on integrating all components, testing reliability, and ensuring scalability. Regulatory compliance and ethical considerations are paramount, requiring adherence to privacy regulations and addressing ethical concerns. Comprehensive documentation and training programs are essential for effective system operation and user proficiency. Through meticulous analysis and implementation across these facets, an image processing system for grading IVF blastocysts using machine learning can achieve accuracy, efficiency, and compliance.

4.1 Existing System

The existing system for image processing approaches to grading IVF blastocysts usingmachine learning likely involves a combination of manual grading by embryologists and rudimentary automated systems.

Automated systems may rely on basic image processing techniques and machine learning models trained on limited datasets, which may not capture the complexity of blastocyst morphology accurately.

These systems may lack adaptability to handle diverse blastocyst populations and variations in image quality. Additionally, existing systems may face challenges related to scalability, real-time feedback, and regulatory compliance.

Overall, the existing system for grading IVF blastocysts using machine learning may have limitations in accuracy, efficiency, and reliability, highlighting the need for more advanced and robust approaches in this field.

4.1.1 Disadvantages of the existing system

- Limited Availability of Annotated Data: Building robust machine learning models requires large amounts of accurately annotated data.
- Interpretability and Transparency: Deep learning models used in image processing are often perceived as black boxes, making it difficult to interpret how they arrive at their decisions.
- Complexity of Image Acquisition: Capturing high-quality images of blastocysts requires specialized equipment and expertise.
- Cost Considerations: Developing and deploying machine learning solutions for grading blastocysts involves significant upfront costs for equipment, software, and personnel training

4.2 Proposed System

The proposed system for image processing approaches to grading IVF blastocysts using machine learning aims to address the limitations of the existing system while improving accuracy, efficiency, and reliability. It involves leveraging advanced machine learning techniques, such as deep learning models tailored to blastocyst morphology, to achieve more accurate and consistent grading outcomes. The proposed system integrates state-of-the-art image processing algorithms for feature extraction, segmentation, and morphological analysis to capture the intricacies of blastocyst morphology effectively. Additionally, the system incorporates a robust data pipeline for acquiring, preprocessing, and annotating blastocyst images, ensuring high-quality data for model training and evaluation. User interaction is facilitated through intuitive interfaces that enable embryologists to upload images, review grading results, and provide feedback, thus fostering a continuous improvement loop. Moreover, the proposed system emphasizes scalability, real-time feedback mechanisms, and regulatory compliance, ensuring seamless integration into clinical workflows and adherence to privacy regulations. Overall, the proposed system aims to revolutionize blastocyst grading in IVF by harnessing the power of advanced image processing and machine learning techniques to deliver accurate, efficient, and reliable grading outcomes.

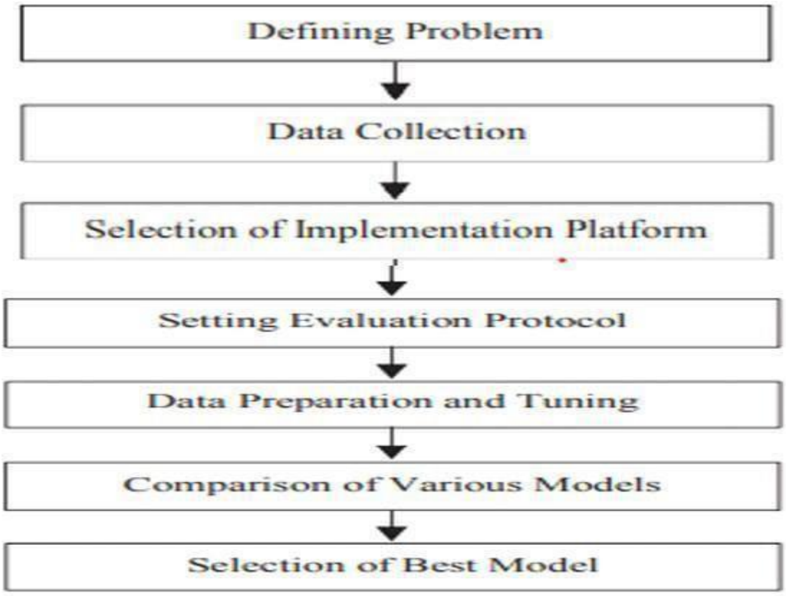


Fig4.2 Block diagram of the system

4.2.1 Advantages

- Improved Accuracy: Utilizing machine learning algorithms enhances the accuracy of blastocyst grading by reducing human error and subjectivity.
- Time Efficiency: Automated image processing speeds up the grading process, allowing for quicker assessments and potentially faster embryo selection.
- Consistency: The system provides consistent grading criteria, reducing variability betweendifferent embryologists and increasing reliability.
- Objective Evaluation: Machine learning removes subjective biases, ensuring fair and standardized evaluations of blastocyst quality.
- PotentialCost Savings: By reducing the need for manual labor and potentially improvingpregnancy success rates, the system may lead to cost savings in IVF procedures.
- Scalability: The system can be scaled to handle large volumes of data, accommodating theincreasing demand for IVF services worldwide.
- Research Opportunities: The collected data can be utilized for further research in understanding factors influencing blastocyst development and improving IVF outcomes.

Chapter 5

SYSTEM METHODOLOGY

5.1 SYSTEM MODEL

A system model is the conceptual model as a result of system modelling that describes and represents a system. A system comprises multiple views such as planning, requirement (analysis), design, implementation, deployment, structure, behaviour, input data, and output data views.

5.1.1 Use Case diagram

A use case diagram is a visual representation of the interactions between actors (users or external systems) and a system under consideration to achieve specific goals or tasks. It is commonly used in software engineering and system design to capture the functional requirements of a system from a user's perspective.

This use case diagram and information provide a visual representation of the interactions between actors and the system components involved in the image processing approach for grading IVF blastocysts using machine learning. It outlines the sequence of actions performed by the embryologist and the machine learning system, as well as the storage of grading results in the IVF clinic database for future reference.

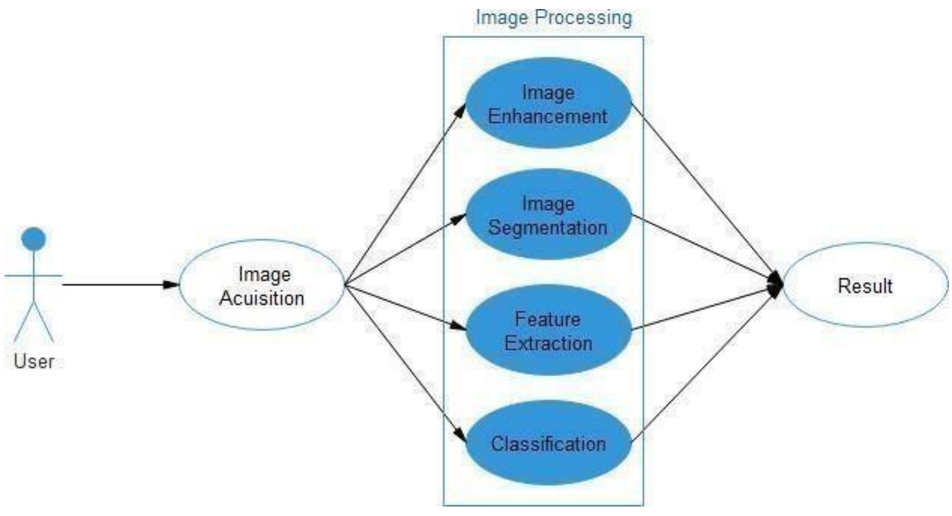
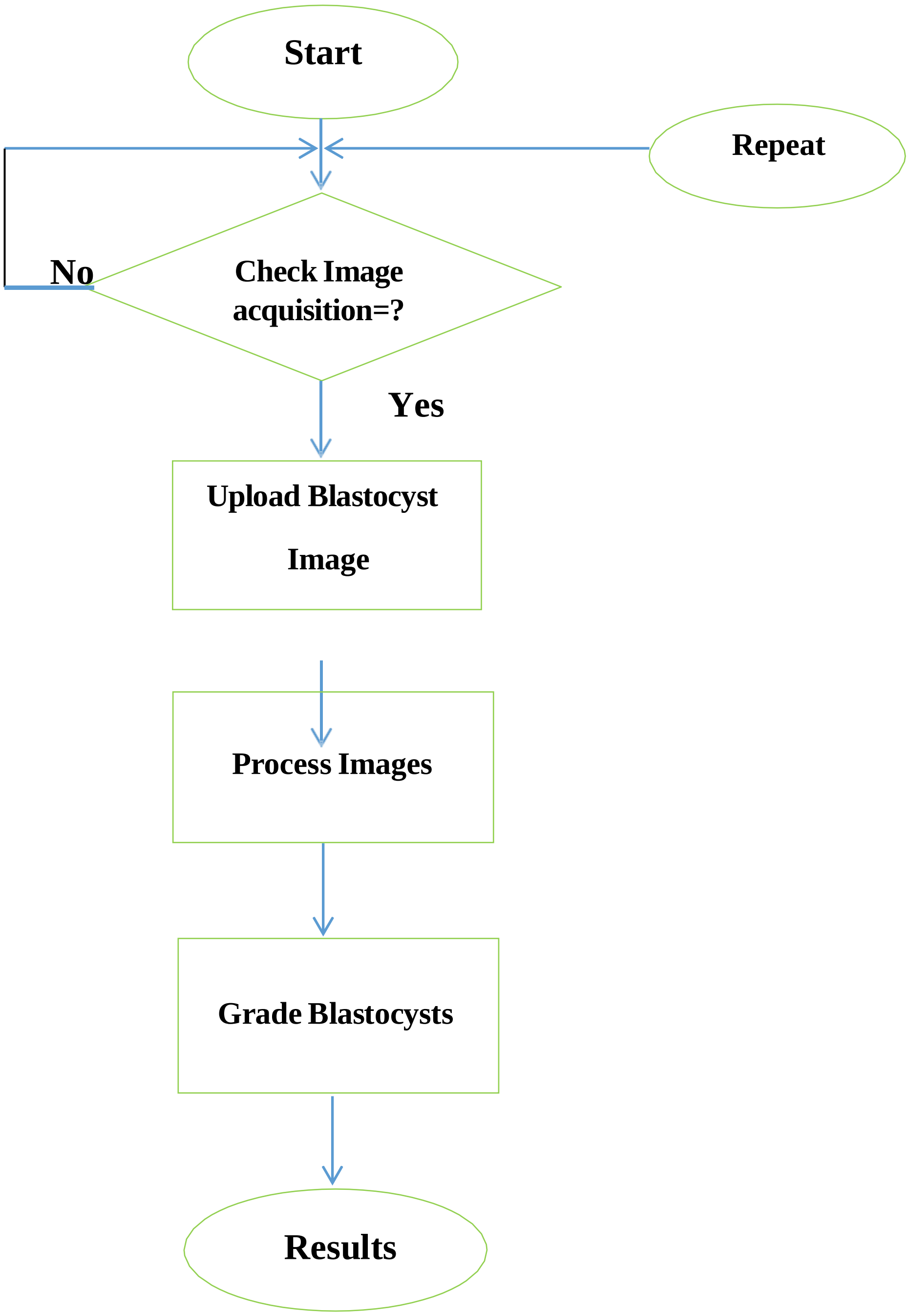


Fig 5.1 Image processing approach for grading IVF blastocyst

5.1.2 Flow Chart

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Fig 5.1.2 Flow chart of working



Description: Figure explains the flow chart of the system. Here continuously Each step in the flow chart represents a stage in the image processing approach for grading IVF blastocysts using machine learning. The flow chart provides a visual representation of the sequential steps involved in the process, from image acquisition to grading and review of results

5.1.3 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

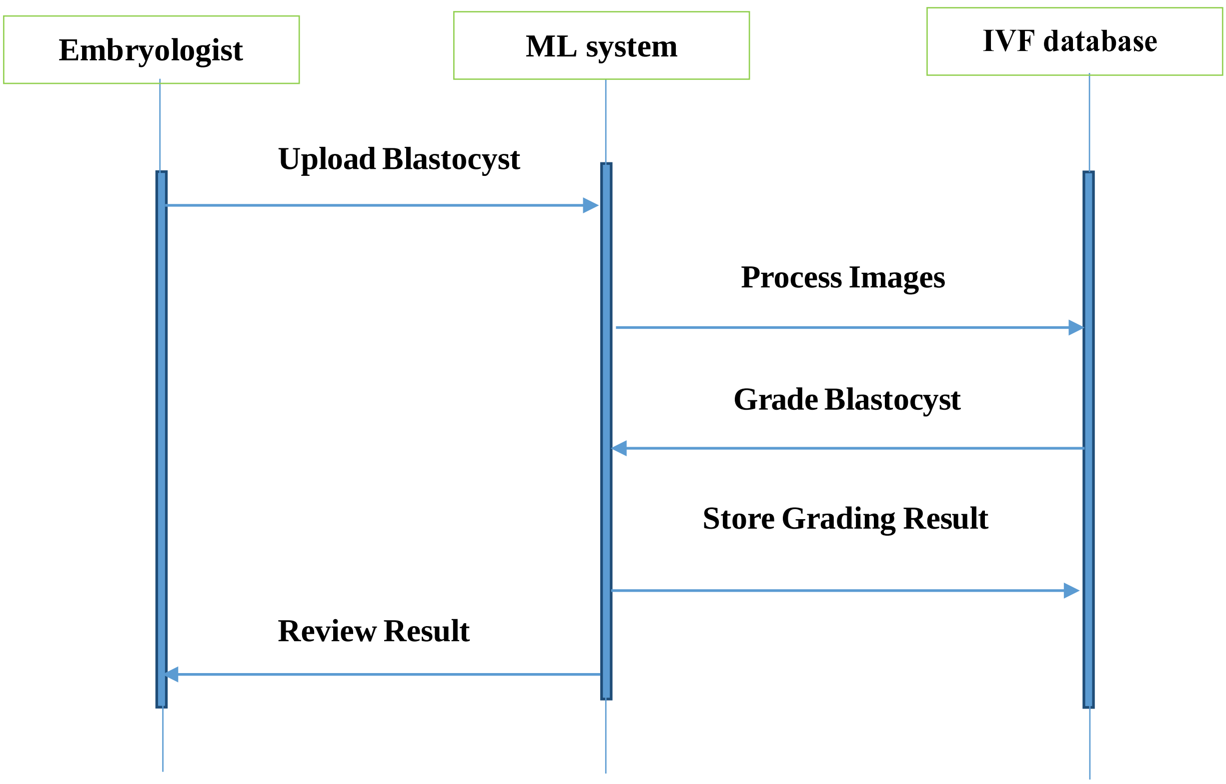


Fig5.1.3Sequence diagram consists of Embryologist,

5.1.4 Class Diagram

Class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions inthe application, and the classes to be programmed.

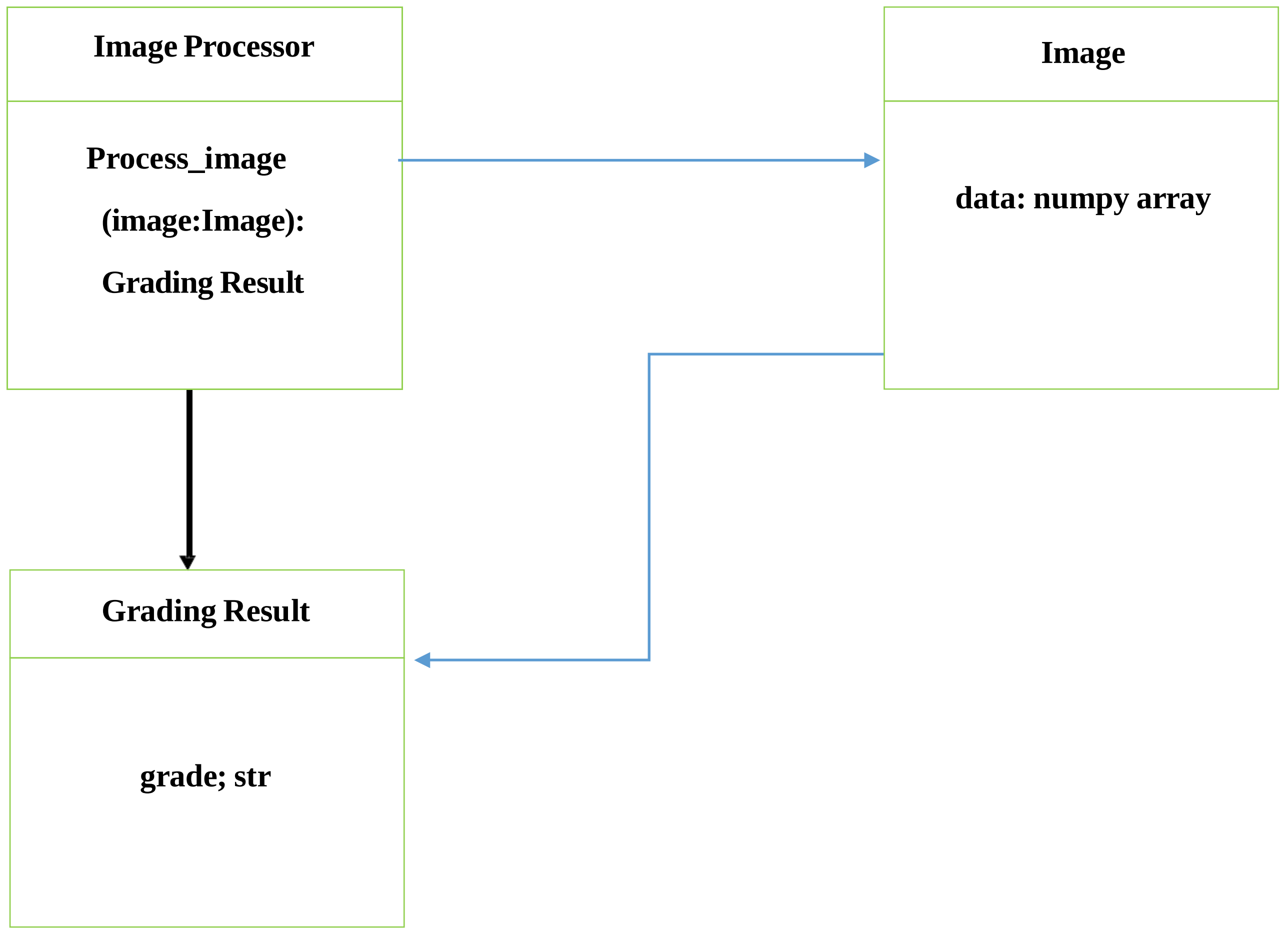


Fig5.1.4 class diagram for the system

5.2 system design

System design refers to modeling of a process. It is an approach to create a new system. It can be defined as a transition from user's view to programmer's view. The system design phase acts as a bridge between the required specification and implementation phase. In this chapter, we discuss problem definition, design considerations, the software process model (waterfall model) used and various dataflow diagrams used for our project.

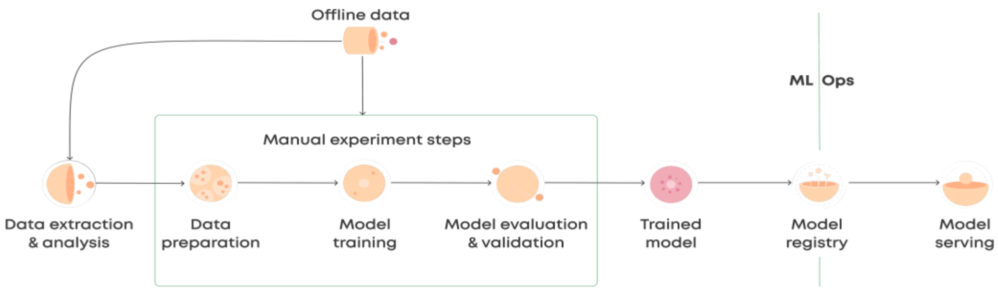


Fig 5.2 System Design

5.3 Software process model used

The software process model is an abstract representation of a software process. Each component of the model provides partial information about the process. Waterfall model approach is used in "Image processing approach for grading IVF blastocyst" application designed by us.

Each process model represents a process from a particular perspective and only provides partial information about that process. They are useful abstractions which can be used to explain different approaches to software development.

5.3.1 Methodology of the project

This model takes the fundamental process activities of specification, development, validation and evolution and represents them as separate process phases.

Requirement Analysis and Definition:

Requirement analysis and definition for an image processing approach for grading IVF blastocysts using machine learning involves identifying stakeholder needs, defining objectives, and understanding constraints. Stakeholders, including clinicians, embryologists, researchers, and patients, provide input through interviews, surveys, or meetings. The primary objective is to enhance accuracy, efficiency, and consistency in blastocyst grading, with secondary goals of reducing subjectivity, increasing automation, and supporting research endeavors. Constraints such as regulatory requirements, budget limitations, and technical constraints are identified and considered. Functional requirements encompass features for image acquisition, preprocessing, feature extraction, model training, grading, and result visualization, while non-functional requirements address performance, scalability, reliability, security, usability, and maintainability. Use cases are specified to outline common interactions with the system, and prototypes are developed for validation and feedback. Requirements are documented comprehensively with prioritization based on importance and feasibility. Stakeholder review and approval ensure alignment with expectations before proceeding to implementation.

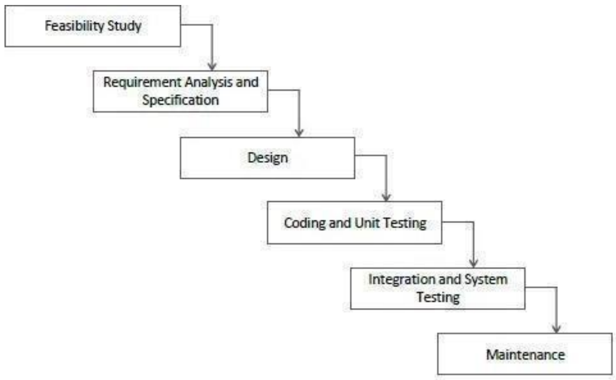


Fig 5.3.1.1 The Schematic Representation of Waterfall Model.

Description: The water fall Model includes requirement specification phase, system and software design, implementation phase, integration and system testing phase.

System and Software Design

The system design process partitions the requirements to either hardware or software systems. It establishes the overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.

The most optimum and appropriate design methods or algorithms have to be selected for design to provide efficiency. System design is prepared by referring to Software requirement specification. Based on this system design implementation is done. This phase very important to prepare zero defect software

Implementation and Unit Testing

During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification. The term "software" includes both computer programs and computer databases. Each unit is tested here for accuracy, relevance and errors. Errors or bugs if any are reported and removed. The term implementation means converting design into programs and computer database.

Integration and System Testing

The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. The aim is to produce a consistent, working software system. After testing, the software system is delivered to the customer. Each and every module is tested separately and verified.

Operation and Maintenance

Normally this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life-cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered.

5.4 System Study: Feasibility study

All frameworks are possible when given boundless asset and unending time. Be that as it may, sadly this condition does not win in reasonable world. So it is both fundamental and reasonable to assess the attainability of the framework at the most punctual conceivable time. Months or years of exertion, a great many rupees and untold expert shame can be deflected if a silly framework is perceived ahead of schedule in the definition stage. Attainability and danger examination are connected from numerous points of view. In the event that venture danger is awesome, the achievability of creating quality programming is lessened. For this situation there are three essential zones of interests.

Performance Feasibility

For the complete functionality of the project work, the project is run with the help of healthy networking environment. Normally, the OS is windows XP. The main theme of this project is to allocate path channels based on the hot spot and clod spot. Performance analysis is done to find out whether our algorithm is more efficient. It is essential that the process of performance analysis and definition must be conducted in parallel. It does not say what has to be done, but about suggests a range of very subtle possibilities for what could be done. A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture.

We measure the parameter called Packet delivery to measure the effectiveness of the approach.

5.4.1 Technical Feasibility

System is only beneficial only if it can be turned into information systems that will meet the organization's technical requirement. Simply stated this test of feasibility asks whether the system will work or not when developed installed, whether there are any major barriers to implementation. Regarding all these issues in technical analysis there are several points to focus on:-

Changes to bring in the system: All changes should be in positive direction, there will be increased level of efficiency and better customer service.

Required skills: Platforms & tools used in this project are widely used. So the skilled manpower is readily available in the industry.

Acceptability: The structure of the system is kept feasible enough so that there should not be any problem from the user's point of view.

Economic Feasibility

Economical analysis is performed to evaluate the development cost weighed against the ultimate income or benefits derived from the developed system. For running this system, we need not have high performance servers. All the functions of implemented through software modules. In this system we are not using any physical devices for connection. So the system is economical feasible enough.

5.5 System Architecture

Framework engineering is the calculated outline that characterizes the structure and conduct of a framework. It characterizes the framework segments or building pieces and gives an arrangement from which items can be obtained and frameworks built up, that will cooperate to execute the general framework.

The system architecture of an image processing approach for grading IVF blastocysts using machine learning consists of several interconnected components designed to efficiently process and evaluate blastocyst images. The architecture typically includes an image acquisition module responsible for capturing high-quality images of IVF blastocysts. These images are then passed through a preprocessing module, which enhances image quality and standardizes characteristics. Subsequently, a feature extraction module identifies relevant features from the preprocessed images, such as size, shape, and texture. These features serve as input to a machine learning model, trained to classify blastocysts into different grades based on the extracted features. The trained model is integrated into a grading algorithm, which applies the learned classification criteria to new blastocyst images, resulting in graded classifications. Additionally, the system may incorporate a user interface module to facilitate interaction with clinicians or researchers, enabling them to input images, view graded classifications, and access additional information. A database module manages image data, labeled features, graded classifications, and other relevant information, ensuring efficient storage, retrieval, and management of data for ongoing analysis and research. The entire architecture is designed to be scalable, reliable, and maintainable, supporting the accurate and objective grading of IVF blastocysts while accommodating future growth and advancements

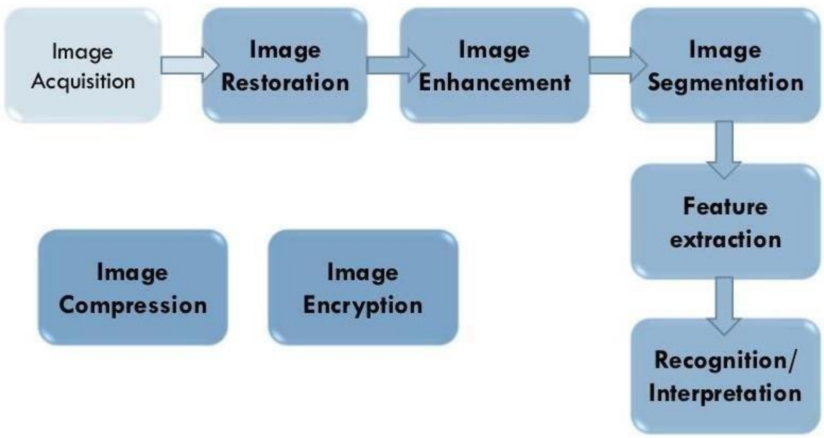


Fig 5.5 System Architecture

Description: Above figure shows system architecture of an image processing approach to grading IVF blastocysts using machine learning involves several interconnected components designed to efficiently process and evaluate blastocyst images.

5.5 Proposed system Architecture

Proposed system will apply new ideas or improvement on the existing system is called as Proposed system.

1. Data Collection and Preparation:

   Acquire a dataset of high-resolution images of IVF blastocysts from clinical sources or research institutions.

   Annotate the images with corresponding grading labels provided by embryologists or fertility specialists.

2. Feature Extraction:

   Extract relevant features from the preprocessed blastocyst images that are indicative of their quality and developmental stage.

   Explore feature extraction methods such as morphological analysis, texture analysis, and edge detection to capture distinguishing characteristics of blastocysts.

3. Machine Learning Model Selection:

   Choose appropriate machine learning algorithms for blastocyst grading, considering factors such as dataset size, complexity of features, and computational resources.

   Experiment with different models, including convolutional neural networks (CNNs), support vector machines (SVMs), decision trees, or ensemble methods, to determine the most suitable approach.

4. Model Training and Evaluation:

   Validate the trained model on the validation set to fine-tune hyperparameters and prevent overfitting.

   Evaluate the final model's performance on the test set using metrics such as accuracy, precision, recall, and F1-score.

5. Integration and Deployment:

   Integrate the grading system into existing IVF clinic workflows or research pipelines, ensuring compatibility with clinical protocols and data management systems.

   Deploy the system on suitable hardware infrastructure, considering factors like computational resources, scalability, and accessibility for end users.

6. Validation and Clinical Testing:

Validate the performance of the grading system through rigorous testing and validation procedures, including comparison with manual grading by expert embryologists.

Conduct clinical trials or case studies to assess the system's effectiveness in real-world settings and its impact on clinical decision-making and patient outcomes.

7. Continuous Improvement and Maintenance:

Monitor the performance of the grading system over time and collect feedback from end users to identify areas for improvement.

Incorporate new data and retrain the machine learning model periodically to adapt to changes in blastocyst characteristics or grading standards.

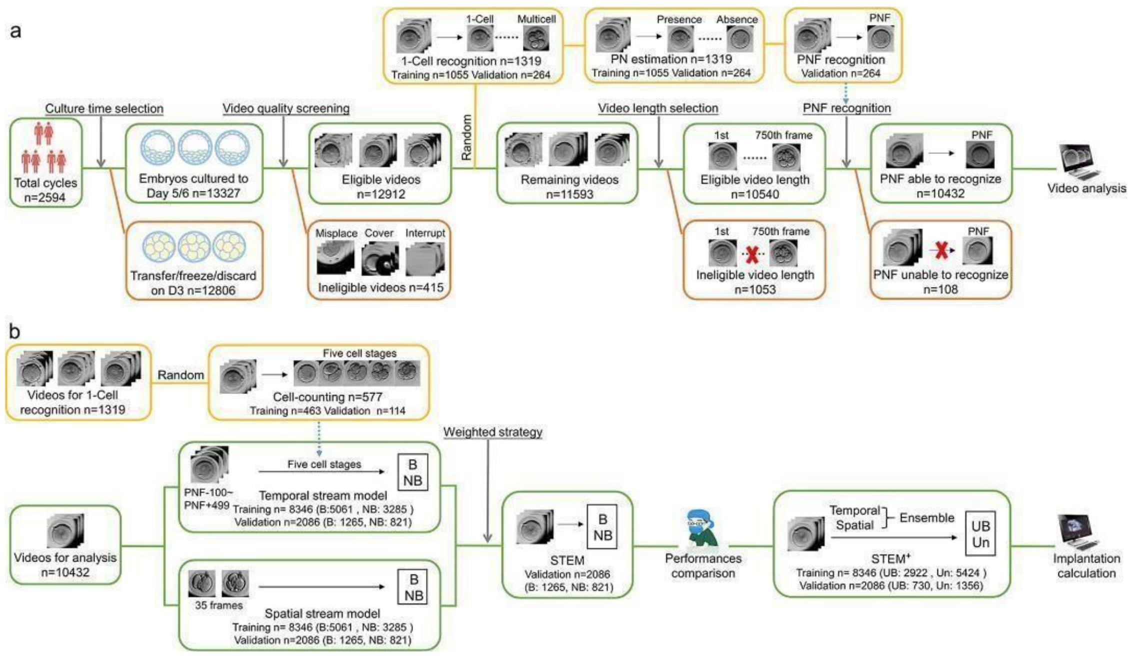


Fig: 5.6 Proposed system Architecture

Description: The block diagram of blastocyst architecture for development of time lapsing image processing

Chapter 6

IMPLEMENTION

The process of putting a decision or plan into effect; execution called as Implementation. This chapter describes the implementation details of this project. This is the code generation phase. The examination and re-examination of the requirements statement is needed to ensure that it is being followed to the letter. The various implementation requirements are elaborated. The operating system choice, the programming choices and details of coding are also explained.

6.1 Implementation Requirements

1. Ovarian Stimulation: The woman undergoes hormonal therapy to stimulate her ovaries to produce multiple eggs, rather than the single egg produced during a natural menstrual cycle.
2. Monitoring: During ovarian stimulation, the woman's response is monitored through blood tests and ultrasound to determine the optimal time for egg retrieval.
3. Egg Retrieval: Once the eggs are mature, a minor surgical procedure called transvaginal ultrasound aspiration is performed to retrieve the eggs from the woman's ovaries. This procedure is usually done under sedation.
4. Fertilization: The retrieved eggs are then fertilized in a laboratory setting. In traditional IVF, this involves mixing the eggs with sperm collected from the male partner. In cases where there are fertility issues, Intracytoplasmic Sperm Injection (ICSI) may be used, where a single sperm is injected directly into each egg.
5. Embryo Culture: The fertilized eggs, now embryos, are cultured in a special incubator for several days to allow them to develop.
6. Embryo Transfer: Typically, one or more embryos are selected and transferred into the woman's uterus. This is done using a thin catheter inserted through the cervix. The number of embryos transferred depends on various factors, including the woman's age and health, as well as any previous IVF attempts.
7. Luteal Phase Support: After embryo transfer, the woman may be given medications such as progesterone to support the uterine lining and improve the chances of successful implantation.
8. Pregnancy Test: About two weeks after embryo transfer, a pregnancy test is done.

6.2 Operating System Requirement

Python is highly portable and can run on various operating systems, including:

Windows: Python is fully supported on Windows operating systems, including Windows 7, 8, and 10. You can install Python from the official Python website or via package managers like Anaconda or Miniconda.

macOS: Python comes pre-installed on macOS, but you can also install alternative versions or manage Python environments using tools like Homebrew, pyenv, or Anaconda.

Linux: Python is well-supported on Linux distributions, including Ubuntu, Debian, CentOS, Fedora, and others. Most Linux distributions come with Python pre-installed, but you can also install it using package managers or from source.

Regardless of the operating system, Python provides a consistent programming environment, allowing you to write code that is portable across platforms. This means that Python code written on one operating system can typically be run on another without modification.

For machine learning tasks in Python, such as data preprocessing, model training, and evaluation, the choice of operating system is largely a matter of personal preference and compatibility with other software dependencies. Many popular machine learning libraries and frameworks, such as scikit-learn, TensorFlow, and PyTorch, are cross-platform and work seamlessly on Windows, macOS, and Linux.

However, it's worth noting that some machine learning tasks may benefit from specific hardware configurations or optimizations that are more readily available on certain operating systems. For example, deep learning tasks that require GPU acceleration may have better support on Linux systems due to more robust drivers and tooling.

In summary.

Python's versatility and cross-platform compatibility make it suitable for machine learning on a wide range of operating systems, allowing developers to choose the environment that best suits their needs and preferences.

6.3 Hardware Selection

Selecting hardware for machine learning tasks, especially when dealing with computationally intensive operations like training deep learning models, is crucial. Here's a simplified approach using Python:

Identify Requirements: Determine the hardware requirements based on the size of your dataset, complexity of your model, and desired training time.

support GPU acceleration. Check if your model can benefit from GPU processing. **Choose GPU:** If GPU acceleration is necessary, select a GPU based on your budget and performance requirements. NVIDIA GPUs are commonly used for deep learning tasks.

Consider CPU: While GPUs are essential for training deep learning models, CPUs still play a role, especially for data preprocessing and post-processing tasks.

Memory (RAM): Ensure you have enough RAM to handle your dataset and model size. Deep learning models can be memory-intensive, especially when working with large datasets.

Storage: Consider the storage requirements for your dataset and model checkpoints. SSDs are preferred over HDDs for faster data access.

6.4 Tools Selection and Software Selection

In machine learning, there are numerous tools and libraries available that facilitate various aspects of the machine learning workflow, from data preprocessing to model deployment. Here are some of the most popular ones:

Python: Python is the most widely used programming language for machine learning due to its simplicity, versatility, and the availability of numerous libraries for data manipulation, visualization, and modeling.

NumPy: NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

Pandas: Pandas is a powerful library for data manipulation and analysis. It provides data structures like Data Frames and Series, which make it easy to handle structured data and perform tasks such as data cleaning, exploration, and transformation.

scikit-learn: scikit-learn is a comprehensive library for machine learning in Python. It provides simple and efficient tools for data mining and data analysis, including algorithms for classification, regression, clustering, dimensionality reduction, and model evaluation.

TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It's primarily used for building and training deep learning models, but it also supports traditional machine learning algorithms. TensorFlow provides high-level APIs like tf.keras for easy model building and training.

PyTorch: PyTorch is another popular open-source machine learning framework developed by Facebook. It's known for its dynamic computation graph, which enables more flexibility in model building and debugging compared to TensorFlow. PyTorch is widely used in both research and production for deep learning tasks.

Here The Code for Implementation

```
import KNeighborsClassifier
 start = time.time()
param_grid = {'n_neighbors':[3,5,7]}
clf = GridSearchCV(KNeighborsClassifier(), param_grid,cv=3,scoring='accuracy')
clf.fit(X,Y)
print (clf.best_params_)
print (time.time() - start)
f1_temp9 = []
acc_temp9 = []
prec_temp9 = []
rec_temp9 = []
rocauc_temp9 = []
prauc_temp9 = []
gmean_temp9 = []
method_temp9 = []
original method :
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X,Y)
pred_9 = model.predict(X_val_9)
# y_score_9 = best_clf.decision_function(X_val_9)
y_score_9 = model.predict_proba(X_val_9)[:,1]
result_f1_9 = f1_score(Y_val_9, pred_9)
result_acc_9 = accuracy_score(Y_val_9, pred_9)
result_conf_9 = confusion_matrix(Y_val_9, pred_9)
result_prec_9 = precision_score(Y_val_9,pred_9)
```

```
result_recall_9 = recall_score(Y_val_9,pred_9)
result_gmean_9 = gmean(result_conf_9)
result_prauc_9 = average_precision_score(Y_val_9, y_score_9)
result_rocauc_9 = roc_auc_score(Y_val_9,y_score_9)
fpr_91, tpr_91, thresholds_91 = roc_curve(Y_val_9, y_score_9)
precision_91, recall_91, thresholds_91 = precision_recall_curve(Y_val_9,y_score_9)


print("Confusion Matrix: %s" % (result_conf_9))
f1_temp9.append(result_f1_9)
acc_temp9.append(result_acc_9)
prec_temp9.append(result_prec_9)
rec_temp9.append(result_recall_9)
gmean_temp9.append(result_gmean_9)
rocauc_temp9.append(result_rocauc_9)
prauc_temp9.append(result_prauc_9)
method_temp9.append("KNN k=5")
print (len(f1_temp9), len(acc_temp9), len(prec_temp9), len(rec_temp9), len(rocauc_temp9), len(prauc_temp9), len(gmean_temp9), len(method_temp9))
total_list = zip(method_temp9,rocauc_temp9, prauc_temp9, f1_temp9, acc_temp9, gmean_temp9, prec_temp9, rec_temp9)
df_res=pd.DataFrame(total_list,columns=['Method','ROCAUC','PRAUC','F1_Score','Accuracy', 'Gme an','Precision','Recall'])
df_res.index = df_res.Method.values
df_res.drop('Method',axis=1,inplace=True)
df_res
f1_temp9 = []
acc_temp9 = []
```

```
prec_temp9 = []
rec_temp9 = []
rocauc_temp9 = []
prauc_temp9 = []
gmean_temp9 = []
method_temp9 = []
print (len(f1_temp9), len(acc_temp9), len(prec_temp9), len(rec_temp9), len(rocauc_temp9),
len(prauc_temp9), len(gmean_temp9), len(method_temp9))
total_list = zip(method_temp9,rocauc_temp9, prauc_temp9, f1_temp9, acc_temp9,
gmean_temp9, prec_temp9, rec_temp9)
df_res=pd.DataFrame(total_list,columns=['Method','ROCAUC','PRAUC','F1_Score','Accuracy',
'Gme an','Precision','Recall'])
df_res.index = df_res.Method.values
df_res.drop('Method',axis=1,inplace=True)
df_res
# Plot the results
%matplotlib inline
from IPython.core.pylabtools import figsize
import matplotlib.pyplot as plt

figsize(10, 5)
ax = plt.subplot(111)

ind = np.arange(df_res.shape[0])
width = 0.2
l = ax.plot(ind, df_res, "-o")
leg = plt.legend(iter(l), df_res.columns.tolist(), loc='center right',fancybox=True)
#leg.get_frame().set_alpha(0.05)
```

```
plt.ylabel('Score')
plt.xlabel('Method')
ax.set_xlim([-0.25, ind[-1]+0.5])
ax.set_xticks(ind)
ax.set_xticklabels(df_res.index)
plt.show()
import numpy as np
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import precision_score, recall_score, f1_score,
accuracy_score, confusion_matrix, average_precision_score, roc_auc_score, roc_curve,
precision_recall_curve
from imblearn.over_sampling import SMOTE
from imblearn.metrics import geometric_mean_score
import time
import matplotlib.pyplot as plt
%matplotlib inline


from google.colab import drive
drive.mount('/content/drive')
# Read CSV file
df_9 = pd.read_csv("/content/drive/MyDrive/Projects/ivf/data.csv")
df_10 = pd.read_csv("/content/drive/MyDrive/Projects/ivf/data.csv")

# Print shape of dataframes
print(df_9.shape, df_10.shape)
# Extract features and target variables
df_10 = np.array(df_10)
Y = df_10[:, 61]
X = df_10[:, :61]
```

```
df_9 = np.array(df_9)
X_val_9 = df_9[:, :61]
Y_val_9 = df_9[:, 61]

print(X.shape, Y.shape, X_val_9.shape, Y_val_9.shape)

# Define geometric mean function
def gmean(cf):
try:
tpr = float(cf[1][1]) / (cf[1][1] + cf[1][0])
tnr = float(cf[0][0]) / (cf[0][0] + cf[0][1])
return np.sqrt(tpr * tnr)
except:
return "DivisionByZero"

# SMOTE tuning parameters
kind_list = ['regular', 'borderline1', 'borderline2', 'svm']
k_neighbors_list = [3, 5, 9]

# Initialize lists for evaluation metrics
f1_temp9 = []
acc_temp9 = []
prec_temp9 = []
rec_temp9 = []
rocauc_temp9 = []
prauc_temp9 = []
gmean_temp9 = []
method_temp9 = []

# GridSearchCV for KNeighborsClassifier
start = time.time()
param_grid = {'n_neighbors': [3, 5, 7]}
clf = GridSearchCV(KNeighborsClassifier(), param_grid, cv=3, scoring='accuracy')
```

```
clf.fit(X, Y)
print(clf.best_params_)
print(time.time() - start)

# Initialize lists for evaluation metrics
f1_temp9 = []
acc_temp9 = []
prec_temp9 = []
rec_temp9 = []
rocauc_temp9 = []
prauc_temp9 = []
gmean_temp9 = []
method_temp9 = []

# Original method: KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=5)
model.fit(X, Y)
pred_9 = model.predict(X_val_9)
y_score_9 = model.predict_proba(X_val_9)[:, 1]

result_f1_9 = f1_score(Y_val_9, pred_9)
result_acc_9 = accuracy_score(Y_val_9, pred_9)
result_conf_9 = confusion_matrix(Y_val_9, pred_9)
result_prec_9 = precision_score(Y_val_9, pred_9)
result_recall_9 = recall_score(Y_val_9, pred_9)
result_gmean_9 = gmean(result_conf_9)
result_prauc_9 = average_precision_score(Y_val_9, y_score_9)
result_rocauc_9 = roc_auc_score(Y_val_9, y_score_9)
fpr_91, tpr_91, thresholds_91 = roc_curve(Y_val_9, y_score_9)
precision_91, recall_91, thresholds_91 = precision_recall_curve(Y_val_9, y_score_9)

print("Confusion Matrix:", result_conf_9)

# Append results to respective lists
```

```
f1_temp9.append(result_f1_9)
acc_temp9.append(result_acc_9)
prec_temp9.append(result_prec_9)
rec_temp9.append(result_recall_9)
rocauc_temp9.append(result_rocauc_9)
prauc_temp9.append(result_prauc_9)
gmean_temp9.append(result_gmean_9)
method_temp9.append("KNeighborsClassifier")

print(len(f1_temp9), len(acc_temp9), len(prec_temp9), len(rec_temp9),
                     len(rocauc_temp9), len(prauc_temp9), len(gmean_temp9),
len(method_temp9))
# Plot the results
total_list = zip(method_temp9, rocauc_temp9, prauc_temp9, f1_temp9, acc_temp9,
 gmean_temp9, prec_temp9, rec_temp9)
df_res = pd.DataFrame(total_list, columns=['Method', 'ROCAUC', 'PRAUC', 'F1_Score',
 'Accuracy', 'Gmean', 'Precision', 'Recall'])
df_res.index = df_res.Method.values
df_res.drop('Method', axis=1, inplace=True)



ax = plt.subplot(111)

ind = np.arange(df_res.shape[0])
width = 0.2
l = ax.plot(ind, df_res, "-o")
plt.legend(iter(l), df_res.columns.tolist(), loc='center right')
plt.ylabel('Score')
plt.xlabel('Method')
ax.set_xlim([-0.25, ind[-
1]+.5]) ax.set_xticks(ind)
ax.set_xticklabels(df_res.inde
x) plt.show()
```

Chapter 7

TESTING

Before the IVF procedure, both partners typically undergo thorough medical evaluations. This includes assessing the woman's ovarian reserve, hormone levels, andthe man's semen analysis. Genetic screening may also be done to detect any potential genetic disorders.

During IVF, the woman undergoes ovarian stimulation to produce multiple eggs. Monitoring this process involves regular ultrasounds and blood tests to track hormone levels and ensure the ovaries are responding appropriately to the medication.

Once the eggs are mature, a minor surgical procedure is performed to retrieve them from the woman's ovaries. The eggs are then placed in a culture medium and examined under a microscope

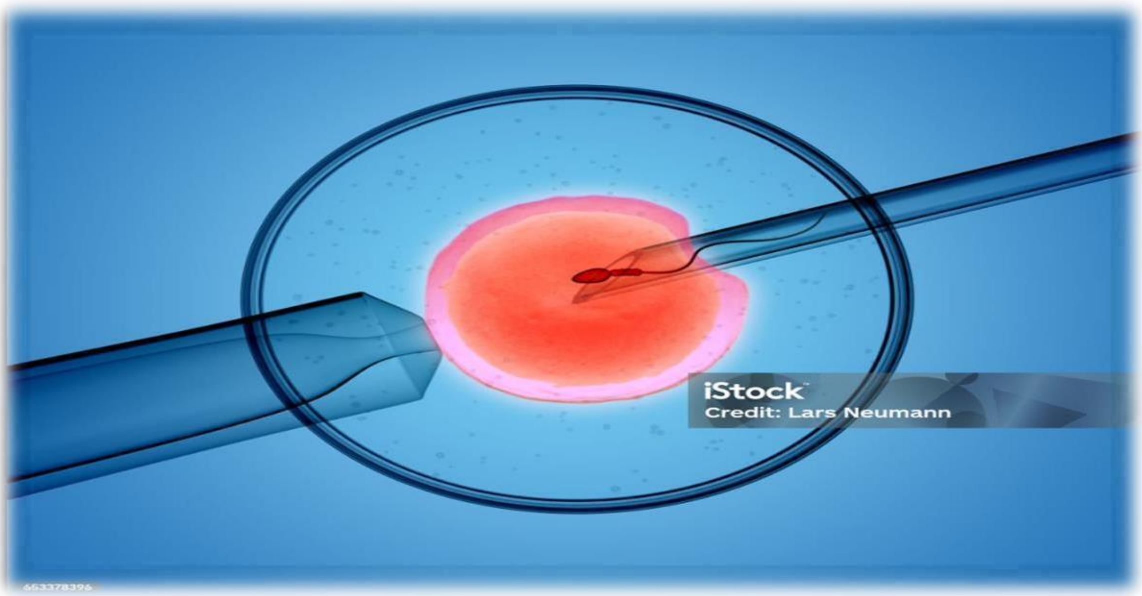


Fig 7.1 Testing blastocyst in laboratory

7.1.1 Unit Testing

Test that input data is properly preprocessed before being fed into the KNN algorithm. This includes handling missing values, normalizing or scaling features, and encoding categorical variables.

Test the calculation of distances between data points. Ensure that the algorithm correctly computes distances using appropriate metrics (e.g., Euclidean distance) and handles different data types (numeric, categorical) appropriately.

- Test the algorithm's performance on imbalanced datasets, where one class may dominate the others. Ensure that the algorithm doesn't bias predictions towards the majority class and handles imbalanced data appropriately.

Table 7.1.1 Unit test procedure for check image

7.1.2 Integration Testing

The IVF system properly collects and preprocesses patient data before passing it to the KNN algorithm. Ensure that data is formatted correctly and that all required fields are present.

Test the output of the KNN algorithm to ensure that it provides predictions in the expected format and that the predictions are meaningful and interpretable within the context of IVF treatment.
Test the integration between the IVF system and the database where patient data is stored. Verify that the IVF system can retrieve patient data from the database and pass it to the KNN algorithm for prediction.

Ensure that the IVF system can update the database with the predictions generated by the KNN algorithm, along with any other relevant information.

Verify that the IVF system's user interface allows users (e.g., clinicians, embryologists) to input patient data and view the predictions generated by the KNN algorithm.

Test the usability and functionality of the user interface, ensuring that it is intuitive, responsive, and provides relevant feedback to users.

Test the IVF system's ability to handle errors gracefully, both in terms of data input errors (e.g., missing or invalid data) and algorithmic errors (e.g., failure to generate predictions).

Verify that the IVF system provides informative error messages and logs errors appropriately for Test the IVF system's performance and scalability when dealing with large volumes of patient data.

Verify that the system remains responsive and efficient even as the size of the dataset grows.

Verify that the IVF system implements appropriate security measures to protect patient data and prevent unauthorized access or tampering.

Test the system for vulnerabilities such as SQL injection, cross-site scripting, and unauthorized data access.

Ensure that the IVF system complies with relevant regulatory standards and guidelines for medical software, such as HIPAA (Health Insurance Portability and Accountability Act) in the United States or GDPR (General Data Protection Regulation) in the European Union.

Test the system's ability to maintain patient confidentiality, secure sensitive data, and provide audit trails for compliance purposes.

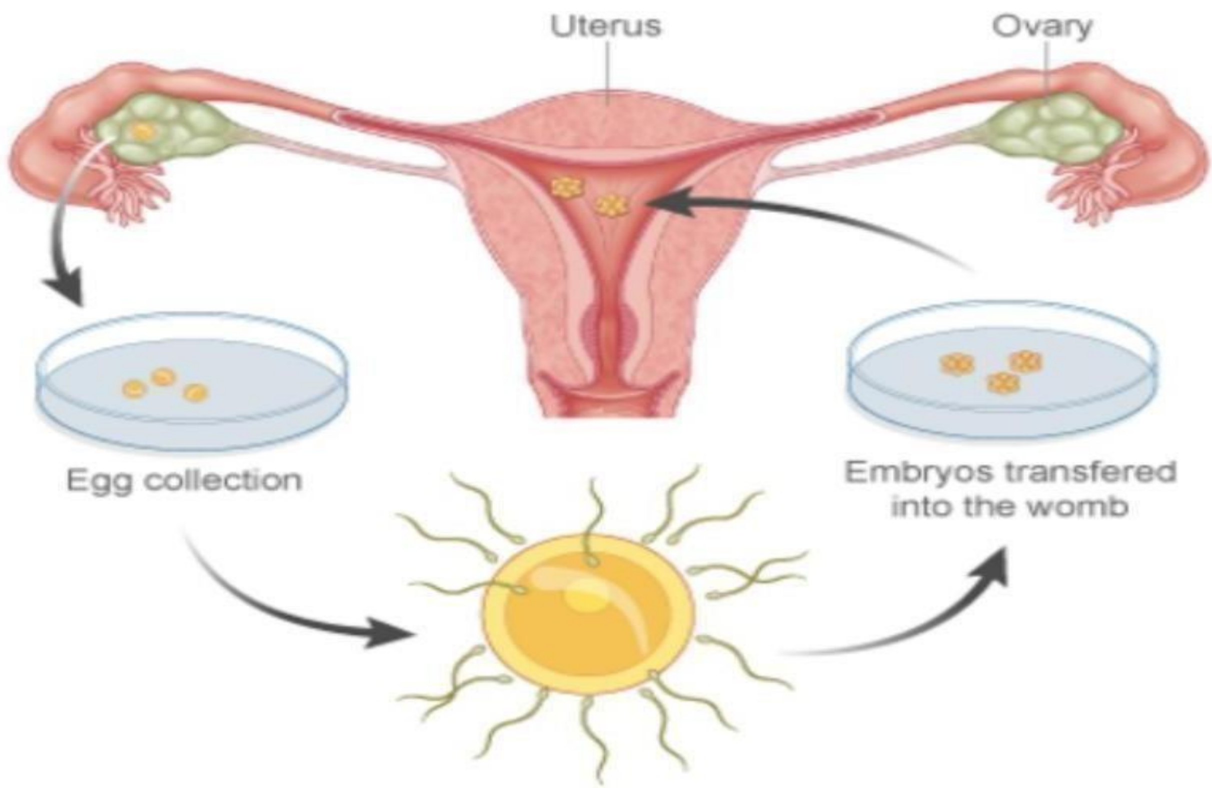


Fig7.1.2 integrated test procedure for check image