Neurocomputing 470 (2022) 427–431

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep learning to detect bacterial colonies for the production of vaccines

Thomas Beznik [a,*,1], Paul Smyth [b], Gael de Lannoy [b], John A. Lee [a,2]

[a] Universite catholique de Louvain, Louvain-La-Neuve, Belgium
[b] GSK Vaccines, Rixensart, Belgium

ARTICLE INFO

Article history:
Received 3 January 2021
Revised 15 March 2021
Accepted 18 April 2021
Available online 24 July 2021

Keywords:
Deep learning
Bacterial colony
Vaccines
Artificial Intelligence
UNet

ABSTRACT

During the development of vaccines, bacterial colony forming units (CFUs) are counted in order to quantify the yield in the fermentation process. This manual task is long, tedious, and subject to errors. In this work, multiple segmentation algorithms based on the U-Net CNN architecture are tested and proven to offer robust, automated CFU counting. It is also shown that the multiclass generalisation with a bespoke loss function allows virulent and avirulent colonies to be distinguished with acceptable accuracy. While many possibilities are left to explore, our results show the potential of deep learning for separating and classifying bacterial colonies.

1. Introduction

Vaccines are considered one of the greatest public health achievements and each year prevent 2 to 3 million deaths, with 86% of the children worldwide getting vaccinated [1]. The recent worldwide outbreak of covid-19 in 2020 has attracted much interest, investment, and research effort in developing vaccines in a tight time frame. One important step in vaccine development is the measure of the number of viable bacterial colonies and the proportion of virulent colonies among them, as a vaccine requires a certain proportion of virulent bacteria to be effective. Samples are taken and applied onto Petri dishes containing a blood agar substrate and after a couple of days the single bacteria have grown into heaps of billions of cells called Colony Forming Units (CFUs) which are visible to the naked eye. The proportion of viable bacteria in the solution can thus be estimated by counting the number of CFUs, since each CFU originated from a single bacterium. The CFUs can be divided into two classes: virulent (also called 'bvg+') and avirulent (also called 'bvg−'). These two classes can be distinguished through visual inspection: virulent colonies lyse the blood cells, i.e., destroy their membrane, which produces a dark halo around the colony, while avirulent colonies do not and thus no halo is observed.

The purpose of this work is to automate the bacterial colony detection and identification process, in order to free time for biologists to concentrate on other tasks. Specifically, the aim is to design a robust automated CFU counting algorithm that can distinguish 'bvg+' and 'bvg−' colonies.

2. State of the art

The automation of colony counting has been on the minds of researchers as early as 1957 [2] and is still being investigated actively to this day [3–5]. The methods can be grouped in three categories.

First, hardware solutions can be applied directly to the plates that count by sweeping the plate with a scanning spot and observing the light transmitted via a photo-tube [2,6]. These solutions are not autonomous as parameters need to be manually set for each plate.

Second, software solutions can process images of the plates using various methods of thresholding (see [7,8,3] for some recent work). The performances of such approaches are often assessed on limited retrospective data. Typically, these approaches struggle to deal with reflections, bubbles, artefacts in the agar, variability in shape and color of the colonies. They can also fail to separate touching colonies.

Finally, modern methods of machine learning can count bacterial colonies in images, following the work of Flaccavento et al. [9],

* Corresponding author.
E-mail addresses: thomas.beznik@relu.eu (T. Beznik), paul.x.smyth@gsk.com (P. Smyth), gael.x.de-lannoy@gsk.com (G.de Lannoy), john.lee@uclouvain.be (J.A. Lee).

[1] T.B. was affiliated to UCL at the time of the work; he works now for the company RELU.

[2] J.A.L. is a Senior Research Associate with the Belgian F.R.S.-FNRS.

https://doi.org/10.1016/j.neucom.2021.04.130
0925-2312/

where random forests are used to identify individual colonies. The results of this method appear reasonably good, but they have been evaluated on only 9 images. The first approach to colony counting with deep learning, using a convolutional neural network (CNN), was published in 2015 [10]. It significantly outperformed other methods. More recently, in 2018, a fully convolutional approach was used [11]. Here, the CNN was applied directly to the full images, rather than to segments of the image. However, their approach did not count the number of colonies, but only separated background from colonies, thus avoiding problems raised by touching colonies.

In summary, there are several gaps in the literature that this work aims to address: first, to our knowledge, no study has been conducted on the classification of the colonies. Second, studies relying on deep learning to count CFUs use old network architectures. Finally, there is little research yet on the use of an end-to-end solution, namely, a deep convolutional network that processes an image of a Petri dish and outputs the colony counts and classes.

3. Data collection and labelling

A total of 108 images of Petri dishes containing bacterial colonies were obtained from GSK laboratories using the laboratory aCOLyte camera (Synbiosis, Cambridge, UK). The plates are placed at a defined location and illuminated by LEDs above and below the plate. An image of size 480x480x3 is acquired (see Fig. 1). There are on average 4.726 'bvg−' colonies and 20.357 'bvg+' colonies in a single Petri dish. We define four classes that the pixels can belong to: 'background', 'bvg+', 'bvg−', and 'border'. The first three are the classes of interest, the 'border' class has been introduced to handle the separation of touching colonies.

The data set was split into four subsets and four trained annotators labelled independently one subset each. A fifth annotator labelled the whole data set for cross-validation. The resulting masks of the four first annotators were compared to the masks obtained by the fifth annotator and the mismatches were classified into two categories: 'detection mismatches' (i.e., the annotators do not agree if a certain object is a colony or not) and 'annotation mismatches' (i.e., the annotators do not agree on the virulence of a colony: 'bvg+' or 'bvg−'). These mismatches were then corrected with the laboratory expert in order to obtain the final masks.

Based on the masks created by the different annotators, we can analyse the variability of the labelling of bacterial colonies, in order to assess consistency of data. Along with the above definitions at the level of objects and colonies, we will also use the mean and standard deviation of the inter-annotator Jaccard index, also known as Intersection over Union (IoU) across images, defined in Eq. (1). For a given image, the inter-annotator IoU is equal to the intersection of two annotators' binary masks, Y_1, Y_2 divided by their union. We looked at the 'colony' binary masks, i.e., the masks which contains both 'bvg+' and 'bvg−' colonies, in order to assess the variability of the selected colony area among annotators.

$$IoU(Y_1, Y_2) = \frac{|Y_1 \cap Y_2|}{|Y_1 \cup Y_2|} \tag{1}$$

Table 1 shows the analysis of inter-annotator IoU for each subset of data prior to correction by the laboratory expert.

We find that the area that is considered to belong to the bacterial colony is variable across annotators. This is reflected in the mean IoU between annotators being relatively low, even though the number of detection errors is small. This variability could be caused by the fact that the exact border of a colony can be unclear in low-resolution images and/or by difficulties with the labelling software. We can also see that the proportion of mismatches between annotators, i.e., compared to the total number of colonies, is low, showing that in general the annotators agreed on the detection and classification of colonies. We observed that detection mismatches were often due to a high density of touching colonies and that annotation mismatches were often caused by poor lighting conditions, which made it difficult to observe the dark halo around 'bvg+' colonies.

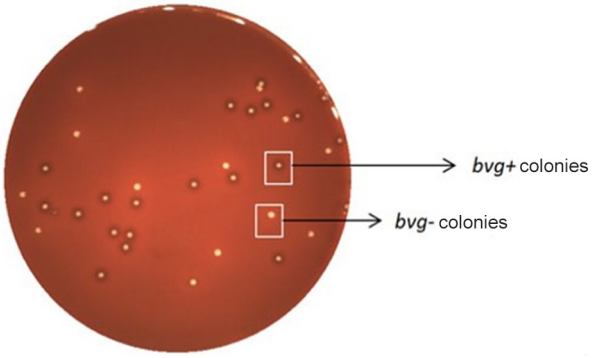

Fig. 1. Image of a Petri dish with 'bvg+' and 'bvg-' colonies.

4. Experiments and results

A first possible difficulty faced in our experiments was the data set size: there are only 108 images and the models will learn from even fewer images due to cross-validation. We chose to continue with this small data set as it could be labelled in a reasonable amount of time and as it would allow us to test if satisfactory performance could be achieved with limited data. Most deep learning architectures for computer vision involve millions of parameters and use very large data sets to train. In our case using these architectures will likely result in over-fitting: we thus need to find architectures that are well-suited for limited-size data sets, use transfer learning, data augmentations, and other techniques to boost performance.

A second difficulty is the class imbalance in the data set: 98.927% of the pixels are 'background', 0.825% are 'bvg+', 0.238% are 'bvg−' and 0.009% are 'border'. We can thus see that there is a very high class imbalance between the 'background' pixels and the other classes. This is due to the fact that the colonies are small and in small numbers, while the rest of the image is considered as 'background'. A trivial model that predicts 'background' for every pixel would obtain a training accuracy of 98.927% without detecting any colony. We must thus take this class imbalance into account in the training and validation phase to avoid this issue; this will mostly be handled by adapting the loss.

As seen in Fig. 1, a third challenge is the presence of noise in the images, mainly the reflections of the LEDs on the border of the agar plate: these reflections could be easily mistaken for colonies and many watershed-based algorithms (see e.g. [12] and references therein) would struggle to discard them. We must also address the challenge of separating touching colonies, which can be complicated even for humans.

The data set is randomly split into training-validation (80%) and test (20%). Random rotations in $[0°, 360°]$ were used to augment the small amount of training data. Two network architectures are explored in the hyperparameter search:

1. Regular U-Net [13]: the hyperparameters that are searched are the number of layers (2, 4 and 6) and the use of batch normalization.

2. U-Net with pre-trained encoder [14]: the only hyperparameter is the pre-trained network (ResNet-50, ResNet-152, Inception-ResNet-v2, or DenseNet-169). The number corresponds to the number of layers of the network.

For those networks, two losses are investigated in the hyperparameter search.

First, for input image Ω, pixel $(x,y) \in \Omega$, and class set C, the weighted cross-entropy is defined as

$$\mathscr{L}_{WCE}(\Omega) = -\frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} \sum_{c\in C} w_c \, r_c(x,y) \, \log(p_c(x,y)) \ , \tag{2}$$

where $p(x,y)$ is the output class probability vector of pixel (x,y), $p_c(x,y)$ is the output class probability for class c, $r(x,y)$ is the one-hot encoded ground truth class vector associated with pixel (x,y), with $r_c(x,y) = 1$ if the pixel is in class c and 0 otherwise, and w_c is the weight associated with class c.

Second, the categorical cross-entropy with soft Dice per channel [15] is defined as

$$\mathscr{L}_{Dice}(\Omega) = w_{CE} \, \mathscr{L}_{CE}(\Omega) + \sum_{c\in C\setminus \text{background}} w_c \, \mathscr{L}_{\mathscr{D}_c}(\Omega) \ , \tag{3}$$

where w_{CE} is the weight associated with the cross-entropy loss, $\mathscr{L}_{CE}$ is the categorical cross-entropy loss and $\mathscr{L}_{\mathscr{D}_c}$ the soft Dice loss.

As we can see, both losses involve weights which are additional hyperparameters. These losses are adapted to the inherent class imbalance.

For all models, the Adam optimizer is used and several learning rates are studied ($1e-3, 1e-4$, and $1e-5$). A mini-batch size of 8 is used for the regular U-Net models and of 4 for the pre-trained U-Net models. Early stopping is used with a patience of 10 epochs. The results were collected in two phases: hyperparameter search and best model evaluation. In the first phase, the impact of all the hyperparameters defined above is assessed through a 4-fold cross-validation. Their performance is evaluated using the 'mAP (mean Average Precision) over multiple IoU thresholds' metric [16]. This metric defines a true positive as an object which is detected with an IoU above a certain threshold, where the IoU is now between the predicted and ground-truth masks. The metric is then computed by averaging the Average Precision over multiple IoU thresholds, typically from 0.5 to 0.95 with a step of 0.05. For the second phase, the hyperparameter set that obtained the best validation 'mAP over multiple IoU thresholds' is selected, it is trained on the full training-validation set and evaluated on the test set. We use the 'MAE (Mean Average Error)' metric to judge the colony counting accuracy of the best model. The colony count is obtained by counting the number of connected components of each class in the mask, ignoring the 'border' pixels.

Due to time constraints, we were not able to perform a full grid-search of the hyperparameter space and thus chose to opt for an "experiment-based" search strategy: we considered several experiments where we vary few hyperparameters and analyze their impact. This gives us only limited insight, as it considers few interactions between hyperparameters, but it gives us the ability to explore a larger range for each hyperparameter. We found that the most influential hyperparameters were the learning rate and the type of architecture. We will here only show a subset of the results of this search. Table 2 presents the results of hyperparameter search on the type of pre-trained encoder for the "U-Net with pre-trained encoder" architecture. We also give the results of the best performing "regular U-Net" model for comparison. We can first observe that several of these models outperform the best "regular U-Net" model. We also see that the models trained with the weighted cross-entropy ('wce') loss are, on average, performing better than those trained with the categorical cross-entropy with soft Dice per channel ('Dice') loss.

The best performing architecture is the ResNet-152 pre-trained U-Net encoder, trained with a learning rate of $1e-4$ and a weighted cross-entropy loss with weights $w_{\text{background}} = 0.01$, $w_{\text{bvg+}} = 0.25$, $w_{\text{bvg-}} = 0.34$ and $w_{\text{border}} = 0.4$. General statistics of the best model are shown in Table 3. We can see that the performances on the 'border' class are worse than those for the 'bvg+' and 'bvg−' classes. For the training set, the precision on this class is 0.758 and the recall is equal to 0.518, which is already not satisfactory. The model obtains a 'border' precision of 0.503 and recall

Table 1
Analysis of the mismatches for the different annotators. Each row describes a different subset of the data. For each subset we report the mean IoU between the annotators with the standard deviation between parentheses (pixel-level results), and the results at the level of objects and colonies: the total number of detection and annotation differences. We also give the total number of colonies in each subset.

Data Subset	Mean IoU (std)	Total Detection Differences	Total Annotation Differences	Total Number of colonies
1	0.476 (0.114)	1	10	699
2	0.509 (0.120)	14	1	734
3	0.328 (0.041)	5	1	420
4	0.483 (0.045)	27	8	813

Table 2
Results of the hyperparameter search on the type of pre-trained encoder, with two different losses and a learning rate of 0.0001. The last row gives the results of the best model from the "Regular UNet" architecture, with 6 layers, batch normalization, and a learning rate of 0.0001. 'Dice' stands for the categorical cross-entropy with soft Dice per channel loss and 'wce' stands for the weighted cross-entropy loss. The numbers in parentheses for the losses give the values of the different weights of these losses: ('background', 'bvg+', 'bvg−', 'border') for the 'wce' loss and (categorical cross-entropy loss, 'bvg+' soft dice loss, 'bvg−' soft dice loss, 'border' soft dice loss) for the 'Dice' loss.

Loss	Pre-trained encoder	Mean train mAP (std)	Mean validationmAP (std)
Dice (0.2, 0.3, 0.31, 0.19)	resnet152	0.656 (0.081)	0.555 (0.040)
	resnet50	**0.700 (0.023)**	**0.583 (0.032)**
	resnetv2	0.654 (0.061)	0.552 (0.032)
	densenet169	0.602 (0.053)	0.556 (0.013)
wce (0.01, 0.25, 0.34, 0.4)	**resnet152**	**0.745 (0.036)**	**0.608 (0.034)**
	resnet50	0.685 (0.017)	0.574 (0.033)
	resnetv2	0.683 (0.093)	0.576 (0.048)
	densenet169	0.702 (0.024)	0.585 (0.021)
Dice (0.2, 0.25, 0.26, 0.29)	N/A	0.742 (0.02)	0.579 (0.02)

Table 3
Statistics of the best model on the training and test sets.

Class	Dataset	MAE	Precision	Recall
Bgv+	Training	0.798	0.952	0.961
Bgv+	Test	1.048	0.906	0.927
Bgv−	Training	0.059	0.955	0.961
Bgv−	Test	0.143	0.919	0.905
Border	Training	N/A	0.758	0.518
Border	Test	N/A	0.503	0.285

of 0.285 on the test set. The model clearly has not learned well to detect border pixels and is thus not good at separating touching colonies (see Fig. 3). We can also observe in Table 3 that the precisions and recalls for the 'bvg+' and 'bvg−' pixels are high, both on the training and test sets. Finally, we can see that the MAE, the average difference in colony counting per image, roughly doubles from the training set to the test set for the 'bvg+' and 'bvg−' colonies: for the test set, the model is on average off by 1.048 'bvg+' colony and 0.143 'bvg−' per image. We can also see that the 'bvg−' MAE is much smaller than the 'bvg+' MAE for the same dataset; the model thus seems to make fewer counting mistakes for the 'bvg−' colonies than for the 'bvg+' for one image, but we believe that this is biased by the fact that there are generally much more 'bvg+' colonies than 'bvg−' in an image.

Figs. 2 and 3 show two outputs of the model on images of the test set. These outputs illustrate the strengths and weaknesses of the model on unseen data. We identify the colonies of interest in the images using blue boxes and an associated number. First, we see that for both images the model detects and classifies the colonies quite well. There is only one class error in Fig. 3, for colony number 5. This results actually from a labelling error; the colony is 'bvg−' indeed, just as the model predicted. We can also observe how the model deals with the separation of colonies for these input images. In Fig. 2, we see that colonies 1, 2, and 3 are well separated with 'border' pixels, but that the colonies at 4 are not completely separated. They will thus be counted as one colony. In Fig. 3, the separation of the colonies at 1 is badly handled by the model and there will thus be a high counting error for this image. On the other hand, colonies 2, 3, and 4 are well separated. The incomplete separation of touching colonies is the most common error of the model. The model often detects some 'border' pixels between touching colonies, but it fails to completely separate them, which results in a counting error. Simple post-processing could greatly improve the separation by for example extending the border. Finally, we see that for both images the model ignores the reflections on the border of the Petri dish. In general, the model is good at detecting, classifying and identifying the correct area of the colonies.

5. Conclusion and future work

We have shown that deep learning algorithms can overcome the drawbacks of existing computer vision solutions for classification of bacterial colonies in images of Petri dishes with access to a limited data set. The U-Net [13] architecture with a pre-trained ResNet-152 [17] encoder, weighted cross-entropy loss, combined with image augmentations, achieves promising results on unseen images. We have shown that this approach can overcome some of the drawbacks of existing computer vision solutions: it is robust to reflections, can consistently detect colonies of various shapes

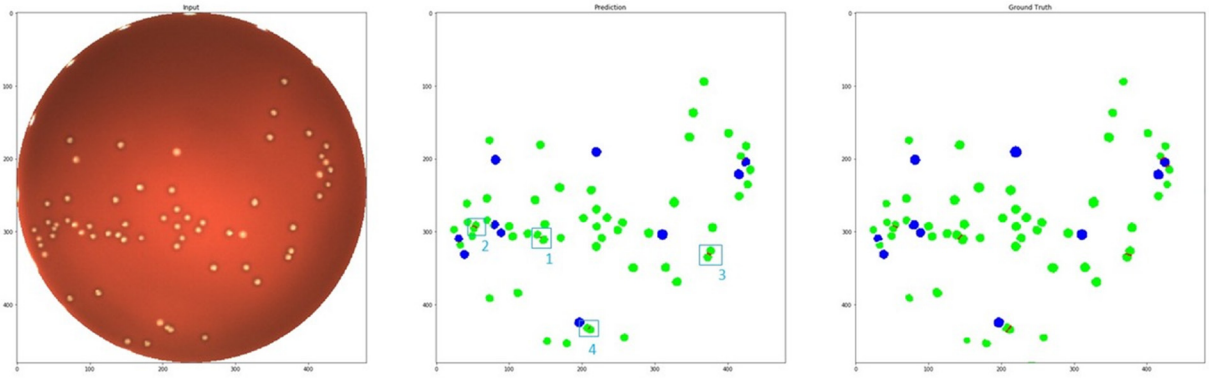

Fig. 2. Left: Test image. Middle: Model prediction. Right: ground truth.

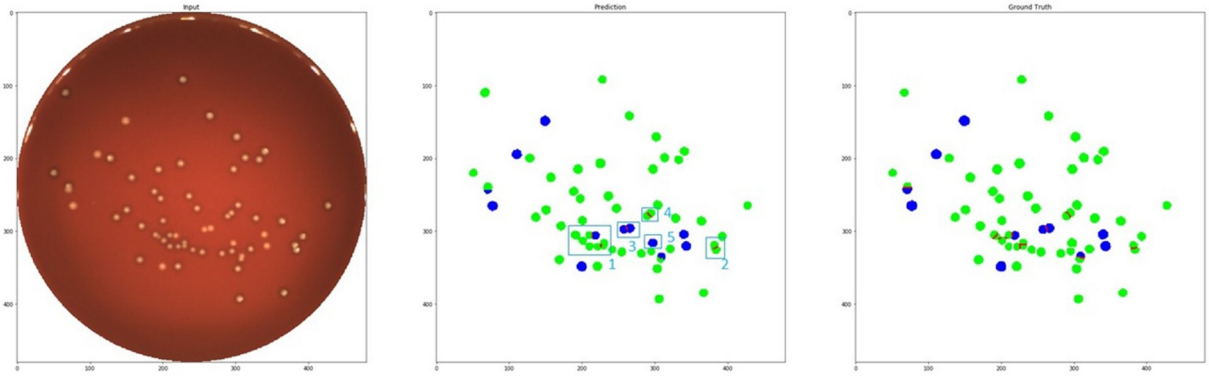

Fig. 3. Left: Test image. Middle: Model prediction. Right: ground truth.

and sizes, and it achieves complete automation. The proposed approach contributes to closing a gap in the literature: no study, to our knowledge, has explored the use of deep learning to solve the problem of CFU counting, in an end-to-end fashion, including the classification of bacterial colonies. These findings open the way to a new approach of automating bacterial colony counting and classification, fitting into the bigger trend towards full laboratory automation. Future research will investigate an improved separation of touching colonies by using specific losses such as the focal loss [18], 'U-Net' loss [13], or changing the task to bounding-box detection. It will also be interesting to evaluate a bigger and more diverse data set, to perform a broader hyperparameter search, and to study the benefit of post-processing methods.

CRediT authorship contribution statement

Thomas Beznik: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft. **Paul Smyth:** Conceptualization, Writing - review & editing, Resources, Supervision. **de Gael Lannoy:** Writing - review & editing. **John A. Lee:** Conceptualization, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gaël de Lannoy and Paul Smyth are employees of the GSK group of companies and report ownership of GSK shares and/or restricted GSK shares.

Acknowledgements

This work was sponsored by GlaxoSmithKline Biologicals SA. All authors were involved in drafting the manuscript and approved the final version. The authors declare the following interest: GDL and PS are employees of the GSK group of companies and report ownership of GSK shares and/or restricted GSK shares. The authors would like to thank Sébastien Cheffert, Alison Pire, and Thomas Haupt for their valuable input. John A. Lee is a Senior Research Associate with the Belgian F.R.S.-FNRS.

References

[1] W.H. Organisation, About 116 million children worldwide receive basic vaccines every year, WHO Infographics..
[2] H. Mansberg, Automatic particle and bacterial colony counter, Science 126 (3278) (1957) 823–827.
[3] A. Torelli, I. Wolf, N. e. a. Gretz, Autocellseg: robust automatic colony forming unit (cfu)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques, Scientific Reports 8 (1) (2018) 7302..
[4] B. Jagga, D. Singh, Image processing based bacterial colony counter..
[5] M. Siragusa, S. Dall Olio, P.M. Fredericia, M. Jensen, T. Groesser, Cell colony counter called coconut, PloS one 13 (11) (2018) e0205823.
[6] N. Alexander, D. Glick, Automatic counting of bacterial cultures-a new machine, IRE Transactions on Medical Electronics (1958) 89–92.
[7] P.-J. Chiang, M.-J. Tseng, Z.-S. He, C.-H. Li, Automated counting of bacterial colonies by image analysis, Journal of Microbiological Methods 108 (2015) 74–82.
[8] N. Uppal, R. Goyal, Computational approach to count bacterial colonies, International Journal of Advances in Engineering & Technology 4 (2) (2012) 364.
[9] G. Flaccavento, V. Lempitsky, I. Pope, P. Barber, A. Zisserman, J. Noble, B. Vojnovic, Learning to count cells: applications to lens-free imaging of large fields..
[10] A. Ferrari, S. Lombardi, A. Signoroni, Bacterial colony counting with convolutional neural networks in digital microbiology imaging, Pattern Recognition 61 (2017) 629–640.
[11] P. Andreini, S. Bonechi, M. Bianchini, A. Mecocci, F. Scarselli, A deep learning approach to bacterial colony segmentation, in: International Conference on Artificial Neural Networks, Springer, 2018, pp. 522–533.
[12] J.B. Roerdink, A. Meijster, The watershed transform: Definitions, algorithms and parallelization strategies, Fundamenta informaticae 41 (1, 2) (2000) 187–228..
[13] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: MICCAI, Springer, 2015, pp. 234–241..
[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.
[15] O. Kodym, M. Španěl, A. Herout, Segmentation of head and neck organs at risk using cnn with batch dice loss, in: German Conference on Pattern Recognition, Springer, 2018, pp. 105–114.
[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755..
[17] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence..

Thomas Beznik received his Master in computer science engineering in 2019, from the Université catholique de Louvain (UCL, Belgium). He currently works as chief product officer at relu (www.relu.eu), a start-up that he co-founded which aims to leverage the power of deep learning in the field of dentistry. His interests are in the application of deep learning in medical imaging, with the aim of partially automating treatments.

Gaël de Lannoy graduated Master in computer science in 2004 and Master in statistics in 2005 both from Université catholique de Louvain (UCL), Belgium with first class honors. He then obtained the Ph.D. degree within the Machine Learning Group at UCL (www.ucl.ac.be/mlg). The main topics covered in his thesis are data science, machine learning and statistics applied to biomedical data. Gaël is now working within the R&D department at GlaxoSmithKline (GSK) Vaccines, leading the global drug substance innovation center.

Paul Smyth received his PhD, MSc and BSci in Theoretical Physics in 2006, 2001 and 2000, respectively, from Imperial College London. He currently works as a senior data scientist at GSK Vaccines. Paul research interests are in the application of machine learning to image, text and biological sequences in vaccines research and development.

John A. Lee received the M.S. degree in Applied Sciences (Computer Engineering) in 1999 and the Ph.D. degree in Applied Sciences (Machine Learning) in 2003, both from the Universite catholique de Louvain (UCL, Belgium). He is a member of the UCL Machine Learning Group and a Research Associate with the Belgian F.N.R.S.. Together with Michel Verleysen, he wrote a monography entitled 'Nonlinear Dimensionality Reduction' published by Springer-Verlag in 2007. His current work aims at developing specific image enhancement techniques for positron emission tomography in the center of Molecular Imaging, Radiotherapy, and Oncology (MIRO).