Developmental Stage Classification of Embryos Using Two-Stream Neural Network with Linear-Chain Conditional Random Field

Stanislav Lukyanenko[1†], Won-Dong Jang[2], Donglai Wei[2], Robbert Struyven[2,5], Yoon Kim[6], Brian Leahy[2,3], Helen Yang[3,4], Alexander Rush[7], Dalit Ben-Yosef[9,10], Daniel Needleman[2,3,8], and Hanspeter Pfister[2]

[1] Department of Informatics, Technical University of Munich, Germany
{[2] School of Engineering and Applied Sciences, [3] Department of Molecular and Cellular Biology, [4] Graduate Program in Biophysics}, Harvard University, USA
[5] University College London, UK [6] MIT, USA [7] Cornell University, USA
[8] Center for Computational Biology, Flatiron Institute, USA
[9] Lis Maternity Hospital, Tel-Aviv Sourasky Medical Center, Israel
[10] Cell and Developmental Biology, Tel-Aviv University, Israel
stanislav.lukyanenko@tum.de

Abstract. The developmental process of embryos follows a monotonic order. An embryo can progressively cleave from one cell to multiple cells and finally transform to morula and blastocyst. For time-lapse videos of embryos, most existing developmental stage classification methods conduct per-frame predictions using an image frame at each time step. However, classification using only images suffers from overlapping between cells and imbalance between stages. Temporal information can be valuable in addressing this problem by capturing movements between neighboring frames. In this work, we propose a two-stream model for developmental stage classification. Unlike previous methods, our two-stream model accepts both *temporal* and *image* information. We develop a linear-chain conditional random field (CRF) on top of neural network features extracted from the temporal and image streams to make use of both modalities. The linear-chain CRF formulation enables tractable training of global sequential models over multiple frames while also making it possible to inject monotonic development order constraints into the learning process explicitly. We demonstrate our algorithm on two time-lapse embryo video datasets: i) mouse and ii) human embryo datasets. Our method achieves 98.1% and 80.6% for mouse and human embryo stage classification, respectively. Our approach will enable more profound clinical and biological studies and suggests a new direction for developmental stage classification by utilizing temporal information.

Keywords: Developmental Stage Classification · Linear-Chain Conditional Random Field · Time-lapse Video · Dynamic Programming.

[†] Works were done during the internship at Harvard University.

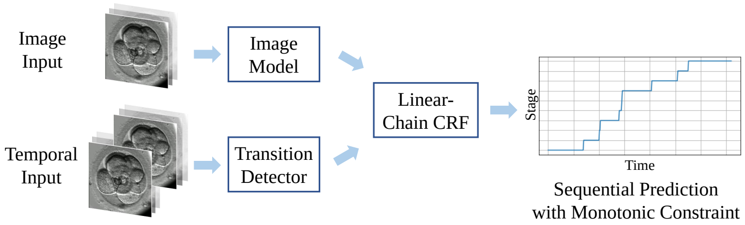


Fig. 1. Developmental Stage Classification of Embryo Time-Lapse Videos. Our two-stream model accepts the current and the previous frames as the input. We feed the current frame into the image model. For the transition detector, we input the concatenation of the current and the previous frames to capture motion information between them. We apply the two-stream model to all the frames in a video and obtain sequential predictions using a linear-chain CRF.

1 Introduction

Biological developments often follow a monotonic order. A mammalian embryo's developmental process is a typical example of the monotonic constraint, which develops through cell cleavages (from 1 cell to multiple cells), morula, and blastocyst. This monotonic constraint does not allow transitions to previous developmental stages, *e.g.*, a transition from 2 cells to 1 cell. Automated developmental stage classification can advance studying an embryo's cellular function, a basic but hard biological problem. Besides, developmental stage classification of embryos is important for *in vitro* fertilization (IVF). To achieve a pregnancy, clinicians select embryos with the highest viability and transfer them to a patient. Division timing is one of the main biomarkers to assess an embryo's viability [11]. The current standard of choosing the most promising embryos is a manual examination by clinicians via a microscope. However, manual inspection is time-consuming and prone to inter-person variability. As such, it is essential to develop a model for automated developmental stage classification.

In automated developmental stage classification for time-lapse videos, difficulties mainly come from overlaps between cells and imbalance between stages. Even though cells are transparent, their overlaps confuse a classifier when identifying their developmental stage. Also, a few developmental stages (*e.g.,* 1, 2 cells) dominate most of the frames in time-lapse videos, which can induce class imbalance in learning. Temporal information is valuable for addressing these two challenges. It can differentiate overlapping cells based on their movements and transitions between stages regardless of their frequencies. Existing developmental stage classification methods [11,9,12] usually classify per-frame stages and apply dynamic programming to make use of the monotonic constraints. However, they do not incorporate temporal information, potentially solving the overlap and imbalance problems. Besides, they do not include dynamic programming in the learning process, making classification models may not learn to maximize the accuracy of dynamic programming.

In this work, we propose a two-stream model for the developmental stage classification of embryos as displayed in Fig. 1. We first introduce a two-stream convolutional neural network (CNN) model, which consists of an image model and a transition detector. While the image model identifies a stage of the current frame, the transition detector returns a high value when the current frame has a different label compared to the previous frame. Unlike the previous methods, we exploit temporal information in our transition detector, which can better suppress the overlap and stage imbalance issues. We build a linear-chain conditional random field (CRF) [16] upon our two-stream model for the monotonic constraints. Unlike conventional methods, our method effectively combines two-stream outputs using linear-chain CRF and enables learning of sequential predictions while constraining the monotonic order. We demonstrate our algorithm's efficacy by comparing it with existing stage classification approaches on two time-lapse video datasets: i) mouse and ii) human embryos.

We have two main contributions. First, our method improves the performance for rare cell stages by combining image and temporal information in a two-stream model. Second, we inject the monotonic constraint into the learning process using linear-chain CRF to optimize the sequential predictions. Our code will be publicly available upon acceptance.

2 Related Work

Developmental Stage Classification of Embryos: Researchers have proposed many stage classification methods due to their importance for IVF. With the emergence of deep learning methods, most state-of-the-art methods rely on CNN. Khan *et al.* [9] adopt CNN for human embryonic cell counting over the first five cell stages. Ng *et al.* [12] introduce late fusion nets, where multiple images are input for CNN, and additionally exploit dynamic programming to ensure a monotonical progression over time. Lau *et al.* [10] detects a region of interest and uses LSTM [5] for sequential classification. Rad *et al.* [13] use CNN to parse centroids of each cell from embryo images. Recently, Leahy *et al.* [11] develop computer vision models that extract five key morphological features from time-lapse videos, including stage classification. They improve a baseline by using multiple focuses, a soft loss, and dynamic programming.

However, most previous methods focus on improving a per-frame prediction and utilize dynamic programming during testing to incorporate the monotonic development order constraint. In this work, we make use of temporal information and directly inject the monotonic condition into the learning process with CRFs for sequential stage prediction.

Two-Stream Models: Researchers widely use two-stream models for action recognition. Two-stream 2D CNN [15] classifies an action by averaging predictions from image and motion branches. 3D-fused two-stream [4] blends features from image and motion information using 3D convolution. I3D [1] replaces 2D convolutions in the two-stream 2D CNN to 3D convolutions to incorporate temporal information better.

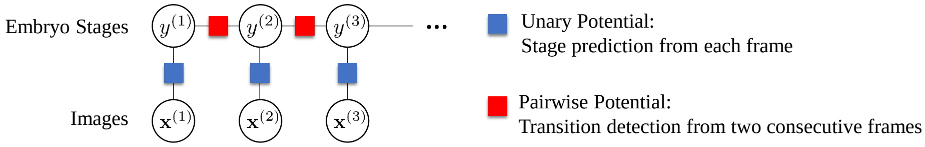


Fig. 2. Linear-Chain CRF Model. For each image, we compute the unary potential using a image model. For pairwise ones, we use predictions from a transition detector.

In their two-stream models, image and motion branches' objectives are the same; predicting an action from the input video. However, the embryo's temporal information could be useful for detecting stage transition timing rather than stage classification. Besides, their architectural designs are for action recognition, which outputs a per-video prediction. Since embryo stage classification requires per-frame classification, the previous two-stream models may not fit sequential prediction. For sequential prediction, one may use recurrent neural networks, *e.g.*, long short-term memory [5]. However, it is hard to incorporate the monotonic constraint of embryo development. Instead, we adopt a linear-chain CRF [16] to encode the constraints.

3 Model

We construct a two-stream approach for the developmental stage classification of embryos. The input is a sequence of frames $X = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}]$, and the output is a sequence of stage predictions $Y = [y^{(1)}, \ldots, y^{(T)}]$. As depicted in Fig. 2, we use features extracted from the two-stream model as input to a linear-chain conditional random field, where the unary potentials are from the image stream, and the pairwise potentials are from the temporal stream. Our parameterization of the pairwise potentials (to be explained below) makes it possible to incorporate the monotonic constraint into the learning process. The entire model is trained end-to-end.

3.1 Two-Stream Feature Encoding

Our model uses temporal information in addition to image data to address the problems of overlapping cells and imbalance between stages, in contrast to many prior works, which often only use image information [9,10,11]. While the temporal information may not be valid for stage classification, it can be useful when there is a stage transition between two frames. To make use of this, we adopt a two-stream approach, which consists of an *image model* and a *transition detector*. While the image model outputs scores (*i.e.,* unary potentials) for each frame's stage, the transition detector outputs transition scores (*i.e.,* pairwise potentials) that recognize the existence of a stage transition between two consecutive frames.

The image model infers a stage from an input frame using ResNet50 [6] pretrained on the ImageNet dataset [3]. Concretely, the unary potential for class $c \in \{1, \ldots, C\}$ (*i.e.,* there are C possible stages) at time step t is given by,

$$\Phi_{\mathrm{I}}(\mathbf{x}^{(t)}; \theta_{\mathrm{I}})_c = \mathsf{softmax}\left(\mathbf{W}_{\mathrm{I}}\, \mathsf{ResNet}(\mathbf{x}^{(t)}) + \mathbf{b}_{\mathrm{I}}\right)_c,$$

where $\mathbf{W}_{\mathrm{I}}, \mathbf{b}_{\mathrm{I}}$ are the parameters of the linear layer that outputs class scores from ResNet features, and the $\mathsf{softmax}(\cdot)$ function normalizes the output to turn them into probabilities.[1]

Our transition detector outputs a score for whether the current frame is in a different stage compared to the previous frame. Even though many two-stream methods [4,15,1] exploit optical flow [7] as temporal information, it cannot distinguish stage transition from cell movements. Hence, we feed two consecutive frames into the detector instead. For the transition detector, we use ResNet50 [6] also pretrained on the ImageNet dataset [3], but we modify the first convolution layer to make it accept two consecutive frames as the input, $\mathbf{x}^{(t-1)}$ and $\mathbf{x}^{(t)}$. The detector returns a probability of stage change existence defined as,

$$\rho_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_k = \mathsf{softmax}\left(\mathbf{W}_{\mathrm{M}}\, \mathsf{ResNet}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) + \mathbf{b}_{\mathrm{M}}\right)_k,$$

where $k \in \{0, 1\}$ indicates whether there was a stage change between $\mathbf{x}^{(t-1)}$ and $\mathbf{x}^{(t)}$. The detector also implicitly parameterizes the pairwise potentials via,

$$\Phi_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_{(c,c')} = \begin{cases} \rho_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_0, & \text{if } c = c' \\ \rho_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_1, & \text{if } c < c' \\ -\infty, & \text{otherwise,} \end{cases}$$

where we penalize inverse transitions with $-\infty$ to incorporate the monotonic constraint. We use these potentials as input to the linear-chain CRF, which enables sequential classification of the input sequences taking into account the pairwise correlations that exist among the output labels.

3.2 Linear-Chain Conditional Random Field

We define a probability distribution over the output sequence Y given the input sequence X with a linear-chain CRF

$$p(Y|X; \theta_{\mathrm{I}}, \theta_{\mathrm{M}}) = \frac{1}{Z(X)} \prod_{t=1}^{T} \exp\left\{\Phi(y^{(t-1)}, y^{(t)}, \mathbf{x}^{(t)}; \theta_{\mathrm{I}}, \theta_{\mathrm{M}})\right\},$$

[1] Since the potentials in a CRF do not need to be probabilities, normalization via the softmax function is not strictly necessary. However, we found the normalization to be helpful for stable training. Note that if the unary potential is defined to be the output of a *log*-softmax function (which is not the case in our approach), the model will reduce to a Maximum Entropy Markov Model.

where $\varPhi$ is a score for transitioning from $y^{(t-1)}$ to $y^{(t)}$, which is given by combining the unary and pairwise potentials from above,

$$\varPhi(y^{(t-1)}, y^{(t)}, \mathbf{x}^{(t)}; \theta_{\mathrm{I}}, \theta_{\mathrm{M}}) = \varPhi_{\mathrm{I}}(\mathbf{x}^{(t)}; \theta_{\mathrm{I}})_{y^{(t)}} + \varPhi_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_{(y^{(t-1)}, y^{(t)})}.$$

Here $Z(X)$ is a normalizing constant,

$$Z(X) = \sum_{y^{(1)}=1}^{C} \dots \sum_{y^{(T)}=1}^{C} \prod_{t=1}^{T} \exp\left\{\varPhi(y^{(t-1)}, y^{(t)}, \mathbf{x}^{(t)}; \theta_{\mathrm{I}}, \theta_{\mathrm{M}})\right\},$$

which can be calculated in $O(TC^2)$ with dynamic programming.

Training: During training, we also found it helpful to minimize the CRF negative log likelihood along with single-model losses derive from the image and transition models. The single-model loss for the image model is defined as,

$$\mathcal{L}_{\mathrm{I}} = \sum_{c=1}^{C} -q_c^{(t)} \log \varPhi_{\mathrm{I}}(\mathbf{x}^{(t)}; \theta_{\mathrm{I}})_c,$$

where $\mathbf{q}^{(t)}$ is the one-hot representation of the ground truth stage at frame t, and the single-model loss for the transition detector is defined as,

$$\mathcal{L}_{\mathrm{M}} = \sum_{k=0}^{1} -(1 - \mathbf{q}^{(t-1)^T} \mathbf{q}^{(t)}) \log \rho_{\mathrm{M}}(\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}; \theta_{\mathrm{M}})_c.$$

Thus the final loss is given by,

$$-\log p(Y|X; \theta_{\mathrm{I}}, \theta_{\mathrm{M}}) + \mathcal{L}_{\mathrm{I}} + \mathcal{L}_{\mathrm{M}},$$

and we perform end-to-end training with gradient-based optimization using the Torch-struct library [14].[2] We use a batch size of four and a learning rate of 0.0001 with the Adam optimizer. To construct a batch, we randomly sample 50 frames from each video and then sort them in a consecutive order. We also perform data augmentation by random resized cropping, rotation, and flipping.

Inference: For prediction, our aim is in obtaining the most likely sequence of labels given a new test video X, *i.e.*,

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, p(Y|X; \theta_{\mathrm{I}}, \theta_{\mathrm{M}}).$$

We obtain this maximum a posteriori sequence with standard dynamic programming (*i.e.,* the Viterbi algorithm). At inference time only, we also smooth the unary potentials from the image model by modifying the potential for class c,

$$\frac{1}{13} \cdot \varPhi_{\mathrm{I}}[c-2] + \frac{3}{13} \cdot \varPhi_{\mathrm{I}}[c-1] + \frac{5}{13} \cdot \varPhi_{\mathrm{I}}[c] + \frac{3}{13} \cdot \varPhi_{\mathrm{I}}[c+1] + \frac{1}{13} \cdot \varPhi_{\mathrm{I}}[c+2],$$

and using the above weighted average as the input to the Viterbi algorithm. This reweighting values, which were found via a search on the validation set, take into account the ordinal nature of the output space (boundaries are zero-padded).

[2] The single-model losses and the CRF negative log likelihood are complementary each other by taking into account local and global predictions, respectively.

Table 1. Accuracies (%) of stage classification methods on the test mouse embryos [2].

Method	Global	Per-Stage	1	2	3	4	5	6	7	8
A. Per-Image Classification Model										
ResNet50 [6]	90.9±0.6	59.4±1.2	99.0	98.3	89.1	90.7	25.1	15.8	26.5	30.4
AutoIVF [11]	96.4±0.1	60.9±3.1	99.8	**99.9**	92.3	**99.9**	4.3	33.5	50.8	6.2
B. Spatiotemporal Classification Model										
Early Fusion [8]	91.9±0.1	57.1±0.2	98.8	**99.9**	74.0	93.2	0.0	14.9	37.9	**37.6**
LSTM [10]	98.0±0.1	45.0±1.7	96.9	98.7	22.9	42.2	0.0	9.2	**90.1**	0.0
Ours	**98.1±0.3**	**76.8±5.4**	**99.9**	**99.9**	**94.9**	**99.9**	**35.1**	**84.6**	71.4	29

4 Experimental Results

We evaluate our method's performance, demonstrating each design choice's effect in the models with ablation studies. We evaluate stage classification algorithms on two embryo datasets: i) mouse and ii) human embryo datasets.

Compared Methods: We compare our method with a general classification model, ResNet50 [6], one state-of-the-art embryo stage classification method, AutoIVF [11], an early fusion method [8] that leverages temporal information, and a sequential model [10] based on LSTM. For a fair comparison, we re-implement AutoIVF using a single focus and the same backbone as ours. The early fusion takes five successive frames as input and learns to predict the middle frame's stage. We adopt the PyTorch 1.7 library to implement all the methods.

Evaluation Metric: We evaluate classification accuracy as the number of correct predictions over the number of data (Global). Since the majority stages, such as 1 cell and 2 cells, can dominate the average accuracy, we calculate the per-stage accuracies and the mean of them (Per-Stage). We train all methods for five seeds and report their average performances with standard deviations.

4.1 Developmental Stage Classification of Mouse Embryos

Dataset: We use the NYU Mouse Embryo dataset consisting of 100 videos of developing mouse embryos [2]. The videos contain 480 x 480 resolution images taken every seven seconds, with a median of 314 frames per embryo, totaling an average length of 36.6 minutes per embryo. The videos have frames with up to 8 cells, *i.e.,* eight developmental stages. For training and evaluation, we randomly split the data 80/10/10 into train, validation, and test videos, respectively. We use the validation set to select hyper-parameters and models for evaluation.

Result: In Table 1, we list overall and per-stage classification performances of the embryo stage classification methods. Our method outperforms all other methods on average for various stages. The frequency imbalance between the stages allows LSTM to achieve comparable results on average over all the data.

Table 2. Scores (%) of stage classification methods on the test human embryos [11].

Method	Global	Per-Stage	1	2	3	4	5	6	7	8	9+	M	B
A. Per-Image Classification Model													
ResNet50 [6]	74.6±1.0	58.2±1.6	97.6	93.8	24.3	80.8	24.1	16.2	19.8	55.2	63.5	70.7	93.9
AutoIVF [11]	77.8±1.2	60.9±2.2	98.2	**96.6**	22.9	88.2	26.5	15.6	22.4	59.3	67.3	77.0	96.1
B. Spatiotemporal Classification Model													
Early Fusion [8]	75.1±0.6	55.7±0.7	97.5	93.4	10.2	84.5	11.5	7.9	12.8	63.5	65.7	72.5	93.7
LSTM [10]	77.1±0.9	61.8±0.9	97.8	92.7	31.4	79.9	21.4	25.4	**28.8**	58.3	**67.6**	**79.3**	**97.0**
Ours	**80.6±0.7**	**66.3±1.9**	**99.4**	96.2	**41.2**	**89.4**	**43.3**	**27.6**	19.7	**69.8**	67.0	78.7	96.7

4.2 Developmental Stage Classification of Human Embryos

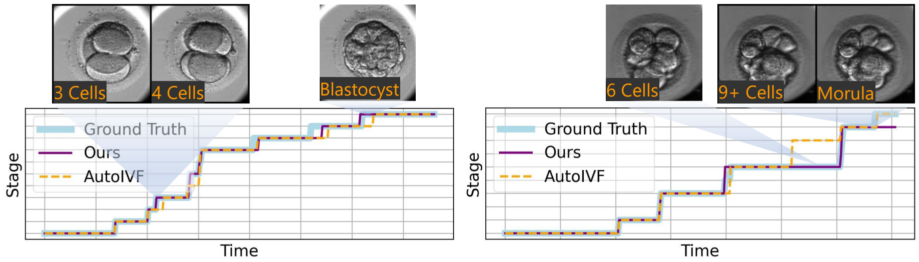
Dataset: We evaluate the stage classification methods on the human embryo dataset [11]. There are 13 stage labels: empty well, 1 cell to 9+ cells, morula (M), blastocyst (B), and degenerate embryo. To focus on the embryo development's monotonicity, we only use 11 stages, excluding frames with the empty well and degenerate embryo labels. The dataset includes 341 training, 73 validation, and 73 test time-lapse videos of embryos. Each video consists of 325 frames on average. As the network input, we crop zona-centered patches from each frame to exclude outside regions of interest and resize the frames to 112 × 112 resolution.

Result: Table 2 benchmarks the developmental stage classification methods. Overall, our approach surpasses the other classification methods. In terms of the mean per-stage accuracy, the performance gain over the existing methods is much higher, which indicates our method notably performs better for rare developmental stages. Since we incorporate the transition detector and use it to force the predictions of our model to be monotonic, our method outperforms the two spatiotemporal methods; Early Fusion and LSTM. Unlike AutoIVF, our model learns the features for the stage change detection, which are helpful for the monotonic predictions.

Our method runs in 268 frames per second on a single TITAN X GPU. Our model has 47M parameters and requires up to 4 GB GPU memory in the inference phase. Fig. 3 visually compares our method with AutoIVF [11]. Our method is better at detecting cell division timings. As one example of failure cases, our model fails to detect the transition between 9+ cells and morula in Fig. 3 (b) since it takes two consecutive frames as the input, which visually have no major difference in this example.

4.3 Ablation Study

We analyze our method's efficacy by conducting an ablation study on the human embryo dataset. To this end, we add one of our components to the baseline at a time. By performing dynamic programming without pairwise potentials, our model improves the baseline's accuracy from 76.7% to 80.0%. Using both unary and pairwise terms in linear-chain CRF, our two-stream model yields 80.6% score, which performs the best. In conclusion, our full setting enables the maximum performance for developmental stage classification.



(a) AutoIVF: 86.6% / Ours: 95.3%

(b) AutoIVF: 85% / Ours: 92.9%

Fig. 3. Qualitative Stage Classification Results. We visualize frames with ground truths (lower left corner), where our method and AutoIVF [11] predict different stages.

5 Conclusion and Future Work

Our method will enable better clinical and embryological studies by improving the accuracies on rare stages, which are infrequent in videos but equally important as frequent stages when analyzing embryos. Since we measure stage transition probabilities, cell division timings predicted by our method are highly interpretable, which will allow tractable inspection in clinical practice. Our future work includes further improving performance on rare stages by combining a stage classifier and a cell detector, developing sequential models for other developmental features of embryos, and experimenting with different ways of acquiring unary and pairwise potentials, *e.g.,* calculating the transition probability over longer sliding windows of frames.

Acknowledgements

This work was funded in part by NIH grants 5U54CA225088 and R01HD104969, NSF Grant NCS-FO 1835231, the NSF-Simons Center for Mathematical and Statistical Analysis of Biology at Harvard (award number 1764269), the Harvard Quantitative Biology Initiative, and Sagol fund for studying embryos and stem cells; Perelson Fund.

References

1. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
2. Cicconet, M., Gutwein, M., Gunsalus, K.C., Geiger, D.: Label free cell-tracking and division detection based on 2D time-lapse images for lineage analysis of early embryo development. Computers in biology and medicine **51**, 24–34 (2014)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

4. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1933–1941 (2016)
5. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. IET (1999)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981)
8. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)
9. Khan, A., Gould, S., Salzmann, M.: Deep convolutional neural networks for human embryonic cell counting. In: European conference on computer vision. pp. 339–348. Springer (2016)
10. Lau, T., Ng, N., Gingold, J., Desai, N., McAuley, J., Lipton, Z.C.: Embryo staging with weakly-supervised region selection and dynamically-decoded predictions. In: Machine Learning for Healthcare Conference. pp. 663–679. PMLR (2019)
11. Leahy, B.D., Jang, W.D., Yang, H.Y., Struyven, R., Wei, D., Sun, Z., Lee, K.R., Royston, C., Cam, L., Kalma, Y.: Automated Measurements of Key Morphological Features of Human Embryos for IVF. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 25–35. Springer (2020)
12. Ng, N.H., McAuley, J.J., Gingold, J., Desai, N., Lipton, Z.C.: Predicting Embryo Morphokinetics in Videos with Late Fusion Nets & Dynamic Decoders. In: ICLR (Workshop) (2018)
13. Rad, R.M., Saeedi, P., Au, J., Havelock, J.: Cell-Net: Embryonic Cell Counting and Centroid Localization via Residual Incremental Atrous Pyramid and Progressive Upsampling Convolution. IEEE Access **7**, 81945–81955 (2019)
14. Rush, A.M.: Torch-struct: Deep structured prediction library. arXiv preprint arXiv:2002.00876 (2020)
15. Simonyan, K., Zisserman, A.: Two-Stream Convolutional Networks for Action Recognition in Videos. In: Neural Information Processing Systems (2014)
16. Sutton, O.: Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction. University lectures, University of Leicester **1** (2012)