A machine learning algorithm for the quantification and uncertainty analysis of the number of spinal microglia trainable in small and heterogeneous datasets

L. Martino[◇], M. M. Garcia[†][*], P. S. Paradas[†], E. Curbelo[‡],

[◇] Universitá degli studi di Catania, Catania, Italy.

[†] Area of Pharmacology, Nutrition and Bromatology, Dep. of Basic Health Sciences,

Faculty of Health Sciences, Universidad Rey Juan Carlos (URJC),

Unidad Asociada de I+D+i al Instituto de Quimica Medica (IQM), CSIC-URJC.

[‡] Universidad Carlos III de Madrid, Madrid, Spain.

Abstract

Counting immunopositive cells on biological tissues generally requires either manual annotation or (when available) automatic rough systems, for scanning signal surface and intensity in whole slide imaging. In this work, we tackle the problem of counting microglial cells in lumbar spinal cord cross-sections of rats by omitting cell detection and focusing only on the counting task. Manual cell counting is however a time-consuming task, and additionally entails extensive personnel training. The classic automatic color-based methods roughly inform of total labeled area and intensity (protein quantification) but do not specifically provide information on cell number. Since the images to be analyzed have a high resolution but a huge amount of pixels contains just noise or artifacts, we first perform a preprocessing generating several filtered images. Then, we design an automatic kernel counter that is a non-parametric and non-linear method. The proposed scheme can be easily trained in small datasets since, in its basic version, it relies only on one hyper-parameter. However, being non-parametric and non-linear, the proposed algorithm is flexible enough to express all the information contained in rich and heterogeneous datasets as well (providing the maximum overfit if required). Furthermore, the proposed kernel counter also provides uncertainty estimation of the given prediction, and can directly tackle the case of receiving several expert opinions over the same image. Different numerical experiments with artificial and real datasets show very promising results. Related Matlab code is also provided.

Keywords: Chronic Disease; Immunohistochemistry; Microtomy; Microglial cells; Automatic counting algorithm; Uncertainty estimation; Kernel smoothers

[*]Corresponding author: miguelangel.garcia@urjc.es

1 Introduction

The problem of counting objects in images or video frames is still one of the relevant tasks many biomedical applications face [10, 16]. Cell counting in images is yet a relevant but unpolished issue for many applications [2, 8]. Cell counting on tissue sections is typically a tedious and time-consuming task, and additionally entails extensive personnel supervision and training. The cells called microglia constitute one major group of glial cells, located throughout the brain and spinal cord of the central nervous system. They are considered resident immune cells within the central nervous system and increase in size and number upon activation [21]. Moreover, they have been suggested to be responsible for initiating altered synaptic and firing activity in neurological disorders, including chronic neurodegenerative diseases (e.g., Alzheimers and Parkinsons disease) and chronic pain [13]. However, one major concern when analyzing the microglial cells is their identification and counting [4, 6, 19].

In this work, we focus only on the simplified task of just counting problem of the microglial cells, skipping to necessarily detect them. We consider the microglial cells in micrographs of lumbar spinal cord cross-sections of rats (see Figure 2). They exhibit a variety of shapes, sizes, and functions depending on their activation state [6, 22]. The problem of manual counting microglial cells in immunohistochemistry is typically a time-consuming task, often requiring a budget for hiring personnel devoted to this task and additionally entails extensive personnel training [2, 5, 21]. Indeed, computing immunopositive cells on biological tissues generally requires the ability to detect the cells and manual annotation [5, 14]. This cell quantification is not possible with other immunoblotting techniques like ELISA (enzyme-linked immunosorbent assay) or western blot for tissue samples to be homogenized [9]. Several specific issues and features of the tackled problem should be taken into account:

Microglial cells are very small and stained a distinguishable dark brown color. On the contrary, size and shape depend on both the cutting plane and activation state [2, 21].

(a) Microglial cells are very small and stained a distinguishable dark brown color. On the contrary, size and shape depend on both the cutting plane and activation state [2, 21]. This is why using shape information to count (and/or detect) them can be misleading for the learning algorithm.

(b) The images to be analyzed have a high resolution, where a huge amount of pixels are just noise or artifacts. The number of pixels forming the microglial cells is extremely smaller with respect to the number of pixels that are not contained in a cell. The difference in order of magnitude, in terms of number of pixels, is approximately 10^4 in favor of non-microglial objects (on average). For instance, Figure 2 depicts just very tiny portions of an entire image, whereas an entire image is given in Figure 1. Namely, most of the input signals in our problem represent virtually "noise", and hence contain useless and/or misleading information.

(c) The image dataset is created by a human expert after a long and tiresome visual inspection. This may unequivocally lead to computing errors. Furthermore, structural uncertainty is sometimes present for some cells. For these cases, even the expert is often not able to provide a clear decision (the expert could just provide an estimation of the uncertainty of being a microglial cell).

(d) The dataset size depends on the inherent effort that demands manual processing, hence the number of stored images (D) available in the dataset is often small. The increase of the dataset depends mainly on costly human work (in terms of effort, time required, budget, etc).

(e) Additionally, the quality of images in the dataset may be highly heterogeneous, depending on the imaging capture system (jointly with human erroneous activities) in the laboratory. Differences in magnification, resolution, brightness, saturation, contrast, and other color shifts might be present in the stored images. Furthermore, the diversity grows if different laboratories share their images in order to increase the size of their datasets. This heterogeneity contributes to the need for the design of suitable algorithms.

Hence, in this work, we design an automatic and adaptive counting method according to the requirements described above. Namely, we propose a flexible counting algorithm that can be easily trained with small datasets and is flexible enough to be able to express all the diversity of images in the database. First of all, in order to address the issue of having a very low signal-to-noise ratio described in point **(b)**, we perform a feature extraction filtering the images according to different color thresholds. This idea is based on the observation given in the point **(a)** above. After the filtering, we obtain binary filtered images with a much higher signal-to-noise ratio. In the second stage, we perform a kernel smoother to solve the regression problem having as inputs the number of objects in those filtered images (counted by a clustering algorithm) and the expert's opinion as outputs. The resulting algorithm, called *kernel counter* (KC), has the following characteristics:

- KC is a *non-parametric* and *non-linear* method, i.e., its flexibility grows with the number D of data/images in the dataset. Indeed, for every possible value of D, the method can work as an interpolator providing the perfect overfitting to the outputs, regardless of the diversity in the data. This characteristic then responds to the requirement **(e)** described above.

- Moreover, in its basic version, the algorithm requires only the tuning of one hyper-parameter. Therefore, the learning task can be performed even in small datasets, fulfilling the condition **(d)**. Note that neural networks, or any other kind of parametric algorithms, can satisfy just one of the two requirements **(e)** or **(d)**, but not both together as the proposed KC does.

- The proposed KC method is also able to provide an uncertainty quantification of the given prediction. This characteristic responds to the requirement **(c)** above.

- The proposed algorithm is a *counter*, indeed the predictions/estimations are always non-negative. Thus, the predictions can easily converter into integers by rounding them.

- The KC method can directly handle the case of receiving several expert opinions over the same image. Namely, different experts provide their counting of the microglial cells of the same image. This is an isotopic multi-output scenario that can be directly handled by the KC [12] (more details are given in Section 5).

We also discuss possible extensions and improvements to increase the robustness of the algorithm. Several additional details are comments in appendices (as the choice of the threshold values). Different numerical experiments show the consistency of the proposed algorithm in different scenarios. We also test the KC method using images obtained by the Department of Basic Health Sciences, Faculty of Health Sciences of the Rey Juan Carlos University, Madrid, obtaining very promising results. Related Matlab code is also provided at `http://www.lucamartino.altervista.org/PUBLIC_CODE_KERNEL_COUNTER.zip`.

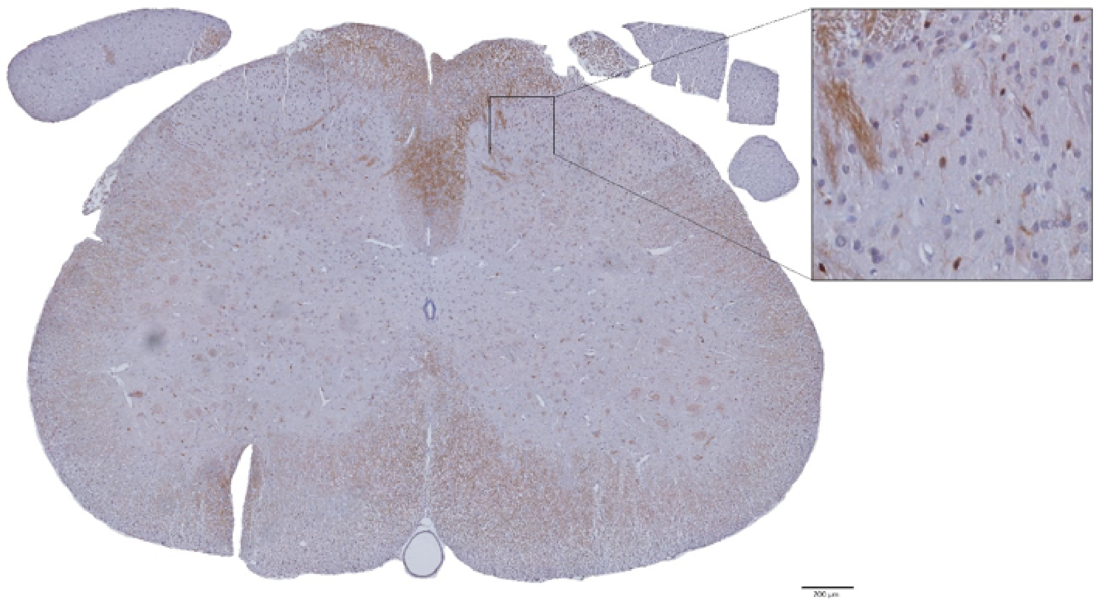


Figure 1: Example of complete image (spinal cord cross-section). The zoom frame indicates part of the ipsilateral dorsal horn with microglial cells (scale = $200\mu m$).

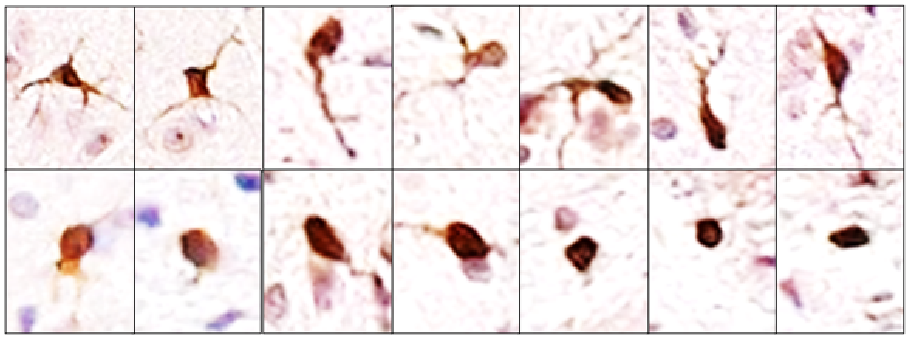


Figure 2: Examples of microglial cells (up: ramified; down: amoeboid). Microglia is always represented by a brownish tinction, whereas blue colored cells usually correspond to the nuclei of neurons.

2 Image filtering: extracting relevant information

The high-resolution images obtained in the laboratory contain huge amount of pixels that are just noise or artifacts. Namely, most of the signal in the input space represents noise in our problem. Moreover, microglial cells are very small and dark objects, with different geometric shape depending on the specific spinal cord cross-section [2, 21]. For all these reasons, we perform a filtering of the images considering threshold color values.
Let us consider a RGB image provided by the laboratory containing possibly the microglial cells[1]. Each pixel is represented by 3 color values,

$$\mathbf{p} = [p_1, p_2, p_3] \in [0, 1]^3,$$

where in 0 to 1 scale, i.e., $p_j \in [0, 1]$, where p_1 represents the amount of red, p_2 represents the amount of green, and p_3 represents the amount of blue. Hence, $\mathbf{p} = [1, 1, 1]$ represents a white pixel and $\mathbf{p} = [0, 0, 0]$ represents a black pixel. We define the vector of threshold color values,

$$\mathbf{t} = [t_1, t_2, t_3] \in \mathbb{R}^3,$$

in order to build a binary (black and white) *filtered image* where only the pixels that satisfy the conditions below

$$\begin{cases} p_1 \leq t_1, \\ p_2 \leq t_2, \\ p_3 \leq t_3, \end{cases} \tag{1}$$

will be considered as black pixels in this binary (black and white) filtered image. Whereas, all the pixels such that at least one condition is not satisfied in (1), i.e., a $p_i > t_i$, are transformed into white pixels. At each d-th image in the database, we apply different threshold vectors $\mathbf{t}^{(k)} \in [t_1^{(k)}, t_2^{(k)}, t_3^{(k)}]$, with $k = 1, ..., T$. Thus, from each medical colored image in the database from the laboratory we extract T binary filtered images. The underlying idea is to count the number of black objects (i.e., clusters of black pixels) within each of these filtered images, denoted by the integer variable $r_{kd} \in \mathbb{N}$. In order to count the black objects in this filtered binary image, we can use any kind of clustering algorithm or any alternative procedure designed for counting objects in binary matrices.

More specifically, let us assume that we have D images in the database and we consider the d-th image to analyze; hence, clearly, $d \in \{1, ..., D\}$. Given the k-th threshold vector $\mathbf{t}^{(k)} \in [t_1^{(k)}, t_2^{(k)}, t_3^{(k)}]$, the number of objects within each filtered image is denoted as $r_{kd} \in \mathbb{N}$, i.e., r_{kd} represents the number of objects (clusters black pixels) in the k-th binary filtered image, obtained filtering d-th image i with the threshold vector $\mathbf{t}^{(k)} \in [t_1^{(k)}, t_2^{(k)}, t_3^{(k)}]$. Hence, we have a correspondence between the thresholds and the number of objects in each d-th filtered image, i.e.,

$$\mathbf{t}^{(k)} \Longrightarrow \{r_{kd}\}_{d=1}^{D}, \qquad k = 1, ..., T.$$

[1] We recall that the approach described here can be employed for completely different types of application, for instance, with images provided by a satellite, a telescope, or any other medical machinery.

Therefore, to the d-th image in the database, we can associate a vector $\mathbf{r}_d = [r_{1d}, r_{2d}, ..., r_{Td}]$ of number of objects in each filtered images. Additionally, for each image in the dataset (provided by the laboratory), the expert provided the expected number of microglial cells $N_d \in \mathbb{N} = \{0, 1, 2, 3, ...\}$. Thus, the vector $\mathbf{r}_d$ is statistically related to N_d, i.e., we have the correspondence,

$$\mathbf{r}_d = [r_{1d}, r_{2d}, ..., r_{Td}] \Longleftrightarrow N_d.$$

Therefore, for each medical image in the database $d = 1, ..., D$, we have T different components of the inputs, $r_{kd} \in \mathbb{N}$, related to the number of microglial cells N_d in the d-th image (that plays the role of outputs in a regression problem). We have now a database formed by the pairs inputs/outputs $\{\mathbf{r}_d, N_d\}_{d=1}^{D}$, i.e., the counting problem can be tackled as a regression problem. Figure 3 depicts a graphical sketch of this image analysis. Figure 4 provides a graphical example with $N_d = 4$ and $T = 4$ different filtered images. Recall that pixels with color values closer to 0 are darker, whereas pixels with color values closer to 1 are clearer. Table 1 summarizes the main notation of the work.

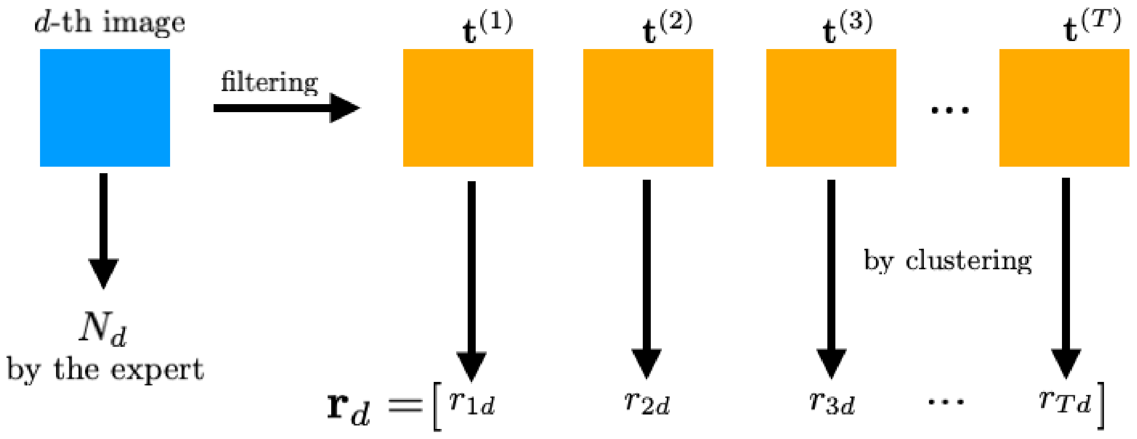


Figure 3: Graphical representation of the analysis performed for the feature extraction in each image. In each image filtered by $\mathbf{t}^{(k)}$, the total number of objects r_{kd} is obtained by clustering.

Table 1: Main notation of the work.

$\mathbf{t}^{(k)}$	k-th threshold vector	Decided by the user
N_d	number of microglial cells of the d-th image in the dataset	Given by the expert
r_{kd}	number of objects in the k-th filtered binary image	Obtained filtering the d-th image with $\mathbf{t}^{(k)}$

When the threshold $t_i^{(k)}$ values are small close to zero (i.e., we keep only dark pixels), we have a small number of objects r_{kd} in the filtered image but, since microglial cells are usually formed

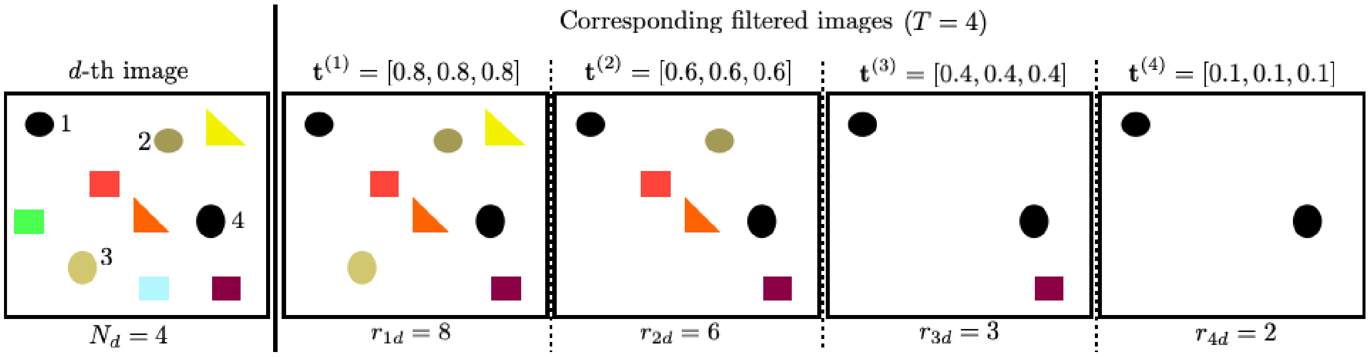

Corresponding filtered images ($T = 4$)
d-th image
$\mathbf{t}^{(1)} = [0.8, 0.8, 0.8]$
$\mathbf{t}^{(2)} = [0.6, 0.6, 0.6]$
$\mathbf{t}^{(3)} = [0.4, 0.4, 0.4]$
$\mathbf{t}^{(4)} = [0.1, 0.1, 0.1]$
1
2
3
4
$N_d = 4$
$r_{1d} = 8$
$r_{2d} = 6$
$r_{3d} = 3$
$r_{4d} = 2$

Figure 4: Illustrative example of a generic d-th image and $T = 4$ corresponding filtered images, with different threshold vectors $\mathbf{t}^{(k)}$. In this graphical example, we have $N_d = 4$ objects of interest (i.e., in our application, microglial cells). The rest of the 6 objects in the image play the role of irrelevant artifacts. In each filtered image, the total number of objects r_{kd} is given.

by dark pixels, most of the r_{kd} objects would be microglial cells, hence we would have a small number of artifacts (false positives). As the threshold values $t_i^{(k)}$ grow closer and closer to 1, there will be a greater chance of getting a larger portion of the microglial cells in the corresponding filtered image, but also a greater number of artifacts. Furthermore, still increasing the threshold values, all the microglial cells would be contained in the filtered images. A further increase of the thresholds will only yield an increase in the false positives/artifacts (since all the microglial cells are already contained in the previous filtered images).

3 A kernel smoother approach for counting microglial cells

Let us assume that $D \geq T$. We also consider a new image that we can study obtaining $\mathbf{r}_{D+1}$ but we have not the number of microglial cells N_{D+1} given by the expert, thus we desire to get an estimator $\widehat{N}_{D+1}$. Given the different images in the database and the new image where N_{D+1} is unknown, the database (training samples) is formed by the pairs inputs/outputs $\{\mathbf{r}_d, N_d\}_{d=1}^{D}$ and the test input where we need a prediction is $\mathbf{r}_{D+1}$ obtained by analyzing the new image, i.e., we have

$$\begin{aligned}
\mathbf{r}_1 &= [r_{11}, r_{21}, \cdots, r_{T1}] \Longleftrightarrow N_1, \quad \text{given by the expert,} \\
\mathbf{r}_2 &= [r_{12}, r_{22}, \cdots, r_{T2}] \Longleftrightarrow N_2, \quad \text{given by the expert,} \\
&\vdots \\
\mathbf{r}_D &= [r_{1D}, r_{2D}, \cdots, r_{Td}] \Longleftrightarrow N_D, \quad \text{given by the expert,} \\
&\text{and} \\
\mathbf{r}_{D+1} &= [r_{1(D+1)}, r_{2(D+1)}, \cdots, r_{T(D+1)}] \Longleftrightarrow N_{D+1} = ?
\end{aligned}$$

Then, the goal is to obtain $\widehat{N}_{D+1}$ as a prediction of the number of microglial cells N_{D+1} in the new image that has not been analyzed by the expert. Recall that each component $r_{k(D+1)}$ represents the number of objects in the k-th filtered image obtained by filtering the new test image using the threshold vector $\mathbf{t}^{(k)}$.

3.1 The kernel counter (KC)

The proposed algorithm, called *kernel counter (KC)*, is composed of the following steps: (a) standardization, (b) weighting, and (c) prediction, which are detailed below.

Standardization. We can standardize each of the $(D+1)$ values for each t, as

$$\bar{r}_{kd} = \frac{r_{kd} - \widehat{\mu}_k}{\widehat{\sigma}_k}, \qquad \widehat{\mu}_k = \frac{1}{D+1}\sum_{d=1}^{D+1} r_{kd}, \qquad \widehat{\sigma}_k = \sqrt{\frac{1}{D}\sum_{d=1}^{D+1}(r_{kd} - \widehat{\mu}_k)^2}, \tag{2}$$

for all $d = 1, ..., D+1$.

Weighting. Denoting as $\bar{\mathbf{r}}_d = [\bar{r}_{1d}, \bar{r}_{2d}, \cdots, \bar{r}_{Td}]$ the vectors with the standardized values, we can compute the D distances with respect to $\bar{r}_{t(D+1)}$,

$$L_d = ||\bar{\mathbf{r}}_d - \bar{\mathbf{r}}_{D+1}||^2 = \sum_{k=1}^{T}(\bar{r}_{kd} - \bar{r}_{k(D+1)})^2, \qquad d = 1, ..., D, \tag{3}$$

and the D different weights with the corresponding normalized weights,

$$\begin{aligned} w_d = \exp\left(-\frac{1}{\eta}L_d\right) &= \exp\left(-\frac{1}{\eta}||\bar{\mathbf{r}}_d - \bar{\mathbf{r}}_{D+1}||^2\right), \\ &= \exp\left(-\frac{1}{\eta}\sum_{k=1}^{T}(\bar{r}_{kd} - \bar{r}_{k(D+1)})^2\right), \qquad \bar{w}_d = \frac{w_d}{\sum_{i=1}^{D} w_i}, \qquad d = 1, ..., D, \end{aligned} \tag{4}$$

where $\eta > 0$ is chosen by the user, or learnt by leave-one-out cross-validation (LOO-CV). Note that, in this weighting step, we convert distances into weights.

Prediction. The kernel smoother estimator is then given by

$$\widehat{N}_{D+1} = \sum_{d=1}^{D} \bar{w}_d N_d. \tag{5}$$

If we desire to get an integer estimation, we can round it, i.e., $\lfloor \widehat{N}_{D+1} \rceil$. Note that, even if the formula above is linear, the estimator performs a non-linear regression with respect to the input vectors $\bar{\mathbf{r}}$ [12]. An estimation of the variance $\widehat{\sigma}^2_{D+1}$ associated to $\widehat{N}_{D+1}$ is given in Eq. (9) below (see also [1]).

The KC estimator has some interesting properties that we discuss below.

Property 1. Note that $\widehat{N}_{D+1} \geq 0$, which is a desired property for a counting algorithm. This is because $\widehat{N}_{D+1}$ is obtained as a linear combination of non-negative quantities, i.e., $N_d \geq 0$, and all the weights are also non-negative, i.e., $\bar{w}_d \geq 0$.

Property 2. We have that $\min_d N_d \leq \widehat{N}_{D+1} \leq \max_d N_d$. Therefore, the increase in the database is also beneficial for increasing the prediction ability and flexibility of the algorithm.

Remark. In this version of the algorithm, we have a unique hyper-parameter to learn that is the non-negative scalar η. It can be learnt by leave-one-out cross-validation (LOO-CV). For this reason, the proposed KC is easy and fast to train. However, other possible KC versions (with more hyper-parameters) are discussed in the next sections below.

Remark. Even if we have only one hyper-parameter to learn, the method is a *non-parametric* regressor, i.e., the complexity of the solution in Eq. (5) grows with D (that is the number of images in the database). Moreover the solution is *non-linear* with respect to the inputs $\bar{\mathbf{r}}$. Hence, the proposed method is able to express the complexity of rich datasets. This kernel procedure allows also the estimation of the variance $\widehat{\sigma}^2_{D+1}$ given in Eq. (9). Furthermore, the extensions with multi-expert's opinions is straightforward as shown in the next section.

3.2 Smoothing of the expert's opinions and variance of the prediction

The KC algorithm can also used to propose a *correction* of the expert's opinions, given all the information in the dataset. Generalizing slightly the previous formulas changing the reference vector, we have

$$L_{dj} = ||\bar{\mathbf{r}}_d - \bar{\mathbf{r}}_j||^2 = \sum_{k=1}^{T} (\bar{r}_{kd} - \bar{r}_{kj})^2, \qquad d = 1, ..., D, \quad j = 1, ..., D. \tag{6}$$

Note that $L_{jj} = 0$ for all j. Fixing now j, we can again define D different weights and the corresponding normalized weights,

$$\rho_{dj} = \exp\left(-\frac{1}{\eta} L_{dj}\right), \qquad \bar{\rho}_{dj} = \frac{\rho_{dj}}{\sum_{i=1}^{D} \rho_{ij}}, \qquad d = 1, ..., D. \tag{7}$$

Note that $\rho_{kj} = \rho_{jk}$ since $L_{kj} = L_{jk}$. Note that $w_d = \rho_{d(D+1)}$ for all d, i.e., if we use $\bar{\mathbf{r}}_{D+1}$ as reference vector, we recover the previous weights in Eq. (4), as expected. The smoothing of the expert's opinion for the j-th image is

$$\widehat{N}_d = \sum_{k=1}^{D} \bar{\rho}_{kd} N_k, \qquad d = 1, ..., D. \tag{8}$$

Variance of the prediction. Considering the smoothing values $\widehat{N}_d$ computed above, we can also estimate the variance in the prediction $\widehat{N}_{D+1}$ in (5) as suggested in [1], i.e.,

$$\widehat{\sigma}^2_{D+1} = \sum_{d=1}^{D} \bar{w}_d \left(N_d - \widehat{N}_d\right)^2. \tag{9}$$

Note that this estimation of the variance is obtained directly by applying a formula without any bootstrap or similar procedures. In the same fashion, the variances of the smoothing values $\widehat{N}_d$ can be approximated by

$$\widehat{\sigma}_d^2 = \sum_{k=1}^{D} \bar{\rho}_{kd} \left(N_k - \widehat{N}_k \right)^2. \tag{10}$$

Estimation of correlations or higher moments could be also provided as suggested in [1].

4 Learning η and extension with more hyper-parameters

The described KC has a unique hyper-parameter η. Note that η controls the underfitting/overfitting trade-off. Indeed, for instance, for big values of η we tend to the underfitting. In the limit case of $\eta \to \infty$, we have that

$$\bar{w}_d = \frac{1}{D}, \quad \widehat{N}_{D+1} = \frac{1}{D}\sum_{d=1}^{D} N_d,$$

so that the prediction is just the arithmetic mean of the number of microglial cells N_d in the different images. As η decreases, we tend to the overfitting. In the limit case of $\eta \to 0$, we have that $\widehat{N}_{D+1}$ is given by the number of microglial cells N_{d^*} corresponding to the nearest neighbor solution, i.e., $\widehat{N}_{D+1} = N_{d^*}$, where

$$d^* = \arg\min_d ||\bar{\mathbf{r}}_d - \bar{\mathbf{r}}_{D+1}||, \qquad d = 1, ..., D.$$

In this scenario, only the value N_{d^*} is taken into account, i.e., $\bar{w}_{d^*} = 1$ whereas the rest of the weights are zero, $\bar{w}_d = 0$ for any $d \neq d^*$. Hence, the KC contains the nearest neighbor algorithm as a special case (see [12] for more details).

An optimal value η^* can be obtained by LOO-CV, trying to minimize an error loss between the predicted and true numbers of cells, or maximizing the coefficient of determination of the linear regression between the predicted and true numbers of cells (any other metric within LOO-CV can be employed). Note that, in any case, as the size D of the dataset grows, the optimal value of η^* decreases, i.e., as $D \to \infty$ then $\eta^* \to 0$. As a good starting point for the LOO-CV procedure (to understand the order of magnitude of η), or as a reasonable proxy of η^*, one can use the following rule of thumb:

$$\widehat{\eta} = \frac{2}{DT}\sum_{d=1}^{D}\sum_{k=1}^{T}(\bar{r}_{kd} - \bar{r}_{k(D+1)})^2, \tag{11}$$

based on the empirical estimator of a variance. This is based on the fact that we are employing Gaussian kernels and $\frac{1}{2}\widehat{\eta}$ plays the role of a variance parameter. The value $\widehat{\eta}$ tends to the underfitting for big values of D (i.e., $\eta^* < \widehat{\eta}$), whereas is already a reasonable value for small D.

As already remarked above, even with just one hyper-parameter to learn, the method is a *non-parametric* and *non-linear* regressor, with respect to the input vectors $\bar{\mathbf{r}}$. The complexity of the final estimator in Eq. (5) grows with D. However, even more flexible versions of KC, with more hyper-parameters, can be easily designed. For instance, we can have one hyper-parameter η_d for each d-th image and, as a consequence, for each weight w_d as shown below,

$$w_d = \exp\left(-\frac{1}{\eta_d}||\bar{\mathbf{r}}_d - \bar{\mathbf{r}}_{D+1}||^2\right), \qquad d = 1, ..., D. \tag{12}$$

Therefore, in this scenario, we have D hyper-parameters η_d, for $d = 1, ..., D$, i.e., also the number of hyper-parameters grows with the number of data D in the dataset. The rules of thumb in this scenario would be $\widehat{\eta}_d = \frac{2}{T}\sum_{k=1}^{T}(\bar{r}_{kd} - \bar{r}_{k(D+1)})^2$, for any d.

5 Increasing the robustness of the KC

In the data-collecting process, the quality of the obtained image can vary according to the employed imaging systems and the human activities that can yield distortion or information loss. For instance, an image can be obtained in different lighting conditions. Moreover, the counting procedure by the expert is also a noisy process. Recall also that the database is quite small since the analysis of the expert's opinion is costly. In this section, we discuss strategies to handle these issues and improve the robustness of the KC.

Adaptive thresholds. The validity of the proposed algorithm is ensured for any choice possible of $\mathbf{t}^{(k)}$, with the unique requirement that the vectors $\mathbf{t}^{(k)}$, with $k = 1, ..., T$, must be different from each other. However, the performance of the algorithm depends on the choice of threshold vectors. See Appendix B for more details. In order to reduce the sensitivity of the image conditions (such as filter type, lighting conditions, etc.), we suggest to covert the threshold values into areas below the approximated densities of each color, and then into the quantile values of each new image. Namely, choosing the threshold vector $\mathbf{t} = [t_1, t_2, t_3]$, we can convert it into a probability vector $\mathbf{a}$,

$$\mathbf{a} = [a_1, a_2, a_3], \qquad \text{where} \qquad a_k \approx \text{Prob}(\text{a pixel has the } k\text{-color} \leq t_i),$$

obtained studying all the histograms of the colors of all the images in the database. Then, given a new test image, analyzing its histograms of color we can compute the approximated quantile values,

$$q_k \approx \text{quantile of order } a_k/100, \quad \text{for} \quad i = 1, 2, 3.$$

Finally, the corresponding threshold vector for analyzing the new image will be

$$\mathbf{t}_{\texttt{new}} = [q_1, q_2, q_3],$$

which is the new vector of the thresholds *adapted* to the new image. This procedure improves the robustness of the KC by analyzing darker or lighter images (due to experimental changes in the laboratory) included in the same dataset. Hence, the adaptive threshold strategy described above has another advantage: it allows us to include the same image *but* with a different degree of clarity for increasing the size dataset. This is a straightforward data augmentation procedure that can be used that also increase the robustness of the proposed algorithm.

Including the expert's uncertainty. The first simple possibility for including the expert's uncertainty (within the proposed KC algorithm) is to consider a "soft" labeling approach. Namely, in the d-th image of the dataset, let us consider that we have a certain number of objects, O, which can be considered possibly a microglial cell by the expert. For each one of these objects, the expert can associate a value p_o between $[0, 1]$ (as a probability) where values close to zero, i.e., $p_o \approx 0$, represent very high uncertainty that the object is a microglial cell, whereas $p_o = 1$ means a complete guarantee that this o-th object is a microglial cell. In this framework, finally, we have

$$N_d = \sum_{o=1}^{O} p_o \in \mathbb{R}^+, \qquad p_o \in [0, 1],$$

which is a positive real number, $N_d \in \mathbb{R}^+$, representing the number of microglial cells in the d-th image of the dataset, including the expert's uncertainty. The rest of the algorithm would remain the same. Another possibility to encode the expert's uncertainty is to include is to add some additional weights $\alpha_d \in [0, 1]$ directly in the final KC estimator

$$\widehat{N}_{D+1} = \sum_{d=1}^{D} \bar{w}_d \ \alpha_d N_d, \tag{13}$$

where if $\alpha_d = 0$ represents maximum uncertainty so that N_d is not considered in the linear combination, whereas if $\alpha_d = 1$ represents zero uncertainty to the value N_d computed by the expert.

Several experts. Let us consider that different experts provide their counting of the microglial cells of the same image. Namely, for the d-th image a j-th expert suggests that the number of microglial cells is N_{dj}. Moreover, let us consider that the total number of experts is E. Therefore, after the filtering, we have the correspondence:

$$\mathbf{r}_d \Longleftrightarrow \mathbf{N}_d = [N_{d1}, N_{d2}, ..., N_{dE}],$$

i.e., we have a vector as output in the regression problem. This multi-output scenario can be directly handled by the KC just considering the different pairs $\{\mathbf{r}_d, N_{dj}\}$ for $j = 1, ..., E$ and $d = 1, ..., D$ as different data points [12]. The total number of data will be then ED.

6 Numerical experiments

In this section, we test the KC algorithm in two synthetic experiments, where we can check and verify the convergence behaviors of the proposed scheme (since the ground-truth values are known). Then, in the last section, we apply the KC algorithm to the dataset obtained by our laboratory. Related Matlab code is also provided.

6.1 First synthetic experiment

In Appendix A, we provide a statistical model that can be employed for synthetic experimental analysis and study the theoretical behavior of the proposed KC algorithm. Thus, we consider the statistical model presented in Appendix A, i.e.,

$$\begin{aligned} &r_{id} = \lfloor \alpha_i N_d + F_i \rceil, \quad \alpha_i \in (0,1], \quad i = 1, ..., T, \quad d = 1, ..., D, \\ &F_i \in \{v_i, v_i + 1, ..., s_i\} \quad s_i, v_i \in \mathbb{N}^+, \quad s_i > v_i. \end{aligned} \tag{14}$$

where $\lfloor b \rceil$ returns the nearest integer to b; the coefficient $\alpha_i \in (0,1]$ depends on the vectors $\mathbf{t}^{(i)}$, i.e., $\alpha_i = \alpha(\mathbf{t}^{(i)})$, and F_i is a discrete random variable which takes non-negative integer values, contained in $[v_i, s_i]$. Eq. (19) shows the relationship between each r_{id} and N_d. Indeed, the value $100\alpha_i\%$ is the percentage of microglial cells N_d in the i-th filtered image (i.e., percentage of true positives), whereas F_i is the number of false positives (artifacts). We consider that in each d-th image, we can have a number of microglial cells between 0 and 200, i.e., $N_d \in [0, 200]$. More precisely, we consider a uniform discrete pmf,

$$N_d \in \mathcal{U}_{\texttt{discrete}}([0, 200]). \tag{15}$$

In this section, we assume that the expert is able to detect *perfectly* all the number of microglial cells N_d in the d-th image, observing the values $N_1, ..., N_D$.
Recall that the value $\alpha_i \cdot 100$ represents the percentages of microglial cells in the i-th filtered image (see App. A), obtained by filtering the d images with the i-th threshold vector $\mathbf{t}^{(i)}$. We consider T threshold vectors $\mathbf{t}^{(i)}$ such as

$$\alpha_i = \frac{i-1}{T-1}, \qquad i = 1, ..., T,$$

i.e., we assume that $\mathbf{t}^{(1)}$ is chosen such that $\alpha_1 = 0$, $\mathbf{t}^{(2)}$ is chosen such that $\alpha_2 = \frac{2}{T-1}$, $\mathbf{t}^{(3)}$ is chosen such that $\alpha_3 = \frac{3}{T-1}$ and so on, until $\mathbf{t}^{(T)}$ that is chosen such that $\alpha_T = \frac{T-1}{T-1} = 1$. Hence, we have always $\alpha_1 = 0$ and $\alpha_T = 1$. Moreover, we assume uniform discrete pmf for the random variable F_i,

$$F_i \sim \mathcal{U}_{\texttt{discrete}}([v_i, s_i]) \in \mathbb{N},$$

where we can write the minimum and maximum values as function of α_i like in Eq. (25), for instance,

$$v_i = \left\lfloor 24\left(\frac{1}{1+\exp(-3\alpha_i)}\right) - 12 \right\rceil, \tag{16}$$

$$s_i = \left\lfloor 1 + 40\left(\frac{1}{1+\exp(-3.5\alpha_i)}\right) - 20 \right\rceil. \tag{17}$$

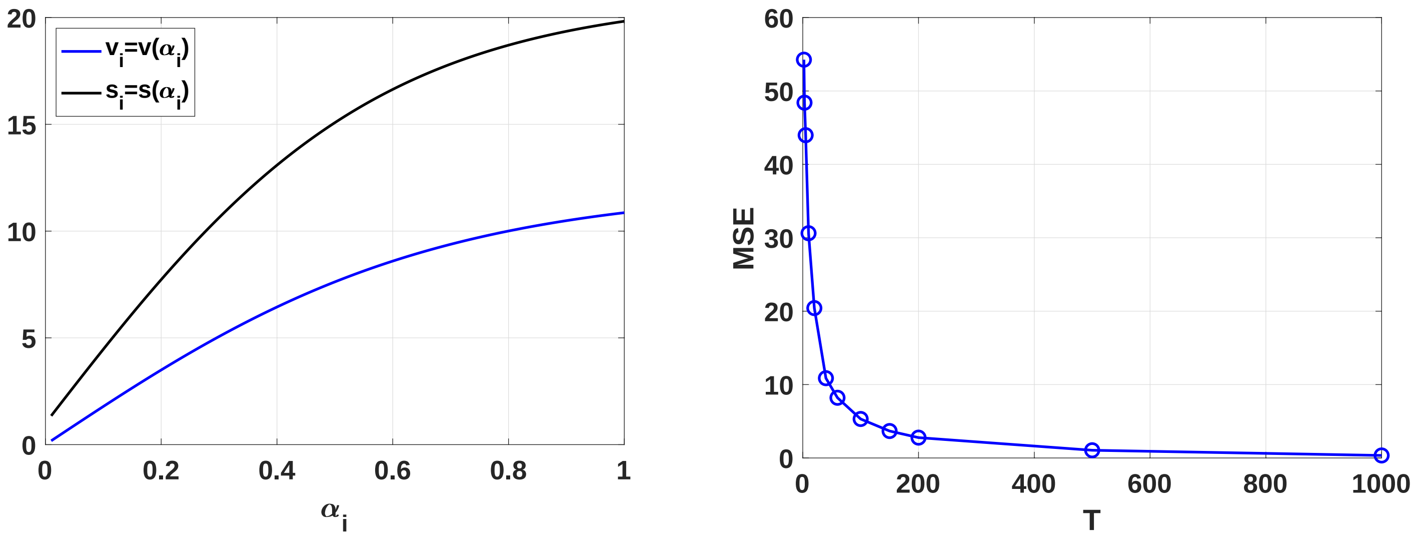
Figure 5(a) depicts these curves. Note that this is equivalent to be function of $\mathbf{t}^{(i)}$. We generate N_d and r_{id} by the model above assuming $D = 10^4$ images in the dataset, $i = 1, ..., D$. In this synthetic experiment, we are considering a high value of D since we are studying the convergence behavior of the proposed algorithm. We also consider different values of

$$T \in \{2, 3, 5, 10, 20, 40, 60, 100, 150, 200, 500, 1000\},$$

i.e., different numbers of threshold vectors $\mathbf{t}^{(i)}$ (and consequently generating different filtered images). Since in this artificial experiment, we know the ground-truth values $N_1, ..., N_D$, and we can compute the mean square error (MSE) in smoothing, i.e.,

$$\text{MSE}(T) = \frac{1}{D}\sum_{d=1}^{D}(\widehat{N}_d - N_d)^2. \tag{18}$$

We have averaged all the results over 10^5 independent runs. We employ LOO-CV for learning η at each run. Figure 5(b) depicts the results. We can observe that the MSE is decreasing quickly as T grows. Note that when $T = 200$, the MSE is already virtually zero.



(a) (b)

Figure 5: **(a)** The two functions $v_i = v(\alpha_i)$ and $s_i = s(\alpha_i)$ in Eqs.(16) and (17). Recall that $\min F_i = v_i$ and $\max F_i = s_i$. **(b)** The mean square error (MSE) in Eq. (18) as function of the number of threshold vectors used, i.e., T.

6.2 Second synthetic experiment

In this section, we consider a more complex version of the previous model,

$$\begin{aligned} &r_{id} = \lfloor \alpha_i N_d + F_i \rceil, \quad \alpha_i \in (0,1], \quad i = 1, ..., T, \quad d = 1, ..., D, \\ &F_i \in \{v_i, v_i + 1, ..., s_i\} \quad s_i, v_i \in \mathbb{N}^+, \quad s_i > v_i, \\ &\widetilde{N}_d = N_d + \epsilon_d, \qquad \epsilon_d \sim \mathcal{U}_{\texttt{discrete}}([-\gamma, \gamma]), \end{aligned} \tag{19}$$

where $\epsilon_d \in [-\gamma, \gamma]$ is a uniform noise variable and γ is a constant integer parameter that indirectly determines the power of this noise perturbation. We assume to observe a noisy measurement $\widetilde{N}_d$, i.e., instead of the true number of microglial cell N_d, as in the previous experiment, so that that the data pairs

$$\{\mathbf{r}_d, \widetilde{N}_d\}_{d=1}^D,$$

where $\mathbf{r}_d = [r_{1d}, ..., r_{Td}]$. Namely, in this experiment, we are assuming that the expert provides *noisy* versions $\widetilde{N}_d$ of the number N_d of microglial cells in the d-th image, so that we can test the robustness of the KC algorithm. In this experiment, we have considered $\gamma \in \{0, 5, 10\}$. Therefore, we use $\widetilde{N}_d$ in the estimators in Eqs. (5) and (8). Note that, setting $\gamma = 0$, we recover to the non-noisy framework in the previous section. We fix $D = 10^4$ and use LOO-CV for learning η at each run. We consider the same equations (15), (16), and (17) for the variables of the model above.

The results are averaged over 10^5 independent runs. At each run, we compute the MSE that is still computed considering the true corresponding values $N_1, ..., N_D$, exactly as in Eq. (18), i.e.,

$$\text{MSE}(T) = \frac{1}{D}\sum_{d=1}^{D}(\widehat{N}_d - N_d)^2. \tag{20}$$

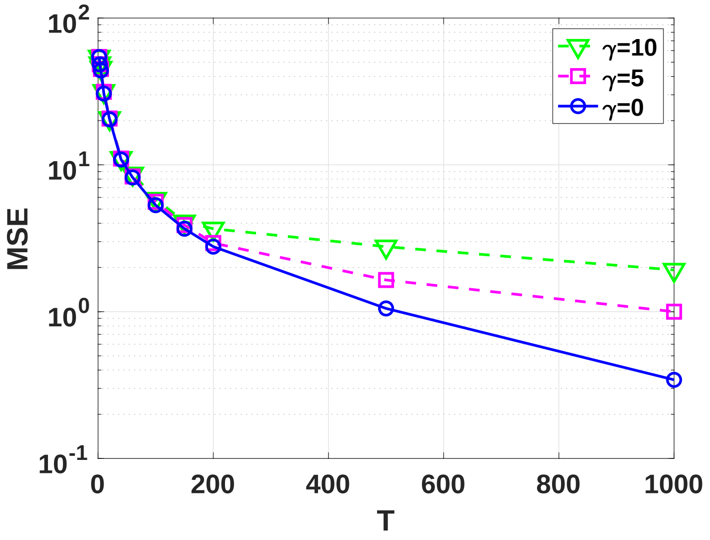
In this way, we can appreciate the deterioration of the performance of the algorithm as γ grows, as shown in Figure 6(a), where the MSE curve is depicted in log-scale to facilitate the visualization. However, the MSE of the algorithm vanishes to zero as T grows, regardless regardless the noise power. This proves the robustness of the proposed KC scheme.

6.3 Counting microglial cells in a real dataset

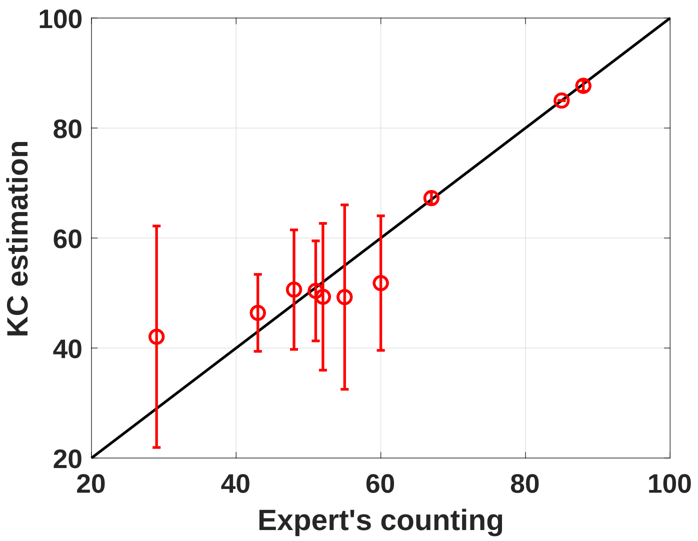
In this section, we present an application of presented automatic procedure applied to a real database. The images have been obtained by the Department of Basic Health Sciences, Faculty of Health Sciences of the URJC in Madrid.

Data collection. Immunoreactivity of spinal cord sections (L3L5) was performed for ionized calcium-binding adapter molecule 1 (Iba-1) as marker of microglia using diaminobenzidine (DAB) immunohistochemistry. Tissue processing and immunohistochemistry are described in [5]. A Zeiss Axioskop 2 microscope equipped with the image analysis software package AxioVision 4.6 was used to make montages of series of photomicrographs at final magnifications of 20x. The immunohistochemistry technique yields a dark brown color reaction. However, heat induced epitope retrieval (HIER), deficient rising/washing step or excessively high concentrations of DAB or hematoxylin counterstain can ultimately cause burgundy, blackish, brownish or dark purple staining artifacts. Subsequent to manual cell counting for $D = 10$, images were preprocessed. Anything other than the grey matter was removed and the background colored in white.

Results. We apply the presented technique to the stored database of D images. We perform a LOO-CV procedure minimizing the L_∞ distance (i.e., minimizing the maximum error) between the predicted number and the true number of microglial cells. We obtain an L_∞ error curve as a function of $1/\eta$, where there is a flat zone of minimum values between $1/\eta \in [1.3, 3.75]$. We



(a)



(b)

Figure 6: **(a)** The MSE curve in log-scale as a function of the number T of used threshold vectors, and different values of $\gamma = 0, 5, 10$. Note that the curve in solid line corresponds $\gamma = 0$ (i.e., without noisy evaluations of N_d) and is the same as in the curve in Figure 5(b) but in log-scale. For any value of γ, the MSE of the algorithm vanishes to zero as T grows, regardless of the power of the noise that affects N_d. **(b)** Results applying the KC to the real dataset. After LOO-CV, we get $\eta^* \approx 2.5$. The coefficient of determination between the numbers of cells given by the expert and the predicted numbers is $R^2 \approx 0.90$. The KC is also able to provide $\widehat{\sigma}_d^2$: the error bars show $\pm 2\sqrt{\widehat{\sigma}_d^2}$ corresponding to the 95% of the probability. Note that the error bar always contains the black line, hence the expert's opinion is always contained in the uncertainty interval.

choose an intermediate value setting $1/\eta^* \approx 2.5$, i.e., $\eta^* \approx 0.40$. With the values of η such that $1/\eta \in [1.3, 3.75]$, the maximum error in prediction never exceeds 25 units. Just to show the robustness of the technique, we also provide the results of LOO-CV procedure minimizing the L_1 distance: $1/\eta* \approx 0.85$, i.e., $\eta^* \approx 1.17$. With this value of η, we obtain an averaged absolute error in LOO-CV procedure of less than 10 units in prediction. Considering all the data points and $\eta^* \approx 0.40$, the average absolute error decreases to 3.68, i.e., less than 4 units. Note that, we are obtaining already remarkable results with a small and heterogenous dataset.

In Figure 6(b), we depict the points with the numbers of cells given by the expert as x-values and the numbers of cells predicted by the KC method as y-values, using $\eta^* \approx 0.40$ obtained with LOO-CV above. Ideally, we would like to have all the points belonging to the black straight line $x = y$ in Figure 6(b). This ideal case would correspond to the perfect prediction of all numbers of microglial cells. We can observe that the points are all very close to the black straight-line with a coefficient of determination of $R^2 \approx 0.90$ (in ideal scenario, we would have $R^2 = 1$). Considering all the data points (instead of leave-one-out as in LOO-CV), we obtain an average absolute error

less than 4 units with a standard deviation of 5.44.
Additionally, The KC algorithm also provides uncertainty estimations of its predictions, i.e., $\widehat{\sigma}_d^2$: the error bars show $\pm 2\sqrt{\widehat{\sigma}_d^2}$ corresponding to the 95% of the probability. Note that the error bar always contains the black line, hence the expert's opinion is always contained in the uncertainty interval. Moreover, the three points virtually belong to the black straight line, having (all of them) uncertainty intervals with almost a zero length. This shows that the algorithm is working extremely well estimating properly the uncertainty. The points with higher uncertainty can suggest a need for the revision of the image by the expert.[2]

7 Conclusions

We have proposed an automatic counter of microglial cells in immunohistochemical images. Automated cell counting is superior to manual cell counting at least in terms of speed efficiency, and the proposed scheme provides similar accuracy. Moreover, conventional manual cell counting is dependent on the researchers expertise and is time-consuming. The proposed KC algorithm is a counter, indeed the predictions/estimations are always non-negative. Thus, the predictions can easily converter into integers by rounding them. KC also provides an uncertainty estimation of the predictions (e.g., see Eq. (9)). In the smoothing case, if a smoothed value presents a high uncertainty, this can indicate a need for revision of the image by the expert. Furthermore, the multi experts scenario can be directly handled by the novel scheme.
The designed method has a unique hyper-parameter to learn, that is the non-negative scalar η, that can be learnt by leave-one-out cross-validation (LOO-CV). For this reason, the proposed KC is easy and fast to train even in small datasets. Even if we have only one hyper-parameter to learn, the method is a *non-parametric* regressor, i.e., the complexity of the solution in Eq. (5) grows with D (that is the number of images in the database). Moreover, the solution is *non-linear* with respect to the inputs. Hence, the proposed method is able to express the complexity of rich datasets. However, other possible KC versions (with more hyper-parameters) have been also discussed. Finally, the results obtained in the different experiments are very promising. Related Matlab code has been also provided in order to facilitate the use by interested practitioners.
Note also that the problem of counting objects in images is still one of the relevant tasks in different applications: for instance, counting cells in microscopic and biomedical images, monitoring crowds in surveillance systems, counting the number of green spots in satellite images [23, 17, 18] etc. Therefore, the proposed approach has a vast range of application, since it can be employed for counting different objects in different types of images, for instance, provided by a satellite, telescope, or drones, to name a few [7, 10, 15, 20, 11].

Acknowledgement

The work was partially supported by Agencia Estatal de Investigación AEI, Spain with project SP-GRAPH, ref. num. PID2019- 105032GB-I00, and project POLI-GRAPH, Grant PID2022-

[2]Related Matlab code is given at: `http://www.lucamartino.altervista.org/PUBLIC_CODE_KERNEL_COUNTER.zip`.

136887NB- I00, by project Starting Grant for Rttb, BA-GRAPH Efficient Bayesian inference for graph-supported data, of the University of Catania (UPB-28722052144), by the project LikeFree-BA-GRAPH funded by PIAno di inCEntivi per la RIcerca di Ateneo 2024/2026 of the University of Catania, Italy, and by Research and Development (R&D) project for young PhD graduates in the Region of Madrid (CAM)-URJC, F861-3483, years 2022-2024 to L.M. P.P. was awarded with a collaboration scholarship from the Spanish Ministry of Education and Vocational Training (call 2021-2022).

References

[1] L. Beleña, E. Curbelo, L. Martino, and V. Laparra. Second-moment/order approximations by kernel smoothers with application to volatility estimation. *Mathematics*, 12(9), 2024.

[2] S. Bradesi. Role of spinal cord glia in the central processing of peripheral pain perception. *Neurogastroenterology & Motility*, 22(5):499–511, 2010.

[3] D. Busby. Hierarchical adaptive experimental design for Gaussian Process emulators. *Reliability Engineering & System Safety*, 94(7):1183–1193, 2009.

[4] P. de Gracia, B. I. Gallego, B. Rojas, A. I. Ramirez, R. de Hoz, J. J. Salazar, A. Trivio, and J. M. Ramirez. Automatic counting of microglial cells in healthy and glaucomatous mouse retinas. *PLOS ONE*, 10(11):1–16, 11 2015.

[5] M. M. Garcia, M. Molina-lvarez, C. Rodriguez-Rivera, N. Paniagua, E. Quesada, J. A. Uranga, M. I. Rodroguez-Franco, D. Pascual, and C. Goicoechea. Antinociceptive and modulatory effect of pathoplastic changes in spinal glia of a tlr4/cd14 blocking molecule in two models of pain in rat. *Biomedicine and Pharmacotherapy*, 150:112986, 2022.

[6] T. R. F. Green, S. M. Murphy, and R. K. Rowe. Comparisons of quantitative approaches for assessing microglial morphology reveal inconsistencies, ecological fallacy, and a need for standardization. *Scientific Reports*, 12(1):18196, Oct 2022.

[7] Ivana Konatar, Tomo Popovic, and Natasa Popovic. Box-counting method in python for fractal analysis of biomedical images. In *2020 24th International Conference on Information Technology (IT)*, pages 1–4. IEEE, 2020.

[8] R. Kongsui, S. B. Beynon, S. J. Johnson, and F. R. Walker. Quantitative assessment of microglial morphology and density reveals remarkable consistency in the distribution and morphology of cells within the healthy prefrontal cortex of the rat. *Journal of Neuroinflammation*, 11(1):182, 2014.

[9] C. H. T. Kwok, Y. Kohro, M. Mousseau, M. S. Brien, J. R. Matyas, J. J. McDougall, and T Trang. Role of primary afferents in arthritis induced spinal microglial reactivity. *Frontiers in Immunology*, 12, 2021.

[10] V. Lempitsky and A. Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 23, 2010.

[11] F. Llorente, L. Martino, D. Delgado-Gomez, and G. Camps-Valls. Deep importance sampling based on regression for model inversion and emulation. *Digital Signal Processing*, 116:103104, 2021.

[12] L. Martino and J. Read. A joint introduction to Gaussian Processes and relevance vector machines with connections to Kalman filtering and other kernel smoothers. *Information Fusion*, 74:17–38, 2021.

[13] L. Minghetti, M. A. Ajmone-Cat, M. A. De Berardinis, and R. De Simone. Microglial activation in chronic neurodegenerative diseases: roles of apoptotic neurons and chronic stimulation. *Brain Research Reviews*, 48(2):251–256, 2005.

[14] R. Morelli, L. Clissa, R. Amici, M. Cerri, T. Hitrec, M. Luppi, L. Rinaldi, F. Squarcio, and A. Zoccoli. Automating cell counting in fluorescent microscopy through deep learning with c-ResUnet. *Scientific Reports*, 11:22920, 2021.

[15] M. Nasor, W. Obaid, et al. Detection and localization of early-stage multiple brain tumors using a hybrid technique of patch-based processing, k-means clustering and object counting. *International Journal of Biomedical Imaging*, 2020, 2020.

[16] K. Ongena, C. Das, J. L. Smith, S. Gil, and Johnston G. Determining cell number during cell culture using the scepter cell counter. *Journal of Visualized Experiments*, 45(2204), 2010.

[17] S. K. Paul, M. T. Chowdhury, M. Nicolescu, and M. Nicolescu. Object detection and pose estimation from rgb and depth data for real-time, adaptive robotic grasping, 2021.

[18] Md Sharifur Rahman and Md Rafiqul Islam. Counting objects in an image by marker controlled watershed segmentation and thresholding. In *2013 3rd IEEE international advance computing conference (IACC)*, pages 1251–1256. IEEE, 2013.

[19] I. Suleymanova, D. Bychkov, and J. Kopra. A deep convolutional neural network for efficient microglia detection. *Scientific Reports*, 13(1):11139, 2023.

[20] D. Heestermans Svendsen, L. Martino, and G. Camps-Valls. Active emulation of computer codes with Gaussian processes - application to remote sensing. *Pattern Recognition*, 100:107103, 2020.

[21] Tuan Trang, Simon Beggs, and Michael W. Salter. Brain-derived neurotrophic factor from microglia: a molecular substrate for neuropathic pain. *Neuron Glia Biology*, 7(1):99108, 2011.

[22] M. Wittekindt, H. Kaddatz, S. Joost, A. Staffeld, Y. Bitar, M. Kipp, and L. Frintrop. Different methods for evaluating microglial activation using Anti-Ionized Calcium-Binding adaptor protein-1 immunohistochemistry in the cuprizone model. *Cells*, 11(11), May 2022.

[23] Z. Zhang, L. Zhang, H. Zhang, Y. Guo, H. Wang, and X. Lu. AMSA-CAFF Net: Counting and high-quality density map estimation from x-ray images of electronic components. *Expert Systems with Applications*, 237:121602, 2024.

A A related statistical model

In this section, we provide a statistical model that can be employed for synthetic experimental analysis and study the theoretical behavior of the proposed KC algorithm. Each filtered image will contain a certain percentage $\alpha \in [0,1]$ of total number of microglial cells N_d (true positives, $\lfloor \alpha N_d \rceil = \#TP$) and also a certain number F of other objects (false positives $F = \#FP$, or number of artifacts in the filtered image) that do not represent microglial cells.

Mathematically speaking, we can express the behavior above with the following statistical model,

$$r_{kd} = \lfloor \alpha_k N_d + F_k \rceil, \quad \alpha_k \in (0,1], \quad k = 1, ..., T, \quad d = 1, ..., D, \tag{21}$$

$$F_k \in \{v_k, v_k+1, v_k+2, ..., s_k\} \quad s_k, v_k \in \mathbb{N}^+, \quad s_k > v_k, \tag{22}$$

where $\lfloor b \rceil$ returns the nearest integer to b; the coefficient $\alpha_k \in (0,1]$ depends on the $\mathbf{t}^{(k)}$, i.e., $\alpha_i = \alpha(\mathbf{t}^{(k)})$, and F_k is a discrete random variable which takes non-negative integer values, contained in $[v_k, s_k]$. The other coefficients $s_k, v_k \in \mathbb{R}^+$, are a positive real values depending also on $\mathbf{t}^{(k)}$, i.e., $s_k = s(\mathbf{t}^{(k)})$, and $v_k = v(\mathbf{t}^{(k)})$. Clearly, s_k affects the support of the random variable F_k such that

$$v_k = \min F_k \in \mathbb{N}^+, \tag{23}$$

$$s_k = \max F_k \in \mathbb{N}^+. \tag{24}$$

We have $v_k < \infty$ whereas s_k could be infinity as well. The probability mass function (pmf) of F_k defined in this support can change depending on the specific application. Note that, with this model, we are assuming that α_k, s_k, v_k and F_k are in some sense stationary quantities/variables, which do not depend on the specific image to analyze. However, $\alpha_k = \alpha(\mathbf{t}^{(k)})$, $v_k = v(\mathbf{t}^{(k)})$ and $s_k = s(\mathbf{t}^{(k)})$ depend on the threshold vector, so that the *signal noise ratio* (SNR) is different in each filtered image. Note that $\#TP = \lfloor \alpha_k N_d \rceil$ and $\#FP = F_k$.
Note that Eq. (21) shows the relationship between each r_{kd} and N_d. Since, $\alpha_k \in (0,1]$ and it multiplies the number of microglial cells N_d in the d-th analyzed image, the value $100\alpha_k\%$ is the percentage of microglial cells in the d-th filtered image (i.e, percentage of true positives), whereas F_k is the number of false positives. Thus, s_k and v_k are the maximum and minimum possible number of false alarms in the image filtered with the threshold vector $\mathbf{t}^{(k)}$. Finally, since $\alpha_k = \alpha(\mathbf{t}^{(k)})$, $s_k = s(\mathbf{t}^{(k)})$ and $v_k = v(\mathbf{t}^{(k)})$, we can also write

$$v_k = v(\alpha_k), \quad s_k = s(\alpha_k), \tag{25}$$

i.e., v_k and s_k can be also seen as functions of the percentage of number of microglial cells in the k-th filtered image. An illustrative example of filtered images and the corresponding r_{kd} values is given in Figure 7. Assuming this model, an upper bound and a lower bound for N_d can be obtained as shown in Appendix C.

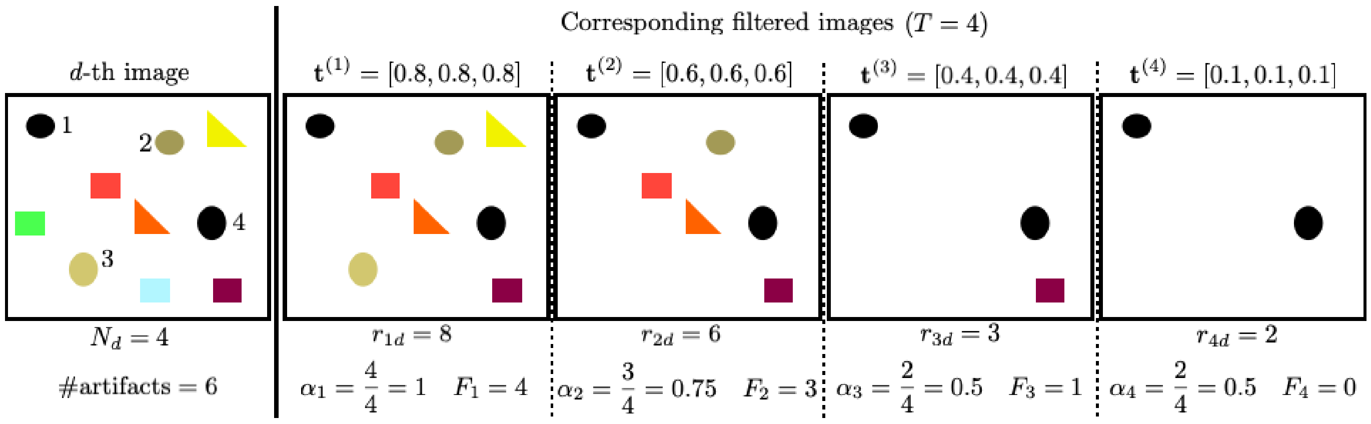

Corresponding filtered images ($T = 4$)
d-th image
$\mathbf{t}^{(1)} = [0.8, 0.8, 0.8]$
$\mathbf{t}^{(2)} = [0.6, 0.6, 0.6]$
$\mathbf{t}^{(3)} = [0.4, 0.4, 0.4]$
$\mathbf{t}^{(4)} = [0.1, 0.1, 0.1]$
1
2
3
4
$N_d = 4$
#artifacts = 6
$r_{1d} = 8$
$\alpha_1 = \frac{4}{4} = 1$ $F_1 = 4$
$r_{2d} = 6$
$\alpha_2 = \frac{3}{4} = 0.75$ $F_2 = 3$
$r_{3d} = 3$
$\alpha_3 = \frac{2}{4} = 0.5$ $F_3 = 1$
$r_{4d} = 2$
$\alpha_4 = \frac{2}{4} = 0.5$ $F_4 = 0$

Figure 7: We recover the illustrative example in a previous figure: a generic d-th image is filtered $T = 4$ times with different threshold vectors $\mathbf{t}^{(k)}$, obtaining the corresponding filtered images. In this example, we have $N_d = 4$ microglial cells. In each filtered images, we also show the total number of objects r_{kd}, the percentage $\alpha_k \in [0, 1]$ of the contained microglial cells, and also the number F_k of other objects (artifacts).

B Choice of the threshold vectors

The proposed algorithm works for any choice possible of $\mathbf{t}^{(k)}$, with the unique requirement that the vectors $\mathbf{t}^{(k)}$, with $k = 1, ..., T$, must be different to each others. However, the performance of the algorithm depends on the choice of threshold vectors,

$$\mathbf{t}^{(k)} \in [t_1^{(k)}, t_2^{(k)}, t_3^{(k)}] \in [0, 1]^3,$$

with $k = 1, ..., T$, that are employed to obtain T different filtered images from each d-th image in the database. This is actually an experimental design and/or active learning problem: see, for instance, [3, 11, 20]. In this section, we discuss some required concepts and a procedure for a proper choice of $\mathbf{t}^{(k)}$.

Note that large values of $t_i^{(k)}$ (close to 1) ensure the presence of the microglial cells, but also a huge number of other artifacts that are false alarm. Small values of $t_i^{(k)}$ (close to 0) ensure to substantially reduce the number of false alarm, however, some microglial cells can be missed. Considering the model in Eq. (21), we have that $\alpha_k = \alpha(\mathbf{t}^{(k)})$ and $s_k = s(\mathbf{t}^{(k)})$, hence the *signal noise ratio* (SNR) is different in each filtered image (different $\#TP$ and $\#FP$). For our purpose, the SNR can be defined as

$$\text{SNR} = \frac{\#TP}{\#FP} = \frac{\alpha N_d}{F}.$$

Clearly, the vectors $\mathbf{t}^{(k)}$ corresponding to larger SNR values are preferable.

Remark 1. *The idea is to choose the vectors $\mathbf{t}^{(k)}$ which provide the higher values of SNR $= \frac{\#TP}{\#FP}$. For instance, this can be directly done by a Monte Carlo search. However, below we divide this search in two steps.*

Another interesting observation is that different vectors $\mathbf{t}^{(k)}$ can provide the same number of

true positives $\#TP$, but different number of $\#FP$. Clearly, given a value $\#TP$, we prefer the vectors $\mathbf{t}^{(k)}$ which provide the smallest value of false positives $\#FP$. We call them as *optimal conditional thresholds.*
Therefore, conceptually the search of proper threshold vectors can be divided in two steps:

- First of all, fixing a value of true positives $\#TP$, find the optimal conditional thresholds, i.e., the thresholds that maximizes the SNR providing the smallest number of false positives $\#FP$.
- Among the optimal conditional thresholds previously obtained, choose the thresholds corresponding with the larger SNRs (possibly SNR > 1).

With the first step, we can build the curve "minimum number of $\#FP$" versus $\#TP$ (or percentage of $\#TP$), as shown in Figures 8 and 9(a). This allows to detect the range of $\#TP$ values with SNR> 1, as depicted in Figure 9(b). Moreover, the division in these two phases, also allows the use of a different payoff function in the second step, instead of maximizing the SNR (i.e., in the proposed procedure, the payoff function is the SNR). The following two subsections describe these two steps.

B.1 Optimal conditional thresholds

First or all, we have to notice that, given a threshold vector $\mathbf{t} \in \mathbb{R}^3$, in the filtered image we have a certain number of true positives $\#TP$ (microglial cells), and a certain number of false positives $\#FP$ (i.e., false alarms). Two different threshold vectors $\mathbf{t}_1, \mathbf{t}_2$ could give the same number of true positives $\#TP$ but different values of false positives $\#FP$. Clearly, given a number of true positives $\#TP$, we prefer the threshold vector which provides the smallest number of false positives (i.e., that minimizes $\#FP$). We call this vector as *optimal conditional threshold vector*, i.e., the optimal vector corresponding to the true positive value $\#TP$. Namely, the optimal conditional vector maximizes the signal-to-noise ratio (SNR), fixing $\#TP$, SNR $= \frac{\#TP}{\#FP}$. In order to find the optimal conditional threshold vectors, we employ a Monte Carlo search:

```
1. For $s = 1, ..., M_{\texttt{runs}}$:
    • Draw $t_{i,s} \sim \mathcal{U}([0,1])$, for $i = 1, 2, 3$.
    • Set $\mathbf{t}_s = [t_{1,s}, t_{2,s}, t_{3,s}]$ and obtain the filtered image corresponding to $\mathbf{t}_s$.
    • Count the number of true positives $\#TP(s)$, and false positives $\#FP(s)$ in this filtered image.
2. For each value $\gamma = 1, 2..., N_d$:
    • Find all the indices $s^*$ such that $\#TP(s^*) = \gamma$, defining a set of indices $\mathcal{S}_\gamma$.
    • For all $s^* \in \mathcal{S}_\gamma$:
        – Find
              $s_\gamma^{\texttt{opt}} = \arg\min_{s^* \in \mathcal{S}_\gamma} \#FP(s^*).$        (26)
        – Then the optimal conditional vector of thresholds (for $r$ true positives) is $\mathbf{t}_{s_\gamma^{\texttt{opt}}} = \left[t_{1,s_\gamma^{\texttt{opt}}}, t_{2,s_\gamma^{\texttt{opt}}}, t_{3,s_\gamma^{\texttt{opt}}}\right]$.
```

Therefore, given a pre-established number of true positives $\gamma = \#TP$ with $1 \leq \gamma \leq N_d$, the optimal conditional threshold vector is $\mathbf{t}_{s_\gamma^{\texttt{opt}}}$: this is the vector which provides the minimum possible number of false alarms $\#FP$ (given the pre-established value $\#TP = \gamma$). Clearly, $\mathbf{t}_{s_\gamma^{\texttt{opt}}}$ is an estimation of the optimal vector due to the Monte Carlo procedure: as $M_{\texttt{runs}} \to \infty$, this approximation improves.

In our specific application, we use $M_{\texttt{runs}} = 10^4$ independent runs. Figure 8 depicts the minimum possible number of false alarm objects ($\#FP$), obtained by the optimal conditional thresholds, versus the percentage of true positives ($\frac{\#TP}{N_d}100\%$). Clearly, there is a trade-off between the number of false alarms and true positives.

B.2 Choosing among the optimal conditional thresholds

Each point of the curve in Figure 8 corresponds to an optimal conditional vector $\mathbf{t}_{s_\gamma^{\texttt{opt}}}$ with $1 \leq \gamma \leq N_d$ and $\#TP = \gamma$. Figure 9(a) shows the same curve in Figure 8 but, in this case, the values of $\#TP$ are directly given in the horizontal axis. Moreover, the straight line $\#FP = \#TP$ is depicted with a solid black line. Clearly, when the blue curve is below the black straight line, we have

$$\text{SNR} = \frac{\#TP}{\#FP} > 1.$$

The corresponding value of the SNR are provided in Figures 9(b). The rectangular region depicted by dashed lines, shows the values of SNR bigger than 1 (obtained by the optimal conditional thresholds for $25 \leq \#TP \leq 53$). Hence, we should choose T points in this interval, e.g., in this

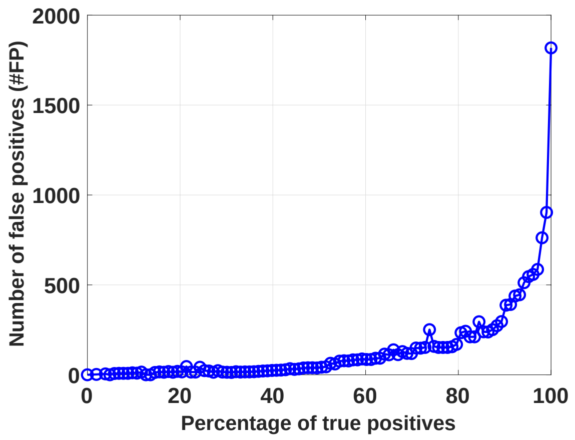


Figure 8: Example of minimum possible number of possible false alarms (false positives) ($\#FP$) versus the percentage of true positives ($\frac{\#TP}{N_d}100\%$) in a specific image of the dataset, obtained by the corresponding optimal conditional thresholds. By increasing the values of thresholds, more $\#TP$ but also more $\#FP$ are obtained. Here, the minimum possible number of false alarms for each given value of $\#TP$ is shown.

example (corresponding to results from our database), $25 \leq \gamma_k \leq 53$, $i = 1, ..., T$, and we use the corresponding optimal conditional vector $\mathbf{t}_{s^{\texttt{opt}}_{\gamma_k}}$.

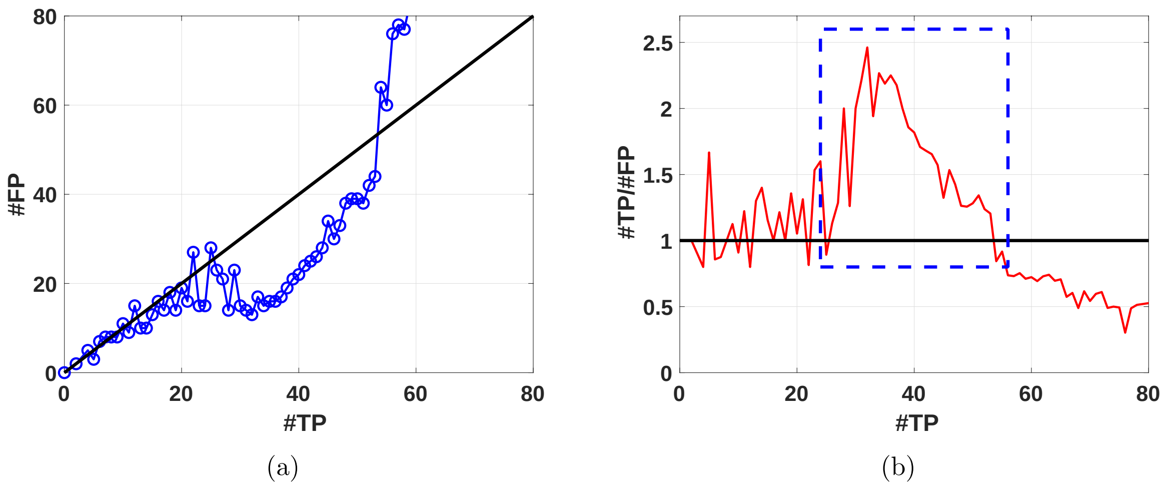


Figure 9: **(a)** The minimum number of false positives ($\#FP$) obtained by the optimal conditional thresholds, versus number of true positives ($\#TP$), in a range of values where we can obtain SNR ≥ 1 (for a specific image of the dataset). The solid black line represents the straight line $\#FP = \#TP$, then when the blue curve is below the black straight line, we have SNR ≥ 1. **(b)** The $SNR = \frac{\#FP}{\#TP}$ corresponding to the values in Figure 8. The rectangular region depicted by dashed lines, shows the values of SNR bigger than 1 (obtained by the optimal conditional thresholds for $25 \leq \#TP \leq 53$).

C Lower and upper bounds

Let us consider both $v_i < \infty$, $s_i < \infty$, finite and known (recall that $0 \leq v_i < s_i$). Moreover, let us also assume the coefficients $\alpha_i \in (0, 1]$, as known value (or approximately known - estimated). In this scenario, if the observations r_{id} are generated according to the model

$$r_{id} = \lfloor \alpha_i N_d + F_i \rceil, \quad \alpha_i \in (0, 1], \quad i = 1, ..., T, \quad d = 1, ..., D,$$

we can obtain some lower and upper bounds. Since $F_i \in [v_i, s_i]$, we have

$$\widehat{N}_{d,\texttt{lower}} = \left\lfloor \frac{r_{id} - s_i}{\alpha_i} \right\rfloor, \qquad \widehat{N}_{d,\texttt{upper}} = \left\lceil \frac{r_{id} - v_i}{\alpha_i} \right\rceil. \tag{27}$$

where $\lfloor b \rfloor$ gives as output the greatest integer less or equal to b, whereas $\lfloor b \rfloor$ returns the least integer greater or equal to b. Namely, if the observations r_{id} are generated according to the model above, we can assert that the true number N_d of microglial cells is contained (with probability 1) in the following interval:

$$\widehat{N}_{d,\texttt{lower}} \leq N_d \leq \widehat{N}_{d,\texttt{upper}}. \tag{28}$$

Therefore, by estimating s_i and v_i (for a certain percentage α_i of microglial cells in the filtered images) we can obtain lower and upper bounds for N_d, using the inequalities above.