جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Automated counting of colony forming units using deep transfer learning from a model for congested scenes analysis

Item Type	Article
Authors	Albaradei, Somayah;Napolitano, Francesco;Uludag, Mahmut;Thafar, Maha A.;Napolitano, Sara;Essack, Magbubah;Bajic, Vladimir B.;Gao, Xin
Citation	Albaradei, S., Napolitano, F., Uludag, M., Thafar, M., Napolitano, S., Essack, M., ... Gao, X. (2020). Automated counting of colony forming units using deep transfer learning from a model for congested scenes analysis. IEEE Access, 1–1. doi:10.1109/access.2020.3021656
Eprint version	Publisher's Version/PDF
DOI	10.1109/ACCESS.2020.3021656
Publisher	Institute of Electrical and Electronics Engineers (IEEE)
Journal	IEEE Access
Download date	2024-04-17 17:04:25
Link to Item	http://hdl.handle.net/10754/665031

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3021656, IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Automated counting of colony forming units using deep transfer learning from a model for congested scenes analysis

Somayah Albaradei[1,2,#], Francesco Napolitano[1,#], Mahmut Uludag[1], Maha Thafar[1,4], Sara Napolitano[3], Magbubah Essack[1], Vladimir B. Bajic[1], Xin Gao[1*]

[1] King Abdullah University of Science and Technology (KAUST); Computational Bioscience Research Center (CBRC); Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division; Thuwal 23955-6900, Kingdom of Saudi Arabia.
[2] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia.
[3] Telethon Institute of Genetics and Medicine; Pozzuoli, Naples, Italy.
[4] College of Computers and Information Technology, Taif University, Taif, Saudi Arabia.

# Equally contributing authors.
* Corresponding author: Xin Gao. E-mail: xin.gao@kaust.edu.sa. Tel: +966-12-8080323.

ABSTRACT Reliable quantification of cellular treatment effects in many bioassays depends on the accuracy of cell colony counting. However, colony counting processes tend to be tedious, slow, and error-prone. Thus, pursuing an effective colony counting technique is ongoing, and varies from manual approaches to partly automated and fully automated techniques. Most fully automated techniques were developed using deep learning (DL). A significant problem in applying DL to this task is the lack of sizeable collections of annotated plate images. For this reason, here we propose an application of Transfer Learning to cell colony counting that can overcome this problem by exploiting models trained for other tasks. To demonstrate this idea's feasibility, we show how a small dataset can be used to transform a DL model designed for counting objects in congested scenes into a specialized cell colony counting model and achieve better performance than existing, more widely-used models.

INDEX TERMS: Artificial intelligence, colony count, machine learning, transfer learning.

I. INTRODUCTION

Quantitative analysis of bacterial populations traditionally requires the counting of colony-forming units (CFUs). Today, such analyses can also be done by more recently developed high-throughput (HT) methods that use fluorescent labeling [1], [2] genome probing microarrays [3], or quantitative PCR [4], [5]. However, these methods have significant drawbacks. For example, they may require special equipment and extensive protocol development; they may not be able to handle environmental samples; and they may not exclude non-viable cells, which is a fundamental feature to assess when cells respond to treatment. For these reasons, manual counting of CFUs from plate scans is still widely used. However, manual counting is time-consuming and may not be practical in an HT context involving a large number of experiments. Moreover, manual counting is prone to errors and biases [6]. Thus, accurate automated counting of CFUs would benefit experimental biologists.

Counting CFUs based on their features is a highly nontrivial task; nonetheless, different approaches have been proposed to tackle this problem using a typical image analysis pipeline, i.e., noise removal, image segmentation, feature extraction, feature selection, object classification, and the actual CFU counting [7]. A key issue is that the feature selection parameters should be flexible, and users are assumed to have prior knowledge of colony characteristics. Other hindrances of proper CFU analysis are low image resolution, high CFU density, background noise, artifacts on the container's boundary, and CFUs located close to the container boundaries. When only the total quantification of colonies is required, estimating the density map help to avoid the difficult task of detecting and segmenting each colony [8]. Estimating the density map is particularly useful when dealing with ambiguous counts (such as in the case presented in this study).

Automated colony counting is usually formulated as a segmentation problem. In this regard, several plugins for the image processing software ImageJ have been proposed [9],[10] that partially automate the counting process based on manually chosen parameters. Similar solutions have also been proposed as integrated software/hardware equipment [11], [12].

Two of the most popular colony counting software are OpenCFU [13] and AutoCellSeg [14]. OpenCFU [13], a stand-alone application, demonstrated improved

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3021656, IEEE Access

performance compared to similar older applications; a significant drawback of OpenCFU lies in the circularity criteria imposed on the detected objects [14]. In fact, OpenCFU is designed to explicitly detect circular CFUs according to a fixed threshold on both the isoperimetric quotient and the aspect ratio of each detected object [13]. AutoCellSeg [14] is another publicly available tool that takes segmentation plausibility criteria into account to overcome the limitations of OpenCFU [13]. However, it enables the end-user to select the object features interactively and correct the obtained result through the graphical interface. These interactions imply that AutoCellSeg is not fully automated.

Deep Learning based techniques have also been proposed. For example, Ferrari et al. [15], [16] proposed a Convolutional Neural Network (CNN) for counting the number of colonies forming on a blood agar plate. They formulated the problem as a classification task for segments obtained through external segmentation software. The model learns to classify each segment based on its estimated cardinality. After outlier rejection, the sum over the assigned cardinalities provides the count for the entire plate. Segments containing seven or more colonies were discarded. This method is adequate for cases with limited bacterial load [16], but not cases with high confluencies, such as the one presented in this study, where numerous areas of confluent cells are present. Recently, several other DL models focus on the identification of individual cells have been proposed [17],[18],[19],[20]. However, these models are based on segmentation methods tailored explicitly for objects showing microscopic cell features, such as nuclei, making it unsuitable for most colony counting tasks. For example, U-Net [17] performed poorly when applied to images used in this study (data not shown).

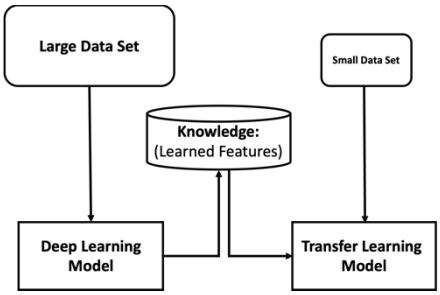

FIGURE 1. An illustration of the general idea of deep transfer learning. A large dataset is used to train a Deep Learning model, and then the trained model is specialized to a different task by fine-tuning it through a smaller dataset. The application of this general framework to the present study is shown in Fig. 2.

The bottleneck associated with using machine learning (ML) and DL to solve this problem is the large number of labeled data needed to generate a model satisfactory prediction accuracy. Thus, we here develop a method that incorporates transfer learning (TL) [21], which effectively mitigates this limitation. That is, TL allows a Neural Network model already trained for one task to be applied to a different, but related task (Fig. 1 depicts the idea of TL). In particular, a model trained using a large dataset from one task can be partially re-trained using a smaller data set from another task. In the context of colony counting, the possibility to rely on a relatively small training set avoids the need for an extensive systematic collection of plate images with manually curated colony counts, which are usually not readily available.

II. MATERIALS AND METHODS

A. DATASET

In this study, we used high-resolution scans of human-induced pluripotent stem cell colonies forming in 14 12-well plates for a total of 164 used wells. The plate scans are provided along with colony counts, which we collected as our ground-truth data. To obtain a dataset of 164 well images, we used the provided software [22] to extract individual images of each well from the plate scans. Subsequently, we split the resulting dataset into 115 (~70%) images to be used for the model development (training and validation) and 49 (~30%) for testing the model. The random split was repeated ten times, yielding ten different training and test sets. Here, we produced the first labeled data set used in this type of ML context.

As a preprocessing step, images were converted to black and white with OpenCV [23] using a simple binary thresholding operation. The threshold value was set to 127 (middle between 0 and 255). Then, each pixel was mapped to 0 (black) if less than the threshold and 255 (white) otherwise. Black and white filter commonly used in CFU counting frameworks [6], [10], [24], reduces the complexity of the images, possibly improving performances also when training ML models with limited data. We verified that unprocessed images tend to show higher RMSE for the tested models (data not shown).

B. TRANSFER MODEL

As the basis of our TL model, we used the CSRNet DL model [25], which was originally developed to count the number of people in congested scenes. The model consists of 10 convolutional layers with fixed filter size (3 x 3) as the front-end and six dilated convolutional layers as the backend. Employing this innovative structure, CSRNet was shown to outperform state-of-the-art counting solutions. CSRNet learns to extract the density map from its input images; while integrating over the map gives the estimated count. The CSRNet model was trained on the ShanghaiTech dataset [26], consisting of 1,198 annotated images of highly congested scenes. We thus sought to apply the trained CSRNet model to the related task of colony counting by specializing it to the analysis of cell plate scans (see Fig. 2 for an overview of the TL strategy).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3021656, IEEE Access

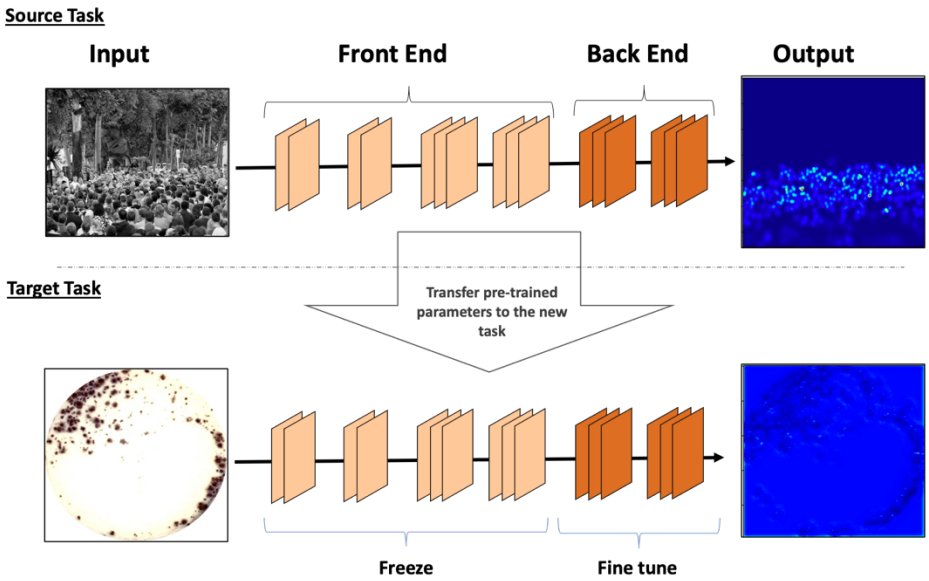

FIGURE 2. Overview of Transfer Learning between counting tasks. The CSRNet model, which was trained to count people in crowds, is adapted to count cell colonies on plates by re-training the last network layers with a reduced dataset.

In such a setting, the ShanghaiTech dataset [25] corresponds to the "Large Data Set" of Fig. 1, while our data set of cell plate images corresponds to the "Small Data Set".

The initial layers in the trained CSRNet model focus on extracting the most basic features, such as curves and edges, and subsequent layers focus on extracting the more specific training data features. Thus, to develop our TL-based method, we froze the first ten layers so that it could be reused in the colony counting task, and fine-tuned the last six layers of the CSRNet based on the plate scans data. In particular, we trained the model using Python with Keras library [27]. We used the root mean square error (RMSE) as the loss function and the Stochastic Gradient Descent (SGD) algorithm as an optimizer. Before training the data, we optimized the maximum number of epochs, batch size, and learning rate hyperparameters by grid search (see Table I).

TABLE I

Hyperparameters grid search for backend tuning. The bold values indicate the best performance.

Hyperparameters	Values
Epoch	[10,**25**,50].
Batch size	[2,**4**,16,32,64]
Optimizer	SGD with learning rate [10^{-2}, $\mathbf{10^{-3}}$, 10^{-4}]

The best hyperparameters (highlighted in bold in Table I) were used to train the model on the ten training/testing splits of the data (see Section II). For each split, 10-fold cross-validation technique was applied. The final loss was computed as the average of all scores across the 10-folds.

We used early stopping [28] to avoid overfitting. In particular, training was terminated after three epochs without improvements on the validation set, which resulted in about 19 to 25 training epochs. We also applied on-the-fly data augmentation techniques to expand the training dataset [29]. The augmentation techniques included color jitter to randomly alter brightness, contrast, saturation, and hue of each image, horizontal/vertical flip, and random rotation.

III. RESULTS

As mentioned, our Transfer Model, based on CSRNet, extracts a density map from the input image, which is used to estimate the count. Fig. 3 shows an example of how the algorithm performed on three different wells with varying colony counts. The averaged RMSE across the ten folds for each of the ten random splits is reported in Table II, which demonstrates the stability of the method with respect to the data subsets. The averaged RMSE across the ten folds for each of the ten random splits of the "Small Data Set" (164 images) is reported in Table II, which demonstrates the stability of the method with respect to the data subsets. As mentioned in Section II, we applied 10-fold cross-validation technique on each split. The average loss of all scores across the 10-folds were reported for training/validation splits. Also, the loss of the separated test set is reported for the 10 splits (see Section II). We then assessed the TL model's relative performance in two steps: first, we assessed the improvements due to TL alone, and second, we compared the Transfer Model to other counting methods. The two steps are described in detail in subsequent sections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3021656, IEEE Access

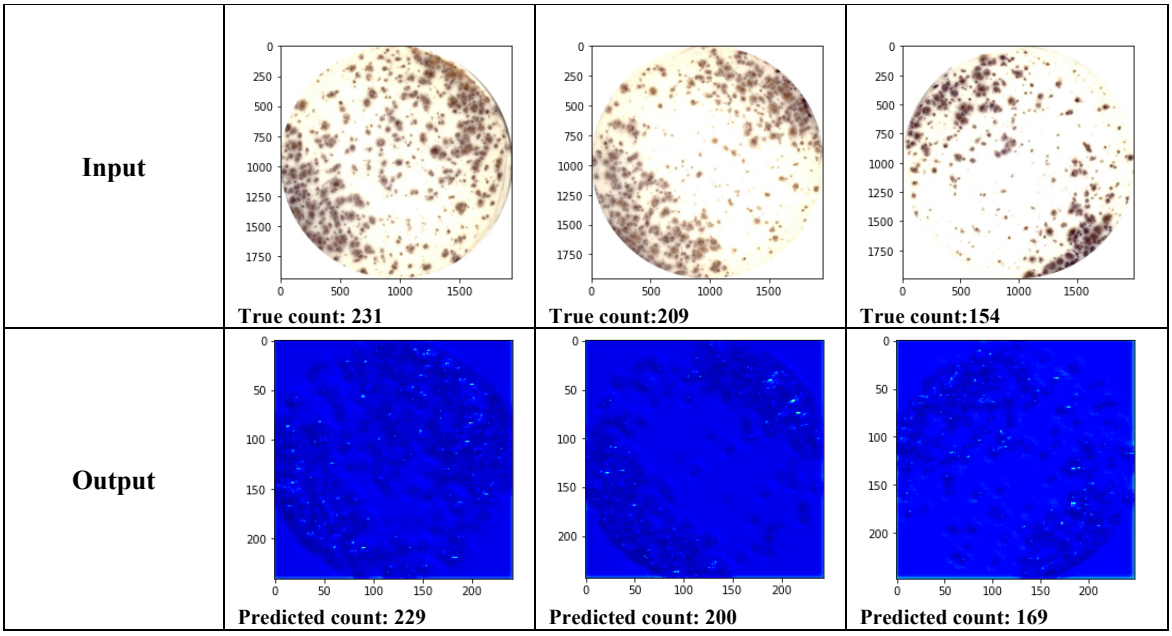

FIGURE 3. **Examples of the density maps generated by the Transfer Model showing the true and predicted counts.**

TABLE II
Mean (± standard deviation) RMSE of the transfer model across 10 random splits for the training and validation sets.

Transfer model	Training	Validation	Testing
Split 1	11.86 (±1.81)	19.86 (±4.25)	24.96
Split 2	13.35 (±1.95)	20.81 (±3.41)	22.93
Split 3	11.04 (±1.70)	19.56 (±3.33)	20.24
Split 4	12.04 (±1.86)	20.58 (±3.00)	23.05
Split 5	12.41 (±2.03)	19.02 (±3.08)	19.26
Split 6	12.44 (±2.05)	19.24 (±2.48)	23.91
Split 7	11.34 (±1.15)	19.90 (±1.12)	22.43
Split 8	11.35 (±1.74)	21.33 (±2.45)	21.03
Split 9	12.42 (±1.75)	23.00 (±1.99)	23.18
Split 10	11.26 (±1.82)	22.45 (±3.96)	22.9

A. COMPARISON WITH CSRNet TRAINED FROM SCRATCH

First, we made sure that TL improved the counting accuracy obtained with the same CSRNet architecture trained from scratch. Towards this aim, we trained all the layers of the CSRNet model using the same splits defined for the Transfer Model. The averaged RMSE values for the training, validation, and test sets across the ten splits are reported in Table III and show significantly lower performance than that achieved with TL (compare with Table II). Fig. 4 provides an illustration of the results. This analysis confirms the usefulness of TL for our application.

TABLE III
Mean (± standard deviation) RMSE for the CSRNet network model trained from scratch.

Without transfer learning	Training	Validation	Testing
Split 1	41.56 (±4.16)	58.36 (±5.50)	69.53
Split 2	44.61 (±3.17)	57.41 (±4.25)	56.95
Split 3	40.08 (±4.75)	57.81 (±5.99)	53.13
Split 4	39.07 (±5.59)	61.19 (±4.89)	53.87
Split 5	39.92 (±4.29)	58.93 (±6.27)	44.54
Split 6	42.61 (±5.52)	61.74 (±6.44)	63.51
Split 7	39.18 (±6.02)	60.19 (±5.72)	48.42
Split 8	39.67 (±2.55)	62.16 (±4.53)	44.25

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3021656, IEEE Access

Split 9	43.76 (±4.45)	58.01 (±5.34)	56.89
Split 10	39.00 (±4.75)	57.55 (±4.53)	58.2

FIGURE 4. Average RMSE for the CSRNet model with and without using transfer learning. Error bars report the standard deviation.

B. COMPARISON WITH AutoCellSeg AND OpenCFU

Concerning the second point, we compared our model's performance with two popular and publicly available tools: 1) AutoCellSeg, which is based on fuzzy a priori information, and 2) OpenCFU, which is based on direct thresholding (see Section I). We applied both tools to our dataset using default parameters. OpenCFU did not require manual intervention. However, AutoCellSeg required a selection of two colony examples: a small one and a large one. We did not observe significant variations in the counts after choosing different examples among the smallest and largest clearly identifiable colonies. RMSE values for both methods are reported in Table IV, along with the average RMSE obtained with the transfer model on the test sets across the ten random splits. As shown in Table IV, the transfer model exhibits the smallest RMSE compared to the other methods.

TABLE IV
Comparison of the different methods based on RMSE. We reported the average (± standard deviation) across the ten random splits for the transfer model.

Method	RMSE
OpenCFU	38.56
AutoCellSeg	27.64
Transfer Model	22.38 (±1.72)

We also sought to assess the tendency of each method to under- or over-estimate the ground-truth count. Towards this aim, for each image, we computed the sum of normalized positive/negative differences between predicted and ground-truth counts. We considered the integer rounded average colony count across all the 10 random splits for the transfer model. The normalized difference is calculated based on the following formula:

$$d = \sum \left| \frac{p - g}{\max(p, g)} \right|$$

where p is the predicted count, and g is the ground-truth count. Fig. 5 depicts the values of d for each of the three methods detailed for over-estimation (p > g) and under-estimation (p < g). We observe that the transfer model fares better than the other two as it exhibits more accurate counting based on the ground-truth counts.

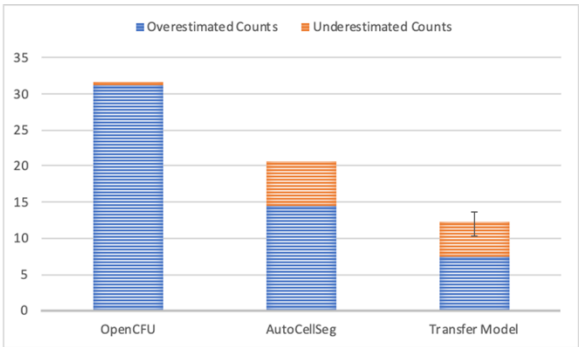

FIGURE 5. Counting errors made by the three tested methods. The chart shows the absolute sum of normalized positive/negative differences between predicted and ground-truth counts.

IV. DISCUSSION AND CONCLUSION

Colony counting is a tedious, slow, and error-prone process. However, experimental biologists make ubiquitous use of manual or partly automated counting techniques to quantify cellular treatments' effects. Thus, work that provides fully automated colony counting or labeled data sets for training ML models is needed. Although DL techniques are revolutionizing many image analysis fields, there is currently no standardized technology to solve colony counting. This may be partly due to the lack of large enough annotated image collections necessary to train a deep neural network for this task. We think that TL techniques could help overcome this problem by exploiting models already trained on related tasks and tuning them for this specific task.

Nonetheless, the use of TL models still implies that at least a small training set is available. In biological studies involving many colony formations, such as the one we considered in this paper, it is possible to tune a model on a small subset and then use it to perform the counting on the remaining images. While this could help speed up the counting process, we believe that the most significant contribution would be making the process systematic and more reproducible. Alternatively, pre-trained models like the one we developed can be used directly to perform CFU counting on new images without additional tuning as long as comparable features are involved. We demonstrated the feasibility of this approach by using, for the first time, a novel data set that we recently made publicly available. However, further validations will be required to assess the extent to which the model can generalize across different experiments. For this reason, we aim to collect more data and test the model's ability to perform counting in different contexts, possibly including

higher resolution images where cell organelles are visible. This will also require further investigation of the network architecture features, and possibly the training of additional layers, which would be feasible with more input data. Exploring such possibilities constitutes the main focus of our future work.

AVAILABILITY

The data can be found at https://github.com/SomayahAlbaradei/tlcc. We also developed a simple web interface that is available at http://www.cbrc.kaust.edu.sa/tlcc/, which takes a colony image and returns the count of colonies.

ACKNOWLEDGMENT

This study is supported by grants from King Abdullah University of Technology (KAUST), grants BAS/1/1606-01-01, FCC/1/1976-17-01, and BAS/1/1624-01-01.

AUTHOR CONTRIBUTIONS

V.B.B., X.G., F.N., and S.A. conceived and designed the study; S.A., M.U., M.T., S.N., and M.E. conducted the main technical development; S.N. helped to provide ground-truth data. All authors contributed to writing the manuscript.

CONFLICT OF INTEREST

All authors declare no conflict of interest.

REFERENCE

[1] L. Blasco, S. Ferrer, and I. Pardo, "Development of specific fluorescent oligonucleotide probes for in situ identification of wine lactic acid bacteria," *FEMS Microbiology Letters,* vol. 225, no. 1, pp. 115-123, 2003.

[2] S. Haba and A. Nisonoff, "IgE-secreting cells in the thymus: correlation with induction of tolerance to IgE," *Proceedings of the National Academy of Sciences,* vol. 89, no. 11, pp. 5185-5187, 1992.

[3] J.-W. Bae *et al.*, "Development and evaluation of genome-probing microarrays for monitoring lactic acid bacteria," *Appl. Environ. Microbiol.,* vol. 71, no. 12, pp. 8825-8835, 2005.

[4] M. Haarman and J. Knol, "Quantitative real-time PCR analysis of fecal Lactobacillus species in infants receiving a prebiotic infant formula," *Appl. Environ. Microbiol.,* vol. 72, no. 4, pp. 2359-2365, 2006.

[5] E. T. Neeley, T. G. Phister, and D. A. Mills, "Differential real-time PCR assay for enumeration of lactic acid bacteria in wine," *Appl. Environ. Microbiol.,* vol. 71, no. 12, pp. 8954-8957, 2005.

[6] S. D. Brugger, C. Baumberger, M. Jost, W. Jenni, U. Brugger, and K. Mühlemann, "Automated counting of bacterial colony forming units on agar plates," *PloS one,* vol. 7, no. 3, 2012.

[7] M. L. Clarke, R. L. Burton, A. N. Hill, M. Litorja, M. H. Nahm, and J. Hwang, "Low-cost, high-throughput, automated counting of bacterial colonies," *Cytometry Part A,* vol. 77, no. 8, pp. 790-797, 2010.

[8] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in neural information processing systems*, 2010, pp. 1324-1332.

[9] Z. Cai, N. Chattopadhyay, W. J. Liu, C. Chan, J.-P. Pignol, and R. M. Reilly, "Optimized digital counting colonies of clonogenic assays using ImageJ software and customized macros: comparison with manual counting," *International journal of radiation biology,* vol. 87, no. 11, pp. 1135-1146, 2011.

[10] P. Choudhry, "High-throughput method for automated colony and cell counting by digital image analysis based on edge detection," *PloS one,* vol. 11, no. 2, 2016.

[11] M. Siragusa, S. Dall'Olio, P. M. Fredericia, M. Jensen, and T. Groesser, "Cell colony counter called CoCoNut," *PloS one,* vol. 13, no. 11, 2018.

[12] O. Chunhachart and B. Suksawat, "Construction and validation of economic vision system for bacterial colony count," in *2016 International Computer Science and Engineering Conference (ICSEC)*, 2016: IEEE, pp. 1-5.

[13] Q. Geissmann, "OpenCFU, a new free and open-source software to count cell colonies and other circular objects," *PloS one,* vol. 8, no. 2, 2013.

[14] A. Torelli, I. Wolf, and N. Gretz, "AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques," *Scientific reports,* vol. 8, no. 1, pp. 1-10, 2018.

[15] A. Ferrari, S. Lombardi, and A. Signoroni, "Bacterial colony counting by convolutional neural networks," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015: IEEE, pp. 7458-7461.

[16] A. Ferrari, S. Lombardi, and A. Signoroni, "Bacterial colony counting with convolutional neural networks in digital microbiology imaging," *Pattern Recognition,* vol. 61, pp. 629-640, 2017.

[17] T. Falk *et al.*, "U-Net: deep learning for cell counting, detection, and morphometry," *Nature methods,* vol. 16, no. 1, pp. 67-70, 2019.

[18] W. Wang *et al.*, "Learn to segment single cells with deep distance estimator and deep cell detector," *Computers in biology and medicine,* vol. 108, pp. 133-141, 2019.

[19] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation," *IEEE Transactions on Medical Imaging,* 2019.

[20] Q. Liu, A. Junker, K. Murakami, and P. Hu, "A novel convolutional regression network for cell counting," in *2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB)*, 2019: IEEE, pp. 44-49.

[21] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*: IGI Global, 2010, pp. 242-264.

[22] F. Napolitano *et al.*, "Automatic identification of small molecules that promote cell conversion and reprogramming," 2020.

[23] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-Based Computer Vision with Intel's Open Source Computer Vision Library," *Intel technology journal,* vol. 9, no. 2, 2005.

[24] J. Austerjost *et al.*, "A smart device application for the automated determination of E. coli colonies on agar plates," *Engineering in Life Sciences,* vol. 17, no. 8, pp. 959-966, 2017.

[25] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091-1100.

[26] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589-597.

[27] F. Chollet, "Keras," *GitHub repository,* 2015.

[28] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*: Springer, 1998, pp. 55-69.

[29] F. Chollet, "Building powerful image classification models using very little data," *Keras Blog,* 2016.