Counting melanocytes with trainable h-maxima and connected component layers

Xiaohu Liu, Samy Blusseau, Santiago Velasco-Forero

Counting melanocytes with trainable h-maxima and connected component layers

Xiaohu Liu [1,2], Samy Blusseau [2], and Santiago Velasco-Forero [2*]

[1] Shanghai Jiaotong University, Shanghai, China
[2] Mines Paris, PSL University, Centre for Mathematical Morphology, Fontainebleau, France

Abstract. Bright objects on a dark background, such as cells in microscopy images, can sometimes be modeled as maxima of sufficient dynamic, called h-maxima. Such a model could be sufficient to count these objects in images, provided we know the dynamic threshold that tells apart actual objects from irrelevant maxima. In this paper we introduce a neural architecture that includes a morphological pipeline counting the number of h-maxima in an image, preceded by a classical `CNN` which predicts the dynamic h yielding the right number of objects. This is made possible by geodesic reconstruction layers, already introduced in previous work, and a new module counting connected components. This architecture is trained end-to-end to count melanocytes in microscopy images. Its performance is close to the state of the art `CNN` on this dataset, with much fewer parameters (1/100) and an increased interpretability.

1 Introduction

Cell counting is a crucial step in biological experiments and medical diagnosis to provide both quantitative and qualitative information on cells. While the process can be tedious, low-efficiency and prone to subjective errors, especially when cell clustering occurs and cells show high variance in shapes and contrasts. Automation offers a quick and accurate estimation of cell quantity in a sample. In this study, we focus on the cell counting task to quantify melanocyte population on fluorescent images with *Tyrosinase-related protein one* (TRP1) as melanocytic marker (see left hand image of Fig. 1). Previous studies [5,6] used deep learning with a U-Net architecture, which associates each input image with a density map, the integral of which yields the cell count. That method achieves satisfactory results but with almost two million parameters while cells, although diverse in shapes and often overlapping, seem to be well approximated by regional maxima over a dark background. Furthermore, the density map is ambiguous regarding what the model considers as a cell, especially in clusters, where the location of cells is often lost. This can be a limitation for users willing

* This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011012212R2 made by GENCI.

to check a posteriori the reliability of the count returned by the model, especially as the deep learning model is a black box and its decisions are hardly explainable.

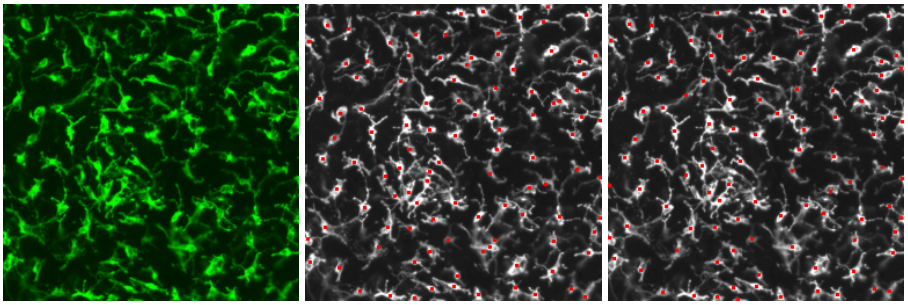
Therefore, this is an interesting study case where it seems that a much simpler model, built on *a priori* knowledge, could achieve similar results with higher interpretability, regarding for example geometrical and contrast criteria used to recognize cells. In this paper, we propose to model cells as regional maxima with sufficient dynamic and size. The dynamic of an extremum is, simply speaking, its depth, and well known morphological methods exist to select extrema with dynamic larger than a threshold h, called h-maxima or h-minima [10]. Hence a simple morphological pipeline is presented, consisting in counting the h-maxima of a size-filtered version of the input image, for a given dynamic h. Since the image resolution is constant in the dataset, as well as the cell size, this approach only depends on the choice of the dynamic parameter, which needs however to be adapted to each image in order to achieve accurate counting. Thus, we take advantage of recent work [15] in which geodesic reconstruction, and therefore the computation of h-maxima, have been implemented as neural layers, allowing to train end-to-end a pipeline where a small Convolutional Neural Network (`CNN`) predicts the optimal dynamic h for the subsequent morphological layer.

The contribution of this paper is threefold. Firstly, a new end-to-end differentiable pipeline combining a trainable `CNN` and an h-maxima operator for cell counting is introduced. Secondly, we define a layer based on geodesical reconstruction, that counts connected components in binary images and is compatible with automatic differentiation [8]. Thirdly, we use a new joint loss function composed of differences on morphological geodesic reconstructions, and a term penalizing miscounts with respect to the true number of cells. This paper is structured as follows. The morphological method to count cells is presented in Section 2. A deep learning architecture based on the morphological method is developed in Section 3. Experiments are conducted in Section 4, the analysis of results in Section 5. Conclusions are drawn in Section 6.

2 Morphological pipeline

In this section we present a simple morphological algorithm for cell counting in images where individual cells are well approximated by regional extrema of significant dynamic, like in Figure 1. We assume images to be mappings of the discrete domain $\Omega \coloneqq ([0, M-1] \times [0, N-1]) \cap \mathbb{N}^2$, $M, N \in \mathbb{N}^*$ to the discrete set of values $\mathcal{V} \coloneqq [0, 1] \cap \epsilon\mathbb{N}$, where $0 < \epsilon < 1$. We will denote by $\mathcal{F}(\Omega, \mathcal{V})$ the set of such functions. The algorithm takes as input an image f, as parameter a number $h \in \mathcal{V}$ and returns an estimated number n_c of cells after running the following steps:

1. Apply an alternate filter $\tilde{f} \leftarrow \varphi_{\mathrm{B}} \circ \gamma_{\mathrm{B}}(f)$ where φ_{B} and γ_{B} are respectively the closing and opening by B, the unit square structuring element
2. Compute the h-extended maxima $M_h \leftarrow \texttt{EMAX}_h(\tilde{f})$, which is a binary image
3. Count the number of connected components of M_h: $n_c \leftarrow \texttt{CCC}(M_h)$



Input $\mathtt{I}$ | Ground truth: 107 cells | Output: $n_c = 107$ for $h^*_{\mathtt{I}} = 54$

Fig. 1. Example of cell counting by identifying cells to h-maxima, in the TRP1 melanocyte dataset. From left to right: input RGB image (resized); ground truth mask for the 107 cells (in red) ; detection results using the best h for the image: $h^*_{\mathtt{I}} = 54$. Note that despite the perfect count, there are some false positives and false negatives (e.g. just below the the top right hand corner).

The definitions required for the key operator $\mathtt{EMAX}_h$ are recalled in the following paragraphs.

Morphological Geodesic Reconstruction Given two functions $f, g \in \mathcal{F}(\Omega, \mathcal{V})$ such that $f \leq g$ the *geodesic dilation* of f under g, noted $\delta^{(1)}_g(f)$, is defined by [10,7]

$$\delta^{(1)}_g(f) := \delta_{\mathrm{B}}(f) \wedge g \tag{1}$$

where $\wedge$ denotes the point-wise minimum operation, B denotes the unitary structuring element defined according to the pixel connectivity. Recall that $\delta_{\mathrm{B}}(f)(x) = \max_{b \in B} f(x + b)$ for all $x \in \Omega$, therefore δ_{B} is an extensive operator, $\delta_{\mathrm{B}}(f) \geq f$, since $0 \in \mathrm{B}$. The geodesic dilation can be iterated and for $p \geq 1$ we note $\delta^{(p+1)}_g(f) := \delta^{(1)}_g\big(\delta^{(p)}_g(f)\big)$.
The *reconstruction by dilation* of f with respect to g, $\mathtt{REC}^{\delta}(f, g)$ is then:

$$\mathtt{REC}^{\delta}(f, g) := \mathtt{REC}^{\delta}_g(f) := \delta^{(k)}_g(f) \tag{2}$$

where k is the first integer such that $\delta^{(k)}_g(f) = \delta^{(k+1)}_g(f)$, hence the dilation iterates until stable.

Regional maxima by reconstruction The reconstruction by dilation in (2) can be used to extract regional maxima [16]. Let us consider an image $f \in \mathcal{F}(\Omega, \mathcal{V})$. $M \subset \Omega$ is a *regional maximum* at level t if M is connected, $f(x) = t \ \ \forall x \in M$, and for any $y \in \Omega \setminus M$ with a neighbor in M, $f(y) < t$. It is well-known (refer to section 6.3.3 in [10]) that the set of all regional maxima of f, denoted by $\mathtt{RMAX}(f)$ is recovered by:

$$\forall x \in \Omega, \quad \mathtt{RMAX}(f)(x) := \begin{cases} 1 & \text{if } f(x) > \mathtt{REC}^{\delta}_f(f - \epsilon)(x) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

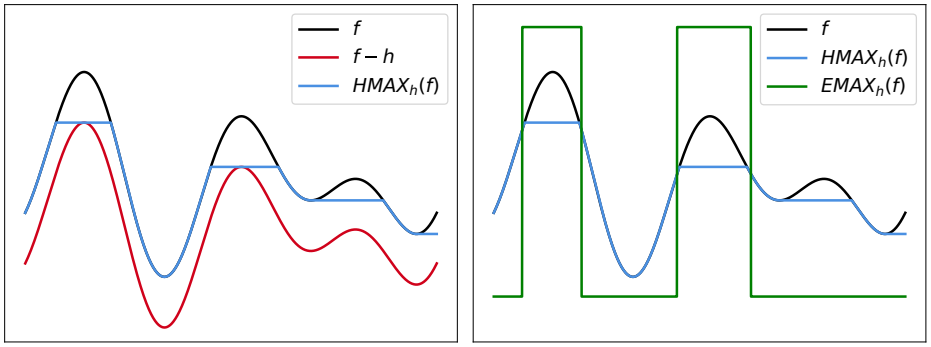
where we recall that ϵ is the minimum positive value in $\mathcal{V}$ and therefore the minimum absolute difference between distinct function values. By construction, every connected component of the binary image $\texttt{RMAX}(f)$ is a regional maximum.

Extended h-maxima by reconstruction The *dynamic* of a regional maximum M (of level t) is the smallest height one needs to come down to reach a higher regional maximum M' (of level $t' > t$) (refer to Section 6.3.5 in [10]). By convention, the dynamic of the global maximum of f can be set to the difference between the maximum and the minimum values of f. The dynamic is usually used to distinguish between irrelevant maxima caused by noise and significant ones corresponding to underlying bright objects.

The *h-maxima transformation* suppresses all maxima with dynamic lower or equal to the given parameter value h, called also *h-domes* in [16]. This can be achieved by performing the reconstruction by dilation of f from $f - h$, i.e:

$$\texttt{HMAX}_h(f) = \texttt{REC}_f^{\delta}(f - h) \tag{4}$$

where $h \in \mathcal{V}$ is a parameter. To each regional maximum of f with dynamic strictly larger than h, corresponds exactly one regional maximum in $\texttt{HMAX}_h(f)$. This is illustrated in Figure 2 (a).



(a) (b)

Fig. 2. (a) $\texttt{HMAX}_h$ produces a regional maximum wherever there is a maximum with dynamic greater than h in f. (b) $\texttt{EMAX}_h$ computes the regional maxima of $\texttt{HMAX}_h(f)$.

Therefore, we define the operator *extended maxima* as the regional maxima of the h-maxima transformation:

$$\texttt{EMAX}_h(f) := \texttt{RMAX}(\texttt{HMAX}_h(f)). \tag{5}$$

By construction, the number of connected components of the binary image $\texttt{EMAX}_h(f)$ is the same as the number of regional maxima with dynamic strictly larger than h in f. This is illustrated by Figure 2 (b). We can see that this

number n_c is a decreasing function of h, ranging from the total number of local maxima (for $h = 0$) to zero (for h exceeding the dynamic of the deepest maximum). This is why we expect that for each image, there is an h value yielding the right number of cells.

3 Integration in a deep learning architecture

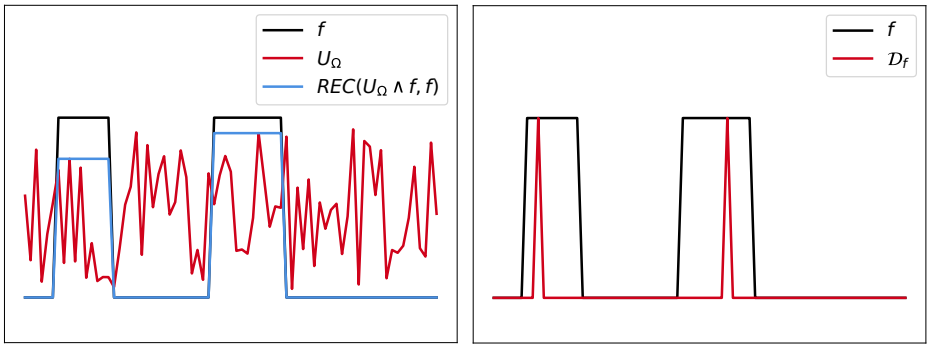
The morphological pipeline described in the previous section depends on one crucial parameter, namely the minimum dynamic for a regional maximum to be considered a relevant object (in our case, a melanocyte). In this section we describe how this pipeline is implemented as a sequence of neural layers, and embedded in an architecture where it is preceded by a `CNN` which predicts an optimal dynamic for each image, in order to count the right number of cells at the output of the network.

Morphological layers This embedding relies on the library `Morpholayers`[13] which implements neural layers performing morphological operators [14,9]. This includes basic operators such as dilations, erosions, openings and closings, useful for the alternate filter in the first step of our pipeline. Furthermore, it also includes the geodesic reconstruction layers recently introduced in the deep learning framework [15], and on which relies most of the operations described in the previous section. It is important to note that the reconstruction by dilation (2), has no parameter because the value k depends on f and g. It can be included as a layer in a neural network since the subdifferential of this operation can be implemented in auto-differentiation software [8]. The Jacobian matrix of (2) has been explicitly calculated in [15]. Therefore, all the necessary layers are available to embed steps 1 and 2 of the morphological pipeline. The embedding of the third step is addressed in the next paragraph.

Counting Connected Components (`CCC`) Layer Counting connected components is an important step in many of the classical methods in data processing [11], especially in images and graphs. In the context of deep learning, this problem has been used to evaluate the generalization capacity of neural networks [4]. However, to be used as a layer inside a network, it must be implemented so that gradient backpropagation can be computed in an auto-derivation software such as Pytorch or Tensorflow [8]. In this subsection, we present an algorithm to count the connected components of a binary image, relying mainly on the geodesic reconstruction defined in (2).

Let $f \in \mathcal{F}(\Omega, \{0,1\})$ be a binary image, and $\mathcal{U}_\Omega \in \mathcal{F}(\Omega, \mathcal{V})$ an injective and positive image, i.e. such that $\forall x, y \in \Omega, \quad x \neq y \Rightarrow \mathcal{U}_\Omega(x) \neq \mathcal{U}_\Omega(y)$ and $\mathcal{U}_\Omega(x) > 0$ (this is possible if and only if the number of values in $\mathcal{V}$ is strictly larger than the number of pixels in Ω, i.e. $\left\lfloor \frac{1}{\epsilon} \right\rfloor + 1 > MN$). Then it is clear that reconstructing $\mathcal{U}_\Omega \wedge f$ under f will label every connected component `CC` of f with the maximum value taken by $\mathcal{U}_\Omega$ in `CC`. Since $\mathcal{U}_\Omega$ is injective, this maximum is

achieved only once in each connected component. Hence, setting to one each pixel $x \in \Omega$ such that $\texttt{REC}^\delta(\mathcal{U}_\Omega \wedge f, f)(x) = \mathcal{U}_\Omega(x)$ produces a binary image where exactly one pixel per connected component of f is lit. This is illustrated by Figure 3.



(a) (b)

Fig. 3. Illustration of the CCC algorithm. (a) $\mathcal{U}_\Omega$ coincides only once with $\texttt{REC}^\delta(\mathcal{U}_\Omega \wedge f, f)$ inside each connected component, and never outside the connected components. (b) Hence $\mathcal{D}_f$ has only two pixels activated, one per connected component of f.

Therefore, we define

$$\mathcal{D}_f := \mathcal{B}\big(\mathcal{U}_\Omega, \texttt{REC}^\delta(\mathcal{U}_\Omega \wedge f, f)\big) \tag{6}$$

where the *binarization operator* $\mathcal{B}$ for two functions $f, g \in \mathcal{F}(\Omega, \mathbb{R})$ is:

$$\forall x \in \Omega, \quad \mathcal{B}(f, g)(x) := \begin{cases} 1 \text{ if } \quad f(x) = g(x) \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

$\mathcal{D}_f$ denotes the final detection result, represented by locations of the isolated connected components. Accordingly, one can count the number of connected components on f by simply summing (6):

$$\texttt{CCC}(f) := \sum_{x \in \Omega} \mathcal{D}_f(x). \tag{8}$$

4 Experiments

Data The TRP1 dataset [6] we used contains two sets, called *set*1 and *set*2, of 76 fluorescent melanocytes RGB images each, with resolution 1024×1024 pixels, which showed a high variability in density and shapes[3]. They come along

[3] The dataset is available at `https://bit.ly/melanocytesTRP1`.

with manual annotations of the cell coordinates. The images were acquired from *in vitro* pigmented reconstructed skin samples submitted to TRP1 fluorescent labeling [3]. Figures 1 and 7 show examples of images and manual annotations. In the present experiments we only worked with the green channel of images, rescaled at a resolution of 255×255 pixels. The image range was between 0 and $V_{\max} = 255$, quantized with a float precision ϵ. Hence the previously described method applies, taking $\mathcal{V} := [0, 255] \cap \epsilon\mathbb{N}$, and binary images with values in $\{0, 255\}$. Applying the morphological pipeline described in Section 2 for a wide range of dynamic values h, allowed to determine for each training image $\mathtt{I}$ the optimal dynamic $h^*_{\mathtt{I}}$, i.e. the one minimizing the difference between true number of cells and the estimated one.

Overall architecture The proposed neural architecture[4] is illustrated in Figure 5. Morphological layers apply an alternate filter (opening followed by closing) by a 3×3 square structuring element, to the resized green channel, which yields the first preprocessed image $\mathtt{I}$. This image is passed as input to a $\mathtt{CNN}$, which estimates the best dynamic $\hat{h}_{\mathtt{I}} = \mathtt{CNN}_\theta(\mathtt{I})$, where θ denotes the parameters of the $\mathtt{CNN}$. This parameter is then used to compute the $\hat{h}_{\mathtt{I}}$-maxima of the filtered image, by first feeding it to the $\mathtt{HMAX}$ reconstruction layer, followed by the $\mathtt{RMAX}$ one to get the binary image $\mathtt{EMAX}_{\hat{h}_{\mathtt{I}}}$. Finally, the number of connected components of the latter is obtained at the output of the proposed connected components counting layer, $\mathtt{CCC}(\mathtt{EMAX}_{\hat{h}_{\mathtt{I}}})$.

$\mathtt{CNN}$ architecture The $\mathtt{CNN}$ architecture is summarized in Fig. 4. In particular, each convolutional layer[5] is followed by a BatchNormalization layer with Relu as activation function. Following [15], global average pooling layer is used after the second Maxpooling layer instead of stacked flatten and fully connected layers, which has the advantage of further reducing the number of parameters of the network. What's more, extracting global feature information is coherent with the objective of our task.

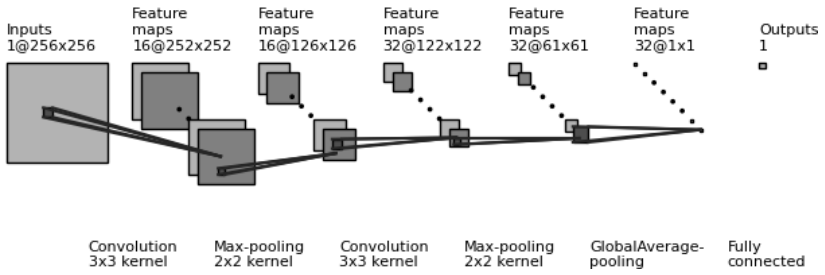


Fig. 4. $\mathtt{CNN}$ architecture used in the proposed pipeline

[4] Code available at https://github.com/peter12398/DGMM2024-comptage-cellule.

[5] Convolutional layers are implemented as a double convolution with twice the same number of filters.

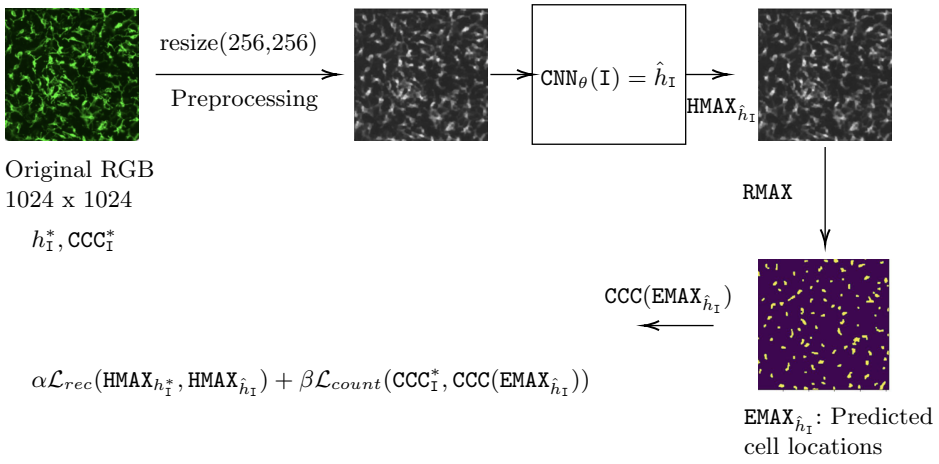


Fig. 5. Proposed end-to-end differentiable pipeline using a CNN and an HMAX layer to count melanocytes. The network is trained by minimizing a loss function that depends on 1) the difference between the geodesic reconstruction with the dynamic $\hat{h}$ estimated by the network, and the best possible one; and 2) the difference between the number of cells and the number of extended maxima of $\texttt{HMAX}_{\hat{h}}$.

4.1 Training protocol

Training data To train the architecture presented earlier, hence fit the parameters of the CNN, we randomly split *set*1, containing 76 images, into 60 training images and 16 validation images. The parameters of the CNN were adjusted through gradient descent on the training set, and the best model weights were chosen through evaluation metrics on the validation set. The whole *set*2, which also contains 76 images, was used for the final test only.

Data augmentation Convolutional operations are translation invariant, but not rotation or flip invariant. Moreover, for the melanocyte image samples in the dataset, the flipped or rotated version still looks like a valid sample. Thus it is reasonable to use flip and rotation as data augmentation methods in the training stage. More precisely, we used both vertical and horizontal flip transformations with probability 0.5 ; a rotation with the same probability and the rotation angle was uniformly selected from $\{90°, 180°, 270°\}$.

Loss function The training phase aims at minimizing the joint loss function composed of a geodesic reconstruction loss and a cell counting loss. More precisely, the geodesic reconstruction loss is the mean squared error (MSE) between the predicted geodesic reconstruction image $\texttt{HMAX}_{\hat{h_\texttt{I}}}$ and the ground truth one $\texttt{HMAX}_{h^*_\texttt{I}}$:

$$\forall \texttt{I} \in \mathcal{F}(\Omega, \mathcal{V}), \quad \mathcal{L}^{\texttt{I}}_{rec} = \frac{1}{|\Omega|} \sum_{x \in \Omega} (\texttt{HMAX}_{\hat{h}_\texttt{I}}(x) - \texttt{HMAX}_{h^*_\texttt{I}}(x))^2. \tag{9}$$

The cell counting loss is the mean absolute error (MAE) between the estimated number of cells $\texttt{CCC}(\texttt{EMAX}_{\hat{h}_\texttt{I}})$ and the true number of cells $\texttt{CCC}^*_\texttt{I}$:

$$\mathcal{L}^\texttt{I}_{count} = \left|\texttt{CCC}(\texttt{EMAX}_{\hat{h}_\texttt{I}}) - \texttt{CCC}^*_\texttt{I}\right|. \tag{10}$$

The joint loss writes:

$$\mathcal{L}^\texttt{I} = \alpha\mathcal{L}^\texttt{I}_{rec} + \beta\mathcal{L}^\texttt{I}_{count} \tag{11}$$

where $\alpha > 0$ and $\beta > 0$ control the importance of each of the terms in the loss function. In our experiments we have used $\alpha = 1, \beta = 0.001$. These were set after five-folds cross validation experiments on *set*1, where 11 ratios β/α were tested, ranging from 10^{-4} to 10^2, including zero[6]. Additional loss terms were also tested, like the binary cross entropy between between $\texttt{EMAX}_{\hat{h}_I}$ and $\texttt{EMAX}_{h^*_I}$, but no improvement was found.

Optimization The joint loss function was minimized with stochastic gradient descent. More precisely, we used the RMSProp optimizer [12], with initial learning rate of 10^{-3}. Moreover, we used a batch size of 16 images and trained for 1600 training epochs. Only the model weights with lowest validation error were saved and used for final evaluation on the test set.

4.2 Evaluation metrics

Following [15], we used two metrics for the test phase to evaluate our proposed pipeline. One intuitive metric is the *average relative error*, which averages the per-image relative counting error:

$$\mathcal{A}_{err}(\mathcal{S}) = \frac{1}{\mathcal{N}} \sum_{\texttt{I}\in\mathcal{S}} \frac{\left|\texttt{CCC}(\texttt{EMAX}_{\hat{h}_\texttt{I}}) - \texttt{CCC}^*_\texttt{I}\right|}{\texttt{CCC}^*_\texttt{I}} \tag{12}$$

where $\mathcal{S}$ denotes the test set with sample number $\mathcal{N}$. However, considering the variability in cell numbers of our dataset, we also monitor the *total relative error* [6,5], which sees the whole dataset as one skin sample, but avoids compensations of errors by taking an absolute difference in each image:

$$\mathcal{T}_{err}(\mathcal{S}) = \frac{\sum_{\texttt{I}\in\mathcal{S}} \left|\texttt{CCC}(\texttt{EMAX}_{\hat{h}_\texttt{I}}) - \texttt{CCC}^*_\texttt{I}\right|}{\sum_{\texttt{I}\in\mathcal{S}} \texttt{CCC}^*_\texttt{I}}. \tag{13}$$

We report evaluation results on both metrics to prevent comparison bias. It is worth noting that finding the right number of cells does not imply correct detections, as false positives can compensate false negatives, like illustrated in Figure 1. However, we have deliberately limited our analysis to counting errors, as counting is the primary objective. All counting methods benefit from this type of compensation, which is only possible if the method is not overly biased towards over or underestimation. Furthermore, future work could analyse reconstruction metrics, as a better reconstruction implies more accurate counting.

[6] Tested values: $\alpha = 1$ and $\beta \in \{0\} \cup 10^{\{-4,-3,-2,-1,0,1,2\}} \cup 5 \times 10^{\{-4,-3,-2\}}$.

5 Results

We present our quantitative results on the test set, along with numbers of parameters and pipeline interpretability, in Table 1, comparing them with the current state-of-the-art method proposed by Lazard *et al*[6]. Although our proposed pipeline does not surpass the latter in the two prediction metrics, the error rates we obtained are close and comparable to their methods. Furthermore, the absolute values of the error rates are around or below 10%, a threshold considered satisfactory by final users, as reported in [6]. It's noteworthy that the number of parameters in our pipeline is significantly lower than the state-of-the-art method using U-net (only less than 1/100 of theirs). This characteristic makes our approach suitable for scenarios with small training sets or with limited computational or storage resources, such as embedded systems or mobile devices.

Method	$\mathcal{A}_{err}$	$\mathcal{T}_{err}$	Network parameter#	Interpretability
Lazard *et al*[6]	9.28%	8.72%	1,760,000	end-to-end not interpretable
Ours	11.8%	9.35%	16,675	partially not interpretable

Table 1. Results of our proposed pipeline compared to the current state-of-the-art method.

Additionally, our proposed method provides better interpretability. While the final output density map of the U-net methods [6] may be considered interpretable, actually density maps do not locate precisely cells, especially when cells overlap. Furthermore, the process to derive a density map from the input melanocyte sample (namely, the U-net architecture) is end-to-end non-interpretable. This makes the criteria used by the network less easy to exploit by the final users. In contrast, for our proposed pipeline, only the `CNN` part (best h prediction module in Fig. 5) is not interpretable; for the other parts, only morphological criteria based on size and contrast are used. Detailed predicted and ground truth results for the dynamics and numbers of cells are displayed in Figure 6. We observe a stronger positive correlation between predicted and true cell numbers than between optimal and estimated h. This further demonstrates the robustness of the proposed algorithm with respect to the dynamic, as errors in the estimation of the optimal h often give an estimated count close to the true one.

Qualitative examples from the test set can be found in Figure 7. These examples already illustrate the soundness of our proposed algorithm. Even in difficult cases (low contrast or non-convex shapes), the predicted cell number and cell locations appear to be in good agreement with the ground truth ones, and the gap between them are always in our expectations under the given difficulty.

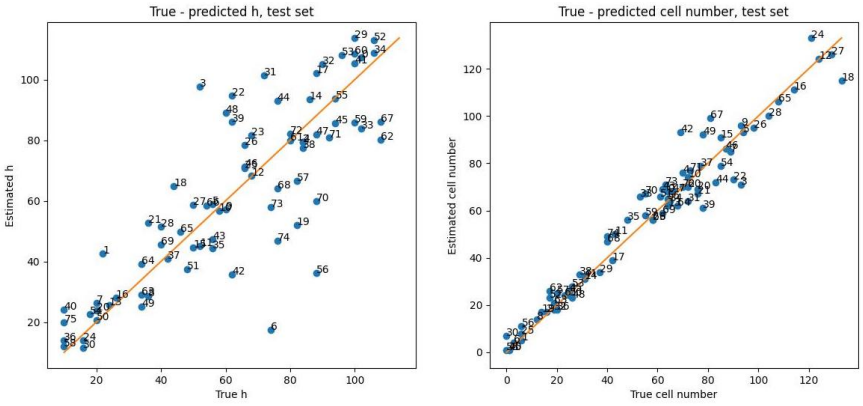


Fig. 6. The predicted $\hat{h}$ is compared to the ground truth h^* on the left, while the right hand plot compares the predicted and true cell numbers. Each dot represents an individual image, identified by its index in the test set. For a detailed analysis, refer to the corresponding discussion in the text of the paper.

6 Conclusion

In this paper, we have presented a new end-to-end trainable pipeline where a `CNN` predicts the dynamic threshold for h-maxima to be recognized as melanocytes, and a layer counts the connected components of the h-maxima images, yielding an estimated numbers of cells. Our pipeline is trained using a novel joint loss function, which comprises the geodesic reconstruction loss and a differentiable cell counting loss. Notably, our approach not only attains comparable results on the test set but also boasts a significantly reduced parameter count (1/100) and superior geometrical interpretability when compared to the state-of-the-art. Future endeavors could involve the prediction of *local* h values for smaller patches and the implementation of contrast enhancement for blurred regions. We posit that the synergy between neural networks and trainable morphological layers opens the door to a broader range of applications, such as counting astronomical objects in astronomical surveys. However, a deeper reflection is necessary to understand which are the best ways to initialize, optimize and regularize morphological layers [2,1]. Should we keep the last sentence?

References

1. Blusseau, S.: Training morphological neural networks with gradient descent: some theoretical insights. In: DGMM Proceedings. Springer-Verlag, Berlin, Heidelberg (2024)
2. Charisopoulos, V., Maragos, P.: Morphological perceptrons: geometry and training algorithms. In: ISMM 2017. pp. 3–15. Springer (2017)
3. Duval, C., Cohen, C., Chagnoleau, C., Flouret, V., Bourreau, E., Bernerd, F.: Key regulatory role of dermal fibroblasts in pigmentation as demonstrated using a reconstructed skin model: impact of photo-aging. PLoS One **9**(12), e114182 (2014)

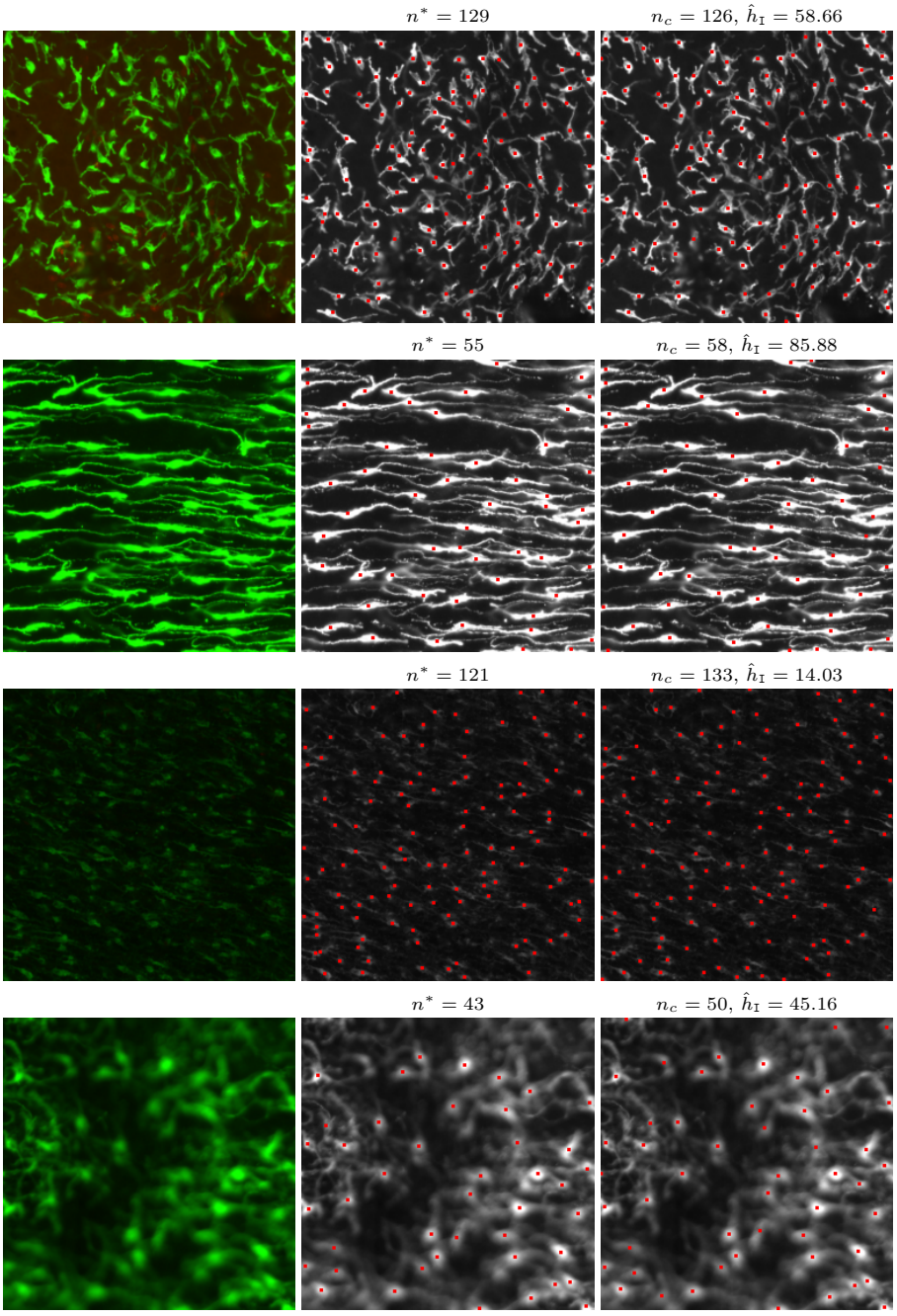


Fig. 7. Examples of results on the test set, with our proposed method. From left to right: Input resized RGB image, green channel image with ground truth cell locations and true number of cells n^*, green channel image with predicted cell locations and their estimated number n_c obtained for the estimated dynamic $\hat{h}_{\mathrm{I}}$.

4. Guan, S., Loew, M.: Understanding the ability of deep neural networks to count connected components in images. In: 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR). pp. 1–7. IEEE (2020)
5. He, S., Minn, K.T., Solnica-Krezel, L., Anastasio, M.A., Li, H.: Deeply-supervised density regression for automatic cell counting in microscopy images. Medical Image Analysis **68**, 101892 (2021)
6. Lazard, T., Blusseau, S., Velasco-Forero, S., Decencière, É., Flouret, V., Cohen, C., Baldeweck, T.: Applying deep learning to melanocyte counting on fluorescent trp1 labelled images of in vitro skin model. Image Analysis & Stereology (2022)
7. Najman, L., Talbot, H.: Mathematical morphology: from theory to applications. John Wiley & Sons (2013)
8. Rall, L.B.: Automatic differentiation: Techniques and applications. Springer (1981)
9. Sangalli, M., Blusseau, S., Velasco-Forero, S., Angulo, J.: Scale equivariant neural networks with morphological scale-spaces. In: International Conference on Discrete Geometry and Mathematical Morphology. pp. 483–495. Springer (2021)
10. Soille, P., et al.: Morphological image analysis: principles and applications, vol. 2. Springer (1999)
11. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM journal on computing **1**(2), 146–160 (1972)
12. Tieleman, T., Hinton, G., et al.: Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
13. Velasco-Forero, S.: Morpholayers (2020), `https://github.com/Jacobiano/morpholayers`
14. Velasco-Forero, S., Pagès, R., Angulo, J.: Learnable empirical mode decomposition based on mathematical morphology. SIAM Journal on Imaging Sciences **15**(1), 23–44 (2022)
15. Velasco-Forero, S., Rhim, A., Angulo, J.: Fixed point layers for geodesic morphological operations. In: British Machine Vision Conference. pp. 1–11 (2022)
16. Vincent, L.: Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE transactions on image processing **2**(2), 176–201 (1993)