

# An automatic counting system of small objects in noisy images with a noisy labelled dataset: computing the number of microglial cells in biomedical images

L. Martino<sup>\*‡</sup>, P. Paradás<sup>\*</sup>, L. Carro<sup>\*</sup>, M. M. Garcia<sup>†</sup>, C. Goicoechea<sup>†</sup>, S. Ingrassia<sup>‡</sup>

<sup>\*</sup> Dep. of Signal Processing, Universidad Rey Juan Carlos (URJC), Madrid, Spain.

<sup>†</sup> Area of Pharmacology, Nutrition and Bromatology, Dep. of Basic Health Sciences, Unidad Asociada I+D+i Instituto de Química Médica (IQM) CSIC-URJC, Spain.

<sup>‡</sup> Dep. of Economics and Business, Università di Catania (Unict), Catania, Italia.

## 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 biomedical images that represent lumbar spinal cord cross-sections of rats. Note that counting microglial cells is typically a time-consuming task, and additionally entail extensive personnel training. We skip the task of detecting the cells and we focus only on the counting problem. Firstly, a linear predictor is designed based on the information provided by filtered images, obtained applying color threshold values to the labelled images in the dataset. Non-linear extensions and other improvements are presented. The choice of the threshold values is also discussed. Different numerical experiments show the capability of the proposed algorithms. Furthermore, the proposed schemes could be applied to different counting problems of small objects in other types of images (from satellites, telescopes, and/or drones, to name a few).

**Keywords:** automatic counting algorithm; image processing; microglial cells; immunohistochemistry; noisy labels

## 1 Introduction

Despite the great progress that image processing have experienced in recent years, cell counting methodology is yet a relevant but unpolished issue for many applications [5, 7, 13, 14, 15]. The problem of counting objects in images or video frames is still one of the relevant task in many real-world applications: for instance, counting cells in microscopic and biomedical images, monitoring crowds in surveillance systems, counting the number of green spots in satellite images [17, 8] etc. More specifically, the study of biomedical images is typically a time-consuming task, and additionally entail extensive personnel training [4, 2, 6, 18]. Indeed, computing immunopositive

cells on biological tissues generally requires either manual annotation [4, 12]. In this work we address the problem of counting microglial cells in micrographs of lumbar spinal cord cross-sections of rats (see Figures 1-2). Microglia constitute one major group of glial cells. They are considered resident immune cells within the central nervous system and increase in size and number upon activation [2, 18]. 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 [10]. However, one major concern when analyzing the microglial cells is their identification and counting. They exhibit a variety of shapes, sizes and functions depending on their activation state. Therefore, current gold-standards for quantifying cells imply undeniable bias when applied to nerve tissue plus, they offer several limitations regarding usability, sensitivity, and robustness.

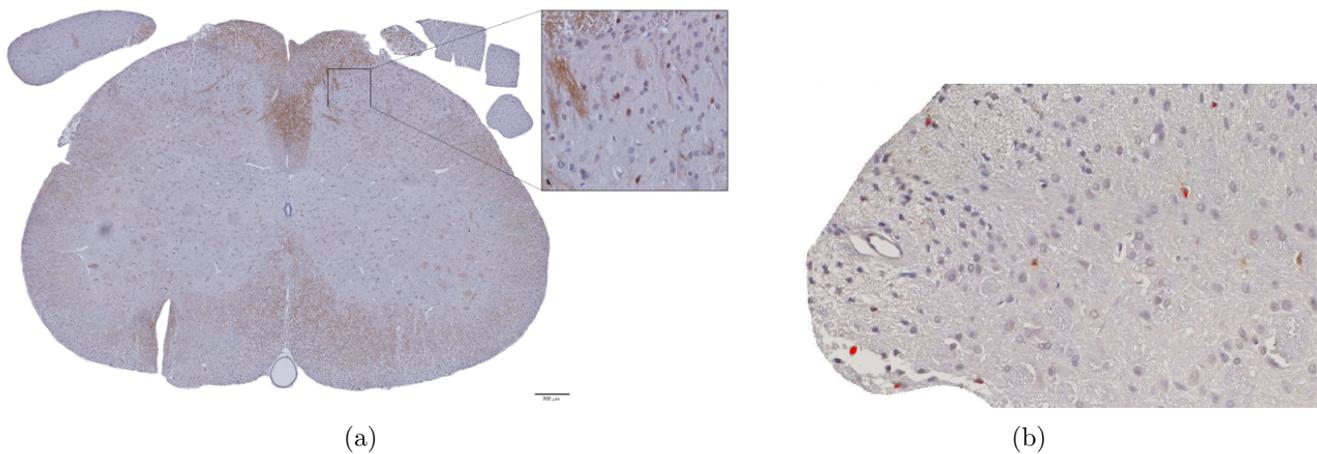


Figure 1: **(a)** Example of complete image (spinal cord cross-section). The zoom frame indicates part of the ipsilateral dorsal horn with microglial cells (scale =  $200\mu m$ ). **(b)** A portion of an image where some microglial cells have been highlighted with red pixels.

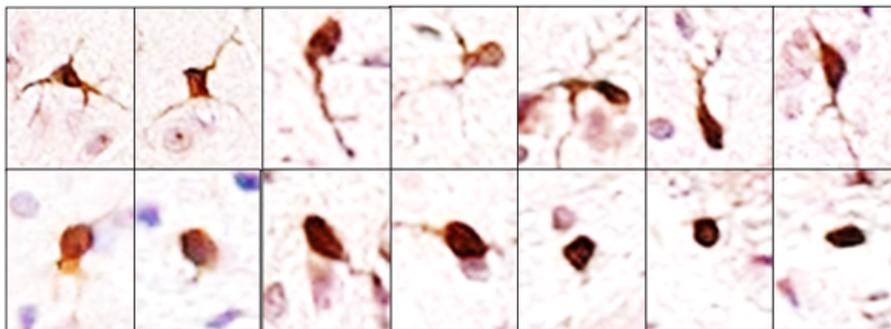


Figure 2: Examples of microglial cells. Microglia is always represented by a brownish tincture, whereas blue colored cells usually correspond to the nuclei of neurons.

Technically speaking, this problem could be addressed and divided in two main steps: firstly a binary classification problem (two classes, “microglial cells” and “non microglia objects”) and

secondly, after the detection, counting the microglial cells. However, in this work, we consider *just* the counting problem of the microglial cells (without necessarily detecting them). Namely, we avoid the hard task of detecting and/or localizing individual objects. Thus, our goal is to accurately estimate the count. The main features for discriminating a microglial cell are the following:

- they are very small objects,
- with a dark color (generally close to black), and
- often they can have an ellipsoidal form.

Nevertheless, the geometric shape of the microglial cell depends on the specific spinal cord cross-section. See Figure 3 for an illustrative example of this issue: a two dimensional representation of a three dimensional object can yield different shapes in an 2D image [2, 18].

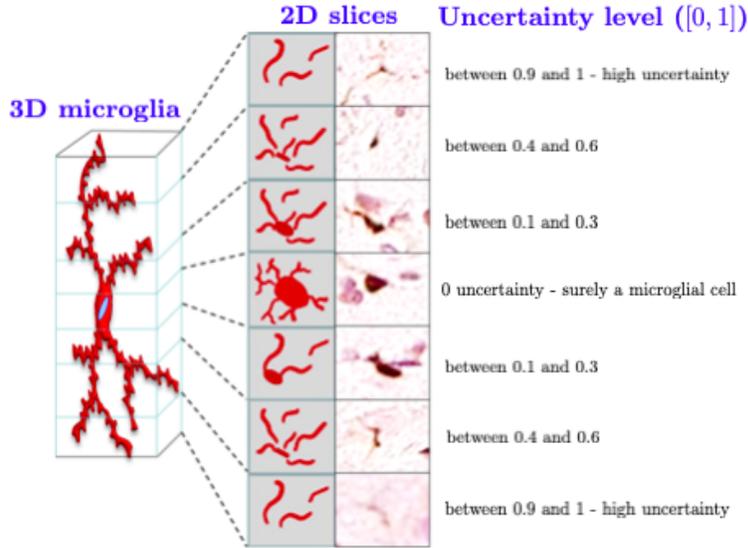


Figure 3: Graphical sketch of a microglial cell in 3 dimensions, and the different scenarios in which microglial cells can appear on a slice (2 dimensional representation). We also show a range of uncertainty level (0 no uncertainty, 1 maximum uncertainty) provided by the expert.

For this reason, we address the problem using only the color information, setting some *thresholds* for the red-green-blue (RGB) images (more precisely, 3 thresholds, one for the red component, one for the green component and one for the blue component). More specifically, we start from a dataset of  $D$  images, labelled by an expert. From each labelled image, we build  $T$  different filtered images, obtained considering different color thresholds. Clearly, each filtered image contains a different number of microglial cells and a different number of false alarm objects (a.k.a., artifacts). See Figure 4 for a graphical representation.

In this work, with this information we build a linear (and then also non-linear) predictor for the number of microglial cells in a possible new image. An additional important feature of the

proposed technique is also the ability to provide a *smoothing version* of the expert’s opinions (noisy labels). In summary, we introduce a novel procedure (based on color thresholds) for automatically counting the number of microglial cells in new image (prediction) or correct the expert’s opinion in an image contained in the dataset (smoothing). Furthermore, the choice of the thresholds is discussed in detail. Several further generalizations and extensions are also described, as well as possible future works. Finally, we employ different numerical simulations to show the consistency of the proposed algorithm in different scenarios. Clearly, the proposed approach has a vast range of application, since it can be employed for different types of images, for instance, provided by a satellite, a telescope, and/or any other sensor. We consider images obtained by the Department of Basic Health Sciences, Faculty of Health Sciences of the Rey Juan Carlos University, Madrid.

The work is structured as follows. Some important features of the problem are highlighted in Section 2. The main technical notations and ideas are introduced in Section 3. Section 4 is devoted to the description of the main formulas of the proposed procedure, and to pose a notion of consistency. Section 5 describes different extensions and improvements. The choice of the thresholds is discussed in Section 6. Some numerical tests and experiments are provided in Section 7. We finish with some conclusions in Section 8.

## 2 Specific features of the problem

It is also important to remark the high complexity of the addressed application due to the following features:

- **Unbalanced dataset.** The number of pixels forming the microglial cells is extremely smaller with respect to the number of pixels that are not contained in a microglial cell. Namely, the microglial cells are very small and represent just few objects within an image, that is plenty of other artifacts as well, in general. 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, Figures 1(b) or 2 depicts just very small portions of an entire image, whereas an entire image is given in Figure 1(a).
- **Noisy labels.** The dataset is created by a human expert using a long (and often exhausting) visual inspection, that can produce several errors including missing cells. But this is not the unique cause of noise, there are two main sources of noise in our problem. The first source of noise is due to the expert’s mistakes and/or oversights. The second source of uncertainty is the structural issue due to the type of “cut” (and correspondence 2D slice), graphically represented in Figure 3. As a consequence, the labels are noisy, even with a perfect fulfillment of the expert.
- **Small dataset.** The number of labelled images in the dataset ( $D$ ) is generally small, since the increase of the dataset depends on costly human work (in terms of effort, time required, etc).

### 3 Main notation and proposed technical approach

Let consider a RGB image provided by the laboratory<sup>1</sup>. Each pixel is represented by 3 color values

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

where  $p_j \in [0, 1]$ , 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. In the same fashion, we define the vector of threshold values,

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

for the colors in the image, in order to build a *filtered image* considering only the pixels such that

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

whereas, in the pixels such that at least one value  $p_i > t_i$ , we set  $p_k^* = 1$  for all  $k = 1, 2, 3$ , i.e., the pixels which do not satisfy (1) are transformed into white pixels. The idea is to count the number of *objects* (i.e., clusters of pixels) within each filtered image, denoted by the variable  $r$ .

Let us assume that we have  $D$  images in the database and we consider the  $d$ -th image to analyze (clearly,  $d \in \{1, \dots, D\}$ ). Furthermore, we employ different threshold vectors  $\mathbf{t}^{(i)} \in [t_1^{(i)}, t_2^{(i)}, t_3^{(i)}]$  with  $i = 1, \dots, T$ , so that we obtain  $T$  different filtered images. The number of objects within each filtered image is denoted as  $r_{id}$ , i.e.,  $r_{id}$  represents the number of objects in the  $i$ -th filtered image, obtained filtering  $d$ -th image in the database with the threshold vector  $\mathbf{t}^{(i)}$ . Hence, we have a correspondence between thresholds and number of objects in each  $d$ -th filtered image, i.e.,

$$\mathbf{t}^{(i)} \implies \{r_{id}\}_{d=1}^D, \quad i = 1, \dots, T.$$

Additionally, for each image provided by the laboratory, the expert provided the expected number of microglial cells  $N_d \in \mathbb{N} = \{0, 1, 2, 3, \dots\}$ .

**Important observation.** Each filtered image will contain a certain percentage  $\alpha \in [0, 1]$  of total number of microglial cells  $N_d$  (*true positives*  $[\alpha N_d] = \#TP$ ) and also a certain number  $F$  of other objects (*false positives*  $F = \#FP$ , or number of “false alarms”) that do not represent microglial cells. When the threshold values are small (i.e., we keep only dark pixels), we have a small number of objects  $r_{id}$  in the filtered image but, since microglial cells are usually formed by dark pixels, most of the  $r_{id}$  objects would be microglial cells, hence we would have a small number of false alarms. As the threshold values grow, 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

---

<sup>1</sup>We recall that the approach described here can be employed for completely different type of application, for instance, with images provided by a satellite, a telescope, or any other medical machinery.

false alarms. Furthermore, still increasing the threshold values, at certain threshold values, all the microglial cells would be contained in the filtered images. A further increase of the thresholds will only yield an increase of the false alarm objects (since all the microglial cells are already contained in the previous filtered images, i.e., we have reached the maximum of  $\#TP = N_d$ ,  $\alpha = 1$ ).

**A possible statistical model.** Mathematically speaking, we can express the behavior above with the following statistical model,

$$r_{id} = \lfloor \alpha_i N_d + F_i \rfloor, \quad \alpha_i \in (0, 1], \quad i = 1, \dots, T, \quad d = 1, \dots, D, \quad (2)$$

$$F_i \in \{v_i, v_i + 1, v_i + 2, \dots, s_i\} \quad s_i, v_i \in \mathbb{N}^+, \quad s_i > v_i, \quad (3)$$

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

$$v_i = \min F_i \in \mathbb{N}^+, \quad (4)$$

$$s_i = \max F_i \in \mathbb{N}^+. \quad (5)$$

We have  $v_i < \infty$  whereas  $s_i$  could be infinity as well. The probability mass function (pmf) of  $F_i$  defined in this support can change depending on the specific application. Note that, with this model, we are assuming that  $\alpha_i$ ,  $s_i$ ,  $v_i$  and  $F_i$  are in some sense stationary quantities/variables, which do not depend on the specific image to analyze. However,  $\alpha_i = \alpha(\mathbf{t}^{(i)})$ ,  $s_i = s(\mathbf{t}^{(i)})$  and  $v_i = v(\mathbf{t}^{(i)})$  depend on the threshold vector, so that the *signal noise ratio* (SNR) is different in each filtered image. Note that  $\#TP = \lfloor \alpha_i N_d \rfloor$  and  $\#FP = F_i$ .

Note that Eq. (2) shows the relationship between each  $r_{id}$  and  $N_d$ . Since,  $\alpha_i \in (0, 1]$  and it multiplies the number of microglial cells  $N_d$  in the  $d$ -th analyzed image, the value  $100\alpha_i\%$  is the percentage of microglial cells in the  $d$ -th filtered image (i.e, percentage of *true positives*), whereas  $F_i$  is the number of *false positives* (i.e., *false alarms*). Thus,  $s_i$  and  $v_i$  are the maximum and minimum possible number of false alarms in the image filtered with the threshold vector  $\mathbf{t}^{(i)}$ . Finally, since  $\alpha_i = \alpha(\mathbf{t}^{(i)})$ ,  $s_i = s(\mathbf{t}^{(i)})$  and  $v_i = v(\mathbf{t}^{(i)})$ , we can also write

$$v_i = v(\alpha_i), \quad s_i = s(\alpha_i), \quad (6)$$

i.e.,  $v_i$  and  $s_i$  can be also seen as functions of the percentage of number of microglial cells in the  $i$ -th filtered image. A summary of the main notation is given in the Table 1. An illustrative example of filtered images and the corresponding  $r_{id}$  values is given in Figure 4.

## 4 A linear estimator for prediction and smoothing

Note also that for each image in the database,  $d = 1, \dots, D$ , we have  $T$  different data,  $r_{i,d} \in \mathbb{N}$ , related to the number of microglial cells  $N_d$  in the  $d$ -th image. Thus, we can define a vector  $\mathbf{r}_d = [r_{1d}, r_{2d}, \dots, r_{Td}]$  which is related to  $N_d$ ,

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

Table 1: Main notation of the work.

|                      |  |                       |
|----------------------|--|-----------------------|
| $\mathbf{t}^{(i)}$   | $i$ -th threshold vector   | decided by the user   |
| $N_d$                | number of microglial cells of the $d$ -th image  | given by the expert   |
| $r_{id}$             | number of objects in the $i$ -th filtered image – filtering $d$ -th image (by $\mathbf{t}^{(i)}$ ) | known by the analysis |
| $\alpha_i, s_i, v_i$ | coefficients in the observation model in Eq. (2)   | unknown               |
| $F_i$                | a discrete random variable which takes non-negative values between $v_i$ and $s_i$                 | unknown               |
| $\#TP, \#FP$         | true and false positives in an image obtained filtering an image in the dataset                    | known by the analysis |

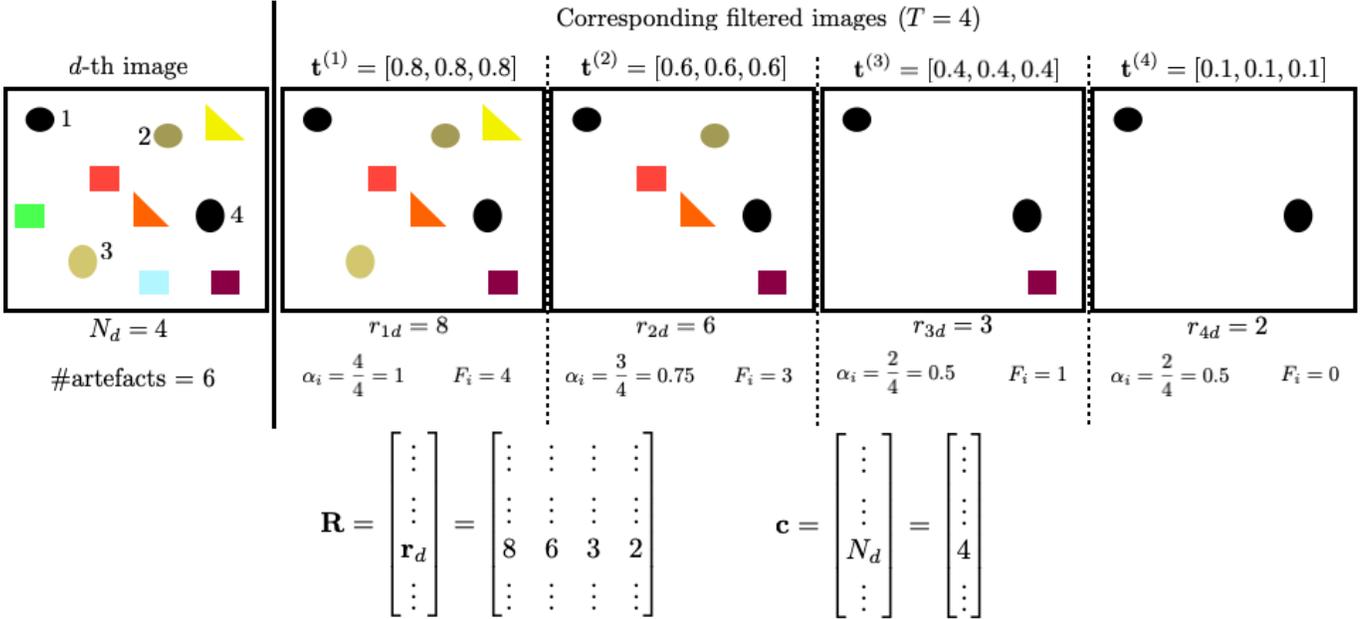


Figure 4: Illustrative example of a generic  $d$ -th image and  $T = 4$  corresponding filtered images, with different threshold vectors  $\mathbf{t}^{(i)}$ . In this illustrative example we have  $N_d = 4$  microglial cells that can be recognized by ellipsoidal shape (however, recall that the shape cannot be used as feature in our real application due to the issue described in Figure 3).

We assume that  $D \geq T$ . Given the different images in the database, we also build a  $D \times T$  matrix,

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_D \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & \cdots & r_{T1} \\ r_{12} & r_{22} & \cdots & r_{T2} \\ \vdots & \vdots & \vdots & \vdots \\ r_{1D} & r_{2D} & \cdots & r_{TD} \end{bmatrix}, \quad (7)$$

and a  $D \times 1$  vector

$$\mathbf{c} = \begin{bmatrix} N_1 \\ N_2 \\ \vdots \\ N_D \end{bmatrix}, \quad (8)$$

See Figure 4, for an example of construction of the vectors  $\mathbf{r}_d$ ,  $\mathbf{c}$  and the matrix  $\mathbf{R}$ . Given a new test image (denoted with the super-index  $*$ ), one idea to estimate the number of microglial cells is to build an estimator of  $N^*$  as a linear combinations of the number of objects  $r_{i*}$  given the  $T$  different filtered images obtained with the threshold vectors  $\mathbf{t}^{(i)}$ , i.e.,

$$\widehat{N}^* = \widehat{\beta}_1 r_{1*} + \widehat{\beta}_2 r_{2*} + \dots + \widehat{\beta}_T r_{T*} = \mathbf{r}_* \widehat{\boldsymbol{\beta}}^\top. \quad (9)$$

where  $\widehat{\boldsymbol{\beta}} = [\widehat{\beta}_1, \dots, \widehat{\beta}_T]$  is found by a Least Squares (LS) procedure [1, 3, 9], i.e.,

$$\widehat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\mathbf{c} - \mathbf{R}\boldsymbol{\beta}\|^2, \quad (10)$$

$$= \arg \min_{\boldsymbol{\beta}} \sum_{d=1}^D (N_d - \mathbf{r}_d \boldsymbol{\beta}^\top)^2. \quad (11)$$

The solution is given by the well-known formula,

$$\widehat{\boldsymbol{\beta}} = (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{c}. \quad (12)$$

## 4.1 Prediction and smoothing of the expert's opinions

We have seen that the prediction of number of microglial cells in a new image will be

$$\widehat{N}^* = \mathbf{r}_* \widehat{\boldsymbol{\beta}}^\top,$$

where  $\mathbf{r}_* = [r_{1*}, r_{2*}, \dots, r_{T*}]$  and each component  $r_{i*}$  represents the number of objects in  $i$ -th filtered image, obtained using the threshold vectors  $\mathbf{t}^{(i)}$ . Moreover, a smoothing version  $\widehat{\mathbf{c}}$  of the expert's opinions contained in the vector  $\mathbf{c}$  can be obtained as

$$\widehat{\mathbf{c}} = \mathbf{R} \widehat{\boldsymbol{\beta}}, \quad (13)$$

$$= \mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{c}. \quad (14)$$

## 4.2 Some notion of consistency in this noisy framework

Let us consider the family of linear estimators of type in Eq. (9). Clearly, as the number of data  $D \rightarrow \infty$ , the estimator  $\widehat{\boldsymbol{\beta}}$  converges to the best coefficient vector  $\boldsymbol{\beta}$  which provides *the best linear estimator* with  $T$  components. In this sense, the consistency is ensured. However, the best linear estimator generally is not the best estimator. Hence, considering a fixed  $T$  and as  $D \rightarrow \infty$ , even

if we obtain the best vector  $\widehat{\boldsymbol{\beta}}$ , we usually have still errors in the prediction  $\widehat{N}^*$ .

However, we can ensure good results in the smoothing scenario. If we consider  $T = D$ , the matrix  $\mathbf{R}$  is square matrix and, in this specific case,

$$\mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top = \mathbf{I}_D,$$

where  $\mathbf{I}_D$  is a  $D \times D$  identity matrix (assuming that  $\mathbf{R}$  has maximum rank and, being a square matrix, then is invertible). Hence, in this scenario, we have  $\widehat{\mathbf{c}} = \mathbf{c}$ , i.e., the algorithm does not *smooth* the expert's opinions but just provides a perfect replica of the expert's opinions [9].

**Remark 1.** *An exhaustive statistical analysis of the estimator in Eq. (12) is not straightforward since all the elements are noisy. First of all, each  $r_{id}$  is noisy and, as a consequence, also the matrix  $\mathbf{R}$ . Secondly, the elements  $N_d$  of the vector  $\mathbf{c}$  are noisy since they are obtained by an expert's opinion, so that they are affected by possible human mistakes.*

However, the result above ensure that, if we use enough threshold vectors (i.e., increasing  $T \rightarrow D$ ), we obtain good performance, according to the expert's opinion. See the proposition below.

**Proposition 4.1.** *Since for a fixed number of data values  $D$  (i.e., images analyzed by the expert), we can increase the number of threshold vectors in order to obtain a square matrix  $\mathbf{R}$  (i.e.,  $T = D$ ). If  $\mathbf{R}$  has maximum rank, then we can always obtain an algorithm able to perfectly reproduce the expert's opinion, i.e.,  $\widehat{\mathbf{c}} = \mathbf{c}$ . Therefore, even when  $D \rightarrow \infty$ , we can always increase  $T \rightarrow \infty$ , such that  $\widehat{\mathbf{c}} \rightarrow \mathbf{c}$ .*

Furthermore, there is another way to see this point: increasing  $T$  means making the algorithm more flexible since we have more parameters involved. Therefore, we can raise the flexibility of the algorithm by just considering more threshold vectors  $\mathbf{t}^{(i)}$  (i.e., more filtered images).

**Remark 2.** *Clearly, it is required that  $\mathbf{R}$  has maximum rank (to be invertible). Namely, the rows  $\{\mathbf{r}_d\}_{d=1}^D$  (or the columns) must be linear independent. Therefore, we should also consider this point when choosing the  $T$  of threshold vectors  $\mathbf{t}^{(i)} \iff \{r_{id}\}_{d=1}^D$ .*

A possible issue is that the filtered images, corresponding to different threshold vectors  $\mathbf{t}^{(i)}$ , could present a high correlation in some cases. For any value of  $T$ , the estimator would still be linear. Clearly, we can have some non-linear extensions as we describe below.

## 5 Further considerations and possible enhancements

### 5.1 Non-linear extensions

**Non-linear bases.** The easiest way to consider a non-linear regression to extend the previous approach, is to include non-linear bases (i.e., non-linear transformations of the number of objects  $r_{id}$ ) into the LS scheme previously described [9].

In order to clarify this point, we consider the use of a quadratic regression function as an example. For this purpose, now let us assume that  $D \geq 2T$ . In this case, we build a  $D \times 2T$  matrix,

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_1^2 \\ \mathbf{r}_2 & \mathbf{r}_2^2 \\ \vdots & \vdots \\ \mathbf{r}_D & \mathbf{r}_D^2 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{21} & \cdots & r_{T1} & r_{11}^2 & r_{21}^2 & \cdots & r_{T1}^2 \\ r_{12} & r_{22} & \cdots & r_{T2} & r_{12}^2 & r_{22}^2 & \cdots & r_{T2}^2 \\ \vdots & \vdots \\ r_{1D} & r_{2D} & \cdots & r_{TD} & r_{1D}^2 & r_{2D}^2 & \cdots & r_{TD}^2 \end{bmatrix}. \quad (15)$$

Note that the number  $T$  of threshold vectors  $\mathbf{t}^{(i)} \iff \{r_{id}\}_{d=1}^D$  is unchanged, although the columns of  $\mathbf{R}$  grows to  $2T$ . The rest of formulas remain virtually unchanged:

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} \hat{\boldsymbol{\beta}}_1 \\ \hat{\boldsymbol{\beta}}_2 \end{bmatrix} = (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{c},$$

and the prediction of number of microglial cells in a new image (denoted with  $*$ ) will be

$$\hat{N}^* = \mathbf{r}_* \hat{\boldsymbol{\beta}}_1^\top + \mathbf{r}_*^2 \hat{\boldsymbol{\beta}}_2^\top,$$

that is, a quadratic estimator based on LS.

Assuming  $D \geq KT$ , the procedure above can be generalized considering different  $K$  generic nonlinearities  $g_k(\mathbf{r})$ , i.e., we would have the following  $D \times KT$  matrix,

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & g_1(\mathbf{r}_1) & g_2(\mathbf{r}_1) & \cdots & g_K(\mathbf{r}_1) \\ \mathbf{r}_2 & g_1(\mathbf{r}_2) & g_2(\mathbf{r}_2) & \cdots & g_K(\mathbf{r}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{r}_D & g_1(\mathbf{r}_D) & g_2(\mathbf{r}_D) & \cdots & g_K(\mathbf{r}_D) \end{bmatrix}. \quad (16)$$

Clearly, this kind of procedure improves the performance of the algorithm if there is a presence of some non-linear behavior in the dataset. However, after that this non-linear behavior is captured by the non-linear estimator, the only way to improve more and more the performance is to increase the  $T$  of threshold vectors  $\mathbf{t}^{(i)} \iff \{r_{id}\}_{d=1}^D$ , and the number  $D$  of images in the dataset [1, 3, 9].

**Non-linear regression methods.** Another more general approach is to consider more sophisticated regression methods such as Gaussian Processes (GPs) and/or Artificial Neural Networks (ANNs), to name a few, considering the following pairs of data points

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

where  $\mathbf{r}_d = [r_{1d}, r_{2d}, \dots, r_{Td}]$ . The application and design of GPs and/or ANNs is left as a future work [9, 11, 16].

## 5.2 Including the expert's uncertainty

One possibility for including the expert's uncertainty (within the proposed algorithm) is to consider a "soft" labelling 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 number  $n_o$  between  $[0, 1]$  where numbers close to zero, i.e.,  $n_o \approx 0$ , represent very high uncertainty, whereas  $n_o = 1$  means a complete guarantee that this  $o$ -th object is a microglial cell. These uncertainty levels are shown in Figure 3. In this framework, finally we have

$$N_d = \sum_{o=1}^O n_o,$$

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. For instance, a value of  $N_d = 2.6$  expresses the uncertainty over a third possible microglial cell. The procedure can be employed to count the percentage of the microglial cells in the  $T$  filtered images, obtaining the vector  $\mathbf{r}_d = [r_{1d}, r_{2d}, \dots, r_{Td}]$  where, in this case,  $r_{id} \in \mathbb{R}^+$  instead of being an integer number. Defining, in the same way,  $\mathbf{c} = [N_1, \dots, N_D]^\top$  (but now  $N_d \in \mathbb{R}^+$ ), the rest of the algorithm remains the same.

### 5.3 Data augmentation: increasing the dataset with surrogate data

Since the analysis and labelling of new images by the expert is generally slow and possibly contains several mistakes (noisy labelling), the number of labeled images  $D$  in the dataset is often small. Moreover, it entails an extensive personnel training and a cost in terms of personal salary, generally. In order to increase the number  $D$  of images in the database, we could create surrogate (“fake”) images with a certain (known) number of microglial cells, by re-using pixels of microglial cells from the images in the database already labeled by the expert.

Clearly, the procedure employed for creating the surrogate images can yield a bias in the results. Therefore, additional theoretical studies and empirical analysis are required for ensuring the consistency of the overall methodology.

## 6 Choice of the threshold vectors

For a finite value of  $T$ , the performance of the algorithm depends on the choice of threshold vectors

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

with  $i = 1, \dots, T$ , i.e., for obtaining  $T$  different filtered images from each  $d$ -th image in the database. However, it is also important to remark that, as  $T$  grows and as  $T \rightarrow D$ , the dependence of the performance on the choice of  $\mathbf{t}^{(i)}$  tends to decrease. In this section, we discuss different considerations regarding a proper choice of  $\mathbf{t}^{(i)}$ .

Note that large values of  $t_k^{(i)}$  (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  $t_k^{(i)}$  (close to 0) ensure to substantially reduce the number of false alarm, however, some microglial cells can be missed. Considering the model in Eq. (2), we have that  $\alpha_i = \alpha(\mathbf{t}^{(i)})$  and  $s_i = s(\mathbf{t}^{(i)})$ , 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  $\mathbf{t}^{(i)}$  corresponding to larger SNR values are preferable.

**Remark 3.** *The idea is to choose the vectors  $\mathbf{t}^{(i)}$  which provide the higher values of  $\text{SNR} = \frac{\#TP}{\#FP}$ . 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  $\mathbf{t}^{(i)}$  can provide the same number of true positives  $\#TP$ , but different number of  $\#FP$ . Clearly, given a value  $\#TP$ , we prefer the vectors  $\mathbf{t}^{(i)}$  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  $\text{SNR} > 1$ ).

With the first step, we can build the curve “minimum number of  $\#FP$ ” versus  $\#TP$  (or percentage of  $\#TP$ ), as shown in Figures 5 and 6(a). This allows to detect the range of  $\#TP$  values with  $\text{SNR} > 1$ , as depicted in Figure 6(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.

## 6.1 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  $\#TP$ ,  $\text{SNR} = \frac{\#TP}{\#FP}$ . In order to find the optimal conditional threshold vectors, we employ a Monte Carlo search:

1. For  $s = 1, \dots, M_{\text{runs}}$ :
  - 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  $\gamma = 1, 2, \dots, N_d$ :
  - 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

$$s_\gamma^{\text{opt}} = \arg \min_{s^* \in \mathcal{S}_\gamma} \#FP(s^*). \quad (17)$$

- Then the optimal conditional vector of thresholds (for  $r$  true positives) is  $\mathbf{t}_{s_\gamma^{\text{opt}}} = [t_{1,s_\gamma^{\text{opt}}}, t_{2,s_\gamma^{\text{opt}}}, t_{3,s_\gamma^{\text{opt}}}]$ .

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

In our specific application, we use  $M_{\text{runs}} = 10^4$  independent runs. Figure 5 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.

## 6.2 Choosing among the optimal conditional thresholds

Each point of the curve in Figure 5 corresponds to an optimal conditional vector  $\mathbf{t}_{s_\gamma^{\text{opt}}}$  with  $1 \leq \gamma \leq N_d$  and  $\#TP = \gamma$ . Figure 6(a) shows the same curve in Figure 5 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 6(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

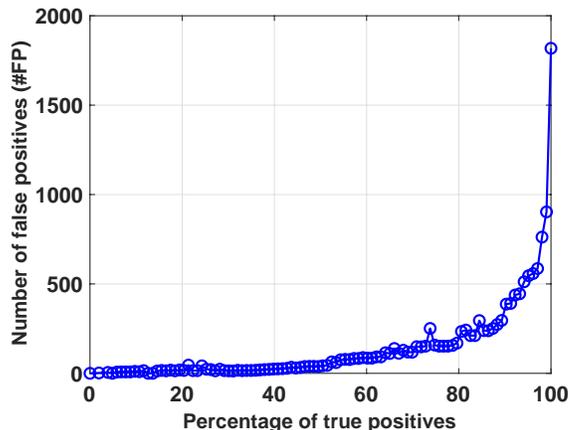


Figure 5: Example of minimum possible number of possible false alarms (false positives) ( $\#FP$ ) versus the percentage of true positives ( $\frac{\#TP}{N_d}100\%$ ) in a specific image of the dataset, obtained by the corresponding optimal conditional thresholds. By increasing the values of thresholds, 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),

$$25 \leq \gamma_i \leq 53, \quad i = 1, \dots, T,$$

and we use the corresponding optimal conditional vector  $\mathbf{t}_{s_{\gamma_i}^{\text{opt}}}$ .

## 7 Numerical experiments

### 7.1 First experiment

We will test with a progressively closer to real-world application. So we consider again the model

$$\begin{aligned} r_{id} &= \lfloor \alpha_i N_d + F_i \rfloor, \quad \alpha_i \in (0, 1], \quad i = 1, \dots, T, \quad d = 1, \dots, D, \\ F_i &\in \{v_i, v_i + 1, \dots, s_i\} \quad s_i, v_i \in \mathbb{N}^+, \quad s_i > v_i. \end{aligned} \quad (18)$$

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,

$$N_d \in \mathcal{U}_{\text{disc}}([0, 200]). \quad (19)$$

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, obtaining a non-noisy vector  $\mathbf{c} = [N_1, \dots, N_D]^\top$ .

Recall that the value  $\alpha_i \cdot 100$  represents the percentages of microglial cells in the  $i$ -th filtered image, obtained filtering the  $d$  images with the  $i$ -th threshold vector  $\mathbf{t}^{(i)}$ . We consider  $T$  threshold vectors  $\mathbf{t}^{(i)}$  such as

$$\alpha_i = \frac{i-1}{T-1}, \quad i = 1, \dots, T,$$

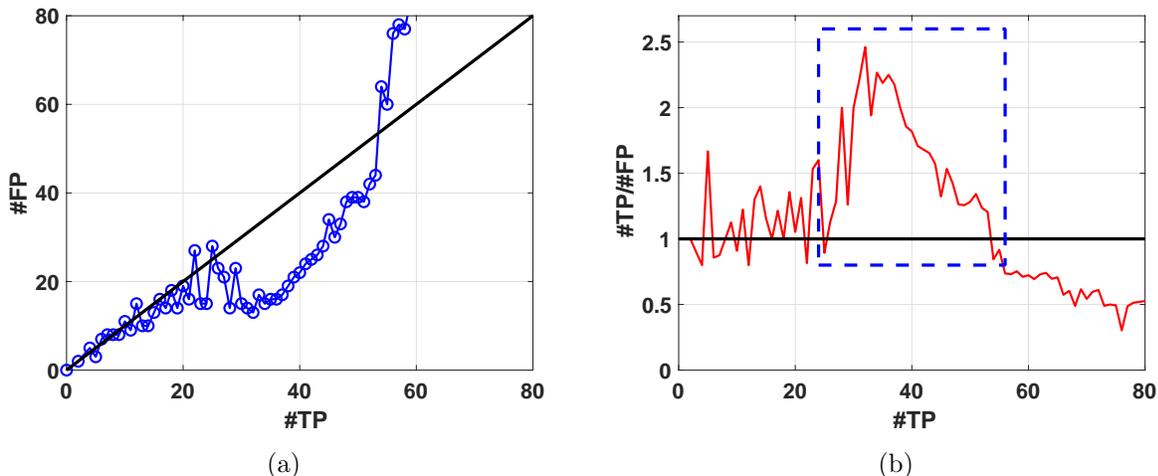


Figure 6: **(a)** The minimum number of false positives ( $\#FP$ ) obtained by the optimal conditional thresholds, versus number of true positives ( $\#TP$ ), in a range of values where we can obtain  $SNR \geq 1$  (for a specific image of the dataset). The solid black line represents the straight line  $\#FP = \#TP$ , then when the blue curve is below the black straight line, we have  $SNR \geq 1$ . **(b)** The  $SNR = \frac{\#FP}{\#TP}$  corresponding to the values in Figure 5. The rectangular region depicted by dashed lines, shows the values of SNR bigger than 1 (obtained by the optimal conditional thresholds for  $25 \leq \#TP \leq 53$ ).

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  $F_i$ ,

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

where we can write that the minimum and maximum values are function of  $\alpha_i$  as in Eq. (6), for instance,

$$v_i = \left\lfloor 24 \left( \frac{1}{1 + \exp(-3\alpha_i)} \right) - 12 \right\rfloor, \quad (20)$$

$$s_i = \left\lfloor 1 + 40 \left( \frac{1}{1 + \exp(-3.5\alpha_i)} \right) - 20 \right\rfloor. \quad (21)$$

Figure 7(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, in  $D = 10^4$  independent runs (i.e., we have  $D = 10^4$  analyzed images). 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). We recall that the matrix  $\mathbf{R}$  has dimensions  $D \times T$  and we can write the smoothing vector  $\hat{\mathbf{c}}_T = \mathbf{R} (\mathbf{R}^T \mathbf{R})^{-1} \mathbf{R}^T \mathbf{c}$  as function of  $T$ , where  $\mathbf{c} = [N_1, \dots, N_D]^T$  in (8). Computing the

mean square error (MSE) in smoothing, i.e.,

$$\text{MSE}(T) = \frac{1}{D} \|\widehat{\mathbf{c}}_T - \mathbf{c}\|^2 = \frac{1}{D} \|\mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{c} - \mathbf{c}\|^2, \quad (22)$$

where  $\|\cdot\|$  is the Euclidean distance. As shown in Figure 7(b), we can observe that the MSE is decreasing quickly as  $T$  grows. Note that when  $T = 200$ , the MSE is already virtually zero. Therefore, by increasing  $T$  we can accurately reproduce the expert’s opinion in the training set. In a new test image, where the number microglial cells and the number of false alarms follow the model given above in Eqs. (18), (19), (20) and (21), the MSE in prediction will be very close to the values obtained in Figure 7(b).

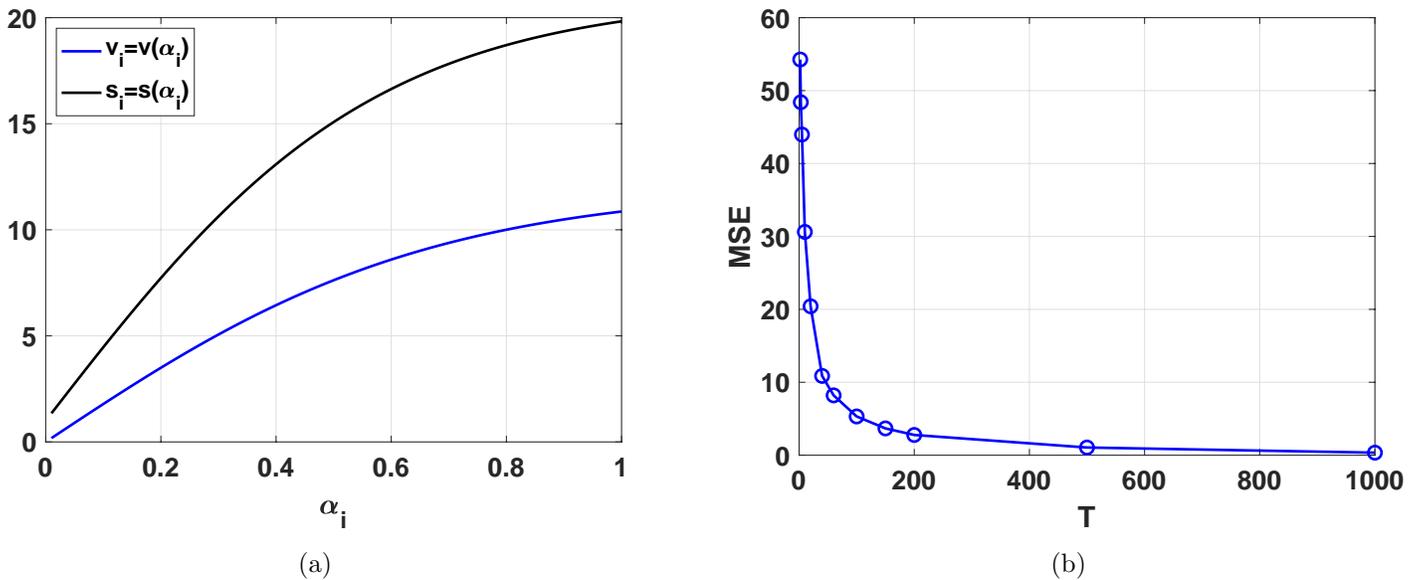


Figure 7: **(a)** The two functions  $v_i = v(\alpha_i)$  and  $s_i = s(\alpha_i)$  in Eqs.(20) and (21). Recall that  $\min F_i = v_i$  and  $\max F_i = s_i$ . **(b)** The mean square error (MSE) in Eq. (22) as function of the number of threshold vectors used, i.e.,  $T$ .

## 7.2 Second experiment: using nonlinear bases

In this section, we repeat the experiment described in the previous section, but also considering quadratic nonlinear bases as in Eq. (15). Therefore, the matrix  $\mathbf{R}$  has dimension  $D \times 2T$  instead of  $D \times T$ , as described in Section 5.1.

Figure 8(a) depicts the results in terms of MSE versus  $T$ . This figure is represented in log-scale, to facilitate the visualization. The MSE curve of the experiment in the previous section is shown with a solid line (this curve is also given in Figure 7(b), but using a standard scale), whereas the MSE curve corresponding to the use of quadratic nonlinear bases is depicted with a dashed line.

We can observe that introduction of the nonlinear bases improves substantially the performance. This improvement is precisely produced by the nonlinear behavior of the bases: for instance, the MSE with  $T = 200$  and the quadratic bases (see the dashed line in Figure 8(a)) is much smaller than the MSE with  $T = 400$  and using only linear bases (see the solid line in Figure 8(a)). In both cases,  $\mathbf{R}$  has dimension  $D \times 400$  (where  $D = 10^4$ ), but employing non-linear bases we obtain a much better results, even using only 200 different threshold vectors instead of 400. Additionally, the MSE with  $T = 200$  and the quadratic bases is close to the MSE obtained with  $T = 1000$  and only linear bases.

### 7.3 Third experiment: considering noisy labels $N_d$

In this section, we consider again the model above in Eqs. (18), (19), (20) and (21). However, now we assume that the expert provides *noisy* approximations of the number  $N_d$  of microglial cells in the  $d$ -th image. Here, instead of a non-noisy vector  $\mathbf{c} = [N_1, \dots, N_D]^\top$ , we have noisy vector

$$\tilde{\mathbf{c}} = [\tilde{N}_1, \dots, \tilde{N}_D]^\top,$$

where

$$\tilde{N}_d = N_d + \epsilon_d, \quad \epsilon_d \sim \mathcal{U}_{\text{disc}}([- \gamma, \gamma]),$$

where  $\epsilon_d$  is a positive uniform noise and  $\gamma \in \{0, \dots, N_d\}$  is a constant integer parameter that determines the power of this noise. Note that, by setting  $\gamma = 0$ , we recover to the non-noisy framework in the previous section. In this example, we set  $\gamma = 5$  and  $\gamma = 10$ . In this scenario, the smoothing vector is given by

$$\hat{\mathbf{c}}_T = \mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \tilde{\mathbf{c}},$$

where we have used the actual observed vector  $\tilde{\mathbf{c}}$  with the noisy data  $\tilde{N}_d$ .

However, the MSE is still computed considering always the true vector  $\mathbf{c} = [N_1, \dots, N_D]^\top$ , exactly as in Eq. (22), i.e.,

$$\text{MSE}(T) = \frac{1}{D} \|\hat{\mathbf{c}}_T - \mathbf{c}\|^2 = \frac{1}{D} \|\mathbf{R} (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \tilde{\mathbf{c}} - \mathbf{c}\|^2. \quad (23)$$

This allows to appreciate the deterioration of the performance of the algorithm as  $\gamma$  grows, as shown in Figure 8(b), 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 the noisy observations of  $N_d$  and regardless the noise power. This proves that the algorithm works properly with noisy labelled images.

## 8 Conclusions

We have introduced an automatic counting algorithm for computing the number of microglial cells in biomedical images. We have designed a linear predictor based on the information provided by filtered images, obtained applying color threshold values to the labelled images in the dataset.

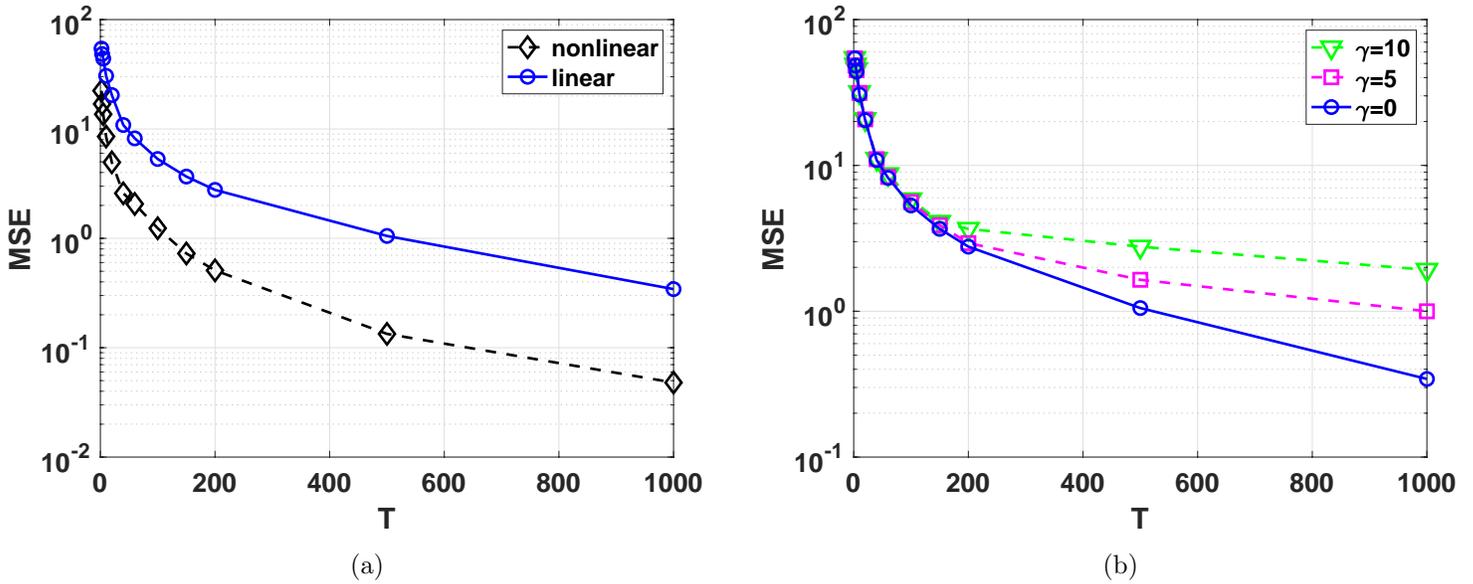


Figure 8: **(a)** The MSE curve in log-scale as function of the number  $T$  of threshold vectors, as in Figure 7(b) (solid line and circles) and using with quadratic nonlinear bases (dashed line and diamonds). **(b)** The MSE curve in log-scale as function of the number  $T$  of used threshold vectors, and different values of  $\gamma = 0, 5, 10$ . Note that  $\gamma = 0$  corresponds to the curve in Figure 7(b) without noisy evaluations of  $N_d$ . In any case, the MSE of the algorithm vanishes to zero as  $T$  grows, regardless the power of the noise in  $N_d$ . Note that the solid line represents the same MSE curve in both figures.

Non-linear extensions and other improvements, jointly with the choice of the threshold values, have been discussed. Different numerical experiments show the capability and consistency of the proposed algorithms, even with noisy labelled images.

The novel approach allows to increase its flexibility and its performance in a very simple and automatic way: considering and adding more threshold vectors and, as a consequence, more columns in the design matrix. Clearly, the proposed schemes could be directly applied to different counting problems of small objects in other types of images. As future lines, we also plan the use of ANNs and the total least squares method due to the presence of noise in the labels (or even more complex techniques as in [12]). Increase the database by artificial images requires additional studies as well.

## Acknowledgement

The authors acknowledge support by Agencia Estatal de Investigación - AEI (project SPGRAPH, ref. num. PID2019-105032GB-I00), Comunidad de Madrid and Universidad Rey Juan Carlos (Proyecto I+D Jvenes Doctores, AUTO-BA-GRAPH, ref. num. F861).

## References

- [1] F. S. Acton. *Numerical Methods That Work*. The Mathematical Association of America, Washington, DC, 1990.
- [2] S. Bradesi. Role of spinal cord glia in the central processing of peripheral pain perception. *Neurogastroenterology & Motility*, 22(5):499–511, 2010.
- [3] R. L. Burden and J. D. Faires. *Numerical Analysis*. Brooks Cole, 2000.
- [4] M. M. Garcia, M. Molina-Ivarez, C. Rodriguez-Rivera, N. Paniagua, E. Quesada, J. A. Uranga, M. I. Rodrogoez-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.
- [5] 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.
- [6] 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.
- [7] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *Advances in neural information processing systems*, 23, 2010.
- [8] F. Llorente, L. Martino, D. Delgado-Gmez, and G. Camps-Valls. Deep importance sampling based on regression for model inversion and emulation. *Digital Signal Processing*, 116:103104, 2021.
- [9] 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.
- [10] 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.
- [11] U. Mitra and H.V. Poor. Neural network techniques for adaptive multiuser demodulation. *IEEE Journal on Selected Areas in Communications*, 12:1460–1470, February 1994.
- [12] 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, 20221.
- [13] Mohamed Nasor, Walid 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.

- [14] 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.
- [15] 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.
- [16] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.
- [17] 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.
- [18] 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.

## A Lower and upper bounds

Let us consider both  $v_i < \infty$ ,  $s_i < \infty$ , finite and known - or estimated - (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

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

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

$$\widehat{N}_{d,\text{lower}} = \left\lfloor \frac{r_{id} - s_i}{\alpha_i} \right\rfloor, \quad \widehat{N}_{d,\text{upper}} = \left\lceil \frac{r_{id} - v_i}{\alpha_i} \right\rceil. \quad (24)$$

where  $\lfloor b \rfloor$  gives as output the greatest integer less or equal to  $b$ , whereas  $\lceil b \rceil$  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:

$$\widehat{N}_{d,\text{lower}} \leq N_d \leq \widehat{N}_{d,\text{upper}}. \quad (25)$$

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