Advanced Methods in Embryo Fertility Classification: A Deep Learning Approach

Doncean Șerban-Gabriel and Panaite Doru-Răzvan

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași

Abstract. In this article two types of embryo fertility classification methods are presented: one that is limited in resources and does not use pre-trained weights (benchmark variant) and one that extends the first variant by removing the constraints to increase model performance (ablation study). Embryo quality assessment is vital in assisted reproductive technology, aiding specialists in quickly selecting the most viable embryos for implantation. The dataset used comes from the "Embryo classification based on microscopic images" competition and is small in size and extremely unbalanced, with a ratio between the two classes of 7:1 in favor of the infertiles embryos. We have used different methods to solve the problem of unbalanced data, with the best results obtained using batch matching during training and using a large number of increments. Compared to the results obtained in the competition, the score we obtained on the private test set is equal to that of the winner of the competition (0.61587 F1 Score), and on the public dataset to that of the 7th place (0.75862 F1 Score).It can be seen that models in the second category manage to achieve better results on both categories, so pretrained resources and weights lead to increased performance.

Keywords: Embryo · Classification · Deep learning · Kaggle Competition · Pre-trained models · TTA · Ensemble · Imbalanced Data · Model Soup.

1 Introduction

1.1 Competition description

The Challenge: Embryo quality assessment is a critical aspect of assisted reproductive technology, helping fertility specialists to select the most viable embryos for implantation more quickly. In this competition, the challenge is to create a robust deep learning model capable of classifying embryos as "good" or "not good" based on their images on day 3 and day 5 of development. By doing so, we help fertility specialists make more informed decisions, increase the efficiency of assisted reproduction procedures and ultimately contribute to the joy of future parents.

The Dataset: Provided by Hung Vuong Hospital in Ho Chi Minh City, this dataset offers a unique observation into the early stages of embryo development. It comprises images of embryos at two crucial time points, day-3 and day-5.

Each image is labeled as either ”good” or ”not good”, reflecting the embryo’s potential for successful implantation.

Evaluation: Submissions were evaluated based on the F1 score, a measure that balances accuracy and recall, making it a popular selection for unbalanced classification tasks. Higher F1 scores indicate better model performance. The objective of the competition is to maximize the F1 score to accurately classify embryo quality.

Test Dataset: The test dataset consists of 180 images, 120 from day 3 and 60 from day 5, which should classify the embryo images as 1 for ”good” or 0 for ”not good” for both day 3 and day 5 stages.

Dataset analyses: The main problem is that the dataset is extremely unbalanced, the bad embryos greatly outnumber the good ones, this being the main challenge. There is a class imbalance of 7:1. In the training set there are:

- 22 good embryos (Day 3)
- 538 bad embryos (Day 3)
- 102 good embryos (Day 5)
- 178 bad embryos (Day 5)

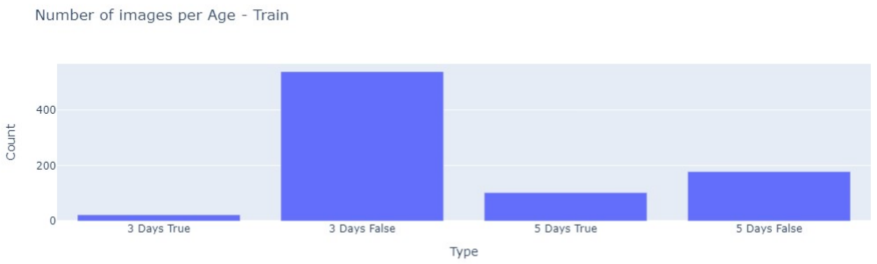
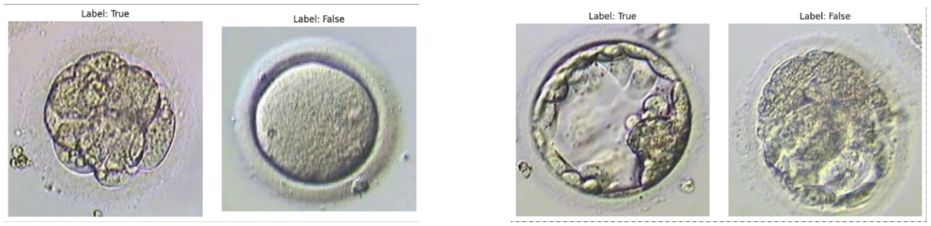


Fig. 1. Train Data



(a) Day 3

(b) Day 5

Fig. 2. Example of images

1.2 Characteristics of Optimal Development

•**Day 3**: At day 3, embryologists use a high-power microscope to take a look at the morphology (a fancy word for "structure") of the embryo. They're looking mainly at two things: the number of cells in the embryo and what they look like.

1. Cell number: An embryo that's dividing well should ideally have between 6 to 10 cells (Day 3 embryos that had 8 or more cells showed a significantly higher live birth rate).[9]
2. Cell appearance: is harder to grade. Embryologists want to see that each cell has a nucleus and that the cells are of equal size. They also check for fragmentation. Up to 20 percent fragmentation is fine.[1]

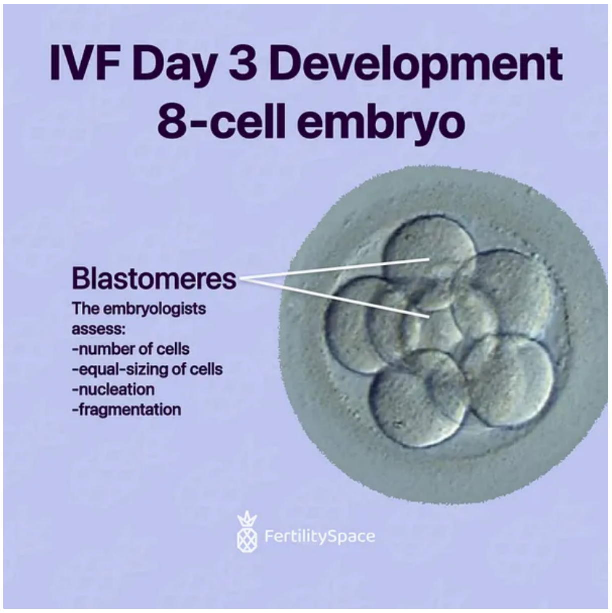


Fig. 3. Information about Day 3

•**Day 5**: At day 5, a healthy embryo will form a blastocyst by now, dividing its cells into sections that will form the foetal matter and placenta. The embryologists use criteria to assess which embryos have developed optimally. The embryo is graded on each of these three factors:[6]

1. The amount that the blastocyst has expanded
2. The quality of the inner cell mass
3. The quality of the trophectoderm

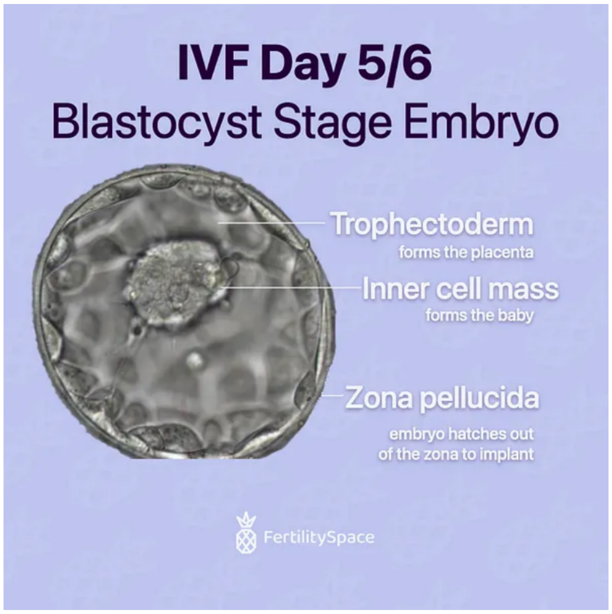


Fig. 4. Information about Day 5

1.3 Our Contribution

In this paper two types of embryo classification methods are presented: one limited in terms of resources (time and memory limited by Kaggle, without using pre-trained weights, etc.) and one that shows an extension of the first one, without these limitations. The direction we focused on was solving data imbalance using various methods (weighted loss function, augmentations, class redefinition, batch matching, changing the prediction threshold, etc.). The results obtained are comparable to the State-Of-The-Art, achieving a score on the private test dataset equal to the winner of the competition (0.61538) and one close to the best score on the public test dataset (0.75862 compared to 0.82758).

2 Related Work

2.1 Literature Study

The competition started on October 1, 2023 and lasted until October 31, 2023. For this reason, there are still no articles with what the others did.

However, a few days after the competition started, on "Discussion", Mot Câu Chuyen Buon Hiu posted a list of "Do's and Don'ts", after he reached 2nd place at the time with a score of 0.71428. These are some of the points he made that have inspired us:

Do:

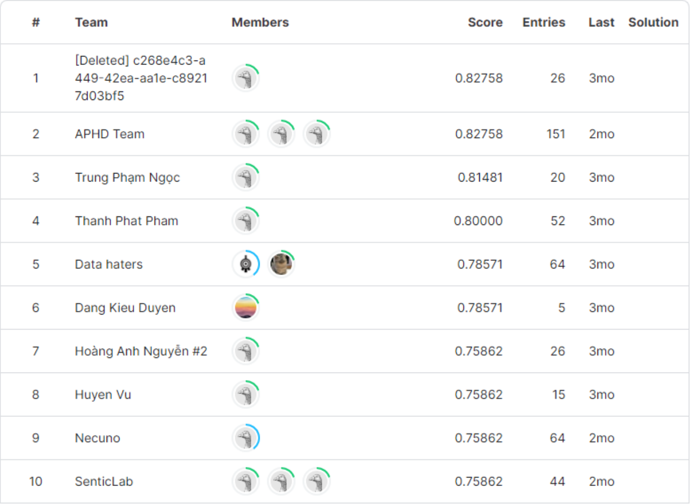
1. ”The EDA may show potential issues with this dataset. Actually, the risk of being imbalanced is not as crucial as the fact that we are having too little data”. This comment led us to take a close look at the data and understand a bit of the medical side behind it. Thus, we added as input to the model the day the image is from, because the embryo looks different in the 2 days.
2. ”Deep models maybe helpful, but not always. My solution implemented as quite shallow architecture and can still achieve the results”. This comment made us careful not to use very large models with many parameters, and to experiment with models of different sizes to see which is the best fit.
3. ”CutMix, cutout helps”. This comment made us try these 2 augmentations, among others.

Don’t:

1. ”Tried very deep architectures. None of them works, the performance is worse than the current solution”. We also tried using very large models, and noticed that it overfitted very quickly, and the results were worse than using smaller models.

2.2 Best competition results

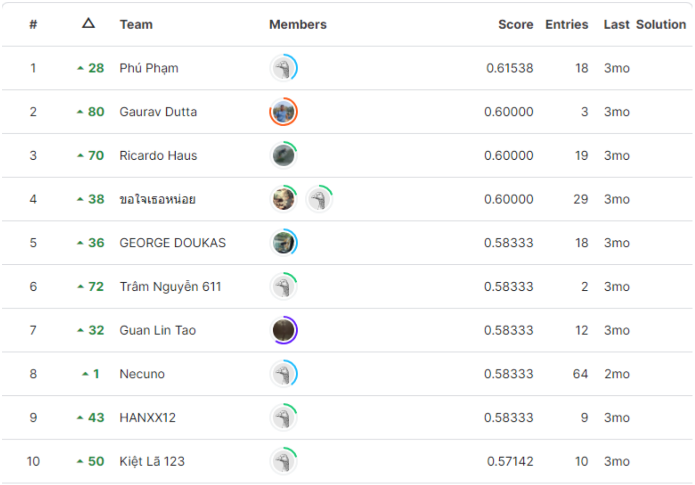
During the competition, the ”public” test result was received when the solution was submitted. This ranking is calculated with approximately 50% of the test data. The first one got a score of 0.82758. These are the top 10 places on the ”public” test set:



#	Team	Members	Score	Entries	Last	Solution
1	[Deleted] c268e4c3-a449-42ea-aa1e-c89217d03bf5		0.82758	26	3mo	
2	APHD Team		0.82758	151	2mo	
3	Trung Phạm Ngọc		0.81481	20	3mo	
4	Thanh Phat Pham		0.80000	52	3mo	
5	Data haters		0.78571	64	3mo	
6	Dang Kieu Duyen		0.78571	5	3mo	
7	Hoàng Anh Nguyễn #2		0.75862	26	3mo	
8	Huyen Vu		0.75862	15	3mo	
9	Necuno		0.75862	64	2mo	
10	SenticLab		0.75862	44	2mo	

Fig. 5. Top 10 - public test set

The results that actually matter are those from the ”private” dataset. The private ranking is calculated with about 50% of the test data, and this ranking reflects the final ranking. Only one of the top 10 on the ”public” dataset also remained in the top 10 on the ”private” dataset. The rest 9 participants moved up from places quite far from the top 10, most coming from 40+. The first one got a score of 0.61538. These are the top 10 places on the ”private” test set:



#	△	Team	Members	Score	Entries	Last Solution
1	▲ 28	Phú Phạm		0.61538	18	3mo
2	▲ 80	Gaurav Dutta		0.60000	3	3mo
3	▲ 70	Ricardo Haus		0.60000	19	3mo
4	▲ 38	ขอใจเธอหน่อย		0.60000	29	3mo
5	▲ 36	GEORGE DOUKAS		0.58333	18	3mo
6	▲ 72	Trâm Nguyễn 611		0.58333	2	3mo
7	▲ 32	Guan Lin Tao		0.58333	12	3mo
8	▲ 1	Necuno		0.58333	64	2mo
9	▲ 43	HANXX12		0.58333	9	3mo
10	▲ 50	Kiệt Lã 123		0.57142	10	3mo

Fig. 6. Top 10 - private test set

3 Methodology

This report presents two different approaches to the problem of classifying healthy embryos from infertile ones in order to make a comparison between the two. The first method (the Benchmark) consists of a series of experiments in which we did not use any additional data or models that were pre-trained on known datasets (e.g. the ImageNet dataset [7]) and constrained the experiments in terms of time and memory to the limitations of Kaggle (30 hours per week and 16 GB of RAM for the GPU). With this approach we want to show that a resource-efficient method can achieve satisfactory results. The second method (the Ablation Study) is an extension to the first and presents experiments where we remove time restrictions and increase the amount of GPU RAM we can use during training and also introduce the possibility of using pre-trained models, which theoretically can achieve better results if they are fine-tuned to our dataset. Also in this section we will present what methods and ideas we have experimented with, but did not have such good results.

3.1 Benchmark Method

Weighted loss function After analyzing the dataset, it is observed that it is very unbalanced between the two classes (it contains 124 healthy embryos, class 1, and 716 infertile embryos, class 0). Thus, a classical training would cause the model to be biased in assigning instances towards class 0 resulting in errors for class 1, so it is necessary to use a method that takes into account the unbalance of the data. For this part, we will use a weighted loss function, so that the classification of an image with a healthy embryo will have a higher weight than one with an infertile embryo to reduce the problem caused by data unbalance. After a simple calculation, we assigned class 0 embryos a weight of 0.18 and class 1 embryos a weight of 1.

Dataset spliting and validation Analysing the dataset, it can be seen that the two main classes can each be divided into two subclasses depending on the age of the embryo in the image, so there are healthy and infertile embryos of 3 and 5 days respectively. If we analyse the distribution of the data in the four classes, we can see that the unbalance between the classes increases, with a ratio of 22:538:102:178 (D3 healthy, D3 infertile, D5 healthy and D5 infertile). In order to be able to evaluate how a training is going, it is necessary that the validation set reflects the training set. Thus, we created the validation fold by extracting 20% of the images from each of the four classes. To measure how well the model performs, in addition to tracking loss we will use the F1 Score metric, good for cases where the data is not balanced which has the following formula:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Data augmentation The dataset provided by the competition is quite small in size, containing only 840 images, and the class with the fewest images (D3 positive) contains only 22. Thus, in order to expand the dataset and increase the generalization power of the model, we used various augmentations, while taking care not to alter the images in such a way that none of them could be understood. To this end we used a Random Augmentation function [2], which set the number of augmentations used and how intensely they should be applied for each image, providing a large diversity. On top of this we used a custom resize function that pads to preserve the aspect ratio of the image, flips and normalized the data. One interesting aspect we noticed when normalizing is that the mean and standard deviation change significantly after applying the augmented pipeline, so we decided to use the mean and standard deviation of the augmented images for normalization. Another important augmentation that we use is CutOut because other participants reported it being effective on this dataset.

Models used One important observation that I have made from the few discussions that exist so far on the competition and the dataset is that large models are not suitable for this task, with other participants in the competition reporting that they perform worse as the size of the models increases. With this in mind, we selected two small models with two different architectures for our experiments:

1. **ResNeSt-14:** ResNeSt, short for Residual Networks with Split-Attention, [11] is a type of deep neural network architecture built upon ResNet (Residual Network) [5], thus facilitating the gradient flow through deeper networks, and using an attention mechanism called split-attention that consists of splitting input features maps into multiple groups that are processed separately by the network, allowing the network to specialize in different aspects of the input data. In each group, a soft attention mechanism is applied to weight the important of features within the group. At the end, the features from all groups are aggregated to produce the output. This method offers comprehensive representation of the input data and improves the capability of learning complex patterns. The version we use is the smallest one, having 10.6 milion parameters.

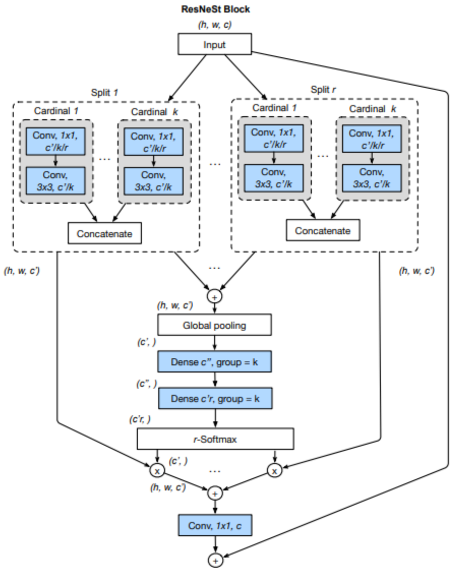


Fig. 7. ResNeSt architecture

2. **MaxVit-Tiny:** MaxVit, short for Multi-Axis Vision Transformer, [8] is a a hybrid approach that combines the strengths of both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) [3] addresing some limitations that both of them have. In this direction, it integrates the local processing abilities of CNNs with the global processing of Vision Transformers, in this way extracting both local and global features in the image. Its architecture follows a hierarchical design like ResNet using a building block that consist of a MBConv (Inverse residual block), a block attention layer (for local interactions) and a grid attention layer (for global mixing), this two combined forming the Multi-Axis Attention block, the network being formed by a seriers of MaxVit blocks. For our experiments, we chose the smallest version of MaxVit, the tiny one, having 30.9 milion parameters.

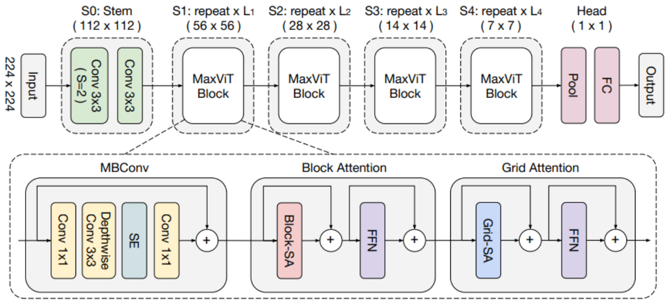


Fig. 8. MaxVit architecture

Loss function and Optimizer For our experiments we used the CrossEntropy loss function combined with weights for the two classes to be assigned, thus alleviating the problem of unbalanced data. Another improvement we made is the use of soft labels which leads to reduced overconfidence and increased generalization power. As an optimizer, we used SAM (Sharpness-Aware Minimization) [4] which not only focuses on loss reduction like classical optimizers, but also on sensitivity to parameter changes, thus being able to generalize better than a classical optimizer, although training time increases as the update is done in two steps.

Using the embryo's day In addition to the image of the embryo as input, we tried to provide before the last classification layer the age of the embryo because, as we saw above, there are significant differences between embryos in the two days. Thus, the model has an extra feature that can be useful in assigning a diagnosis.

Test Time Augmentations An additional technique we use to increase model performance is Test Time Augmentations, which involves generating a number of images for each image in the test set using the pipeline of augmentations used in training, passing each generated image through the model, and the class assigned to the most images being the result for the original image. This technique is beneficial if the model uses many augmentations and is accustomed to augmented data, as in this case, while giving it multiple perspectives for the same image. In our experiments, we generated a number of 20 augmented instances for each image.

Hyperparameters Regarding the hyperparameters, we used a batch size of 16 images, a learning rate of 0.0001, an image size of 224 by 224, 2 augmentations selected by the RandAugment function. We trained for 1000 epochs with 4 dataloader workers. In addition, I also used a decay for weights of 0.005.

3.2 Ablation Study

Using pre-trained weights and larger models The biggest extension that this method brings to the benchmark is the use of pre-trained weights, this eases the training process as the model is already used to recognizing a number of features and can also lead to better performance especially in models pre-trained on large datasets, being able to recognize complex patterns. Also, the training time is reduced, not needing to compute from scratch if the task is similar, but only adapting the model to the context used (fine-tuning). In this case, we experimented with models that are pre-trained on the ImageNet dataset. We also tried to use a larger model, i.e. instead of MaxVit-tiny let's use the small version, which has 68.9 million parameters, double the previous version to see if a larger model can achieve better performance.

Splitting the images into two datasets Another idea we approached was that instead of using weights for the loss function, to use two separate datasets for class 0 images and class 1 images. Thus, we used stronger augmentations for those in class 1 to balance the dataset and, during training, we extracted an equal number of images from each dataset, resulting in a batch with an equal number of images from each class and solving the problem of unbalancing the dataset.

Ensemble model A classical method of improving the performance of a model involves creating a set of models trained on different folds (cross-validation). Thus, we re-partitioned the dataset into 5 folds with the same share of images from each of the 4 classes we identified and trained each fold separately with the same hyperparameters. At the end, we made inference using all 5 folds and decided the result by majority vote. This method is expensive in terms of time and memory resources, but leads to increased performance and is often used in competitions of this type.

Changing the prediction threshold Influenced by the unbalanced data, the model may tend to predict one class over another. Thus, using the SoftMax function, at the end a percentage is generated for each of the two classes. Implicitly, which one has the majority vote is considered the image label. To balance the problem of unbalanced training, we can lower this threshold so that the minority class needs a lower threshold to be elected. For our experiments, we chose a detection threshold for class 1 of 0.58 (we observed that despite it is the minority class, because of our improvements it tends to select class 1 more).

Other tried methods

- **Using 4 classes** As presented above, from the 2 classes that are annotated on the dataset we can derive 2 others using the day the image was taken. Thus, we tried to train using 4 classes (D3 healthy, D3 infertile, D5 healthy, D5 infertile), using 8 datasets and balancing the batches accordingly. Also, we tried a variant where we train one model for each of the days, each with two classes.
- **Model soup** Model soup [10] in machine learning refers to an ensemble technique where multiple independently trained variations of a model are combined. This approach often yields better performance than single models, leveraging diversity to reduce overfitting and enhance prediction robustness. We tried using the ensemble folds we trained to combine them into a model using this technique, thus reducing the inference time and the memory needed for it.

3.3 Comparison between methods

Compared to the benchmark version, we can see that using the enhanced version the training time decreases even if we use a larger model because more memory allows us to use a larger batch size (we increased it from 16 to 32) (of course at a cost in terms of memory). Also, the model for the enhanced version reaches a high score much faster because it uses pre-trained weights and can identify complex patterns much faster than the benchmark. In the case of inference, the execution time is similar even if the model is larger (not considering the variant where we use an assembly where the inference time is 5 times larger), both variants can run in a short time on the CPU.

4 Experimental Results

4.1 Comparation with State-Of-The-Art

The results obtained on the private score of the extended version competition using MaxVit-Small with batch matching using two datasets manage to compare with State-Of-The-Art, obtaining the same score (0.61538). For the public score, the best score we obtained is 0.75862 using an ensemble and modifying the threshold for model output prediction to 0.58, a score close to State-Of-The-Art (0.82758).

The table below shows the various approaches we have tried, with the public set and private set scores and where they would have ranked in the final rankings of the competition.

Method	Public Score	Private Score	Public Place	Private Place
SOTA public	**0.82758**	0.54545	**1**	16
SOTA private	0.69565	**0.61538**	29	**1**
ResNeSt benchmark	0.59459	0.48275	67	45
MaxViT benchmark	0.58823	0.54545	71	16
MaxViT extended	0.62857	**0.61538**	56	**1**
Ensemble - best fold	0.73333	**0.60869**	15	**2**
Ensemble - threshold 0.58	**0.75862**	0.40000	**7**	77
MaxViT 4 classes	0.55555	0.42857	79	73
ResNeSt 2 models	0.62857	0.52173	56	28
Model Soup	0.70967	**0.57142**	25	**10**

4.2 Discussion

It can be seen the large difference in score obtained for the private test dataset and the public test dataset in the competition, this is due to the fact that the former was used for the final evaluation once at the end of the competition, while the latter was used throughout the competition for evaluation. Thus, the participants overfitted on the public set, this was very noticeable when the final evaluation was done and the one who was at that time in 1st place dropped to 16th place and the winner moved up from 29th place.

At the same time, we can see that the model that does not use pretrain (MaxViT benchmark) manages to achieve a good score on the private dataset (0.54545) and has a score close to the one obtained on the public dataset (0.58823), so it generalizes well on both sets (compared to the existing results).

The two models involving training on multiple folds (ensemble and soup model) both obtain good results on the public dataset, although for the ensemble there is overfitting on the public dataset due to the change in threshold (this is explained by the fact that the submission was made during the run of the competition, when the private dataset evaluation was not possible).

5 Conclusions

The task proposed by the Embryo competition is a difficult one, mainly because of the unbalanced dataset and its small size. We tried different methods to solve this problem and the most effective ones turned out to be the use of weights for the loss function, the equalization of the batches as number of images in each class and the use of model combinations (ensemble and soup model).

Our results are comparable to State-Of-The-Art, scoring equal to State-Of-The-Art on the private evaluation set and equal to 7th place in the competition on the public evaluation set. These results can be improved by experimenting with other augmentations or with other models, and there is a possibility that smaller models than the ones we tried may perform better.

Comparing the results obtained by the benchmark models and the extended models, we observe that the first category manages to achieve comparable results for the private evaluation set, but does not come close to the performance on the public dataset, but it should be kept in mind that since the best score for the public evaluation set obtained by us was obtained during the competition to have unintentionally overfitted on it by changing the threshold. Even so the result obtained by the soup model is clearly superior to that obtained on all benchmark models, hence the need for resources to achieve a good result. If we were to compare the best result obtained on a single fold using pretrained, it can be seen that the difference between the benchmark and this one remains considerable, so we can conclude that pretrained weights improve model performance.

References

1. Carolyn Kay, M.: All about ivf embryo grading (2020), https://www.healthline.com/health/infertility/embryo-grading
2. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical data augmentation with no separate search. CoRR **abs/1909.13719** (2019), http://arxiv.org/abs/1909.13719
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR **abs/2010.11929** (2020), https://arxiv.org/abs/2010.11929
4. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. CoRR **abs/2010.01412** (2020), https://arxiv.org/abs/2010.01412
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
6. O'Neill, C.: Ivf cycle embryo development day-by-day - what are your embryos up to? (2022), https://fertilityspace.io/blog/ivf-cycle-embryo-development-day-by-day-what-are-your-embryos-up-to
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. CoRR **abs/1409.0575** (2014), http://arxiv.org/abs/1409.0575
8. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: MaxViT: Multi-Axis Vision Transformer. arXiv e-prints arXiv:2204.01697 (Apr 2022). https://doi.org/10.48550/arXiv.2204.01697
9. T.W.Sadler: Langman - Embriologie Medicală. Medicala CALLISTO (2019)
10. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., Schmidt, L.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time (2022)
11. Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Zhang, Z., Lin, H., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., Smola, A.J.: Resnest: Split-attention networks. CoRR **abs/2004.08955** (2020), https://arxiv.org/abs/2004.08955