Deep Convolutional Neural Networks for Human Embryonic Cell Counting

Aisha Khan[1], Stephen Gould[1], and Mathieu Salzmann[1,2]

[1] College of Engineering and Computer Science
The Australian National University, Canberra, AU
[2] CVLab, EPFL, Lausanne, Switzerland

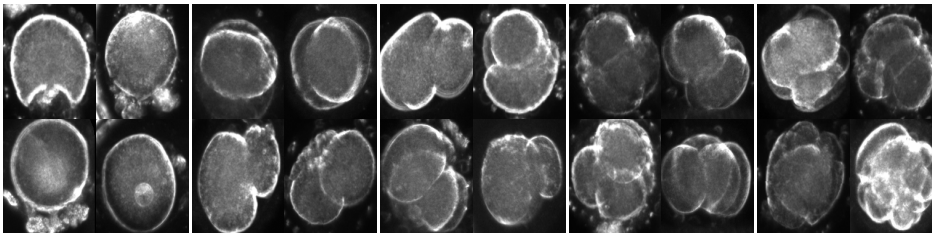
Abstract. We address the problem of counting cells in time-lapse microscopy images of developing human embryos. Cell counting is considered as an important step in analyzing biological phenomenon such as embryo viability. Traditional approaches to counting cells rely on hand crafted features and cannot fully take advantage of the growth in data set sizes. In this paper, we propose a framework to automatically count the number of cells in developing human embryos. The framework employs a deep convolutional neural network model trained to count cells from raw microscopy images. We demonstrate the effectiveness of our approach on a data set of 265 human embryos. The results show that the proposed framework provides robust estimates of the number of cells in a developing embryo up to the 5-cell stage (i.e., 48 hours post fertilization).

1 Introduction

Counting the number of objects in an image is an important and challenging computer vision problem that arises in many real-world applications ranging from crowd monitoring to biological research. In biological research counting cells is a fundamental first step for further analysis (e.g., cell mitosis detection and cell lineage analysis). In this paper we focus on the problem of determining the number of cells in time-lapse microscopy images of developing human embryos. We are primarily interested in images of embryos up to the 5-cell stage, which have been used in other works for computing biomarkers (e.g., cell timing parameters) to assess embryo viability in the context of *in vitro* fertilization (IVF) treatments [14, 19].

Manual cell counting, is an extremely tedious process that is prone to error and subject to intra- and inter-individual variability. Automating the process has the benefit of reducing time and cost, minimizing errors, and improving consistency of results between individuals and clinics. To simplify the task and improve robustness, many researchers stain the cells prior to automatic counting [1, 4, 5, 20]. However, cell staining is not feasible for many applications (such as IVF embryo assessment).

Counting non-stained cells in dark-field microscopy images is difficult because of constraints in the imaging process. For example, the exposure time, the light intensity and the transparency of the specimen all cause variations in the image



(a) (b) (c) (d) (e)

Fig. 1. Examples of developing embryos: (a) one-cell stage, (b) two-cell stage, (c) three-cell stage, (d) four-cell stage and (e) 5-or-more cell stage.

quality and result in faint cell boundaries. Analysis of human embryonic cells is further challenged by the fact that the cells exhibit variability in appearance and shape. Also, each embryo grows (cells undergo divisions) in a compact manner where cells severely overlap with each other. Moreover, cells are surrounded by distracting noise such as extra cellular material (fragments) attached to the growing embryo and surrounding gel material (see Fig. 1 and Fig. 2 (e)–(f) for examples). All these difficulties make hand crafted algorithms for automated cell counting brittle.

In this paper, we utilize non-stained dark-field microscopy images of developing human embryos. Our goal is to automatically count the number of cells in the developing embryos up to the 5-cell stage (with higher cardinality being grouped into a "5-or-more" category). We do so using a convolutional neural network (CNN) that can learn from vast amount of training data to overcome the difficulties presented above. Our network significantly outperform the previous state-of-the-art is this task.

2 Related Work

Counting objects in images is typically achieved by training either an object detector or directly a counter. Detection-based methods first localize the individual objects in the input image and then simply count them. In the context of cell biology, this strategy was employed in [5, 7, 15]. One of the main drawbacks of this approach, however, is that it requires training data labeled with the locations of the objects of interest, which is time-consuming and expensive to acquire, if possible at all. A different, yet related approach, consists of predicting object density instead of precise object locations. This was proposed by Lempitsky and Zisserman [13] in the general context of computer vision and by Xie et al. [20] for microscopy images. While effective, this again suffers from the fact that it requires detailed annotations, if not of the objects themselves, then of their local density.

By contrast, directly learning a counter bypasses the hard detection problem and reduces labelling cost by only requiring the true number of objects in

each training image. When the number of such objects can be arbitrarily large, and thus not all possible numbers can be observed during training, such as for people counting in crowds [21], regression approaches are typically best-suited. By contrast, when the number of object instances that can appear in an image is small, classification becomes the method of choice. This approach has recently become popular in the context of human embryonic cell counting. For example, Wang et al. [18] proposed a 3-level classification method to predict the embryo cell stage without performing detection. In Khan et al. [8], a conditional random field (CRF) framework was developed to count the cells in sequences of microscopy images. Thanks to their accuracy, these methods have become a key ingredient in cell detection and early-stage embryo development analysis algorithms [7, 9, 15]. Following their success, in this paper, we introduce a method to directly count the number of cells in an early-stage embryo without requiring any detection phase.

Traditional counting methods rely on hand crafted feature descriptors [7, 8, 9, 15, 18]. Recently, however, learning features via deep convolutional neural network (CNN) models [12] has proven highly effective for a variety of tasks, such as recognizing handwritten digits [3], handwritten characters [2], faces [17] and natural images [11], and, in the biomedical domain, detecting cell mitosis [1, 4]. CNNs were also recently applied to the problem of cell counting [16, 20]. In both cases, however, the networks were trained to perform in well-controlled environments, with clean background and little cell overlap on synthetic images. In practice, however, and in particular in the case of human embryos, the images background contains a lot of noise, such as fragments, and the cells of the growing embryo greatly overlap each other.

In this paper, we therefore address the cell counting problem in this challenging scenario. To this end, we introduce a CNN-based counting approach that requires minimal annotations, i.e., only the number of cell in each image. Furthermore, we incorporate temporal information via a CRF, which smoothes the individual CNN predictions across the entire sequence. Our experiments demonstrate that our approach outperforms, by a large margin, the state-of-the-art method on the challenging task of counting cells in early stage human embryo development.

3 Deep Learning Based Cell Counter

Our goal is to count cells directly from the microscopy images of developing human embryos. We formulate cell counting as a classification problem and employ an end-to-end deep CNN framework. The main objective for our cell counting model is to learn a mapping $F : \mathbb{R}^{m \times n} \rightarrow \mathcal{L}$, where $m \times n$ is the size of the microscopy image and $\mathcal{L} = \{1, \ldots, N^{max}\}$ is the cell cardinality of the image, with the last label corresponding to $N^{\max}$-or-more cells in the embryo. In practice, we use $N^{\max} = 5$.

Our framework uses raw pixel intensity as input and, in contrast to previous cell counting approaches [5, 13, 20], only uses the cell count in each training image

as annotation, without requiring any information about the objects shape, size and location. In this setting, our goal is to learn a mapping from an input image to the number of cells in this image, which we propose to do with a CNN, as discussed below.

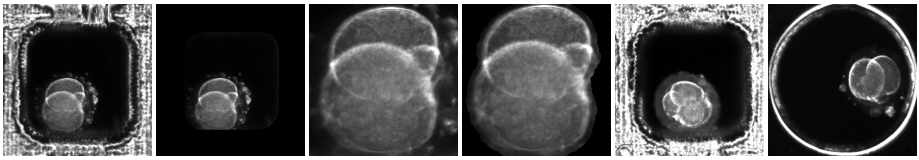
3.1 CNN-based Embryonic Cell Counting

In this work, we make use of images of developing embryos that were acquired using the *Eeva System*^{TM}, for capturing microscopy images developed by Auxogyn, Inc. Embryos are placed in a petri dish inside an incubator and image acquisition software acquires a single-plane image every five minutes over a five day period. Below, we first present our approach to obtaining the training data and then discuss our CNN framework.

Computing embryo bounding boxes. The microscopy images that we used as input contain a well boundary, which we remove by applying a pre-calculated boundary mask (see Fig. 2(b) for boundary-removed image). These images also contain extracellular material and noise that could easily confuse a classifier. To reduce the impact of this noise, we introduce a fully automatic approach to select a region of interest by computing a bounding box that encloses the embryo. To this end, we first determine the largest connected component in the thresholded boundary-removed intensity image and compute the centroid of this component. Each image contains one embryo only (in our application only one embryo is grown per well as shown in Fig. 2), so the centroid can be computed as the point within the component with maximum shortest distance to the region boundary [10]. We then crop the image around this centroid to obtain an image of size 151×151 pixels. The result is shown in Fig. 2(c). The dimension of the bounding box reflects the size of a fully developed embryo and is determined by the known optical setup of the image acquization system. After processing all the training images in this manner, we normalize the results by subtracting the mean intensity taken over the whole dataset. The same mean is subtracted from test images.

Cell counting is invariant to rotation. Therefore, we generate additional training instances by applying arbitrary rotations and mirroring to the original data. This reduces overfitting and, as shown in our experiments, improves accuracy.

Our CNN framework. We follow the architecture of Krizhevsky et al. [11]. Our CNN model contains eight layers (five convolutional ones and three fully connected ones). The first convolutional layer (Conv1) filters the 151×151 input image with 96 kernels of size 11×11. Conv2 has 256 filters of size 5×5, Conv3 and Conv4 have 384 filters of size 3×3, and the last Conv5 layer has 256 filters of size 3×3. The fully connected layers have 4096 neurons each with 50% dropout ratio used during training. Max pooling layers with a 3×3 kernel size are used after Conv1, Conv2 and Conv5. We employ a Rectified Linear Unit (ReLU) activation function after every convolutional and fully connected layer.



(a) (b) (c) (d) (e) (f)

Fig. 2. Example of (a) raw intensity image; (b) boundary-removed image; (c) cropped image; and (d) masked cropped image [10]. Notice that the noise at the right of the image (c) is removed in (d) but that some boundary pixels are also affected. Also shown is (e) surrounding gel material; and (f) fragments attached to the embryo.

Furthermore, we apply local response normalization to Conv1 and Conv2. The output layer encodes an ($N^{\max} = 5$)-way classification problem for which we use a softmax function to produce a distribution over the five class labels. The network maximizes the multinomial logistic log loss of the softmax output.

We use the Caffe [6] package to implement our model. All the configuration settings are standard other than those specified here. In particular, we initialize the learning rate to 0.005 and decrease it by a factor of 10 after half of the iterations. The momentum term is set to 0.9, and the weight decay to 0.0005. The network is trained for 5,000 iterations, taking three hours on an Nvidia K40 GPU. These parameters are obtained by cross-validation.

3.2 Enforcing Temporal Constraints

The CNN framework described above provides a cell count for each frame individually. However, in sequential data, as ours, adjacent frames are more likely to have the same number of cells. Considering neighboring frames can thus further improve the cell count over a complete time-lapse sequence. Following the work of Khan et al. [10], we capture this by making use of a conditional random field (CRF) with a unary and pairwise terms.

Formally, we represent the number of cells at time t with a discrete random variable N_t. Each variable N_t for $t \in \{1, \ldots, T\}$ can take a label from the set $\mathcal{L}$=$\{1, \ldots, N^{\max}\}$. Let $N_t \in \mathcal{L}$ denote the label assigned to frame t. Then we can define the energy of a complete labeling over all frames as

$$E(N_1, \ldots, N_T) = \sum_{t=1}^{T} \psi_t^{\mathrm{U}}(N_t) + \sum_{t=1}^{T-1} \psi_{t,t+1}^{\mathrm{P}}(N_t, N_{t+1}), \tag{1}$$

where the unary term (ψ_t^{U}) represents a score for each cell count in each frame and is obtained from the log of the softmax output of our CNN-based cell counter. The pairwise term (ψ_t^{P}) enforces consistency between neighboring frames by imposing a biological constraint encoding the fact that the cell count should be monotonically non-decreasing over time and is expressed as

$$\psi_{t,t+1}^{\mathrm{P}}(N_t, N_{t+1}) = \begin{cases} 0, & \text{if } N_t \leq N_{t+1} \\ \infty, & \text{if } N_t > N_{t+1} \end{cases}. \tag{2}$$

Table 1. Cell stage prediction performance. Here, the average is computed as the arithmetic mean of the one to five cell predictions, and the overall represents the fraction of correct instances.

Experiments	Cell Stage Prediction (%)						
	1-cell	2-cell	3-cell	4-cell	5-cell	Avg.	Overall
RawImg	98.48	89.36	60.44	77.33	89.08	82.94	87.36
RawImgCRF	98.87	94.18	63.07	81.67	91.11	85.78	90.25
CroppedImg	99.19	94.51	70.41	82.90	92.89	87.98	91.45
CroppedImgCRF	**99.48**	**97.60**	**73.91**	**87.30**	**95.23**	**90.70**	**94.05**
Khan et al. [8]	95.66	87.93	23.91	65.91	73.12	69.31	77.93

We search for the most likely number of cells in each frame, and ultimately the most likely sequence. This corresponds to the assignment that minimizes $E(N_1, \ldots, N_T)$, which can be obtained efficiently by dynamic programming.

4 Experiments

We evaluated our approach on 265 time-lapse image sequences consisting of a total of 148,993 frames (with 21%, 24%, 4%, 23%, 28% of samples for 1 to 5-or-more cell cardinality, respectively). The sequences capture the embryos from 53 different patients and show a high degree of variation, such as extra cellular material artifacts and cell reabsorption. We used a 10-fold cross validation strategy and report the cell stage prediction accuracy, computed as the percentage of frames where the correct number of cells was predicted. We compare the results of our approach against the cell counting results of Khan et al. [8]. To obtain ground-truth, we manually annotated the sequences with the number of cells in each frame.

Table 1 compares the results of our approach with different types of input, with and without using the CRF, and against those of the state-of-the-art method of Khan et al. [8]. Note that, independently of the kind of input and of the use of a CRF, our approach always significantly outperforms Khan et al. [8]. In particular, our basic CNN cell counter, with raw intensity images as input, yields on average 13.63% and overall 9.43% improvement over Khan et al. [8]. The performance further improves by 5.04% on average and 4.09% overall by training our CNN with images cropped around the bounding box. The use of a CRF yields additional boost of roughly 3% both in average and overall accuracy. We note that the computational cost of the CRF is minimal and that our running time is well within the 5-minute interval between frames. Analyzing the performance for each cardinality reveals that our approach can reliably predict the correct number of cells up to the 5-cell stage. The improvement over the state-of-the-art method of Khan et al. [8] is highest in the 3-cell case (from 23.91% to 73.91%), which is the most challenging one due to the small amount of data available for this stage. Note that this lack of data also explains why

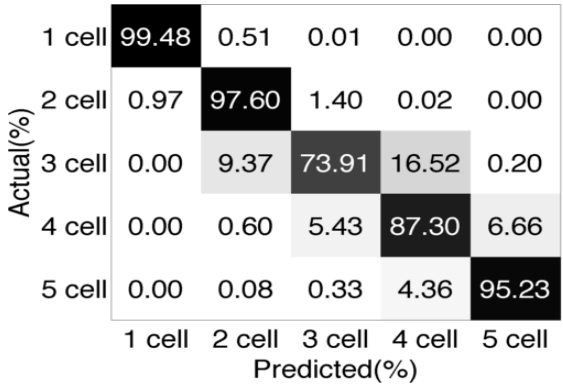


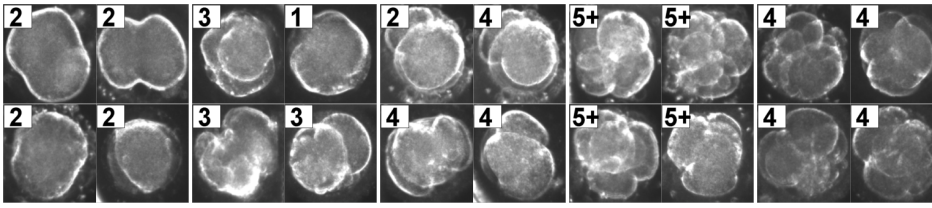
Fig. 3. Confusion matrix (CroppedImgCRF) %.

Table 2. Ablation analysis. Here, the average is computed as the arithmetic mean of the one to five cell predictions, and the overall represents the fraction of correct instances.

Experiments	(%) Avg.	Overall
CroppedImg w/o aug	80.32	85.82
CroppedImg w/o augCRF	83.65	89.08
CroppedImg w Mirror	83.32	87.79
CroppedImg w MirrorCRF	86.31	91.15
BRmImg	84.54	88.20
BRmCRF	88.09	92.12
MaskedImg	85.98	90.71
MaskedImgCRF	88.98	93.26

the 3-cell stage remains comparatively lower than the other stages even with our approach. The confusion matrix in Fig. 3 shows that our errors typically occur with adjacent classes. We show some error cases in Fig. 4.

In Table 2, we analyze the influence of several components of our method via an ablation study. To this end, we first evaluate the impact of data augmentation. Our results show that augmenting the data by mirroring and rotation (CroppedImg vs CroppedImg w/o aug) increases the overall performance by 5.63%. In particular, we observed a substantial improvement in the 3-cell case (17.62%), which, as mentioned above suffers from data scarcity. Note that only using mirroring augmentation (CroppedImg w Mirror and CroppedImg w MirrorCRF) improves over no augmentation at all, but still does not reach the accuracy when using rotation and mirroring. In addition to data augmentation, we also study the influence of our image pre-processing steps, described in the method section, on our results. In particular, we observed an improvement of 0.84% by training the CNN with the boundary-removed images (RawImg vs.



(a) (b) (c) (d) (e)

Fig. 4. Examples of cell counting error with CroppedImgCRF variant for 1-cell stage (a), 2-cell stage (b), 3-cell stage (c), 4-cell stage (d) and 5-or-more cell stage (e). Here, predicted cell count is mentioned on top left of each image.

Table 3. Cell detection analysis. Here, the average is computed as the arithmetic mean of the one to four cell predictions, and the overall represents the fraction of correct instances.

	Cell Stage Prediction (%)						
Experiments	1-cell	2-cell	3-cell	4-cell	5-cell	Avg.	Overall
Khan et al. [9]	96.89	89.77	16.48	49.37	65.32	63.57	72.05
detection w CroppedImgCRF	100.00	99.47	89.20	76.06	90.54	91.05	92.18

BRmImg). As a pre-processing step, the method of Khan et al. [8] subtracts the background from the images by performing embryo segmentation [10] (see Fig. 2 (d)). To compare the impact of this background subtraction to our simpler bounding boxes, we trained a CNN with such background subtracted images. This resulted in performance drop of 0.74% (MaskedImg vs. CroppedImg). The drop can be explained by the errors in the embryo masks, which occasionally include background and exclude foreground. These results suggest that our simpler bounding box based approach is more robust to these phenomena.

To summarize, all of the variants of our method yield significantly higher accuracy than the state-of-the-art (Khan et al. [8]) on this dataset. While the alternative method of Wang et al. [18] also performs cell counting directly from the masked microscopy images, they only evaluate on a subset of our data, and, more importantly, only up to the 4-cell case. A direct comparison is therefore not truly possible. However, we believe that the fact that their accuracy on the 4-cell stage was substantially lower than ours (-16.44%) illustrates the superiority of our approach. Also, comparisons done by Khan et al. [8] showed that their method performed better than Wang et al. [18].

Finally, we also study the impact of our improved cell counts on the task of cell detection. To this end, we employed the method of Khan et al. [9] and replaced their cell counter with ours. For cell detection, we used the 35 sequences, consisting of 19,147 frames, in which the ground-truth cell locations were manually annotated. In Table 3, we can observe a substantial improvement (20.13%) in cell detection accuracy, thanks to our better counting strategy.

5 Conclusion

Previous approaches to cell counting in microscopy images of early-stage embryo development [7, 8, 9, 15, 18] have put a lot of effort in designing features that are well-suited for the task. These approaches, however, do not scale up to the large variability observed in ever growing datasets. In this paper, we have therefore proposed to directly learn the relevant features from images. To this end, we have introduced a deep CNN approach to cell counting in microscopy images. Our experiments have demonstrated that our basic CNN counter outperforms previous cell counting methods by a margin of 16.12% in overall accuracy. Furthermore, we have shown that this performance could be significantly improved by automatically computing a bounding box enclosing the embryo and by incorporating temporal information via a CRF. Altogether, our approach yields state-of-the-art results on the task of counting human embryonic cells in microscopy images. In the future, we plan to apply deep learning to analyze complete embryo sequences and find correlations with embryo viability. This will help embryologists to identify new biomarkers and, eventually, improve IVF success rates.

Bibliography

[1] T. Chen and C. Chefdhotel. Deep learning based automatic immune cell detection for immunohistochemistry images. In *MLMI*. 2014.

[2] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Conv. neural network committees for handwritten character classification. In *ICDAR*, 2011.

[3] D. C. Ciresan, U. Meier, J. Masci, L. Maria Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI*, 2011.

[4] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *MICCAI*. 2013.

[5] G. Flaccavento, V. Lempitsky, I. Pope, P. Barber, A. Zisserman, J. Noble, and B. Vojnovic. Learning to count cells: applications to lens-free imaging of large fields. *Microscopic Image Analysis with Applications in Biology*, 2011.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[7] A. Khan, S. Gould, and M. Salzmann. A linear chain markov model for detection and localization of cells in early stage embryo development. In *WACV*, 2015.

[8] A. Khan, S. Gould, and M. Salzmann. Automated monitoring of human embryonic cells up to the 5-cell stage in time-lapse microscopy images. In *ISBI*, 2015.

[9] A. Khan, S. Gould, and M. Salzmann. Detecting abnormal cell division patterns in early stage human embryo development. *MLMI*, 2015.
[10] A. Khan, S. Gould, and M. Salzmann. Segmentation of developing human embryo in time-lapse microscopy. In *ISBI*, 2016.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.
[13] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010.
[14] M. Meseguer, J. Herrero, A. Tejera, K. M. Hilligse, N. B. Ramsing, and R. Jose. The use of morphokinetics as a predictor of embryo implantation. *Human reproduction*, 2011.
[15] F. Moussavi, W. Yu, P. Lorenzen, J. Oakley, D. Russakoff, and S. Gould. A unified graphical model framework for automated human embryo tracking. In *ISBI*, 2014.
[16] S. Seguí, O. Pujol, and J. Vitria. Learning to count with deep object features. In *CVPR Workshops*, 2015.
[17] D. Strigl, K. Kofler, and S. Podlipnig. Performance and scalability of gpu-based convolutional neural networks. In *PDP*, 2010.
[18] Y. Wang, F. Moussavi, and P. Lorenzen. Automated embryo stage classification in tlm video of early human embryo development. In *MICCAI*. 2013.
[19] C. Wong, K. Loewke, N. Bossert, B. Behr, C. D. Jonge, T. Baer, and R. R. Pera. Non-invasive imaging of human embryos before embryonic genome activation predicts development to the blastocyst stage. *Nature Bio.*, 2010.
[20] W. Xie, J. A. Noble, and A. Zisserman. Microscopy cell counting with fully convolutional regression networks. In *DLMIA*, 2015.
[21] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015.