An Optimal Approach in Selecting Embryo for In-Vitro Fertilization (IVF) Based on Deep Learning

Bita Nasiri (MSc)[1], Nacer Farajzadeh (PhD)[2]*, Jalil Ghavidel Neycharan (PhD)[2]

[1]Artificial Intelligence and Machine Learning Research Laboratory, Azarbaijan Shahid Madani University, Tabriz, Iran
[2]Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran

ABSTRACT

Background: About 14% of couples experience infertility, and In Vitro Fertilization (IVF) has become one of the most widely used treatment options. However, the overall success rate of IVF remains relatively low, at around 30%. At present, viable embryos are typically selected on the fifth day based primarily on morphological assessment, a method that is both subjective and limited in accuracy. Considering the substantial financial, physical, and emotional costs associated with failed IVF attempts, there is a pressing need for more reliable and effective embryo selection techniques.

Objective: This study aimed to increase the accuracy of embryo selection in IVF based on a deep learning-based transfer with the GoogLeNet architecture.

Material and Methods: In this experimental study, a retrospective dataset of embryo images was used to develop and evaluate a deep learning-based classification model in the following main phases: data preprocessing, model implementation, and evaluation. Embryo images were standardized through cropping and normalization to ensure consistency across different imaging systems. The GoogLeNet architecture, pre-trained on the ImageNet dataset, was utilized and further modified to adapt to the specific task of embryo viability classification.

Results: Evaluation on the test dataset demonstrated that the proposed model achieved strong predictive performance, with accuracy, precision, recall, and F1-score all reaching 97%. This performance surpasses that of existing baseline techniques, highlighting the model's effectiveness.

Conclusion: The proposed transfer learning-based approach using GoogLeNet shows significant potential for improving embryo selection in IVF, thereby reducing the emotional and financial strain associated with repeated IVF failures.

Keywords

Embryo; Blastocyst; In Vitro Fertilization; Image Processing; Deep Learning; Machine Learning

Introduction

Infertility affects nearly 14% of couples worldwide, making In Vitro Fertilization (IVF) a critical assisted reproductive technology, particularly among individuals of advanced reproductive age compared with younger couples [1-3]. Despite incremental IVF use, its success rates have been relatively low at approximately 30%, often resulting in considerable emotional and financial burdens on patients [1]. A key determinant of IVF success is accurate embryo selection for transfer. Traditional embryo evaluation at the blastocyst stage relies on morphological

*Corresponding author:
Nacer Farajzadeh
Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran
E-mail: n.farajzadeh@azaruniv.ac.ir

Received: 2 October 2024
Accepted: 23 November 2025

Copyright: © Journal of Biomedical Physics and Engineering
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 Unported License, (http://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited non-commercially.

assessment by embryologists, which is inherently subjective, labor-intensive, and prone to human error [1, 4]. Embryos progress through multiple developmental stages, including morula and blastocyst, with the blastocyst comprising distinct structures, such as the Inner Cell Mass (ICM) and Trophectoderm (TE) [5]. The embryo transfer time significantly affects implantation potential, as early transfer may reduce success, while late transfer can lead to premature hatching [5]. Furthermore, repeated removal of embryos from incubators for manual observation can negatively affect their development [2, 6].

Recent advances in Artificial Intelligence (AI) and deep learning have enabled objective, reproducible, and automated methods for embryo evaluation, thereby reducing the variability inherent in manual assessment. AI-driven systems can extract and quantify morphological and kinetic features from embryo images or time-lapse videos, improving the prediction of embryo viability [7-10]. A wide range of studies have applied supervised learning techniques, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), to classify embryos and predict IVF outcomes.

For example, Bormann et al. [1] developed a pre-trained Deep Convolutional Neural Network (DCNN) based on the ImageNet dataset to classify embryos as blastocysts or non-blastocysts, achieving 91% accuracy using a genetic algorithm. Their training dataset included 2,440 annotated embryo images captured 113 hours post-insemination. Khosravi et al. [4] introduced the STORK framework (they used 50,000 steps for training the DNN and subsequently evaluated the performance of their DNN (called STORK)) using 10,148 time-lapse embryo images from the Weill Cornell Medical Center. By combining Google's Inception model with decision trees, they predicted blastocyst quality and incorporated maternal age as a predictive factor. Similarly, Liao et al. [6] employed DenseNet201 together with a Spatial–Temporal Ensemble Model (STEM) to predict blastocyst formation from time-lapse videos, and in a separate analysis, to assess embryo usability.

Kaufmann et al. [11] utilized a convolutional neural network to select embryos for transfer, considering factors like age, number of eggs recovered, and number of transferred embryos. Despite multiple training attempts, their method achieved a maximum accuracy of 58.8%. Thirumalaraju et al. [12] utilized a video dataset of embryos to classify them based on morphological quality using deep convolutional neural networks. In this article, the embryo grading system is based on the Gardner classification system [7]. They employed architectures such as Inception-v3, ResNET, Inception-ResNet-v2, and Xception, with Xception achieving the highest accuracy of 91.47%. Patil et al. [8] trained a CNN on a dataset of embryo images from multiple days to classify embryos based on shape, cell division, and size. Patil et al. [10] utilized the Find-S algorithm to classify embryos and select features based on cell characteristics. Bormann et al. [13] employed a deep CNN for automatic embryo classification, comparing its performance to that of ten embryologists. Their study showed that neural networks outperformed human experts in classification tasks.

Chen et al. [14], used transfer learning with the ImageNet dataset and the ResNet50 architecture to classify embryos graded using the Gardner system that were taken 112 to 116 hours (5 days) or 136 to 140 hours (6 days) after fertilization. Their method achieved an accuracy of 75.36% in grading embryos based on blastocyst, trophectoderm, and inner cell mass. According to Hernandez-Gonzalez et al. [15], three different Bayesian network models were used to classify embryos based on the probability of transfer into two groups of datasets, one of which uses only embryo features and the other uses both embryo features and cycle features. A simple Bayesian network,

Tree Augmented Naïve Bayes (TAN), and K-dependence Bayesian Network (KDB) were used to build the model in this article. Embryos were classified on the third day by Petersen et al. [16], based on their growth status using a decision tree and the KIDScore D3 algorithm. Success rate was based on gestational sacs or fetal heartbeats when sacs were unavailable. The presence of two pronuclei in the embryo is the first criterion in this study. Another dataset of 11218 images was used to evaluate the performance of the KIDScore algorithm. The Area Under the Curve (AUC) of this system is 64.2% when one embryo is transferred and 65.8% when two embryos are transferred.

Silver et al. [17], utilized unlabeled videos to train a convolutional neural network encoder and labeled videos to train a Long Short-Term Memory (LSTM) neural network. The system outperformed embryologists in grading embryo images, evaluated by five experts from different countries. Cao et al. [18], employed a deep convolutional neural network to select embryos based on their morphological features, utilizing a Region Of Interest (ROI) extractor to examine blastocysts and classify embryo images. A convolutional neural network and a Recurrent Neural Network (RNN) were used by Kragh et al. [19] to grade the appearance of embryos, focusing on their inner cell mass and trophectoderm status, which was used from 90 hours post insemination until the blastocyst expanded to its maximum size. They achieved accuracies of 71.9% for detecting the inner cell mass and 76.4% for detecting the trophectoderm. Durairaj et al. [20], employed a Multilayer Perceptron (MLP) and data from 250 patients with 27 features, such as female age, Body Mass Index (BMI), duration of infertility, tubal causes, sperm concentration, number of oocytes retrieved, and number of transferred embryos to predict IVF success rates. Their model achieved a 73% accuracy rate. Liu et al. [21], utilized Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF) algorithms to predict pregnancy success using features such as age, BMI, endometrial thickness, type of infertility, serum progesterone level, and serum estradiol level on the day of embryo transfer. Pregnancy success was defined by sac or fetal heartbeat within 4-5 weeks post-transfer. Random forest achieved the highest accuracy of 61%. To predict live births, Goyal et al. [22], employed ensemble learning, machine learning, and deep learning with features, such as age, number of previous cycles, type and cause of infertility, and the number of transferred embryos. Random forest, voting-hard classifier, voting-soft classifier, and Adaboost were used in ensemble learning, and k-nearest neighbor, multilayer perceptron, and decision tree were used in machine learning. In this article, the network is trained in two modes: with and without feature selection. Random forest without feature selection outperformed other methods.

In addition to supervised learning, unsupervised techniques have been applied to uncover hidden patterns in embryo development. Kanakasabapathy et al. [23], utilized a dataset comprising images captured by various systems, including time-lapse microscopes and clinical microscopes, resulting in images with varying qualities. They employed adversarial learning to classify embryos into blastocyst and non-blastocyst groups, and utilized the Scale-Invariant Feature Transform (SIFT) for feature extraction. Thakkar et al. [24], utilized Multiple Linear Regression (MLR) and Support Vector Regression (SVR) for embryo status prediction using user-based and item-based collaborative filtering techniques. They used user- and item-based collaborative filtering with Pearson correlation and nearest-neighbor methods. Tran et al. [25], developed IVY, a fully automated deep learning model to predict pregnancy (fetal heart) from time-lapse embryo videos. The IVY model scores videos on a scale from 0 to 1 for pregnancy prediction, without requiring manual annotation by embryologists.

Although AI and deep learning methods have shown great potential in improving embryo selection, many studies suffer from limited datasets [3, 4, 12, 19, 23], suboptimal generalization, and relatively high false-positive rates. These limitations can lead to the transfer of non-viable embryos and reduced implantation success. Therefore, this study proposes a deep learning-based model using a modified GoogLeNet architecture and transfer learning to classify blastocyst images. The current study aimed to address the shortcomings of prior methods by improving precision, recall, and accuracy while reducing false positives, ultimately enhancing the effectiveness and reliability of embryo selection in IVF.

Material and Methods

This experimental study was designed to automate embryo selection for transfer. Our approach integrates image preprocessing with a tailored deep learning architecture to enhance predictive performance. Our proposed approach comprises image preprocessing and the proposed architecture.

Preprocessing

Preprocessing entails adjusting and standardizing images before feeding them into the network. Given that images in the dataset were captured using three different devices, standardization is essential. Preprocessing is divided into three stages, as follows:

1. Resizing

Images captured by various devices vary in size, necessitating standardization to 224×224, as required by the proposed method's architecture. This resizing not only ensures uniformity but also reduces computational complexity within the convolutional neural network.

2. Converting to a tensor

Conversion to a tensor involves transforming the input image into a matrix of pixels and subsequently into a three-dimensional tensor encompassing red, green, and blue channels, with dimensions of 3×224×224 (Figure 1).

3. Normalization

Data normalization is crucial for network performance by emphasizing differences in images rather than common features. This process involves computing the mean (μ) and standard deviation (σ) across all three-color channels (red, green, and blue) and normalizing the images using Equation (1), where X is the original input value and Z is the normalized output.

$$Z = \frac{X - \mu}{\sigma} \qquad (1)$$

Table 1 shows normalized image examples using Equation (1) to illustrate the process. These examples highlight the importance of normalization in our method. The mean and standard deviation values for the RGB (Red, Green and Blue) channels were as follows: red channel (μ=0.406, σ=0.225), green channel (μ=0.456, σ=0.224), and blue channel (μ=0.485, σ=0.229).

Proposed Approach

Our system's initial architecture is based on GoogLeNet [26] and a pre-trained version on the ImageNet dataset. After preliminary

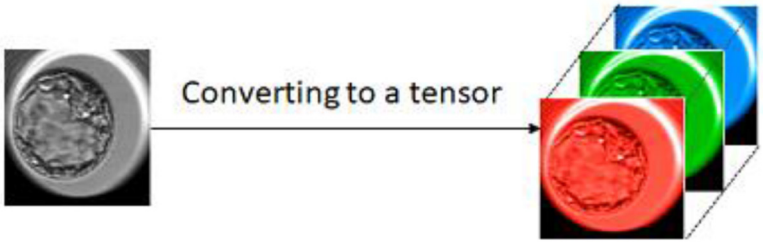
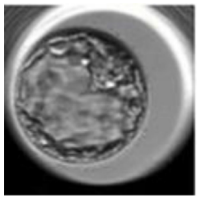
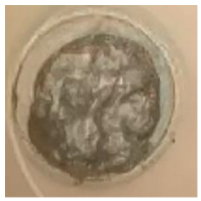
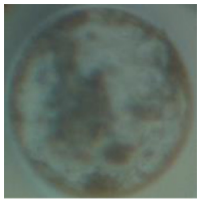
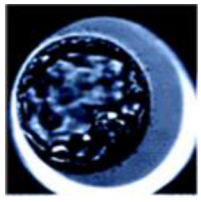
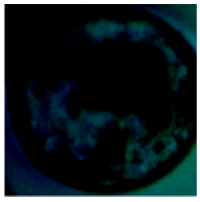


Figure 1: The conversion process from an image to a tensor.

Table 1: Illustration of the Normalization Process Applied to Embryo Images

Images before normalization			
Images after normalization			

experiments comparing GoogLeNet's performance with several other popular deep learning models, such as ResNet50 and DenseNet121, we decided to use GoogLeNet as the base architecture. GoogLeNet achieved superior accuracy and generalization on our dataset. Additionally, GoogLeNet's relatively lightweight structure and Inception modules allowed for efficient training and reduced overfitting, which were important considerations given the size of our dataset and the variability across imaging devices.

Informed by prior work on modifying Inception modules [27] and the GoogLeNet architecture [28], we implemented a series of adjustments to the pre-trained GoogLeNet model. To identify the optimal architecture, we systematically evaluated 35 modifications to the GoogLeNet model, including the addition or removal of Inception modules (4B, 4C, 5A) and convolutional layers, as well as combinations of these changes. Each modification was initially tested over 3 epochs, and the top four performers were trained for 300 epochs. The final architecture was selected based on the highest validation and test accuracy among these top candidates. Detailed results of all 35 modifications, including training, validation, and test accuracies, are provided in Tables 2 and 3 to ensure reproducibility. As shown in Figure 2, the GoogLeNet architecture includes convolutional layers and Inception modules [26], whose layer names (e.g., 4B, 4C, 5A) correspond directly to those referenced in the Tables 2 and 3. For example, modifications such as "DEL (Deletion) 4B 4C" or "ADD (Addition) 5A" indicate experiments, in which the Inception modules 4B, 4C, or 5A were removed or added, respectively. This alignment between the architecture diagram and the detailed results Table 2 ensures clarity and reproducibility, allowing readers to trace the structural changes tested in our study.

Figure 3 illustrates our proposed module, comprising seven Inception modules. An essential component of GoogLeNet, the Inception module captures multi-scale features by processing inputs through parallel convolutional pathways with different filter sizes (1×1, 3×3, and 5×5). The network can effectively extract both local and global spatial patterns because these pathways are concatenated along the channel dimension. Larger filters are preceded by a 1×1 convolution to lower computational complexity through dimensionality reduction. The three-step process for each convolutional layer in the architecture is as follows: employing learned filters for the convolution operation, batch normalization to stabilize activations by modifying mean and variance, and then introducing non-linearity using a ReLU (Rectified Linear Unit)

Table 2: Performance of 35 architectural modifications of GoogLeNet after 3 epochs.

	STATE (EPOCH=3)	TRAIN_ACC	VAL_ACC	TEST_ACC
	WITHOUT CHANGE	54.936	55.644	58.92
1)	ADD 4B	52.489	52.145	55.22
2)	ADD 4C	56.238	53.394	55.13
3)	ADD 5A	57.394	55.893	61.392
4)	DEL 4B	52.354	52.145	53.398
5)	DEL 4C	53.06	55.394	60.593
6)	DEL 5A	55.738	57.434	61.59
7)	ADD 4B 4C	52.135	50.854	55.33
8)	ADD 4B 5A	53.28	49.521	60.293
9)	ADD 4C 5A	54.874	55.31	61.392
10)	ADD 4B 4C 5A	45.011	44.815	47.668
11)	DEL 4B 4C	53.156	55.852	61.99
12)	DEL 4B 5A	52.979	50.396	55.563
13)	DEL 4C 5A	54.29	55.185	58.095
14)	DEL 4B 4C 5A	54.093	54.727	56.962
15)	ADD TWO MORE 4B	53.270	53.353	57.79
16)	ADD TWO MORE 4C	52.364	50.146	58.461
17)	ADD TWO MORE 5A	52.416	53.519	55.72
18)	ADD TWO MORE 4B 4C 5A	52.645	46.147	52.798
19)	ADD 4B DEL 4C	55.686	50.646	57.828
20)	ADD 4B DEL 5A	52.489	51.603	56.062
21)	ADD 4C DEL 4B	55.905	53.644	60.493
22)	ADD 4C DEL 5A	57.05	52.686	56.862
23)	ADD 5A DEL 4B	55.030	52.686	49.00
24)	ADD 5A DEL 4C	55.582	48.98	58.694
25)	ADD 4B 4C DEL 5A	52.812	50.437	53.431
26)	ADD 4B 5A DEL 4C	49.74	44.315	46.968
27)	ADD 4C 5A DEL 4B	51.156	54.269	59.760
28)	DEL 4B CONV2	49	47.730	51.865
29)	ADD 4B 4C CONV2	51.468	20.283	19.454
30)	ADD 4B DEL CONV2	42.23	45.481	51.699
31)	DEL 4B 4C CONV2	42.595	41.608	43.471
32)	DEL 4C CONV2	49.396	49.146	52.665
33)	ADD THREE MORE 5A	54.634	49.271	53.231
34)	ADD TWO MORE 5A DEL 4B	50.833	49.313	53.231
35)	ADD TWO MORE 5A DEL 4B CONV2	45.22	45.273	49.434

ACC: Accuracy, VAL: Validation, ADD: Addition, DEL: Deletion

Table 3: Performance of selected GoogLeNet modifications after 300 epochs.

STATE (EPOCH=300)	TRAIN_ACC	VAL_ACC	TEST_ACC
WITHOUT CHANGE	90.033	90.504	93.471
DEL 4B 4C	91.314	91.295	97.468
DEL 5A	91.262	90.92	95.969
ADD 5A	91.314	90.587	95.436
ADD 4C 5A	89.408	85.756	92.139
Without pre-training	85.513	85.423	91.27

ACC: Accuracy, VAL: Validation, DEL: Deletion, ADD: Addition

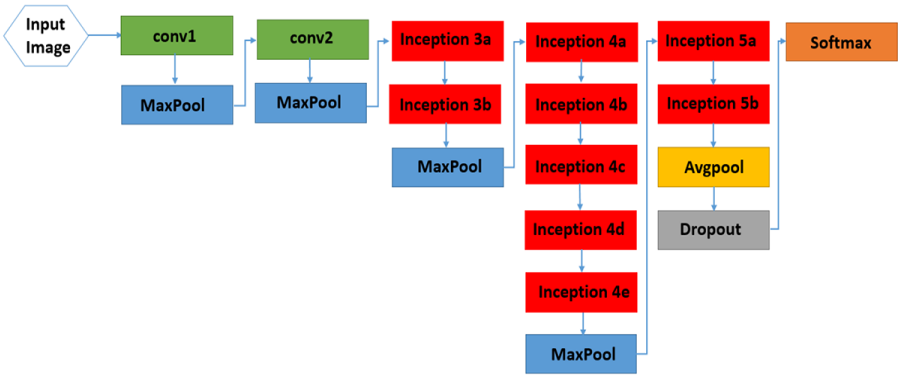


Figure 2: GoogLeNet architecture

activation function. This improves gradient flow and training stability. Table 4 provides detailed specifications of our proposed architecture.

Our proposed architecture requires preprocessed images with dimensions of 3×224×224 pixels (channels × height × width). Upon receiving the input image, we apply a convolutional layer with 64 filters of size 7×7 and a stride of 2, yielding 64 feature maps of size 112×112. Subsequently, a 3×3 Max-pooling layer with a stride of 2 is employed to reduce the feature map size to 56×56.

Following this, a 1×1 convolutional layer is applied, succeeded by a 3×3 convolutional layer with 192 filters and a stride of 2, resulting in 192 feature maps of size 56×56. Another 3×3 max-pooling layer with a stride of 2 is utilized to further reduce the feature map size to 28×28.

Two Inception modules are then employed to generate 480 feature maps of size 28×28. Subsequently, a 3×3 max-pooling layer with a stride of 2 is applied to decrease the feature map size to 14×14. Three additional Inception modules are integrated into the network to produce 832 feature maps of size 14×14.

Next, a 3×3 max-pooling layer with a stride of 2 is utilized to reduce the feature map size to 7×7, followed by the incorporation of two more Inception modules to generate 1024 feature maps of size 7×7. Subsequently, a 7×7 average pooling layer with a stride of 1 is applied to reduce the feature map size to 1×1.

To prevent overfitting, random dropout is applied to 20% of the network weights. The final layer of the network employs a softmax activation function for multi-class classifica-

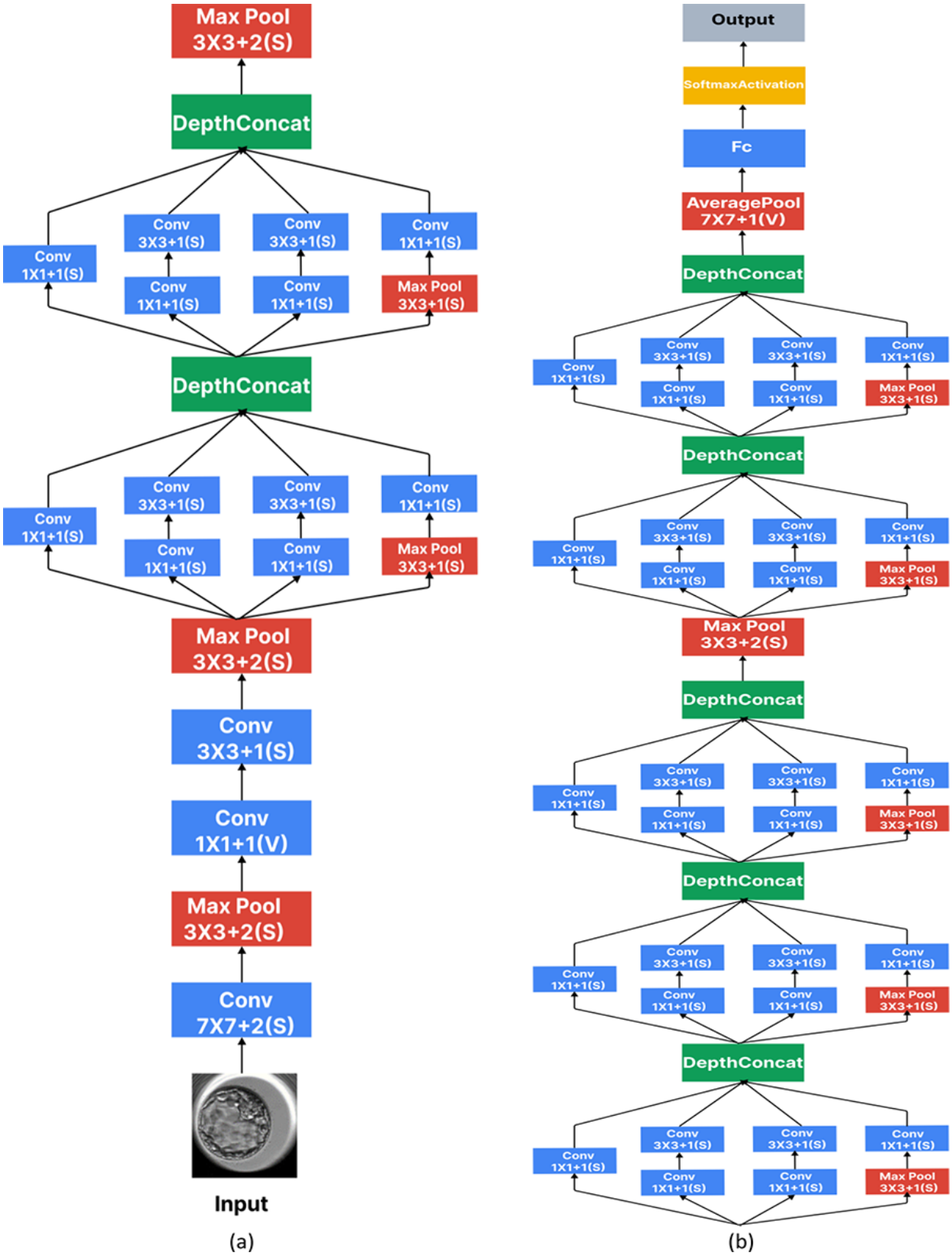


Figure 3: The proposed modified GoogLeNet architecture, showing (**a**) the initial layers and Inception modules and (**b**) the final layers.

Table 4: Details of proposed architecture.

Type	Patch size	Stride	Output Size	Depth	#1×1	#3×3 reduce	#3×3	#3×3 reduce	#3×3	Pool proj
Convolution	7×7	2	112×112×64	1						
Max Pool	3×3	2	56×56×64	0						
Convolution	3×3	1	56×56×192	2		64	192			
Max Pool	3×3	2	28×28×192	0						
Inception			28×28×256	2	64	96	128	16	32	32
Inception			28×28×480	2	128	128	192	32	96	64
Max Pool	3×3	2	14×14×480	0						
Inception			14×14×512	2	192	96	208	16	48	64
Inception			14×14×528	2	112	144	288	32	64	64
Inception			14×14×832	2	256	160	320	32	128	128
Max Pool	3×3	2	7×7×832	0						
Inception			7×7×832	2	256	160	320	32	128	128
Inception			7×7×1024	2	384	192	384	48	128	128
Avg. Pool	7×7	1	1×1×1024	0						
Dropout (20%)			1×1×1024	0						
Linear			1×1×5	1						
Softmax			1×1×5	0						

#: The number of filters used in each convolutional layer within the Inception module

tion, categorizing embryos into five classes of blastocysts and non-blastocysts

1. Modified Transfer Learning (Third Approach)

In the third experiment, we proposed a modified version of the GoogLeNet architecture to enhance performance. We retained the general structure of the Inception modules but implemented several key modifications, primarily in the classification head. The original fully connected layer was replaced with a new sequence: a dense layer (512 units, ReLU activation), a dropout layer (rate=0.2), and a final dense layer with 5 output neurons and a softmax activation function.

We added batch normalization layers after each convolutional layer in the modified Inception modules to improve learning stability and convergence speed. Some filters in the Inception modules were modified based on empirical testing to better capture embryo features. We tested 35 variations and selected the most accurate one. These changes helped improve feature extraction tailored to embryo morphology.

In summary, our model preprocesses images before passing them through convolutional and max-pooling layers, followed by Inception modules and average pooling. Random dropout is applied, and a softmax activation function is used for classification. In the subsequent section, we evaluate the performance of this proposed model.

Results

Dataset

Our dataset images were acquired 113 hours

post-insemination from 374 patients at the fertility center of Massachusetts General Hospital in Boston. Image diversity improves network robustness. Embryologists classified these images into five groups based on Gardner's modified classification system and morphological growth status, with groups 1 and 2 comprising non-blastocysts and groups 3 to 5 comprising blastocysts. Table 5 presents a sample of each state of imaging systems.

The dataset comprised a total of 2805 embryo images captured by three distinct imaging systems:

• Clinical time-lapse imaging system (2440 images): Class 1 (473 images), Class 2 (411 images), Class 3 (473 images), Class 4 (366 images), Class 5 (717 images).

• Portable 3D-printed imaging system (69 images): Class 1 (10 images), Class 2 (3 images), Class 3 (3 images), Class 4 (9 images), Class 5 (44 images).

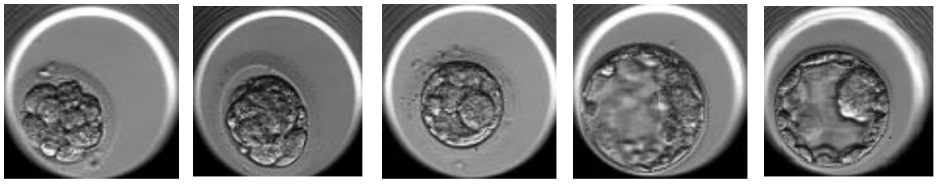
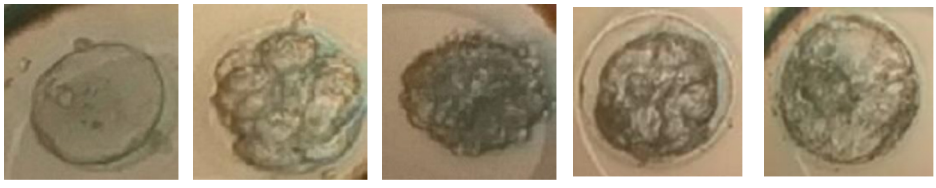
• Smartphone-based imaging system (296 images): Class 1 (90 images), Class 2 (9 images), Class 3 (5 images), Class 4 (81 images), Class 5 (111 images).

Across all systems, the total number of images per class was: Class 1 (573 images), Class 2 (423 images), Class 3 (481 images), Class 4 (456 images), and Class 5 (872 images).

Diverse sources enhanced generalizability. The incorporation of images from different imaging systems enhances the potential robustness of the network. However, the dataset was naturally imbalanced, as the majority of the images were sourced from the clinical time-lapse system. To solve this, we used data augmentation techniques such as random rotations, flips, and Gaussian noise. We also utilized stratified sampling to ensure that the training and testing sets had the same class proportions. The dataset used in this study is available as supplemental material in the work of Kanakasabapathy et al. [23] and can be obtained directly from their publication or by contacting the corresponding authors.

While the inclusion of images from diverse imaging systems enhances the potential

Table 5: Sample images from the five classes across different imaging systems

Imaging systems	Class 1	Class 2	Class 3	Class 4	Class 5
Clinical time-lapse imaging system					
Portable 3D-printed imaging system					
3D-printed Smartphone-based imaging system					

versatility of our model, a key limitation is the relatively small number of images from the 3D-printed and smartphone-based systems compared to the clinical time-lapse system. This imbalance may have biased the model toward patterns specific to the clinical images, potentially affecting its performance on data from lower-cost or portable devices. To mitigate this, we employed data augmentation and stratified sampling. Future work involving a larger, more balanced multi-source dataset would improve the model's generalizability and ensure robust performance across all device types, particularly in low-resource settings.

Evaluation Measure

We compute the True Positive (TP) rate, True Negative (TN) rate, False Positive (FP) rate, and False Negative (FN) rate for each image. Our evaluation metrics include accuracy, precision, recall, and F1-score.

Network fitting and parameter tuning

The dataset is randomly partitioned into three segments: training, validation, and testing, with an 80:20 ratio for training and testing, respectively. Additionally, 20% of the training data is allocated for validation. Data augmentation techniques, including resized cropping, random rotation, random vertical and horizontal flipping, and adding Gaussian noise, are applied to enhance model generalization. Images were shuffled to preserve class balance across sets. Training utilizes a batch size of 128 and 300 epochs, with an Adam optimizer employing a learning rate of 0.001 and a categorical cross-entropy loss function. The model weights that achieved the best performance on the validation set were retained for final evaluation.

Experimental Methodology

We used Python 3.9 to implement the proposed algorithm, with PyTorch employed for implementing the convolutional neural network, and the PIL (Python Imaging Library) library used for image manipulation. The implementation was executed on a system featuring an Intel Core i7 6500U CPU and a NVidia GeForce 930Mx GPU, leveraging the cuDNN deep neural network library for parallel processing on the GPU.

The first experiment presents the results of the GoogLeNet architecture implementation, and the second experiment presents the results of transfer learning in the GoogLeNet architecture. In the third experiment, we present the results of modified transfer learning on the GoogLeNet architecture. All statistical comparisons between models were conducted using the Wilcoxon signed-rank test with bootstrapped 95% confidence intervals.

1. GoogLeNet architecture

In the first experiment, we evaluated the performance of the GoogLeNet architecture. Figure 4 illustrates the system's performance during training and validation epochs. The final evaluation on the test dataset yielded a multi-class accuracy of 91.27%, precision of 91.62%, recall of 91.27%, and F1-score of 91.23%. Detailed evaluation results for each class on the test dataset are presented in Table 6. Figure 5 shows confusion matrices for classification performance.

2. Transfer learning in GoogLeNet architecture

The second experiment involved employing transfer learning in the GoogLeNet architecture using the ImageNet dataset. Figure 4 showcases the system's performance during training and validation epochs. The final evaluation on the test dataset resulted in a multi-class accuracy of 93.47%, precision of 93.68%, recall of 93.47%, and F1-score of 93.43%. Evaluation results for each class on the test dataset are outlined in Table 6. Figure 5 illustrates the confusion matrix with correct classifications on the diagonal and misclassifications off-diagonal. Notably, this architecture outperformed the

standard GoogLeNet architecture when transfer learning was not employed.

3. Approach architecture

The third experiment involved examining the model by modifying transfer learning in the GoogLeNet architecture using the ImageNet dataset. Figure 4 illustrates the system's performance in each epoch during training and validation modes. The final evaluation on the test dataset yielded a multi-class accuracy of 97.47%, precision of 97.47%, recall of 97.47%, and F1-score of 97.46%. Evaluation results for each class on the test dataset are presented in Table 6. The confusion matrix

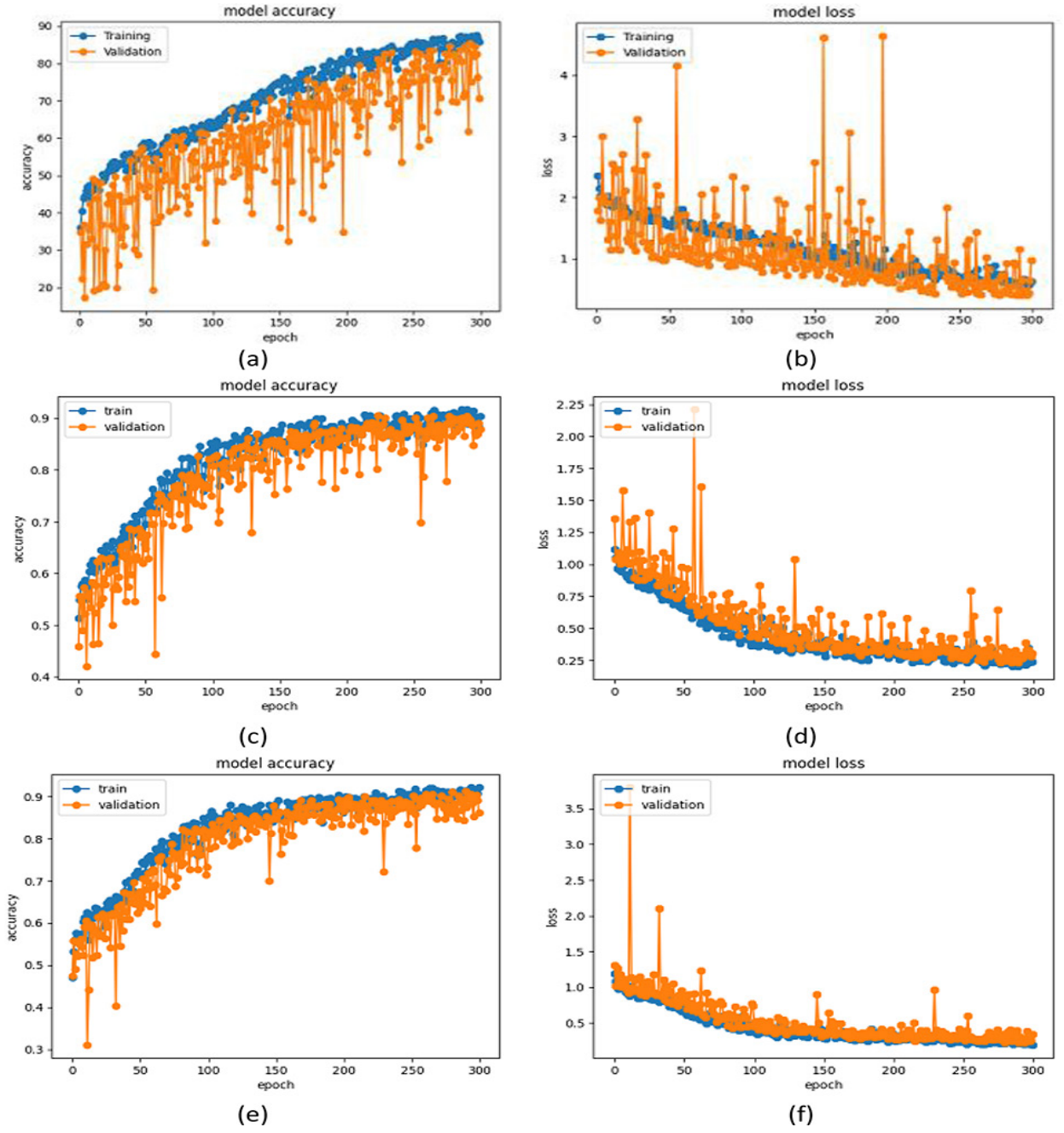


Figure 4: System performance during training and validation. (**a, b**) Accuracy and loss for the baseline GoogLeNet architecture. (**c, d**) Accuracy and loss for the pre-trained GoogLeNet architecture. (**e, f**) Accuracy and loss for the proposed modified architecture

Table 6: Performance Comparison of GoogLeNet Architectures

Metric	Class	GoogLeNet (Exp 1)	Transfer Learning (Exp 2)	Modified Transfer Learning (Exp 3)
Precision (%)	Class 1	91.19	92.85	**98.14**
	Class 2	95.12	93.08	97.02
	Class 3	83.48	**98.14**	97.67
	Class 4	90.89	87.22	95.82
	Class 5	**95.55**	95.03	97.99
	Macro avg.	91.25	93.27	97.33
	Weighted avg.	91.62	93.68	97.47
Recall (%)	Class 1	95.07	**97.11**	98.47
	Class 2	82.46	94.96	**98.59**
	Class 3	**97.54**	83.16	95.61
	Class 4	82.93	94.09	95.40
	Class 5	93.94	96.52	98.43
	Macro avg.	90.39	93.17	97.30
	Weighted avg.	91.27	93.47	97.47
F1-score (%)	Class 1	93.09	94.93	**98.30**
	Class 2	88.34	94.01	97.80
	Class 3	89.97	90.03	96.63
	Class 4	86.73	90.53	95.61
	Class 5	**94.74**	**95.77**	98.21
	Macro avg.	90.57	93.05	97.31
	Weighted avg.	91.23	93.43	97.46
Accuracy (%)	Class 1	95.07	97.11	**98.47**
	Class 2	82.46	94.96	98.59
	Class 3	**97.54**	83.16	95.61
	Class 4	82.93	94.09	95.04
	Class 5	93.94	**96.52**	98.43
Overall accuracy		**91.27**	**93.47**	**97.47**

for our proposed model (Figure 5c) shows a strong diagonal, indicating a high number of correct classifications and minimal confusion between classes.

Comparatively, our proposed architecture demonstrated significantly higher accuracy and performance when compared to previous methods and other proposed architectures. Our suggested approach demonstrated excellent performance, achieving 97.47% accuracy, precision, and recall along with a 97.46% F1-score. These findings outperform those in [12], which achieved an accuracy of 91.47% using the Xception architecture [23] (a precision of 94.14%, recall of 94.88%, and F1-score of 91.51% using a traditional machine learning approach). Even our own baseline transfer learning experiment using GoogLeNet (without architectural modifications) achieved a slightly lower accuracy of 93.47%. Table 7

compares the proposed method with previous studies on embryo classification.

Grad-CAM heatmaps were created for a single representative embryo from each class to better understand how our suggested model makes decisions. Figure 6 shows the model focuses on biologically relevant areas, such as the inner cell mass, consistent with embryologist criteria. These results suggest that our architecture modifications, combined with transfer learning and targeted preprocessing, provide a more robust and accurate solution for embryo classification. This represents a step forward in automating embryo

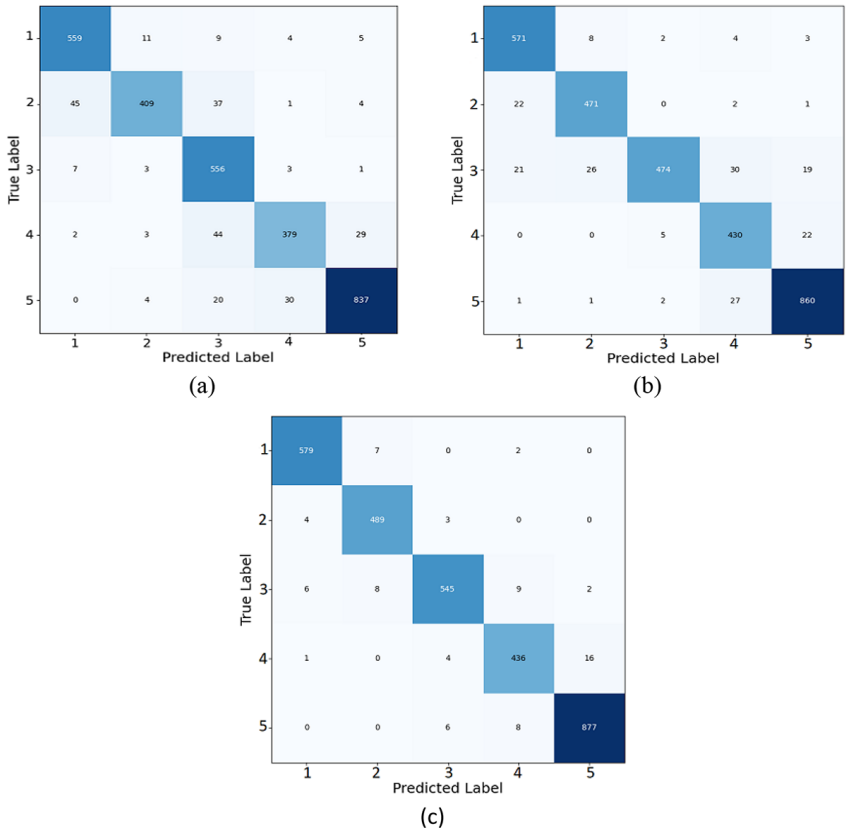


Figure 5: Confusion matrix of our methods: (**a**) GoogLeNet architecture, (**b**) pre-trained GoogLeNet architecture, (**c**) approach architecture

Table 7: Comparison of proposed method with existing studies on embryo classification

Study / Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Proposed Method	**97.47**	**97.47**	**97.47**	**97.46**
GoogLeNet (Our 1st Exp.)	91.27	91.27	91.62	91.23
Transfer Learning (Our 2nd Exp.)	93.47	93.68	93.47	93.43
Xception [1]	90.97	-	-	-
Xception [12]	91.47	-	-	-
CNN [8]	-	75.62	88.15	81.15
Adaptive adversarial [23]	91.51	94.88	92.14	-

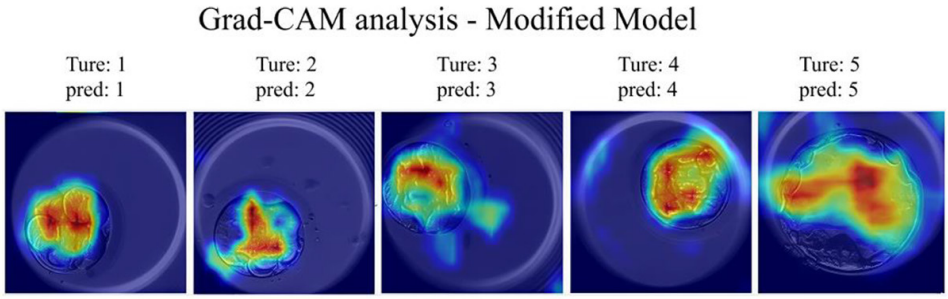


Figure 6: Grad-CAM (Gradient-weighted Class Activation Mapping) heatmap for representative embryo images from each class, generated using our proposed model.

selection in IVF, offering improved reliability and consistency over prior approaches, which is critical for clinical decision-making.

Our model outperformed the standard GoogLeNet and transfer learning models, according to statistical analysis using the Wilcoxon signed-rank test (P-value<0.0001). This was corroborated by bootstrapped 95% CIs, which showed that our model's accuracy (97.47%, CI=[96.94%, 98.00%]) outperformed the GoogLeNet baseline (90.24%, CI=[90.24%, 92.37%]) and the transfer learning model (93.47%, CI=[92.57%, 94.34%]).

Discussion

This study developed a deep learning-based technique that significantly improves the accuracy of embryo selection, potentially reducing the risk of IVF failure. Our modified GoogLeNet architecture achieved a classification accuracy of 97.47%, outperforming other deep learning models and conventional methods. This result surpasses the performance of adversarial learning (94.88% precision [23]) and the Xception architecture (91.47% accuracy [12]), highlighting the effectiveness of our architectural modifications. We attribute this performance improvement to our customized preprocessing and architectural modifications, which were designed to enhance feature extraction specific to blastocyst morphology.

A more accurate and reliable classification model was produced by combining transfer learning with batch normalization, dropout, and customized classifier layers, especially when applied to a variety of imaging sources. The accuracy of AI-assisted embryo selection in IVF is greatly increased by these changes.

Data augmentation and standardized preprocessing (resizing, normalization) were used to improve data diversity and model consistency across imaging systems to address data scarcity, a common limitation in medical imaging. By taking these actions, prediction reliability was improved and false positives were decreased.

Our ultimate objective was to refine the GoogLeNet architecture to increase accuracy and decrease false positive rates in embryo selection. The findings point to a significant advancement in the automation and standardization of IVF decisions. To further improve performance, future research should investigate the analysis of embryos at earlier stages (such as pronuclei), evaluate the effects of laboratory conditions, and look into other preprocessing techniques.

Conclusion

This study successfully developed a deep learning framework for automated embryo selection, demonstrating that a modified GoogLeNet architecture with targeted preprocessing can achieve high classification accuracy (97.47%). By providing a more objective and reliable alternative to manual morphological assessment, our model has the potential to significantly improve IVF success rates and

reduce the emotional and financial burden on patients. Future work should focus on validating this model with larger, multi-center datasets and exploring its application to earlier developmental stages, such as the pronuclei stage, to further enhance its clinical utility.

Acknowledgment

The authors extend their heartfelt gratitude to the participants of this study and the fertility center of Massachusetts General Hospital in Boston for generously providing access to the invaluable dataset, without which this research would not have been possible. The study followed ethical guidelines and protected participants' rights.

Authors' Contribution

All authors made significant contributions to the conception, design, implementation, and writing of this study. B. Nasiri was in charge of data preprocessing, model development, and preparing the manuscript; N. Farajzadeh helped interpret the results. N. Farajzadeh and J. Ghavidel Neycharan oversaw the project and offered critical revisions. All authors reviewed and approved the final manuscript.

Ethical Approval

Ethical approval was not required for this study because it used a publicly available dataset from a previously published work. No new human participant data were collected.

Conflict of Interest

None

References

1. Bormann CL, Kanakasabapathy MK, Thirumalaraju P, Gupta R, Pooniwala R, Kandula H, et al. Performance of a deep learning based neural network in the selection of human blastocysts for implantation. *Elife.* 2020;**9**:e55301. doi: 10.7554/eLife.55301. PubMed PMID: 32930094. PubMed PMCID: PMC7527234.
2. Merican ZZ, Yusof UK, Abdullah NL. Review on Embryo Selection Based on Morphology Using Machine Learning Methods. *Int J Advance Soft Compu Appl.* 2021;**13**(2):44-59.
3. Kragh MF, Karstoft H. Embryo selection with artificial intelligence: how to evaluate and compare methods? *J Assist Reprod Genet.* 2021;**38**(7):1675-89. doi: 10.1007/s10815-021-02254-6. PubMed PMID: 34173914. PubMed PMCID: PMC8324599.
4. Khosravi P, Kazemi E, Zhan Q, Malmsten JE, Toschi M, Zisimopoulos P, et al. Deep learning enables robust assessment and selection of human blastocysts after in vitro fertilization. *NPJ Digit Med.* 2019;**2**:21. doi: 10.1038/s41746-019-0096-y. PubMed PMID: 31304368. PubMed PMCID: PMC6550169.
5. Afnan MA, Rudin C, Conitzer V, Savulescu J, Mishra A, Liu Y, Afnan M. Ethical implementation of artificial intelligence to select embryos in in vitro fertilization. In Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society; USA: Association for Computing Machinery; 2021. p. 316-26.
6. Liao Q, Zhang Q, Feng X, Huang H, Xu H, Tian B, et al. Development of deep learning algorithms for predicting blastocyst formation and quality by time-lapse monitoring. *Commun Biol.* 2021;**4**(1):415. doi: 10.1038/s42003-021-01937-1. PubMed PMID: 33772211. PubMed PMCID: PMC7998018.
7. Gardner DK, Lane M, Stevens J, Schlenker T, Schoolcraft WB. Blastocyst score affects implantation and pregnancy outcome: towards a single blastocyst transfer. *Fertil Steril.* 2000;**73**(6):1155-8. doi: 10.1016/s0015-0282(00)00518-5. PubMed PMID: 10856474.
8. Patil SN, Wali U, Swamy MK, Nagaraj SP, Patil N. Deep learning techniques for automatic classification and analysis of human in vitro fertilized (IVF) embryos. *J Emerg Technol Innov Res.* 2018;**5**(4):100-6.
9. Berntsen J, Rimestad J, Lassen JT, Tran D, Kragh MF. Robust and generalizable embryo selection based on artificial intelligence and time-lapse image sequences. *PLoS One.* 2022;**17**(2):e0262661. doi: 10.1371/journal.pone.0262661. PubMed PMID: 35108306. PubMed PMCID: PMC8809568.
10. Patil SN, Wali UV, Swamy MK. Selection of single potential embryo to improve the success rate of implantation in IVF procedure using machine learning techniques. In International

Conference on Communication and Signal Processing (ICCSP); Chennai, India: IEEE; 2019. p. 0881-6.

11. Kaufmann SJ, Eastaugh JL, Snowden S, Smye SW, Sharma V. The application of neural networks in predicting the outcome of in-vitro fertilization. *Hum Reprod.* 1997;**12**(7):1454-7. doi: 10.1093/humrep/12.7.1454. PubMed PMID: 9262277.
12. Thirumalaraju P, Kanakasabapathy MK, Bormann CL, Gupta R, Pooniwala R, Kandula H, et al. Evaluation of deep convolutional neural networks in classifying human embryo images based on their morphological quality. *Heliyon*. 2021;**7**(2):e06298. doi: 10.1016/j.heliyon.2021.e06298. PubMed PMID: 33665450. PubMed PMCID: PMC7907476.
13. Bormann CL, Thirumalaraju P, Kanakasabapathy MK, Kandula H, Souter I, Dimitriadis I, et al. Consistency and objectivity of automated embryo assessments using deep neural networks. *Fertil Steril*. 2020;**113**(4):781-7. doi: 10.1016/j.fertnstert.2019.12.004. PubMed PMID: 32228880. PubMed PMCID: PMC7583085.
14. Chen TJ, Zheng WL, Liu CH, Huang I, Lai HH, Liu M. Using deep learning with large dataset of microscope images to develop an automated embryo grading system. *Fertility & Reproduction*. 2019;**1**(01):51-6. doi: 10.1142/S2661318219500051.
15. Hernández-González J, Inza I, Crisol-Ortíz L, Guembe MA, Iñarra MJ, Lozano JA. Fitting the data from embryo implantation prediction: Learning from label proportions. *Stat Methods Med Res*. 2018;**27**(4):1056-66. doi: 10.1177/0962280216651098. PubMed PMID: 27242336.
16. Petersen BM, Boel M, Montag M, Gardner DK. Development of a generally applicable morphokinetic algorithm capable of predicting the implantation potential of embryos transferred on Day 3. *Hum Reprod*. 2016;**31**(10):2231-44. doi: 10.1093/humrep/dew188. PubMed PMID: 27609980. PubMed PMCID: PMC5027927.
17. Silver DH, Feder M, Gold-Zamir Y, Polsky AL, Rosentraub S, Shachor E, et al. Data-driven prediction of embryo implantation probability using IVF time-lapse imaging [Internet]. arXiv [Preprint]. 2020 [cited 2020 Jun 1]. Available from: https://arxiv.org/abs/2006.01035.
18. Cao Q, Liao SS, Meng X, Ye H, Yan Z, Wang P. Identification of viable embryos using deep learning for medical image. In Proceedings of the 5th International Conference on Bioinformatics Research and Applications; USA: Association for Computing Machinery; 2018. p. 69-72.
19. Kragh MF, Rimestad J, Berntsen J, Karstoft H. Automatic grading of human blastocysts from time-lapse imaging. *Comput Biol Med.* 2019;**115**:103494. doi: 10.1016/j.compbiomed.2019.103494. PubMed PMID: 31630027.
20. Durairaj M, Thamilselvan P. Applications of artificial neural network for IVF data analysis and prediction. *Journal of Engineering, Computers, and Applied Sciences*. 2013;**2**(9):11-5.
21. Liu R, Bai S, Jiang X, Luo L, Tong X, Zheng S, et al. Multifactor Prediction of Embryo Transfer Outcomes Based on a Machine Learning Algorithm. *Front Endocrinol (Lausanne)*. 2021;**12**:745039. doi: 10.3389/fendo.2021.745039. PubMed PMID: 34795639. PubMed PMCID: PMC8593232.
22. Goyal A, Kuchana M, Ayyagari KPR. Machine learning predicts live-birth occurrence before in-vitro fertilization treatment. *Sci Rep*. 2020;**10**(1):20925. doi: 10.1038/s41598-020-76928-z. PubMed PMID: 33262383. PubMed PMCID: PMC7708502.
23. Kanakasabapathy MK, Thirumalaraju P, Kandula H, Doshi F, Sivakumar AD, Kartik D, et al. Adaptive adversarial neural networks for the analysis of lossy and domain-shifted datasets of medical images. *Nat Biomed Eng.* 2021;**5**(6):571-85. doi: 10.1038/s41551-021-00733-w. PubMed PMID: 34112997. PubMed PMCID: PMC8943917.
24. Thakkar P, Varma K, Ukani V, Mankad S, Tanwar S. Combining user-based and item-based collaborative filtering using machine learning. In Information and Communication Technology for Intelligent Systems; Singapore: Springer; 2019. p. 173-180.
25. Tran D, Cooke S, Illingworth PJ, Gardner DK. Deep learning as a predictive tool for fetal heart pregnancy following time-lapse incubation and blastocyst transfer. *Hum Reprod.* 2019;**34**(6):1011-8. doi: 10.1093/humrep/dez064. PubMed PMID: 31111884. PubMed PMCID: PMC6554189.
26. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S,

Anguelov D, et al. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition; USA: CVPR; 2015. p. 1-9.

27. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition; USA: CVPR; 2016. p. 2818-26.
28. Basha SS, Ghosh S, Babu KK, Dubey SR, Pulabaigari V, Mukherjee S. Rccnet: An efficient convolutional neural network for histological routine colon cancer nuclei classification. In 15th International Conference on Control, Automation, Robotics and Vision; Singapore: ICARCV; 2018. p. 1222-7.