

APPLYING DEEP LEARNING TO MELANOCYTE COUNTING ON FLUORESCENT TRP1 LABELLED IMAGES OF IN VITRO SKIN MODEL

TRISTAN LAZARD^{✉,1}, SAMY BLUSSEAU¹, SANTIAGO VELASCO-FORERO¹, ÉTIENNE DECENCIÈRE¹, VIRGINIE FLOURET², CATHERINE COHEN² AND THÉRÈSE BALDEWECK²

¹Mines Paris, PSL University, Centre for Mathematical Morphology, 35 rue St Honoré, 77300 Fontainebleau, France, ²L'Oréal Research and Innovation, 1 avenue Eugène Schueller, 93600 Aulnay-sous-Bois, France
e-mail: tristan.lazard@minesparis.psl.eu, samy.blusseau@minesparis.psl.eu, santiago.velasco@minesparis.psl.eu, etienne.decenciere@minesparis.psl.eu, catherine.cohen@rd.loreal.com, virginie.flouret@rd.loreal.com, therese.baldeweck@rd.lorcountingreal.com

(Received November 9, 2021; revised September 9, 2022; accepted October 26, 2022)

ABSTRACT

Cell counting is an important step in many biological experiments. It can be challenging, due to the large variability in contrast and shape of the cells, especially when their density is so high that the cells are closely packed together. Automation is needed to increase the speed and quality of the detection. In this study, a cell counting method is developed for images of melanocytes obtained after fluorescent labelling with TRP1 (Tyrosinase-related protein 1) of 3D reconstructed skin samples. Following previous approaches, a strategy based on predicting the local cell density, by means of a convolutional neural network (a U-Net), was adopted. The method showed great efficiency on a test set of 76 images, with an assessed counting error close to 10% on average, which is a commonly accepted target in cytology and histology. For comparison purposes, we have made our dataset publicly available.

Keywords: cell counting, convolutional neural networks (CNN), deep learning, fluorescence microscopy, histology, immunohistochemistry (IHC), reconstructed skin.

INTRODUCTION

In order to investigate human skin responses to molecules of interest, *in vitro* biological models are needed to mimic human skin behavior. An *in vitro* 3D pigmented reconstructed skin, containing keratinocytes, fibroblasts and melanocytes, has been developed to study pigmentation mechanisms (Duval *et al.*, 2014). It offers the possibility to select, manage and treat cells in order to create customized experimental setups.

Cutaneous pigmentation is currently assessed by measuring melanin content on Fontana Masson histological color images and by quantifying melanocyte population on fluorescent images labelled for Tyrosinase-related protein 1 (TRP1), a melanocytic marker. These quantifications are usually done in a semi-automatic way, which is time consuming and prone to user induced variability. Here, we aim at developing an automatic cell counting method for TRP1 images. Variability appears between different production batches of skin models leading to various melanocyte shapes, densities and dendrites as shown on Fig. 1. Therefore the proposed method has been designed to encompass a wide range of image variability.

The contributions of this paper are: 1. The adaptation of an existing cell counting solution based on deep learning and local density estimation; 2. a specific metric adapted to counting approaches in large field images; 3. A new challenging data set for cell counting.

RELATED WORK

Counting cells automatically in microscopy images is a challenge (Schmitz *et al.*, 2014) that has motivated the development of many specific methods in the computer vision community. The first approaches involved image processing methods based on color and texture descriptors combined with morphology filtering (Berge *et al.*, 2011; Chadha *et al.*, 2020). However, over the last decade, the state of the art has been achieved by deep learning based approaches, and more precisely by Convolutional Neural Networks (CNNs) (Khan *et al.*, 2016; Cohen *et al.*, 2017; Xie *et al.*, 2018a; Falk *et al.*, 2019; Han *et al.*, 2019; Xie *et al.*, 2018b; Paulauskaite-Taraseviciene *et al.*, 2019; Liu *et al.*, 2019; Sierra *et al.*, 2020; Zheng *et al.*, 2020; He *et al.*, 2021). In object counting, it is common to distinguish between detection-based methods (Arteta *et al.*,

2016; Laradji *et al.*, 2018; Xie *et al.*, 2018b; Falk *et al.*, 2019), which are designed to precisely locate objects before counting them, and regression-based methods (Cohen *et al.*, 2017; Xie *et al.*, 2018a; He *et al.*, 2021), which directly output a number of cells without necessarily detecting their precise locations. Since accurately locating objects is harder than counting them, especially when they are partially occluded or overlapping, regression-based methods seem preferable when only a density estimation is required. As a matter of fact, they show the best results in cell counting (Cohen *et al.*, 2017; Xie *et al.*, 2018a; He *et al.*, 2021). The most successful ones consist in learning a mapping between input images and density maps, as originally proposed by Lempitsky and Zisserman (2010). The number of objects of interest is then recovered by integrating these density maps.

The seminal work¹ of Xie *et al.* (2018a), proposed to do so with a fully convolutional neural network, laying the basis of a new shift in the field of cell counting. The success of this approach lies in three key advantages. First, it benefits from the flexibility of regression-based methods in the handling of overlapping cells. Second, density maps can be easily interpreted as they provide a rough location of cells, contrary to other regression methods like (Cohen *et al.*, 2017). This is critical for real life users who need to check the soundness of the algorithm on their data. Third, the method comes with all the strengths of CNNs, namely great accuracy, automatic learning of features and the possibility to run the network on any size of image, including very large ones, avoiding the need to cut images into patches which often entails border effects. Regarding the very counting performance, progress were made since the study of Xie *et al.* (2018a), but at the cost of lower explainability (Cohen *et al.*, 2017) or a much more complex setting (He *et al.*, 2021) for a marginal improvement. In the literature these methods are usually applied to benchmark datasets where cells are mostly convex.

In the present study, we propose a successful application of the paradigm initiated by Xie *et al.* We customize this method using recent progress in deep learning but keeping it simple, easily interpretable for users, and apply it to data coming from a real life lab routine, previously untested and showing non convex cells with very high variability.

DATASET

In order to identify *in vitro* molecules impacting the pigmentary function of the skin, a 3D model of pigmented reconstructed skin has been developed (Duval *et al.*, 2014). It comprises an epithelium made of keratinocytes, harboring at its basal layer the melanocyte population which is responsible for the pigments generation. The epidermis is built on a dermal equivalent containing the fibroblast population embedded in a contracted collagen gel. The model offers the possibility to select, combine and treat different cell populations in order to build specific experimental setups to study pigmentation.

Different *in vitro* pigmented reconstructed skin samples were produced and sliced in quarters for different analysis purposes (radius: 10mm). One quarter was submitted to TRP1 fluorescent labeling according to Duval *et al.* (2014), see Fig. 2. As can be seen in the the zoomed image, only the fluorescent melanocytes are visible. They appear as dendritic cells in unlabeled keratinocyte epithelium (not visible).

Histological images were acquired with a Nanozoomer Hamamatsu microscope producing large field images ($\approx 13mm \times 13mm$, 30000×30000 pixels) with high resolution ($0.4523 \times 0.4523 \mu m^2 / pixel$). Three channels RGB images were saved, but only the green channel was used since the fluorescent signal was recorded in this channel. From each large field image, an area of 1024×1024 pixels was extracted in a region near the center of the sample (center of the disk) to avoid the edge effects of the biological tissue.

In order to take into account different experimental conditions, we selected images from studies performed between 2011 and 2017. Within each study, various experimental conditions were realized in triplicates (3 samples per condition). To establish the data set for this work, we made the decision to retain from each study only one sample per experimental condition.

Our dataset is composed of two sets of 76 colour images each, one for training and the other for testing. Each image is 1024×1024 pixels large and at the resolution mentioned above (see Fig. 1). Each image was analyzed by experts who marked the coordinates of each cell, providing a ground truth in location as well as a number of cells per image, as shown in Fig. 1. The first set of 76 images, that we named *Set 1*, was used for training, whereas *Set 2*, composed of the remaining 76 images, was kept as a test set. Both sets are available online².

¹Although the paper was published in a journal issue of 2018, it was first published online in 2016.

²Our dataset is available at the following link: <https://cloud.minesparis.psl.eu/index.php/s/c50xQFENFZ6I5h>

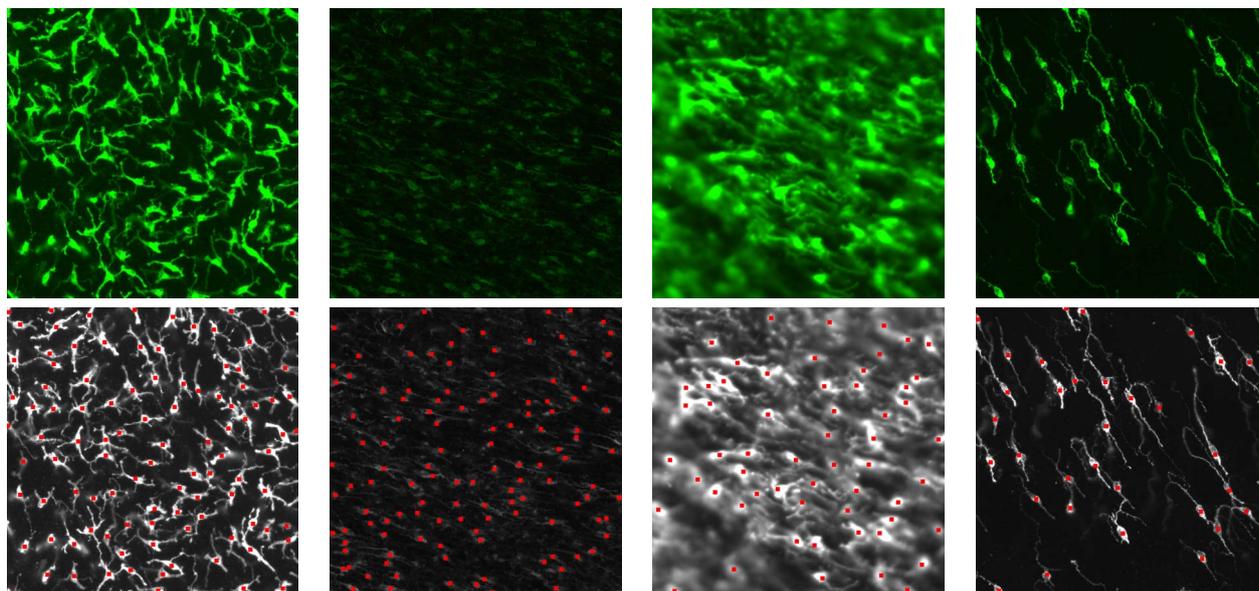


Fig. 1. **Top row:** Four examples of 1024×1024 images cropped from high resolution skin samples (Resolution: $0.4523 \times 0.4523 \mu\text{m}^2 / \text{pixel}$). They illustrate the high variability of the data to analyse. **Bottom row:** Corresponding coordinates of manually annotated cells (in red on grayscale images for better visibility).

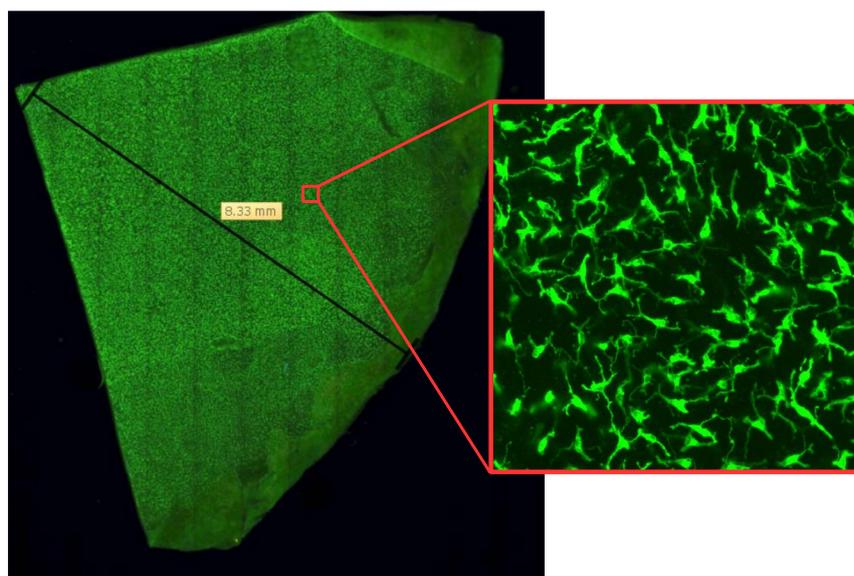


Fig. 2. Example of a quarter skin sample labelled by the TRP1 (Tyrosinase-related protein 1) and zoom on a detail where melanocytes are visible.

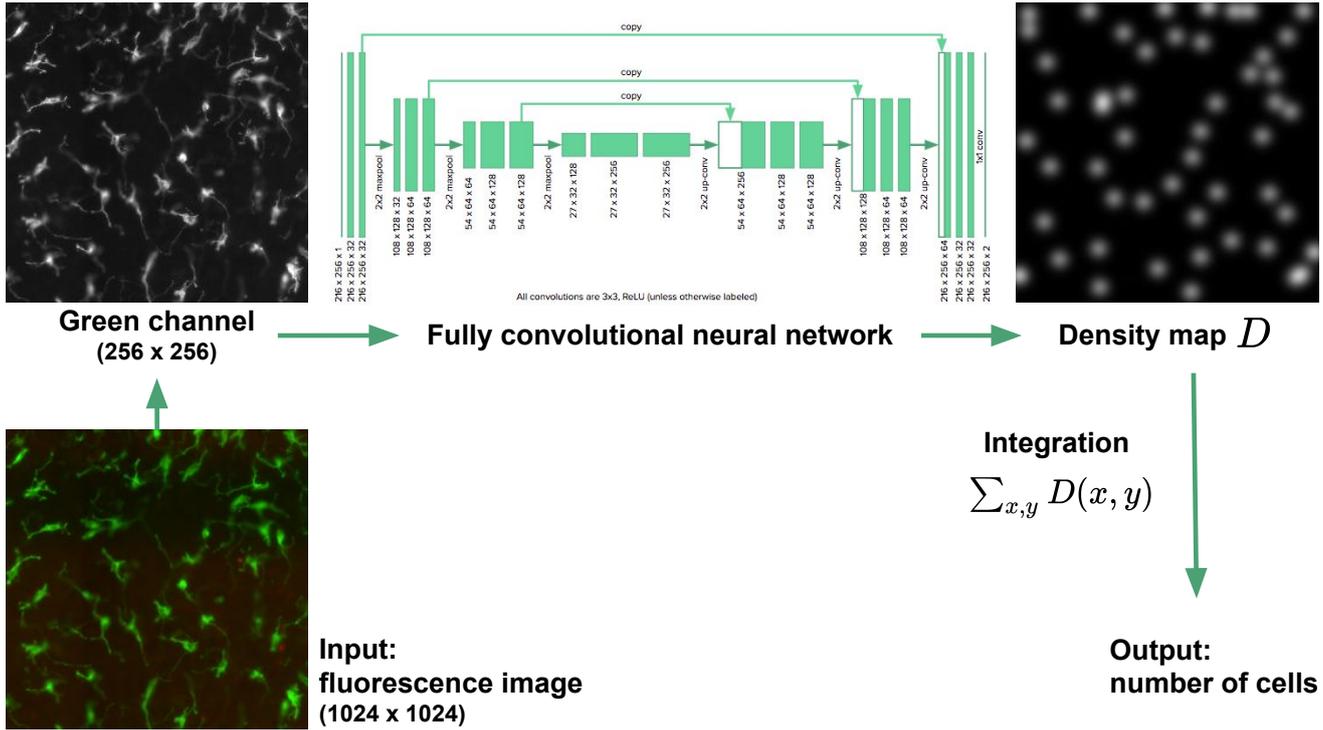


Fig. 3. Illustration of the proposed method: input images are mapped by a Fully Convolutional Neural Network to density maps containing the density information.

METHODS

Inspired by successful examples of object counting in the deep learning literature (Xie *et al.*, 2018a; He *et al.*, 2021), we propose to train a deep and Fully Convolutional Neural Network (FCNN) to map input cell images to density maps, as shown in Fig. 4. More precisely, we used a U-net architecture (Ronneberger *et al.*, 2015), the details of which are provided below. The link between counting and density maps is that the sum of a density map pixel values equals the number of objects to count. This approach has the advantage to provide an interpretable object, the density map, to support the count inference, hence allowing us and the future users of the algorithm to control its soundness. Moreover, inferring a density map allows a more compliant behaviour as it is able to infer intermediate densities in case of uncertainty, for example when two cells overlap or if one has a peculiar shape.

INPUT IMAGES AND ANNOTATIONS

Since almost all the visual information is contained in the green channel of our images, we chose to use the latter as input of the network to avoid unnecessary complexity. Note that tests were also run using all three channels in case information coming from the blue and red channels might help, but this brought

no improvement and even impaired performance. The spatial resolution was also reduced by four, as images were downscaled from 1024×1024 to 256×256 pixels (Fig. 4a). The reason for this choice was to increase the batch size during training (see Section 4.3) and, again, tests with full resolution images and smaller batches showed lower performances.

The ground-truth density images, representing the output expected from the network, were built from the manual annotation of the cells coordinates (Fig. 4b), as follows. A 2D isotropic Gaussian function with covariance matrix $\sigma^2 I_2$ was normalized (to have integral 1) and translated at each cell location, before all these translated Gaussian functions were summed up. Then the resulting image was multiplied by 100. In that way, the integral of the resulting image (see Fig. 4c) was set to be $100C$, where C is the true number of cells contained in the corresponding input image. This factor 100 was used to ensure the density images were not too "flat", i.e. that their maxima were not too small. Indeed, we observed that this numerical trick helped the training process to converge (see Section 4.3). The parameter σ , which rules the width of the Gaussian blobs, was also to be set. It is not obvious to choose its optimal value for the method *a priori*, although one can assume it should roughly match the typical dimension of cells in the images.

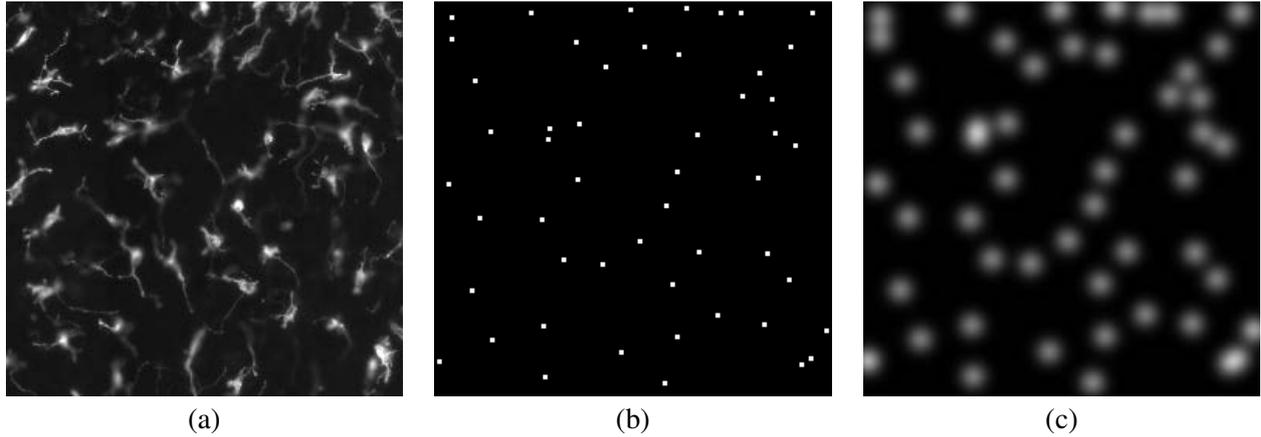


Fig. 4. An example of input and associated density map used to train the FCNN. (a) Green channel, 256×256 pixels. (b) Cells coordinates marked manually. (c) Corresponding target density map.

After trying several values on the training set, we finally opted for $\sigma = 6$ pixels for the downsampled 256×256 pixels images, which corresponds approximately to $10\mu\text{m}$. Fig. 4 shows an example of density image built this way.

ARCHITECTURE

The FCNN we designed for the image-to-density-map task is a U-net (Ronneberger *et al.*, 2015) with the following custom features, that are illustrated by Fig. 5. The first layer counts 32 filters and the network includes three downsampling steps followed by the corresponding three upsampling steps. In the contraction path of the network, the double convolutions followed by a max-pooling are replaced by a 3×3 convolution with stride 1 followed by a 3×3 convolution with stride 2. In the expansion path, the upsampling is achieved by a bilinear interpolation followed by a 3×3 convolution. The activation function following each convolution is a *leaky ReLU* defined by $f_{\alpha}(x) = \mathbb{1}_{\mathbb{R}^{-}}(x)\alpha x + \mathbb{1}_{\mathbb{R}^{+}}(x)x$, with $\alpha = 0.2$. Besides, a batch normalization is applied on each output of the leaky ReLU. Regarding the depth of the U-net (the number of downsampling steps in the contracting path), the choice of setting it to three was not arbitrary. Experiments were carried out to assess the effect of depth on accuracy. They showed that the performance increased with depth up to three, but no significant improvement was observed for a depth of four.

TRAINING PROTOCOL

Training data. To train the U-net presented earlier, we used the first set of 76 images. This set was split into a *training set* of 60 images, and a *validation set* of 16 images. As usual, the parameters of the network were updated based on the evaluation of the

loss function on the training set, whereas the evaluation on the validation set helped monitor the generalization ability of the network.

Data augmentation. It is reasonable to assume that the organisation of melanocytes in a skin sample is isotropic in the focal plane. This means that for any image like one of Fig. 1, its rotated and flipped versions still look like realistic data. Furthermore, we observed in the data that some cells appear blurry when they are slightly out of the focal plane of the microscope. Both observations allow us to artificially produce new images from the available ones. Convolutional neural networks are translation invariant but, in general, not invariant to rotations or mirror symmetries. This means that an image and its rotated or mirrored versions are seen as different by the network. Hence, adding to the dataset rotated, mirrored or blurred versions of the initial images allows to train the network with more data than initially available and may therefore increase the performances of the algorithm. We implemented this data augmentation by uniformly sampling over the set of combined transformations mentioned above. The blurring was done through a Gaussian convolution, the scale of the Gaussian kernel being uniformly sampled in $[0.5, 2.5]$.

Loss function. The training phase consisted in minimizing the L^1 distance between each image \hat{y} output by the network and the corresponding expected density map y , that is

$$\ell(y, \hat{y}) = \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (1)$$

where y_i and \hat{y}_i denote the values at pixel i of both images respectively, and $N = 256^2$ is the number of pixels in each image. Note that our actual target is that the estimated number of cells \hat{c} be as close as

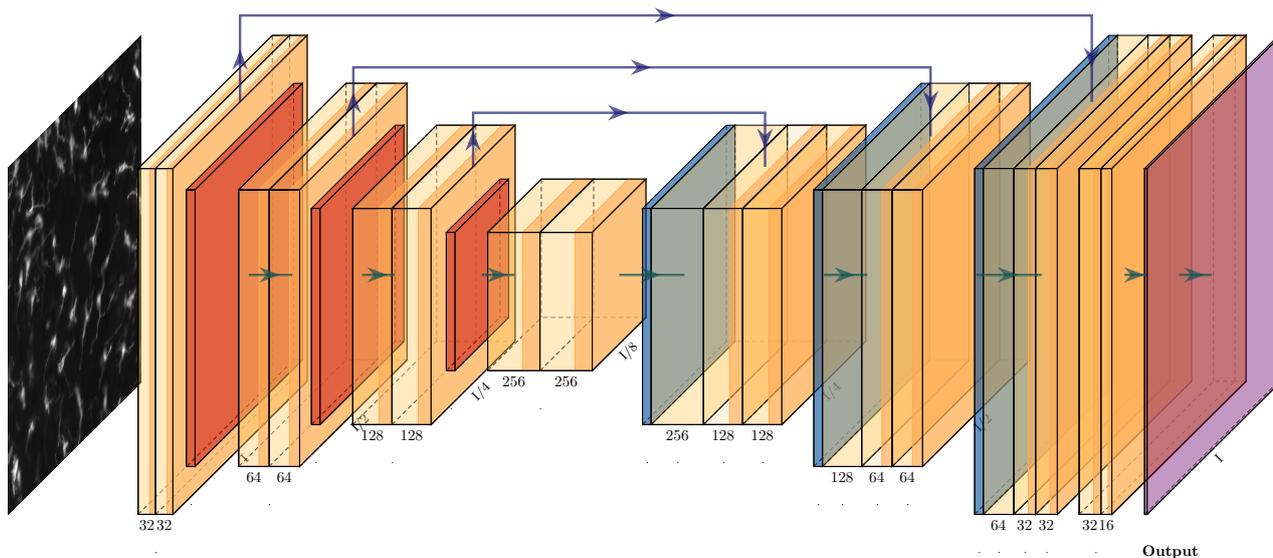


Fig. 5. The U-net architecture we used.

possible to the real number c , but since $c = \sum_{i=1}^N y_i$ and $\hat{c} = \sum_{i=1}^N \hat{y}_i$, we get by triangular inequality that $|c - \hat{c}| \leq \ell(y, \hat{y})$. Therefore, it is sufficient to minimize $\ell(y, \hat{y})$ for our purpose and, as said earlier, we get more by doing so: indeed, density maps can represent uncertainty and they are interpretable.

Optimization. The minimization of the loss function with respect to the parameters of the network is based on the principle of stochastic gradient descent. More precisely, we used the Adam optimizer (Kingma and Ba, 2015) with a learning rate $\eta = 2 \cdot 10^{-4}$ and PyTorch default momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The update of the network parameters was done after each evaluation on batches of 16 images, based on the average loss over the batch. The optimization process was carried on until no (or too little) improvement was observed on the validation loss for 2500 epochs.

EVALUATION METRICS

Average relative error. As a quality measure, we compute the relative counting error achieved by the algorithm. A natural way to do so is to take the ratio of the absolute difference between estimated and true number over the same true number of cells, taken as reference. Mathematically, this writes $|c - \hat{c}|/c$, where c and \hat{c} denote respectively the true and the estimated number of cells in an image. To assess the global performance of the algorithm, we could compute this ratio for each test image and take the average over images. For a test set composed of N_S images, this *average relative error* writes

$$A_{err}(S) = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{|c_i - \hat{c}_i|}{c_i}, \quad (2)$$

where c_i and \hat{c}_i are the true and estimated numbers of cells for image i .

Although intuitive and often used (He *et al.*, 2021), this measure is questionable. First it completely ignores the different numbers of cells to be found in different images. Suppose for example a set of two images, one with 100 cells and the other one with only one cell, and that the algorithm estimates 90 cells in the first image and does not see any in the second. In that case $A_{err}(S) = 55\%$, which seems very unfair, especially considering that the images come from much larger samples and that the split into sub-images is rather arbitrary. Second, the relative error for an image is not defined as soon as $c_i = 0$, which does occur. In this case, we may count this as a relative error of 100% if the algorithm counted at least one cell in this image, and zero otherwise, which is again questionable as it does not distinguish between over-counting by one, two or more cells. Finally, it does not evaluate over- and under-estimations the same way: counting one cell when there were two yields a ratio of 50%, whereas counting two when there was only one yields a relative error of 100% (and likewise errors even larger than 100% can occur).

Total relative error. For all these reasons, we propose an alternative quality measure that we call *total relative error*. It consists in summing all the errors made over all images and taking the ratio over the total number of cells to be counted in the set of images. This is close to pretending that the images of the test set

come from one large skin sample, and that the target number is the total number of cells in this sample - except that in this measure the absolute errors are summed and therefore over- and under-estimations can never compensate, contrary to what would happen in a large skin sample. This new quality measure now writes:

$$T_{err}(S) = \frac{\sum_{i=1}^{N_S} |c_i - \hat{c}_i|}{\sum_{i=1}^{N_S} c_i}. \quad (3)$$

Although we believe this measure makes more sense than the average one (eq. 2), we report both in the Results section to avoid any choice that could seem biased.

RESULTS

In Fig. 6, we show the output density maps and associated counted number of cells for four different input images, coming from the test set. Qualitatively, these examples already show the soundness of the deep learning algorithm, even in very difficult cases. Indeed, the output density maps seem always very correlated to the expected ones, and the gap between estimated and true number of cells is always within what we could expect at a given difficulty level.

The quantitative results of the approach are reported in Table 1. As we can see, there is a significant overfitting to the training set. However, this did not impair the performance on the validation set (the training would have been stopped earlier otherwise), and the errors on both validation and test sets are still under or about 10%, which was considered as satisfactory by the final users. Note also that it is consistent with the best reported results in cell counting on different benchmark datasets: between 8% and 15% in He *et al.* (2021).

Table 1. Summary of the average and total relative errors in cells counting on the different image sets, for the deep learning approach.

	Training	Validation	Test
A_{err} (%)	0.3	8.36	9.28
T_{err} (%)	0.4	10.4	8.72

Furthermore, these errors are an overestimation of the overall error that would be made on a large skin sample, where local over and under detections can compensate.

Finally, we show how the method can be simply embedded and applied in a lab routine for melanocyte density estimation. Indeed, the algorithm can be applied directly to large images (to a whole skin sample in theory, up to RAM limitations). Fig. 7 is

an example of large image (30000×26624 pixels) split into four sub-images for memory reasons. The cells could be counted just like in small images, as the resolution is the same, and the density of cells could be estimated locally as well as globally. The estimated number of cells of the whole quarter sample is about 16,700, and the density about $245 \text{ cells.mm}^{-2}$, with a maximum of $440 \text{ cells.mm}^{-2}$ for the lower left hand corner and a minimum of 40 cells.mm^{-2} for the top right hand corner (note that a mask of the quarter sample was computed beforehand so the densities exclude dark background area surrounding the sample). These computations took four minutes, including loading images and writing outputs, with an Intel Xeon CPU E5-2640 v3, 2.60GHz processor base frequency (and without GPU).

DISCUSSION

The method presented here shows a great setup simplicity, as it requires a relatively small annotated set, and implementation and training of the U-Net architecture are straightforward. At the same time, the obtained performance on our dataset matches both the users' expectations and the cell counting community standards. However, better performance could probably be achieved by increasing the sophistication of the approach. For example, adding multiscale auxiliary networks at the output of the right hand blocks of the U-Net, as proposed in He *et al.* (2021), looks promising since an improvement of about three points in average relative errors is reported for two challenging cell datasets. But this comes with more parameters to train and more hyperparameters to set, since two more losses need to be weighted in the overall loss function. Replacing isotropic Gaussian functions by anisotropic ones following the cells main orientations, like in Sierra *et al.* (2020), could also be an option, but this would require a more complex manual annotation, as the bounding box of each cell is needed.

CONCLUSION

In this paper, a cell counting method has been adapted to deal with a challenging database. This confirms the interest of adopting a regression approach to estimate a local cell density: it reaches a satisfactory trade-off between annotation effort and counting precision. We also show how to apply the method to large field images, which opens new perspectives for whole skin samples characterisations.

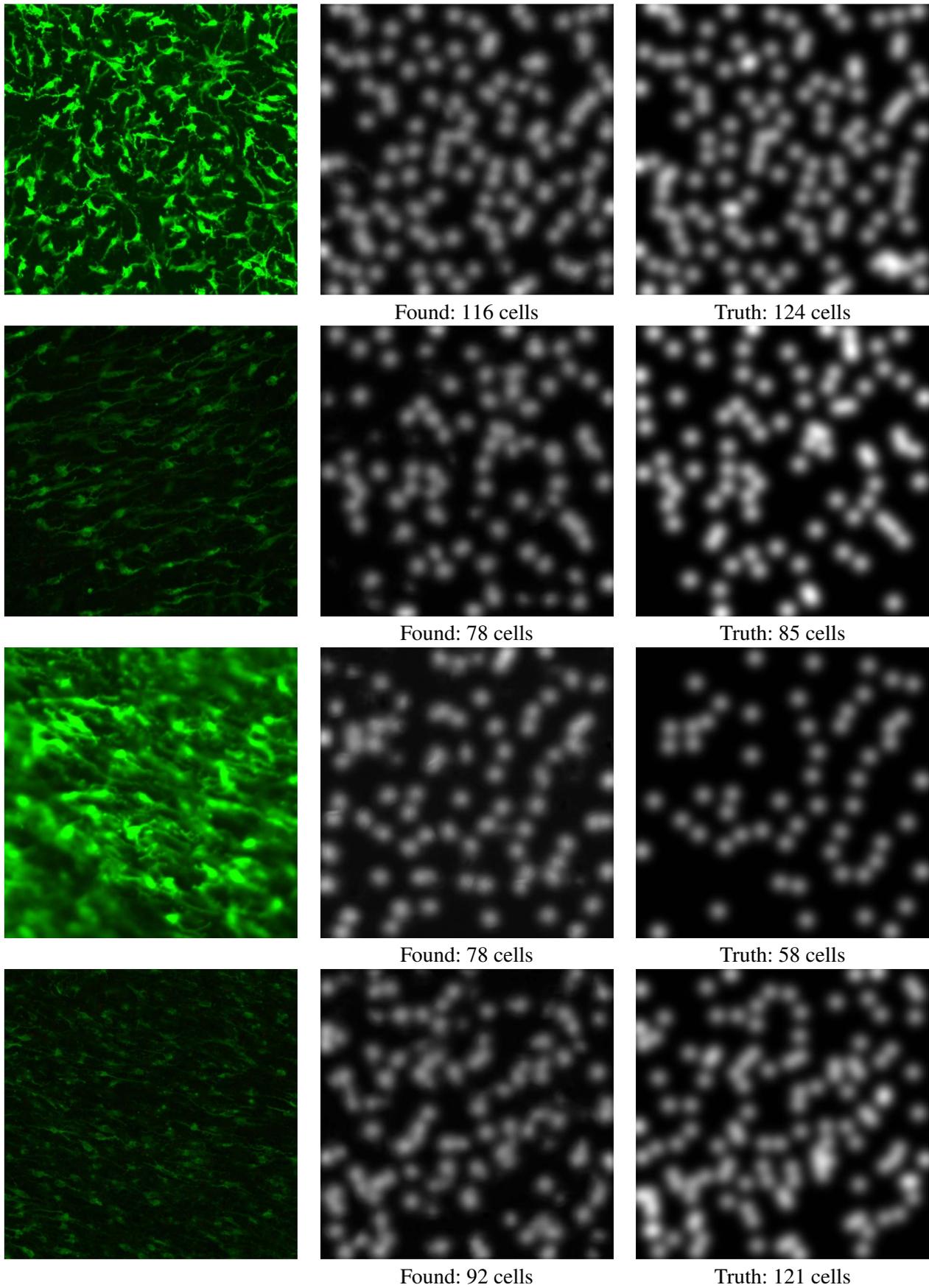


Fig. 6. Examples of results obtained on images from the test set. From left to right: input images, corresponding output with the counted number of cells, ground truth density map with the correct number of cells.

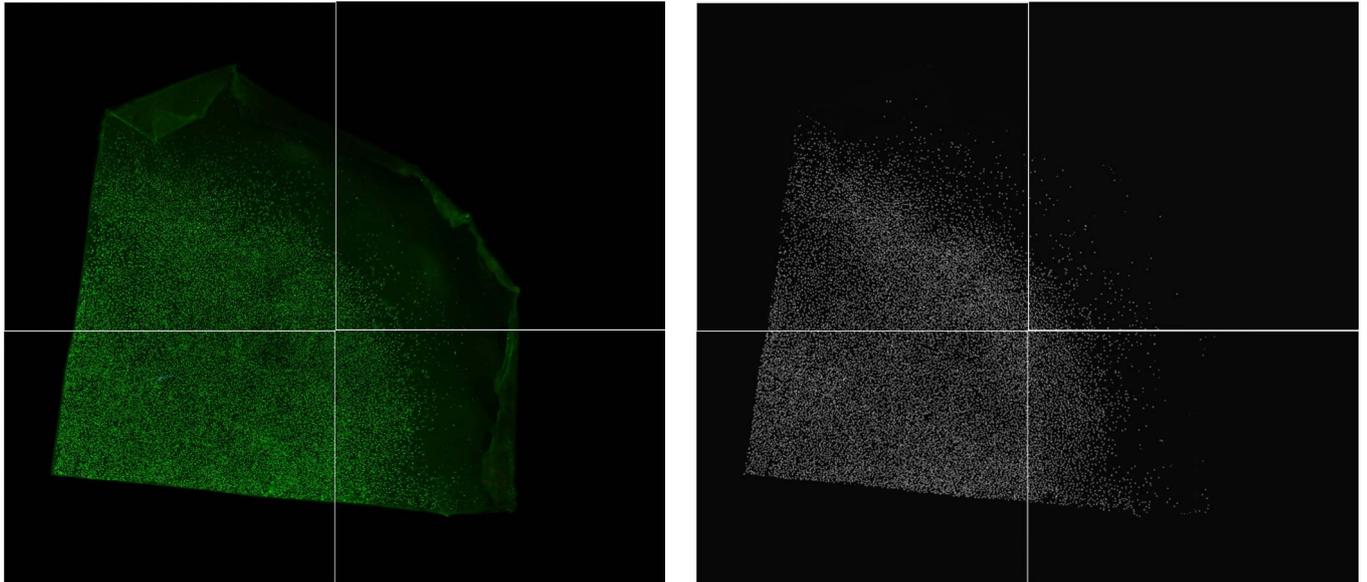


Fig. 7. Example of a quarter of skin sample, covering a 30000×26624 pixels image split into four sub-images, on which our algorithm was applied.

Moreover, the database used in this study is shared with the research community.

We believe that the next challenge in this type of problem is reducing even more the annotation effort. To that aim, we envision two possible research paths: 1) using active learning during annotation, as to precisely adapt the effort to the task at hand; 2) using generative adversarial networks.

REFERENCES

- Arteta C, Lempitsky V, Zisserman A (2016). Counting in the wild. In: Leibe B, Matas J, Sebe N, Welling M, eds., Proceedings of the European Conference on Computer Vision (ECCV). Cham: Springer International Publishing.
- Berge H, Taylor D, Krishnan S, Douglas TS (2011). Improved red blood cell counting in thin blood smears. In: I S Biomed Imaging. IEEE.
- Chadha GK, Srivastava A, Singh A, Gupta R, Singla D (2020). An automated method for counting red blood cells using image processing. *Procedia Comput Sci* 167:769–78.
- Cohen JP, Boucher G, Glastonbury CA, Lo HZ, Bengio Y (2017). Count-ception: Counting by fully convolutional redundant counting. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops.
- Duval C, Cohen C, Chagnoleau C, Flouret V, Bourreau E, Bernerd F (2014). Key regulatory role of dermal fibroblasts in pigmentation as demonstrated using a reconstructed skin model: impact of photo-aging. *PLoS One* 9:1–25.
- Falk T, Mai D, Bensch R, Çiçek Ö, Abdulkadir A, Marrakchi Y, Böhm A, Deubner J, Jäckel Z, Seiwald K, *et al.* (2019). U-net: deep learning for cell counting, detection, and morphometry. *Nat Methods* 16:67–70.
- Han S, Lee S, Fu C, Salama P, Dunn KW, Delp EJ (2019). Nuclei counting in microscopy images with three dimensional generative adversarial networks. In: Angelini ED, Landman BA, eds., *Medical Imaging 2019: Image Processing*, vol. 10949. International Society for Optics and Photonics, SPIE.
- He S, Minn KT, Solnica-Krezel L, Anastasio MA, Li H (2021). Deeply-supervised density regression for automatic cell counting in microscopy images. *Med Image Anal* 68:101892.
- Khan A, Gould S, Salzmann M (2016). Deep convolutional neural networks for human embryonic cell counting. In: Hua G, Jégou H, eds., *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. Cham: Springer International Publishing.
- Kingma DP, Ba J (2015). Adam: A method for stochastic optimization. In: Bengio Y, LeCun Y, eds., *ICLR*.
- Laradji IH, Rostamzadeh N, Pinheiro PO, Vazquez D, Schmidt M (2018). Where are the blobs: Counting by localization with point supervision. In: *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Lempitsky V, Zisserman A (2010). Learning to count objects in images. *Adv Neur In* 23:1324–32.

- Liu Q, Junker A, Murakami K, Hu P (2019). A novel convolutional regression network for cell counting. In: 2019 IEEE 7th International Conference on Bioinformatics and Computational Biology (ICBCB).
- Paulauskaite-Taraseviciene A, Sutiene K, Valotka J, Raudonis V, Iesmantas T (2019). Deep learning-based detection of overlapping cells. In: Proceedings of the 3rd International Conference on Advances in Artificial Intelligence, ICAAI. New York, NY, USA: Association for Computing Machinery.
- Ronneberger O, Fischer P, Brox T (2015). U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, eds., Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham: Springer International Publishing.
- Schmitz C, Eastwood BS, Tappan SJ, Glaser JR, Peterson DA, Hof PR (2014). Current automated 3d cell detection methods are not a suitable replacement for manual stereologic cell counting. *Front Neuroanat* 8:27.
- Sierra JS, Pineda J, Viteri E, Tello A, Millán MS, Galvis V, Romero LA, Marrugo AG (2020). Generating density maps for convolutional neural network-based cell counting in specular microscopy images. *J Phys Conf Ser* 1547:012019.
- Xie W, Noble JA, Zisserman A (2018a). Microscopy cell counting and detection with fully convolutional regression networks. *Comput Method Biomec Imaging Visualization* 6:283–92.
- Xie Y, Xing F, Shi X, Kong X, Su H, Yang L (2018b). Efficient and robust cell detection: A structured regression approach. *Med Image Anal* 44:245–54.
- Zheng Y, Chen Z, Zuo Y, Guan X, Wang Z, Mu X (2020). Manifold-regularized regression network: A novel end-to-end method for cell counting and localization. In: Proceedings of the 2020 the 4th International Conference on Innovation in Artificial Intelligence, ICIAI 2020. New York, NY, USA: Association for Computing Machinery.