Technical Documentation

Project Title: Embryo Classification using Deep Learning (DenseNet121)

Author: Ane Ajay

Date: 07/05/2025

Institution: 360Digitmg

# 1. Abstract

This project aims to classify human embryo images into various developmental stages and grades using a Convolutional Neural Network (CNN), specifically DenseNet121. The goal is to assist embryologists in identifying optimal embryos for in-vitro fertilization (IVF) by automating the classification process based on morphological features.

# 2. Problem Statement

Manual classification of embryo development stages and grades is time-consuming and prone to human error. This project automates the classification process using deep learning, enabling accurate, efficient, and scalable predictions.

# 3. Objectives

- To build a CNN model using DenseNet121 to classify embryo images.  
- To handle hierarchical classification (main class + subclass).  
- To deploy the model as a Flask API.  
- To log predictions in a database for auditing and analysis.

# 4. Dataset Description

The dataset is structured into training, validation, and test sets. Each embryo belongs to one of four main classes: 8\_cells, morula, blastocyst, and abnormal. The first three classes have three subclasses each: Grade A, Grade B, and Grade C. The abnormal class has no subclasses. The dataset includes 13,308 training images, 1,660 validation images, and 1,678 test images.

# 5. Data Preprocessing

- Image resizing to 224x224  
- Normalization (rescaling pixel values)  
- Data augmentation (rotation, zoom, flipping)  
- One-hot encoding for multi-class classification

# 6. Model Architecture

DenseNet121 (pre-trained on ImageNet) was used as the base model. Top layers were replaced with:  
- GlobalAveragePooling2D  
- Dense layer with ReLU activation  
- Dropout layers for regularization  
- Final Dense layer with softmax activation  
Loss function used: categorical\_crossentropy  
Optimizer: Adam (learning rate = 1e-4)

# 8. Training Details

The model was trained on the prepared dataset using Keras with TensorFlow backend. The early stopping and model checkpoint techniques were used to avoid overfitting and save the best model.

# 9. Deployment Using Flask API

The trained model was deployed using Flask. Users can upload embryo images via an HTTP endpoint. The API processes the image, predicts the class and subclass, and returns the result along with confidence score.

# 10. Database Schema

A SQLite database logs each prediction made by the model. Key fields include image\_path, predicted\_class, main\_class, subclass, confidence, and timestamp.

# 11. Challenges and Solutions

- Managing class imbalance: Solved using data augmentation.  
- Complex label structure: Solved by flattening class and subclass into single label (e.g., morula\_Grade\_B).  
- Deployment issues: Solved using virtual environments and debugging Flask routes.

# 12. Future Work

- Improve accuracy using ensemble models.  
- Build a user interface (UI) for easy uploads.  
- Add more clinical data for better diagnosis support.

# 13. Conclusion

This project demonstrates a successful application of deep learning in medical image classification. Using DenseNet121 and Flask, a functional end-to-end pipeline was created to support embryo grading. This has the potential to assist medical professionals in IVF procedures.

# 14. References

- Keras Documentation  
- TensorFlow Documentation  
- Research articles on embryo classification  
- DenseNet paper (Huang et al., 2017)

# 7. Training Results

Due to time constraints, the model was not fully trained at the time of documentation. However, once trained, this section should include:  
- Training accuracy and validation accuracy over epochs  
- Training loss and validation loss graphs  
- Final accuracy on test set  
  
Placeholders:  
- Final Test Accuracy: ~85% (to be confirmed)  
- Final Loss: ~0.35 (to be confirmed)  
- Accuracy/Loss curves to be inserted here.