Medical Image Analysis (2022)

Contents lists available at ScienceDirect

Medical Image Analysis

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

Learning to count biological structures with raters' uncertainty

Luca Ciampi[a,1,*], Fabio Carrara[a,1,*], Valentino Totaro[b,c], Raffaele Mazziotti[b,e], Leonardo Lupori[b,c], Carlos Santiago[d], Giuseppe Amato[a], Tommaso Pizzorusso[b,c], Claudio Gennaro[a]

[a] *Institute of Information Science and Technologies (ISTI-CNR), Pisa, Italy*
[b] *Institute of Neuroscience (IN-CNR), Pisa, Italy*
[c] *BIO@SNS lab, Scuola Normale Superiore, Pisa, Italy*
[d] *Institute for Systems and Robotics (ISR/IST), Lisbon, Portugal*
[e] *Department of Neuroscience, Psychology, Drug Research and Child Health (NEUROFARBA), University of Florence, Italy*

ARTICLE INFO

Article history:
Received -
Received in final form -
Accepted -
Available online -

Communicated by -

2020 MSC: 68T07, 92C55

Keywords: Automatic cell counting, Counting with uncertainty, Deep Learning, Biomedical image analysis, Microscopy images, Multi-rater data, Perineuronal nets

ABSTRACT

Exploiting well-labeled training sets has led deep learning models to astonishing results for counting biological structures in microscopy images. However, dealing with weak multi-rater annotations, i.e., when multiple human raters disagree due to non-trivial patterns, remains a relatively unexplored problem. More reliable labels can be obtained by aggregating and averaging the decisions given by several raters to the same data. Still, the scale of the counting task and the limited budget for labeling prohibit this. As a result, making the most with small quantities of multi-rater data is crucial. To this end, we propose a two-stage counting strategy in a weakly labeled data scenario. First, we detect and count the biological structures; then, in the second step, we refine the predictions, increasing the correlation between the scores assigned to the samples and the raters' agreement on the annotations. We assess our methodology on a novel dataset comprising fluorescence microscopy images of mice brains containing extracellular matrix aggregates named perineuronal nets. We demonstrate that we significantly enhance counting performance, improving confidence calibration by taking advantage of the redundant information characterizing the small sets of available multi-rater data.

1. Introduction

Detection and counting of biological structures are among the earliest fields revolutionized by artificial neural networks now dominating state of the art. Several vision models (mostly convolutional networks) have been successfully adopted to localize, segment, and count cells or other structures from microscopy images and even provide counting-density estimation particularly effective in "crowded" scenarios. However, the success of these methods often assumes the availability of a representative set of images with well-labeled biological structures. Whereas, in most cases, those structures can be unambiguously flagged by human raters, here we investigate cell counting under the assumption of weak multi-rater labels, that is, in the presence of non-negligible disagreement between multiple raters. This often occurs when trying to detect and count cells with non-trivial patterns on a large scale, where several factors

*Corresponding authors.
e-mail: luca.ciampi@isti.cnr.it (Luca Ciampi), fabio.carrara@isti.cnr.it (Fabio Carrara)
[1] They contribute equally to this work.

can produce weak labels; raters can incur errors due to fatigue or inexperience (common when hiring less-experienced raters to reduce labeling time) or have different judgments that can span from conservative to liberal when assigning labels.

More reliable labels can be obtained by naively averaging the decisions taken by several raters on the same data, i.e., multi-rating can be leveraged to create stronger singular annotations. However, such data are expensive to obtain and often available only in small quantities. On the other hand, given the scale of training sets needed for deep learning methodologies and the counting task, we consider here the case in which few expert raters, on a limited labeling budget, tend to label new data rather than label the same images more than once. This results in large, single-rater weakly labeled datasets very likely to contain errors, and only small multi-labeled subsets Campagner et al. (2021).

In this setting, we propose a two-stage counting methodology for biological structures, where each stage is devised to fully exploit the annotations in each data subset. The first stage adopts existing solutions on weakly-labeled data to detect and count cells. Specifically, we compare three common CNN-based methodologies already present in the literature — a) *segment and count*, b) *detect and count*, and c) *count by density estimation*. The goal is to investigate their counting ability when trained with data characterized by significant label noise from errors introduced by raters, and to derive *uncalibrated* scores from the models' output that have not been designed to correlate with the quality of the predictions. In the second stage, using a small set of multi-rater data, we define a rescoring model that refines predictions of the first stage, increasing the correlation between the scores assigned by the model to the predictions and the raters' agreement on the sample labels. We refer to scores produced in this stage as *calibrated* scores, in contrast with the uncalibrated ones previously assigned; these final scores can eventually be used to filter low-quality predictions. Advantages in operating in two-stage are twofold: i) the localization of objects are decoupled from their scoring, thus obtaining an overall improved counting model when the latter is fine-tuned even on a few multiple raters' judgments, and, ii) we can easily swap the first stage with any state-of-the-art localization and counting method, making the pipeline model-agnostic and "future proof" — any subsequent work can simply plug-in the best detector and still use the proposed pipeline when multi-rater data is available.

We evaluate the various stages of our pipeline on a novel weakly-labeled dot-annotated dataset that we publicly release (Ciampi et al., 2021a). It consists of a collection of fluorescence microscopy images of mice brain slices containing Perineuronal Nets (PNNs), extracellular matrix aggregates surrounding the cell body of a large number of neurons throughout the nervous system. Multiple expert raters have labeled a small part of the dataset; nonetheless, the maximum agreement between raters is roughly 70%, highlighting the need for an automated counting technique that accounts for uncertain patterns. We show through experimental evaluation that our proposed two-stage pipeline, independently from the specific implementation of each stage, can improve the performance of several state-of-the-art counting methods on multiple ground-truth settings, from liberal to conservative ones.

To summarize, the main contributions of this work are

- the proposal of a two-stage pipeline that improves biological structures counting in multi-rater weak-labels settings,
- the introduction of a novel dot-annotated dataset for cell counting in microscopy images (specifically, perineuronal nets) comprised of a large weakly-labeled single-rater subset and a small multi-rater subset, and
- the public release of the pretrained models for automatic perineuronal nets counting in fluorescence images.

We organize the rest of the paper as follows. We review related work in Section 2. In Section 3, we describe the datasets used in our experiments. Section 4 formalizes the proposed methodologies, while Section 5 outlines the performed experiments showing the obtained results. Finally, Section 6 concludes the paper suggesting some insights on future directions. Code and trained models are publicly available at `https://`

github.com/ciampluca/counting_perineuronal_nets.

2. Related Work

2.1. Visual Counting.

Visual counting aims at estimating the number of object instances, like people (Boominathan et al., 2016) or vehicles (Ciampi et al., 2021b), in images or video frames (Lempitsky and Zisserman, 2010). Current solutions are formulated as supervised deep learning-based problems belonging to one of two main categories: counting by *detection* and counting by *regression*. Detection-based approaches, such as Amato et al. (2019, 2018) and Laradji et al. (2018), require prior detection of the single instances of objects. On the other hand, regression-based techniques like Oñoro-Rubio and López-Sastre (2016), Ciampi et al. (2020) and Zhang et al. (2016) try to establish a direct mapping between the image features and the number of objects in the scene, either directly or via the estimation of a target map, such as a density or a segmentation map, i.e., a real-valued or integer-valued function, respectively. Regression techniques show superior performance in crowded and highly-occluded scenarios but often lose the ability to locate objects precisely.

2.2. Microscope Cell Counting.

Counting biological structures like cells in microscopy images is a crucial step to diagnose many diseases (Venkatalakshmi and Thilagavathi, 2013). Several automatic cell counting methods have been proposed over the years to facilitate this tedious and challenging task. Compared to a typical counting task, microscopy images present different challenges, such as low image contrast, significant cell shape and count variance, and superposition of cells, leading to occlusions. As such, both detection-based and regression-based methods have been proposed. In the former category, Arteta et al. (2016a) introduced a tree-structured discrete graphical model exploited to select and label a set of non-overlapping regions in the image by global optimization of a classification score. More recently, Paulauskaite-Taraseviciene et al. (2019) exploited the Mask R-CNN instance segmentation framework (He et al., 2020) to detect overlapping cells, whereas Dou et al. (2017) used a CNN to segment biological structures from 3D medical images. A comprehensive survey about deep learning algorithms used in medical image analysis, including cell detection in microscopy images, is given by Litjens et al. (2017).

Recent efforts also focused on regression-based approaches that cope better with overlapped objects and crowded scenarios. For example, Guo et al. (2021) proposed SAU-Net, an extension of the U-Net segmentation network (Ronneberger et al., 2015) with a Self-Attention module for counting by density regression. In Aich and Stavness (2018), another regression-based counting model is introduced, enhanced by regulating activation maps from the final convolution layer of the network with coarse ground-truth activation maps generated from simple dot annotations. More, in Cohen et al. (2017), the authors proposed a novel deep neural network architecture adapted from the Inception family (Szegedy et al., 2015) of networks called Count-ception. In Huang et al. (2020), the so-called CSRNet (Li et al., 2018), a regression-based CNN suitable for counting objects in several contexts, is employed to estimate cell densities in immunohistochemically stained sections of breast tissue. Jiang and Yu proposed two different regression-based cell counting approaches (Jiang and Yu, 2021, 2020b), again, based on the estimation of density maps. Finally, authors in He et al. (2021) presented another regression model based on density estimation where auxiliary convolutional neural networks are employed to assist in the training of intermediate layers. Other regression-based strategies have also been devised to deal with densely concentrated cells but still generating individual cell detections, such as Falk et al. (2018), Tofighi et al. (2019), Koyuncu et al. (2020) and Xie et al. (2018). These approaches first generate intermediate maps that indicate the likelihood of each pixel being the center of a cell in the image, and then convert them into detections by applying some form of Non-Maximum Suppression (NMS).

Concerning the automatic counting of PNNs, previous solutions are often based on brittle hand-crafted computer vision pipelines, such as in Slaker et al. (2016). To the best of our

knowledge, we are the first to use deep-learning solutions to address the counting of perineuronal nets and its specific challenges, such as the extreme inter-image variance of the number and the non-trivial appearance of PNNs that cause difficulty to precisely count them, even for human experts.

2.3. Learning with multi-rater data

When dealing with multi-rater data, most existing methodologies apply simple strategies like majority voting to obtain a unique set of ground-truth labels. However, approaches exploiting multi-rater data more effectively exist and are not new; in their seminal work, Dawid and Skene (1979) proposed an Expectation-Maximization algorithm to estimate raters' error-rates in multinomial multi-rater data. More recent works aim at modeling raters' reliability for aggregating or filtering labels, such as Rodrigues et al. (2013) and Zhang and Obradovic (2012). We refer the reader to Zheng et al. (2017) for a review of approaches and challenges in inference with multi-rater data. The recent trend is instead increasingly exploiting multi-rater data, when possible, to increase data efficiency; in the biomedical context, Wei et al. (2021) proposed a curriculum learning approach on samples with increasing raters' agreement for histopathology image classification, while Mirikharaji et al. (2021) tackles skin-lesion segmentation by building multiple models (one for each set of raters' labels) and then aggregating models predictions. To the best of our knowledge, the only proposed counting approach dealing with multi-rater data is Arteta et al. (2016b), where authors train a supervised algorithm to count antarctic penguins in images dot-annotated by non-professional volunteers; multi-rater labels are mainly exploited to estimate the object scale, which varies wildly in their dataset (the diameter of a penguin varies between 15 and 700 pixels) and is instead fixed in our scenario. When dealing with constant scale objects, as in our microscopy images scenario, their solution resembles Falk et al. (2018), a segmentation-based approach adopted and compared in this work. Moreover, instead of requiring large multi-rater training sets, our approach is designed to train on a large single-rater set plus a small multi-rater set, lowering the total labeling cost.

3. Datasets

In this section, we describe the employed datasets, summarized in Table 1. We consider four publicly available single-rater datasets widely used in the context of the microscope cell counting task that we exploit for comparing the adopted counting architectures against the state of the art. Those will serve as baselines for our counting framework. Then, we illustrate our novel collection of fluorescence microscopy images containing perineuronal nets labeled by multiple professional raters, which we use for the experimental evaluation of our two-stage counting pipeline.

3.1. VGG Cells Dataset

This public dot-annotated dataset was introduced by Lempitsky and Zisserman (2010). It contains 200 RGB synthetic images simulating bacterial cells from fluorescence-light microscopy at various focal distances. Images have a fixed size of 256×256×3 pixels, and the cells are designed to be clustered and occluded with each other.

3.2. MBM Cells Dataset

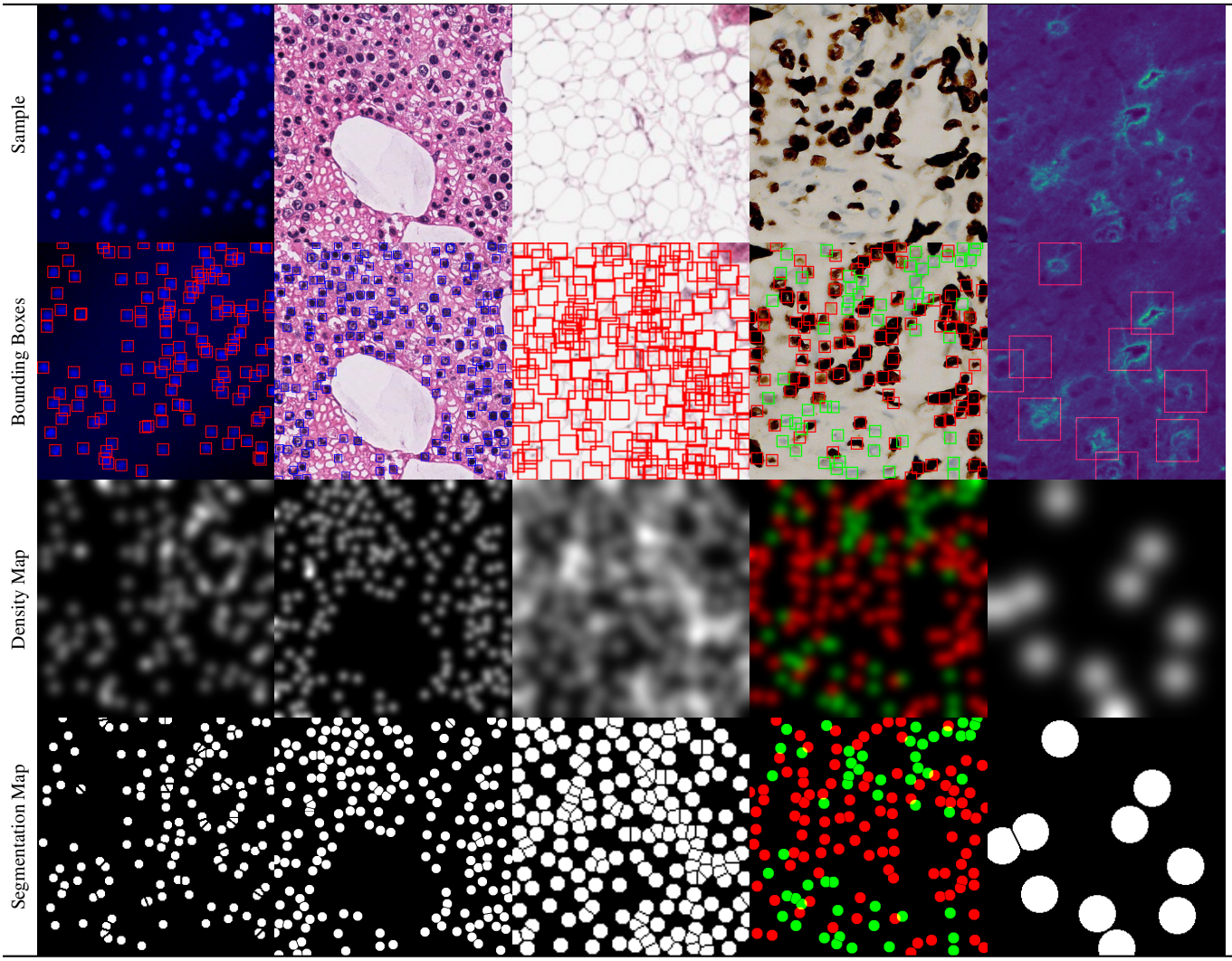
The *Modified Bone Marrow (MBM)* dataset contains 44 RGB dot-annotated microscopy images of human bone marrow with various cell types stained blue. The original dataset was collected by Kainz et al. (2015), acquiring 11 microscopy images from the human bone marrow tissues of 8 different patients. The original images are 1200×1200×3 pixels in size, but authors in Cohen et al. (2017) split each of them into four images with the size of 600×600 pixels.

3.3. ADI Cells Dataset

The *Adipocyte (ADI)* dataset is a human subcutaneous adipose tissue dot-annotated collection of microscopy images introduced by (Cohen et al., 2017). It consists of 200 Regions Of Interest (ROI) of 150×150×3 pixels in size sampled from high-resolution histology slides representing adipocyte cells. The average cell count across all images is 165 ± 44.2, and the size of the biological structures can vary dramatically, representing a challenging test case for automated cell counting procedures.

Table 1: **Summary of datasets.** We report some numerical characteristics on the top of the table. Below, we show a dataset image sample (for PNN, we show a 640x640 crop) and, in the last three rows, the associated targets exploited during training. Specifically, the targets are generated from dot annotations using different procedures: i) *bounding boxes* are produced by generating squares with side s, ii) *density maps* are built by superimposing Gaussian kernels G_σ, and iii) *segmentation maps* are generated drawing discs with radius r separated by background ridges. Bounding boxes, Gaussian kernels and discs are centered in the dot-annotated locations; the s, σ, and r parameters are fixed and dataset-specific, depending on the typical object size in the images. Targets in the multi-class BCData dataset are shown in false colors.

					PNN	
	VGG Lempitsky and Zisserman (2010)	MBM Kainz et al. (2015)	ADI Cohen et al. (2017)	BCData Huang et al. (2020)	1 rater (PNN-SR)	7 raters (PNN-MR)
subjects	none (synthetic)	8	N/A	394	1	1
images	200	44	200	1,338	25	12
size	256×256	600×600	150×150	640×640	≥8184×6163	2000×2000
objects	35,192	5,553	29,684	181,074	34,620	2,351
obj./img.	176±61	126±33	148±32	135±68	1,385±590	196±43



3.4. BCData

The *Breast tumor Cell Dataset (BCData)* (Huang et al., 2020) is a recent collection of 1,338 images based on Ki-67 staining with 181,074 dot-annotated cells divided into two classes (positive and negative tumor cells, i.e., malignant and not malignant, respectively). Unlike other datasets, BCData is not only large in scale concerning the labeled objects but also considering the number of different unique patient cases (that are 394). The size of each image is fixed to 640×640×3 pixels, and the authors divided the dataset into training, validation, and testing split at a ratio of approximately 6:1:3 (803, 133, and 402 images, respectively).

3.5. PNN Dataset

Perineuronal Nets (PNNs) are extracellular matrix aggregates surrounding the cell body of many neurons throughout the nervous system; their alterations are associated with several physiological processes and pathological conditions, e.g., psychiatric disorders such as schizophrenia (Berretta et al., 2015). This contributed to the increasing interest in PNNs research spanning various conditions and animal models, including rodents (Napoli et al., 2020; Boggio et al., 2019; Fawcett et al., 2019), primates (Mueller et al., 2016), and even human brain samples (Rogers et al., 2018). We collect and publicly release (Ciampi et al., 2021a) a novel dataset of fluorescence microscopy images of mice brain slices containing annotations for perineuronal nets. Specifically, we obtained 50μm brain slices from C57BL6/J adult mice (transcardially perfused with 4% paraformaldehyde). PNNs were stained with a green fluorescent marker by sequentially incubating them with biotinylated Wisteria floribunda Lectin (WFA) and streptavidin Alexa Fluor™ 488 conjugate. We acquired images with a fluorescence microscope (Zeiss Apotome.2). PNNs were manually annotated by neuroscientists and biologists from the laboratory of Prof. Pizzorusso, a leading expert in the field of the PNNs since 2002 (Pizzorusso et al., 2002). For a detailed description of the experimental procedures for generating the samples in the dataset, we refer the reader to (Ciampi et al., 2021a).

The dataset is composed of two subsets — a large single-rater subset (*PNN-SR*) and a smaller multi-rater subset (*PNN-MR*) — described below and depicted in Figure 1.

1) *PNN-SR*: consists of 25 images having different sizes ranging from 8184×6163 to 15120×9477 pixels. The extreme size of the images makes their use impracticable by AI-based Computer Vision tools unless dividing them into smaller regions. Among all the images, there are roughly 34k annotated PNNs, varying from a few dozens to some thousand per image, depending on the considered portion of the brain. An expert manually created annotations by putting a dot over the centroid of each identified PNN. Since PNNs are often not easy to find and are subject to different judgments depending on the rater, the training labels are sure to contain errors. Thus, this subset can be considered *weakly-annotated*.

2) *PNN-MR*: comprises 12 microscopic images of 2000×2000 pixels representing different portions of a mouse brain, with a total of 2,532 dot-annotated PNNs. The main peculiarity of this subset is that the annotation procedure has been performed by seven different raters, showing a remarkable discrepancy between the various judgments. As shown in Fig. 1.C, more than 40% of the PNN has not been annotated by the majority of raters (3 or less of the 7 raters), expressing the difficulty of achieving error-free assessments by a single rater.

4. Methodology

Most counting approaches, both regression- or detection-based, can obtain good detections of the objects and a good prediction of the total count when using well-labeled training sets, as already demonstrated by cell counting literature. However, under the presence of weak labels, these models tend to detect also low-confidence or spurious patterns with high confidence for multiple reasons; for example, regression-based models such as density-map estimators do not model the confidence of a detected pattern, thus disabling any filtering step, and detection-based approaches often assign overestimated detection scores to maximize recall that does not correlate with the "objectness" of the pattern. Although it is feasible to modify current models to better express this correlation, training

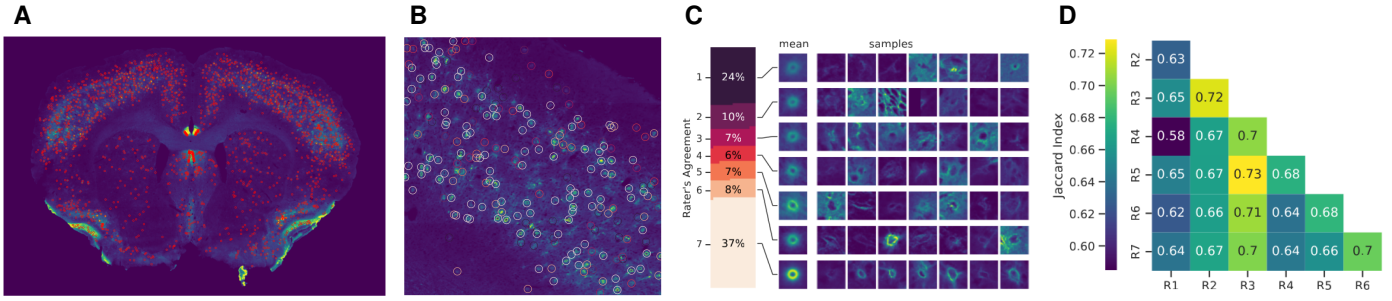


Fig. 1: **PNN Dataset.** *From Left to Right*: **A.** A sample from the single-rater subset (PNN-SR) with dot annotations in red. **B.** A sample from the multi-rater subset (PNN-MR, labeled by 7 raters); the color of the circles encodes the number of raters' that identified that PNN following the legend on the adjacent figure. **C.** Breakdown of the PNN-MR subset by raters' agreement level. We show some sample patches centered in the locations identifying PNNs, together with the mean patches; percentages represent the fraction of the total number of PNNs localized by i raters, for $i = 1, \ldots, 7$. **D.** Jaccard Index between the PNN sets found by each rater.

them would necessitate large multi-rater datasets (where each pattern is labeled with a degree of objectness or quality) that are expensive to obtain.

Here, we assume to have access to a large weakly-labeled single-rater dataset and only a small multi-rater subset. With the former, we exploit the power of existing counting solutions, and with the latter, we devise an additional rescoring stage to cope with the problems discussed above. Specifically, we model the counting task as a process (depicted in Fig. 2) comprised of two stages, each having its separate training phase. The first stage follows standard approaches producing a set of coordinates localizing objects in the input image. In the second stage, we consider the objects previously localized, and we assign them an "objectness" score that correlates with the raters' agreement on their detection, i.e., a higher score indicates a higher probability that most or all human raters detect that object. To do so, we define a scorer module that inputs a small cropped patch containing the previously localized objects and outputs a scalar score. We train it in a supervised fashion with a small set of multi-rater data, where the agreement between multiple raters reflects the pattern's certainty. In practice, the output of the scorer model provides a new "objectness" score that practitioners can use to exclude or include samples from the total count.

We describe the two stages in more detail below.

4.1. Localization Stage

For this stage, we assume to have a collection of N images with dot annotations $\mathcal{X} = \{(I_1, \hat{L_1}), \ldots, (I_N, \hat{L_N})\}$, where I_i is the i-th image and $\hat{L_i}$ is the set of coordinates of the structures to be counted in image I_i labeled by a human rater. We assume $\mathcal{X}$ is large and may have weak labels, e.g., it may contain spurious (false positives) and missing annotations (false negatives).

A localization model f_θ applied to the input image I produces a set of coordinates $L = \{p_1, \ldots, p_C \mid p_i \in \mathbb{R}^2\}$ localizing the objects to be counted. This model is trained using location data $\mathcal{X}$ and can be implemented following several different strategies; here, we test three successful approaches from the literature, that are *segmentation*, *detection*, and *density estimation*, described below.

4.1.1. Localization by Segmentation

For this approach, we follow Falk et al. (2018), i.e., we first produce a segmentation map $S = f_\theta(I) \in [0, 1]^{H \times W}$ for the input image I having height H and width W. S is then thresholded and further processed to extract connected components. The centroids of those components form the output localizations L. This solution can accommodate variable-shaped objects, but segmentation annotations are usually very expensive to produce. Here, we generate the target segmentation maps $\hat{S} \in [0, 1]^{H \times W}$ by imposing a disc centered in the dot-annotated position. The radius of the disc is fixed and depends on the typical object size in the dataset. A narrow ridge separates overlapping discs. In case of multiple object classes, the network outputs one segmentation map per class, and target generation is performed independently for each class. An example of a target segmentation map is reported in Table 1. The model is trained to minimize the weighted binary cross-entropy between pixels of the output and target maps; more weight is assigned to

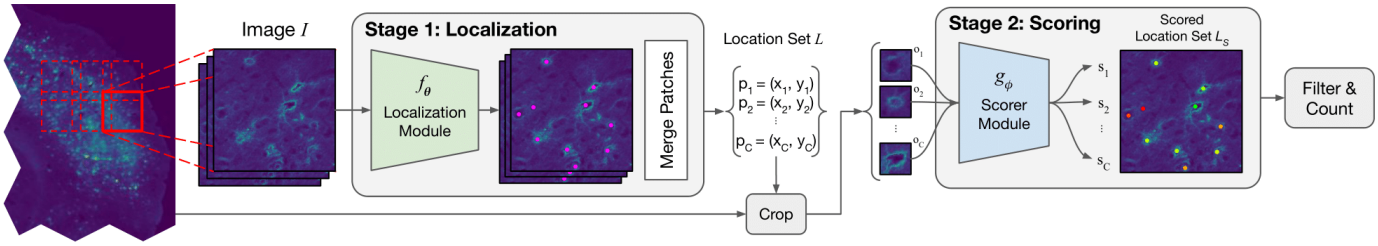


Fig. 2: **Proposed counting pipeline.** We model the task as a two-stage process. In the first one, we detect the objects exploiting a localization model f_θ, previously trained on a large collection of dot-annotated images that may have weak labels. In the second stage, we employ a scorer model g_ϕ that assigns to the objects localized in the previous step an "objectness" score, which we correlate with the pattern uncertainty quantified by the agreement's level.

more important pixels of the map, such as background ones near foreground objects. More details of the generation procedure of segmentation targets are available in Falk et al. (2018). As in Falk et al. (2018), we implement f_θ as a standard U-Net (Ronneberger et al., 2015). In the following, we will refer to this method as *S-UNet*.

4.1.2. *Localization by Detection*

For this approach, we employ the Faster-RCNN model for visual object detection (Ren et al., 2017); f_θ produces a list of bounding boxes following the standard two-stage detection paradigm. In the first step, a Region Proposal Network (RPN) generates the region proposals that might contain objects, slicing pre-defined region boxes (called anchors); in the second step, these priors are refined, performing a regression to the coordinates of bounding boxes precisely localizing the objects inside these Regions of Interest (RoIs). The centers of the boxes comprise the final localization of the entities. Targets are produced by generating squared bounding boxes centered in the dot-annotated data with a fixed side, again, depending on the typical object size in the dataset. We implement f_θ as a Faster-RCNN network with a Feature Pyramid Network module and a ResNet-50 backbone. From now on, we will refer to this method as *FRCNN*.

4.1.3. *Localization by Density Estimation*

We also tested density-estimation approaches known for delivering excellent counting performances, especially in "crowded" scenarios. Using this approach, we learn a regression model producing a density map $D = f_\theta(I) \in \mathbb{R}^{H \times W}$ from an input image of height H and width W. Each pixel of D corresponds to the *quantity* of the objects present at that precise point. Thus, the notion of density map loosely corresponds to the physical/mathematical notion of density; the number of objects n in an image sub-region $P \subseteq I$ is estimated by integrating D over P, i.e., summing up pixel values in the considered region, $n = \sum_{p \in P} D_p$. Although these approaches are not intended to localize objects, a coarse localization can be obtained by analyzing the estimated density map, in particular by finding the top-n maximum local peaks of it, as already done in Xie et al. (2016). During training, the target density maps are produced by superimposing Gaussian kernels G_σ centered in the dot-annotated locations; the spread parameter σ is fixed and depends on the typical object size in the dataset. In case of multiple object classes, the network outputs one density map per class, and target generation is performed independently for each class. An example of a target density map is reported in Table 1. The model is trained by minimizing the mean squared error loss between target and output density maps. We implement f_θ exploiting the Congested Scene Recognition Network (CSRNet) (Li et al., 2018), a CNN for density estimation comprised of a modified VGG-16 network (Simonyan and Zisserman, 2015) for feature extraction and a series of dilated convolutional layers (Yu and Koltun, 2016) to extract deeper information of saliency and, at the same time, maintaining the output resolution. We will refer to this method as *D-CSRNet*.

Once objects are localized, the "objectness" of each prediction needs to be quantified to permit filtering of false positives or negatives that inevitably leak from labeling errors. We cope with this task in the subsequent separate stage, but for compari-

son, we also consider deriving objectness scores from the three accounted localization models alone as a baseline. Among the considered models, *FRCNN* is the only one that natively outputs a score $\in [0, 1]$ stating the probability of containing an object inside the bounding box that we can use as objectness score. On the other hand, *S-UNet* and *D-CSRNet* do not provide directly a score that can be used for filtering predictions. For *S-UNet*, we derive a score $\in [0, 1]$ associated with each localized object by taking the maximum value of the corresponding connected component found in the predicted segmentation map S. Regarding the *D-CSRNet* model, we instead infer a score by taking the local maximum peak values of the predicted density map. However, none of these scores are defined to correlate with the pattern uncertainty that instead needs to be explicitly modeled. We do that in the second stage of our pipeline.

4.2. Scoring Stage

The goal of this stage is to define a model that scores the certainty of a pattern; higher scores should represent objects localized by most human raters, while lower scores should indicate dubious patterns.

Given the coordinates p of an object in image I localized with one of the approaches in the previous stage, we define a scorer model g_ϕ that assigns to the object a scalar objectness score $s = g_\phi(o)$, with o the squared sub-patch of the image I centered in p containing the object. To train g_ϕ, we assume to have a small set of images where objects have been labeled by K different raters; this produces a training set $\mathcal{X}' = \{(o_1, a_1), \dots, (o_M, a_M)\}$, where $o_i \in \mathbb{R}^{l \times l}$ is the image sub-region containing the i-th localized object, and $a_i \in \{0, \dots, K\}$ is the raters' agreement, i.e., the number of raters who localized that object. Regions containing no localized objects ($a = 0$) are used as negative samples during training. In the prediction phase of the entire pipeline, g_ϕ is fed with patches extracted from the input image using the coordinates found by the previous localization stage.

Although this rescoring stage is novel in counting pipelines, we can formulate it as well-known problems and implement it following existing solutions. Below we propose several methodologies that can be adopted for training the g_ϕ model. It is worth noting that the s score takes on different values depending on the adopted method.

4.2.1. Agreement Regression (AR)

A simple baseline is directly regressing scores from the input patches. In this formulation, g_ϕ produces a scalar output and is trained to directly regress the normalized raters' agreement a/K from the object patch. Specifically, we minimize

$$\mathcal{L}(\mathcal{X}'; \phi) = \frac{1}{2} \sum_{(o,a) \in \mathcal{X}'} \left(\frac{a}{K} - g_\phi(o) \right)^2 , \tag{1}$$

where o is a squared image patch containing a localized object and a/K is the fraction of raters localizing that object.

4.2.2. Agreement Classification (AC)

Another simple baseline comprises classifying the input patches in agreement levels. In this formulation, we consider the $K+1$ agreement values $a \in \{0, \dots, K\}$ (including the 0 value as background samples) as separate classes into which objects can be classified. The model g_ϕ produces a $(K+1)$-way softmax output that is trained with standard cross-entropy loss

$$\mathcal{L}(\mathcal{X}'; \phi) = - \sum_{(o,a) \in \mathcal{X}'} \log(g^a_\phi(o)) , \tag{2}$$

where $g^i_\phi(o)$ indicates the i-th output of the model. The final scalar score s is obtained as the (normalized) expected value of the class over the output categorical distribution

$$s(o) = \frac{1}{K} \sum_{i=0}^{K} i \cdot g^i_\phi(o) . \tag{3}$$

4.2.3. Agreement Ordinal Regression (OR)

We formulate the scoring problem as an ordinal regression problem with $K + 1$ ordered categories from the lowest to the highest agreement. Similarly to agreement regression, g_ϕ produces a scalar output but is trained following Pedregosa et al. (2017). Along with model parameters, a set of K ordered thresholds $\Theta = \{\theta_i\}_{i=0}^{K-1}, \theta_0 < \theta_1 < \cdots < \theta_{K-1}$ are defined as learnable parameters. Given the model scalar output $s = g_\phi(o)$, we model

$$P(a \le k | o) = \sigma(\theta_k - s) \quad k = 0, \dots, K-1 , \tag{4}$$

where σ is the sigmoid function, and thus

$$\begin{aligned} y_k(o) &= P(a = k|o) \\ &= P(a \leq k|o) - P(a \leq k-1|o) \\ &= \begin{cases} \sigma(\theta_0 - s) & \text{if } k = 0\,, \\ \sigma(\theta_k - s) - \sigma(\theta_{k-1} - s) & \text{if } k = 1 \dots K-1\,, \text{and} \\ 1 - \sigma(\theta_{K-1} - s) & \text{if } k = K\,. \end{cases} \end{aligned} \tag{5}$$

The models parameters ϕ and thresholds Θ are optimized by minimizing the negative log likelihood of observed samples

$$\mathcal{L}(\mathcal{X}'; \phi, \Theta) = -\sum_{(o,a)\in\mathcal{X}'} \log(y_a(o))\,. \tag{6}$$

The values of θ_i are optionally clipped after each update to kept them ordered. Once trained, we discard Θ and adopt only g_ϕ to output the score s for an object.

4.2.4. Agreement Rank Learning (RL)

Here, we model agreement by learning to rank a tuple of samples with increasing agreement values. Our formulation instantiates a standard pairwise learning to rank approach Burges et al. (2005) with a custom sample loss definition. Specifically, we still define a model with a scalar output $s = g_\phi(o)$, but we employ a different training scheme; given a $(K+1)$-tuple of ordered samples $O = (o_0, \dots, o_K)$ containing one sample per agreement class (i.e., o_i has an agreement value $a_i = i$), we ask our model to produce scores $s_i = g_\phi(o_i)$ that are sorted $s_0 < s_1 < \cdots < s_K$. Translating this constraint in a loss function for the single tuple, we obtain a class-balanced pairwise margin loss

$$\mathcal{L}(o_0, \dots, o_K; \phi) = \frac{1}{K}\sum_{i=1}^{K} \max(m - g_\phi(o_i) + g_\phi(o_{i-1}), 0)\,, \tag{7}$$

where m is a margin hyper-parameter empirically set to 0.1. A dataset of tuples is obtained by repeatedly drawing $K+1$ random samples, one for each agreement class, from the training set $\mathcal{X}'$. This has the advantage to produce large training datasets even when dealing with a small initial multi-rater dataset. The batch loss is obtained as the mean loss over a batch of tuples.

5. Experiments and Results

In this section, we describe the experiments performed to validate our approach and discuss the obtained results. We divided them into three parts. First, we evaluate the considered counting architectures, i.e., the segmentation-based *S-UNet*, the detection-based *FRCNN*, and the density-based *D-CSRNet* approaches, against standard single-rater cell counting benchmarks. The aim is to demonstrate that they work plausibly fine, i.e., they produce comparable results against the state-of-the-art, warding off that results provided by our counting pipeline are not due to a weak poorly-trained baseline. Then, we evaluate the first stage of our pipeline, i.e., the localization stage. Specifically, we perform experiments on our novel PNN dataset, training the three adopted counting architectures with single-rater data having significant label noise from errors introduced by raters. The goal is to detect and count perineuronal nets under this weakly labeled setting, deriving *uncalibrated* scores from the models' output that have not been designed to correlate with the quality of the predictions. Finally, we perform experiments with our multi-rater PNN-MR subset to validate our proposed second stage, i.e., the score calibration stage. Here, we refine predictions of the previous stage, producing *calibrated* scores that increase the correlation with the raters' agreement. We compare it to several baselines and show that it improves counting performances when dealing with uncertain patterns. We report training and implementation details in Appendix B.

5.1. Evaluation of the adopted counting architectures

We evaluate the three adopted counting approaches against the state of the art using VGG Cells, MBM Cells, ADI Cells, and BCData counting benchmarks described in Section 3. All these collections of images are single-rater, i.e., the final available labels belong to a single rater. Even when multiple raters have been employed during the annotation procedure, the final annotations are squashed into a single label per object. In other words, multi-rater annotations are not leveraged if not for the creation of stronger annotations at the expense of the dataset scale.

We follow the evaluation protocol introduced by Lempitsky and Zisserman (2010) and adopted by most subsequent works; we consider a testing subset fixed for all the experiments (100 images for VGG Cells and ADI Cells, and 10 images for MBM

Cells) and training and validation subsets of varying size (N images for each subset) to simulate lower or higher numbers of labeled examples. Following previous work, we set N to 16, 32, and 50 for VGG Cells, to 10, 25, and 50 for ADI Cells, and to 5, 10, 15 for MBM Cells. Concerning BCData, we instead use the training, validation and testing splits provided by Huang et al. (2020). As performance metric, we compute the mean absolute (MAE) counting error

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}\left|c^n_{\mathrm{gt}} - c^n_{\mathrm{pred}}\right|, \tag{8}$$

where N is the number of test images, and c^n_{gt} and c^n_{pred} are the ground-truth and the predicted count of the n-th image, respectively. For VGG Cells, MBM Cells, and ADI Cells, we repeat the experiment 10 times, randomly sampling different splits for each configuration, and we report the mean and standard deviation of the evaluation metric. To check the consistency of the results on these random splits, we also re-implemented the original FCRN-A method presented in Xie et al. (2016), thus performing an exact head-to-head comparison with the same samples being used for training and testing (we report training details in Table B.10). Concerning BCData, we report the mean and the standard deviation of the MAE calculated between 10 runs over the 402 images comprising the test split, changing the random initialization seed each time.

Table 2 reports results on the four datasets. The density-based solution *D-CSRNet* performs best among tested solutions on the VGG dataset and comparably to state of the art. The other two adopted methods, i.e., the segmentation-based *S-UNet* and the detection-based *FRCNN*, exhibit slightly larger errors, consistently with their inherent limitations when applied to "crowded" scenarios with occluded objects like VGG Cells. On the same grounds, *D-CSRNet* achieves best performances on the BCData dataset, and the detection-based *FRCNN* approach is the one that faces more difficulties. On the other hand, in the MBM and the ADI datasets, where the challenges are more related to the object shape variations, all the approaches show competitive results, outperforming state-of-the-art solutions in some cases (e.g., *S-UNet* in MBM and *FRCNN* in ADI). Overall, the tested approaches perform in line with state-of-the-art, and thus we proceed to adopt them in the first localization stage of our pipeline.

5.2. Localization Stage Evaluation

For these experiments, we apply the three previously evaluated solutions to our novel PNN dataset, and we investigate their counting ability in the presence of weakly-labeled data, i.e., under significant label noise introduced by errors of raters.

We consider the large single-rater subset PNN-SR to train the models, whereas we evaluate them on the multi-rater subset PNN-MR. Since the training set contains weak labels, for each solution, we derive a scalar score s from the models' output that can be used to filter low-quality predictions. We refer to the scores obtained in this stage as *uncalibrated* scores, since they have not been designed to correlate with the quality of predictions (we do this in the subsequent scoring stage). For *S-UNet* we set s as the maximum value of the connected component found in the segmentation map. For *FRCNN*, we set s as the classification score that the network already outputs together with the regressed bounding box coordinates localizing the object. Finally, for *D-CSRNet*, we consider as s the value of the higher local peak in the density maps localizing the object.

During the training phase, we split the data into training and validation parts. We do not adopt the common per-image split strategy, as the number of PNNs vastly varies depending on the particular considered brain slice (i.e., image), and thus this strategy would produce unbalanced splits. Instead, we split each image vertically in half, including one half in the training set and the other in the validation set in an alternate fashion.

Due to the extreme size of the images, we process them in patches. During the training phase, we crop squared randomly localized patches from training images. We experiment with different patch sizes of 256, 320, 480, 640, and 800 pixels. At validation time, we divide and process the image in regularly-spaced overlapped patches of the same size used during training (see also Fig. 2), we reconstruct the global output by combining patch predictions, and we compute metrics at the entire image level. For segmentation-based and density-based solutions, image-level maps are obtained by stitching back together the

Table 2: **Comparison of the adopted architectures on standard single-rater counting benchmarks.** For VGG, MBM and ADI we vary the training and validation subsets (N images for each subset), repeating the experiments 10 times. For BCData we use the splits provided by Huang et al. (2020), performing 10 runs changing the seed each time. Mean±st.dev. of MAE is reported.

(a) **VGG Cells** (200 images in total - 100 test images).

Method	N = 16	N = 32	N = 50
Arteta et al., 2016a	N/A	5.06 ± 0.2	N/A
GMN (Lu et al., 2019)	N/A	3.6 ± 0.3	N/A
Lempitsky and Zisserman, 2010	3.8 ± 0.2	3.5 ± 0.2	N/A
VGG-GAP-HR (Aich and Stavness, 2018) *	N/A	2.95**	2.67
SAU-Net (Guo et al., 2021) †	N/A	N/A	2.6 ± 0.4
FCRN-A (Xie et al., 2016)	3.4 ± 0.2	2.9 ± 0.2	2.9 ± 0.2‡
FCRN-A (Xie et al., 2016) §	4.7 ± 0.7	3.3 ± 0.2	3.1 ± 0.1
Count-Ception (Cohen et al., 2017)	2.9 ± 0.5	2.4 ± 0.4	2.3 ± 0.4
CCF (Jiang and Yu, 2020a)	2.8 ± 0.1	2.6 ± 0.1	2.6 ± 0.1
C-FCRN+Aux (He et al., 2021) $		2.3 ± 2.2	
S-UNet (Falk et al., 2018) *(our)*	7.7 ± 2.0	6.8 ± 1.0	4.5 ± 1.0
D-CSRNet (Li et al., 2018) *(our)*	3.7 ± 0.3	2.9 ± 0.3	2.5 ± 0.1
FRCNN (Ren et al., 2017) *(our)*	9.3 ± 0.7	7.5 ± 0.6	7.0 ± 0.4

* They did not report standard deviation. ** They used a validation subet of $100 - N$ images. † They did not use a test subset, but only a $100 - N$ images validation subset. ‡ Reported in their work as N = 64. $ They used a 5-fold cross validation-based evaluation protocol considering the whole dataset. § Re-implemented in this work

(b) **MBM Cells** (44 images in total - 10 test images).

Method	N = 5	N = 10	N = 15
Xie et al., 2018 ‡		36.3 ± 19.4	
FCRN-A (Xie et al., 2016) $	28.9 ± 22.6	22.2 ± 11.6	21.3 ± 9.4
FCRN-A (Xie et al., 2016) §	15.6 ± 4.3	12.4 ± 4.0	12.2 ± 2.9
Marsden et al., 2018 *	23.6 ± 4.6	21.5 ± 4.2	20.5 ± 3.5
Count-Ception (Cohen et al., 2017)	12.6 ± 3.0	10.7 ± 2.5	8.8 ± 2.3
CCF (Jiang and Yu, 2020a) *	9.3 ± 1.4	8.9 ± 0.9	8.6 ± 0.3
C-FCRN+Aux (He et al., 2021) **		6.5 ± 5.2	
SAU-Net (Guo et al., 2021) †	N/A	N/A	5.7 ± 1.2
Jiang and Yu (Jiang and Yu, 2021)	8.2 ± 1.1	6.9 ± 0.9	6.0 ± 0.6
Jiang and Yu (Jiang and Yu, 2020b)	-	-	6.0 ± 0.2
S-UNet (Falk et al., 2018) *(our)*	5.5 ± 1.9	5.9 ± 4.2	5.7 ± 0.9
D-CSRNet (Li et al., 2018) *(our)*	9.4 ± 3.4	7.2 ± 2.0	6.4 ± 1.4
FRCNN (Ren et al., 2017) *(our)*	9.3 ± 1.4	9.0 ± 1.4	8.6 ± 0.8

* They used 14 test images. ** They used a 5-fold cross validation-based evaluation protocol considering the whole dataset. † They did not use a test subset, but only a $44 - N$ images validation subset. ‡ They used a train/test split of 8/3 using full-size images. $ Implemented by (Cohen et al., 2017). § Re-implemented in this work.

(c) **ADI Cells** (200 images in total - 100 test images).

Method	N = 10	N = 25	N = 50
FCRN-A (Xie et al., 2016) §	21.1 ± 4.7	13.1 ± 0.7	11.3 ± 1.1
Count-Ception (Cohen et al., 2017)	25.1 ± 2.9	21.9 ± 2.8	19.4 ± 2.2
CCF (Jiang and Yu, 2020a)	16.9 ± 1.9	14.5 ± 0.4	14.5 ± 0.4
SAU-Net (Guo et al., 2021) †	N/A	N/A	14.2 ± 1.6
Jiang and Yu (Jiang and Yu, 2021)	13.8 ± 0.7	11.6 ± 0.4	10.6 ± 0.3
Jiang and Yu (Jiang and Yu, 2020b)	-	-	10.1 ± 0.1
S-UNet (Falk et al., 2018) *(our)*	16.6 ± 5.5	13.6 ± 1.8	13.7 ± 4.9
D-CSRNet (Li et al., 2018) *(our)*	12.6 ± 1.3	10.8 ± 1.5	8.8 ± 1.0
FRCNN (Ren et al., 2017) *(our)*	10.0 ± 0.9	9.1 ± 0.7	8.7 ± 0.8

† They did not use a test subset, but only a $200 - N$ images validation subset. § Re-implemented in this work.

(d) **BCData** (1,338 images in total - 803 train, 133 val, 402 test); positive and negative cells are malignant and not malignant tumor cells, respectively.

Method	Positive	Negative	All
(Sirinukunwattana et al., 2016) * †	9.1	20.6	14.8
CSRNet (Li et al., 2018) (integr.) * $	9.2	24.8	14.8
U-CSRNet (Huang et al., 2020) (integr.) *	10.0	18.0	14.0
CSRNet (Li et al., 2018) (detect.) * $	7.7	14.1	10.9
U-CSRNet (Huang et al., 2020) (detect.) *	6.8	14.1	10.5
S-UNet (Falk et al., 2018) *(our)*	8.3 ± 0.5	19.7 ± 0.9	14.4 ± 0.7
D-CSRNet (Li et al., 2018) *(our)*	8.3 ± 0.9	16.6 ± 1.4	12.5 ± 0.9
FRCNN (Ren et al., 2017) *(our)*	10.3 ± 0.4	30.9 ± 2.3	20.6 ± 1.3

* They did not report standard deviation. $ Implemented by (Huang et al., 2020), they used ResNet-50 (He et al., 2016) instead of VGG-16 (Simonyan and Zisserman, 2015) for feature extraction. † Implemented by (Huang et al., 2020).

patch-level maps and taking the mean pixel values in the overlap areas. For the detection-based solution, we perform non-maximum suppression of all the bounding boxes predicted in the overlap areas.

In Fig. 3, we show the results obtained by the three solutions (one per column) on the whole multi-rater PNN-MR subset in terms of MAE when varying the patch size and the threshold on the scalar score s. As depicted, patch size does not significantly influence the performance of *FRCNN* and *D-CSRNet*. Thus, for these models, we suggest opting for bigger patch sizes that reduce processing overhead. On the other hand, the *S-UNet* solution is more sensitive to this aspect; due to artifacts in the overlap regions of the segmentation map, different patch sizes induce different score distributions that respond differently to score thresholding. For *S-UNet*, the best performance is obtained with smaller patch sizes together with more conservative threshold values. Note that the density-based solution *D-CSRNet* is the most strained by weakly-labeled training data, achieving the worst overall performance primarily due to a low recall. Moreover, as expected, score thresholding is not effective since the density peak value employed as the score is not expected to locate the PNN precisely and correlate with its "visual quality". Even when counting is performed via density map integration, instead of peak localization and counting, the best counting performance that *D-CSRNet* achieves is an MAE of 90.99. We plot additional metrics in Fig. A.7 in Appendix A.

So far, we experimented on the entire PNN-MR subset, thus

Table 3: **PNN-MR: Performance on different agreement levels.** We report the mean ± st.dev. of MAE considering four sets of ground-truth labels for the whole PNN-MR dataset composed by objects labeled by any number of raters (Any), at least 50%, at least 70%, or all raters, respectively, to simulate different counting policies, from liberal to conservative ones. Models are trained once on the weakly-labeled PNN-SR dataset.

	Raters' Agreement			
	Any ($a \geq 1$) 2351 obj.	$\geq 50\%$ ($a \geq 4$) 1384 obj.	$\geq 70\%$ ($a \geq 5$) 1234 obj.	100% ($a = 7$) 880 obj.
S-UNet	27.6 ± 24.8	15.1 ± 14.1	15.8 ± 11.6	13.8 ± 12.8
FRCNN	27.8 ± 21.6	15.5 ± 13.4	13.3 ± 12.6	7.8 ± 9.9
D-CSRNet	91.5 ± 43.6	21.1 ± 23.0	15.3 ± 20.1	10.9 ± 8.9

including every PNN found by at least one rater in the ground-truth set. Next, we illustrate how the trained models behave when asked to localize only PNNs on which at least a raters agree on their presence. Specifically, we define four sets of ground truth labels for PNN-MR comprising PNNs labeled by at least 1, 4, 5, and 7 out of 7 raters, respectively, simulating different counting policies, from more liberal to more conservative ones; the choice of 4 and 5 raters reflects the rater's agreement above 50% and 70%, respectively, which are two thresholds widely adopted to legitimize labels. In Table 3, we report the results obtained on these four sets by the three tested models in their most effective combination of patch size and threshold values. We observe that models tend to correctly identify and count the PNNs found by more raters, as these are also the clearer and easier-to-spot samples. Although all the models deliver similar performance at the higher agreement levels, at the lowest agreement level ($a \geq 1$), the *D-CSRNet* tends to have an higher MAE due to its inability to achieve a high recall on low-agreement samples. We observe that *FRCNN* tends to achieve lower MAEs when increasing rater's agreement a with respect to the other tested models. We also note that all models show high variability in MAE values computed on different test set images. We deem this is due to particular brain regions where PNNs appear dimmer and thus more difficult for both models and human raters to cope with (see Figure 4). We report additional metrics in Table A.4 in Appendix A.

5.3. Scoring Stage Evaluation

Here, we perform experiments to evaluate the proposed additional scoring stage. The goal is to produce new "objectness" scores that correlate with the raters' agreement; we refer to scores produced in this stage as *calibrated* scores, in contrast with the uncalibrated ones derived in the previous stage. This stage requires a small multi-rater dataset to be used as a training set, and thus we adopt the PNN-MR dataset for both the training and testing phases. To this end, we randomly split the images comprising PNN-MR into train, validation, and test sets following the widely employed 70/15/15 proportion.

First, we assess this scoring stage in a stand-alone way, considering it independently from our overall counting pipeline. We implement the scorer model g_ϕ as a small convolutional network with 8 Conv-GroupNorm-ReLU blocks followed by average pooling and a linear projection producing the desired number of outputs (8 for Agreement Classification, 1 for the rest). We train each calibration methodology presented in Section 4.2 by providing, as inputs, small patches around PNNs centered in the locations provided by the ground-truth labels. Due to the limited size of the dataset, we perform five runs with randomly generated splits. In Fig. 5, we report the distribution of (z-normalized) scores obtained by the tested methods on the PNN patches of the test splits. In addition, we also report the distributions of the uncalibrated scores obtained from the localization models as described in Section 5.2. We notice that scores obtained by Agreement Ordinal Regression (OR) and Agreement Rank Learning (RL) strategies behave best in terms of correlation with the raters' agreement a achieving the highest Pearson's correlation index r of 75%. Those are also the most data-efficient methods, as they operate on pairs or tuples of samples, while Agreement Regression (AR) and Classification (AC) seem to suffer from the limited number of samples. Thus, opting for OR or RL is suggested, as multi-rater data is often limited. Moreover, the mean scores per agreement level of OR and RL tend to follow a steep regression line. In contrast, in other methods like *D-CSRNet* and AR, the score distributions of nearby agreement levels are more overlapped and often with

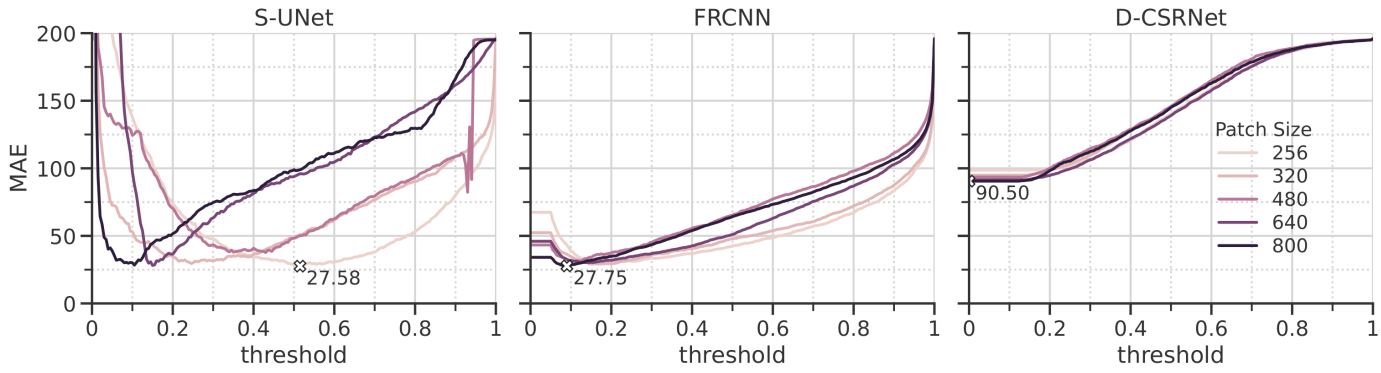


Fig. 3: **PNN-MR: Impact of patch size and score thresholding** on standard segmentation-based (S-UNet), detection-based (FRCNN), and density-based (D-CSRNet) approaches (without score calibration).

non-monotonic means.

Next, we evaluate the effect of the scoring stage on the whole counting pipeline, e.g., when the scoring stage is fed with outputs of the localization stage. We first localize PNNs in test images with the three localization models trained in Section 5.2; for this operation, we set a high-recall threshold for each model, such that filtering can be postponed after rescoring by g_ϕ. We then score the locations found using the trained scoring models and evaluate counting performance for each combination of localization and scorer models. Figure 6 compares the achieved performance in terms of MAE when choosing the best threshold value for the rescored predictions. As baseline, we report also the counting performance of using uncalibrated scores, i.e., without using the proposed rescoring stage. Again, we report results for different ground-truth settings defined by the minimum desired raters' agreement. *S-UNet* combined with Agreement Classification (AC) achieves the best counting performance among most ground-truth configurations, whereas Agreement Ordinal Regression (OR) is the second-best rescoring solution. Despite providing significant boosts compared to other tested rescoring approaches, AC can suffer when multi-rater samples are unbalanced (or missing) among agreement levels, which is fairly common in this application. In those cases, rank-based methods (OR or RL) are known to behave better under these scenarios. However, we leave to future work the evaluation of sample efficiency and of robustness to class unbalance. Note that for *S-UNet* and *FRCNN*, score calibration generally improves the counting performance, specifically for the former where we achieve MAE reductions up to 11.07. On the other hand, when adopting the *D-CSRNet* localization method, we achieve an improvement only on the highest agreement test set; this is mainly due to the limited recall of *D-CSRNet* on the PNN dataset.

6. Conclusion

In this work, we tackled the task of counting biological structures from microscopy images under the assumption that training datasets are characterized by weak multi-rater labels, i.e., in the presence of non-negligible disagreement between multiple raters. This often occurs in medical images where intrinsically non-trivial patterns can produce weak annotations due to raters' judgment differences, even among experts. More robust annotations can be obtained by aggregating and averaging the decisions provided by multiple raters regarding the same data. However, the scale of the counting task and the limited resources dedicated to the labeling process put a damper on this solution. While supervised training with well-defined training sets has been widely studied, dealing with weak multi-rater annotations per image remains a relatively unexplored problem. We considered here the case in which few expert raters mostly annotate novel data and check only a small portion of already labeled images, i.e., to have large, single-rater weakly labeled datasets and only small subsets labeled by multiple raters.

In this setting, we proposed a two-stage counting strategy,where each stage is devised to make the best of the annotations available in each data subset. The first stage

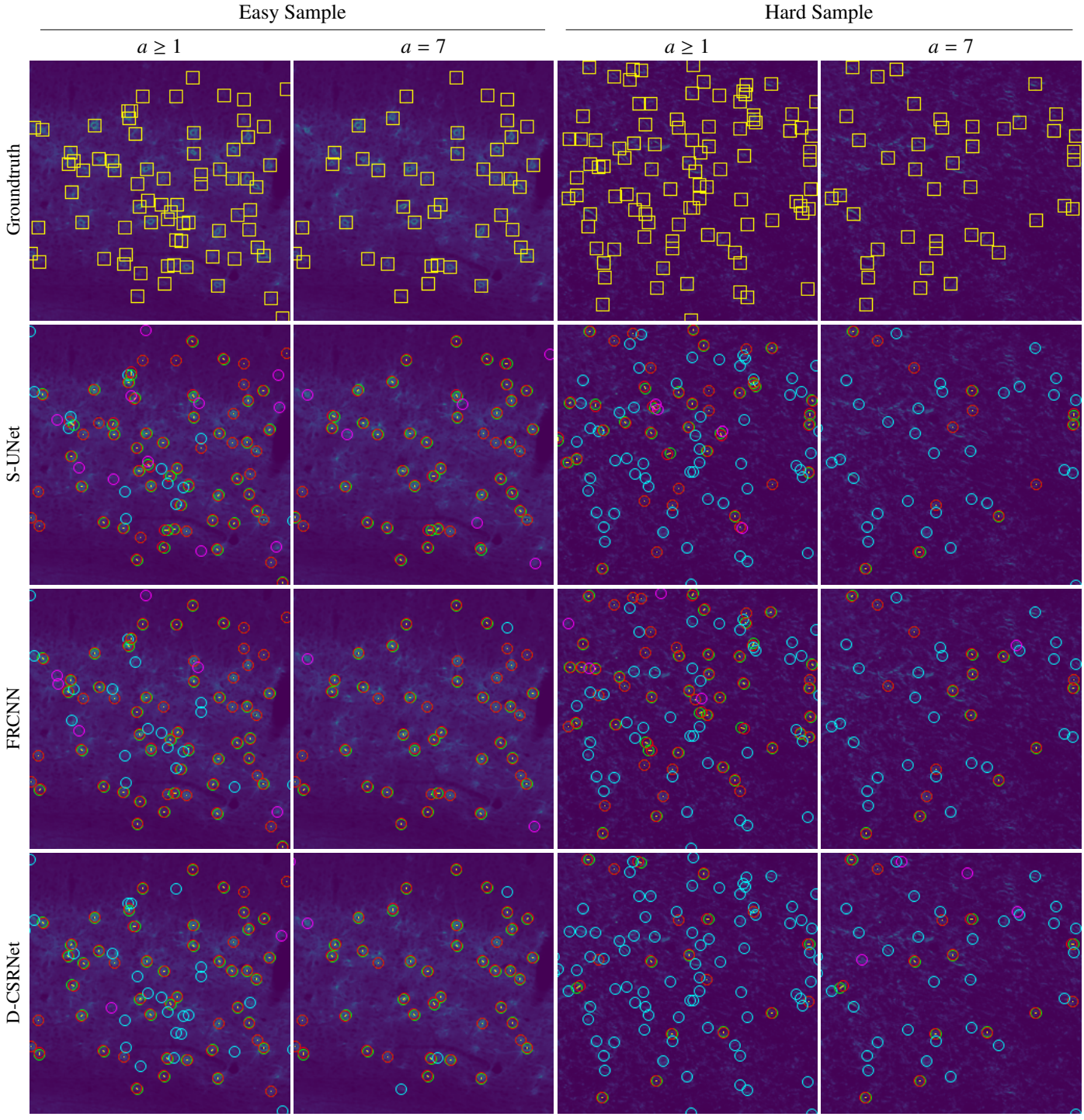


Fig. 4: **PNN-MR: Examples of localization predictions of tested models.** We show portions of two samples; an easy one (first two columns) and a hard one (last two columns) with different ground truths defined by including only PNNs with a minimum raters' agreement a. In the first row, we highlight the ground truth in yellow squares. In the rest of the rows, we indicate false positives in purple, false negatives in cyan, and true positives in green, with the corresponding ground-truth position drawn in red and connected via a thin yellow line. Best viewed in electronic format.

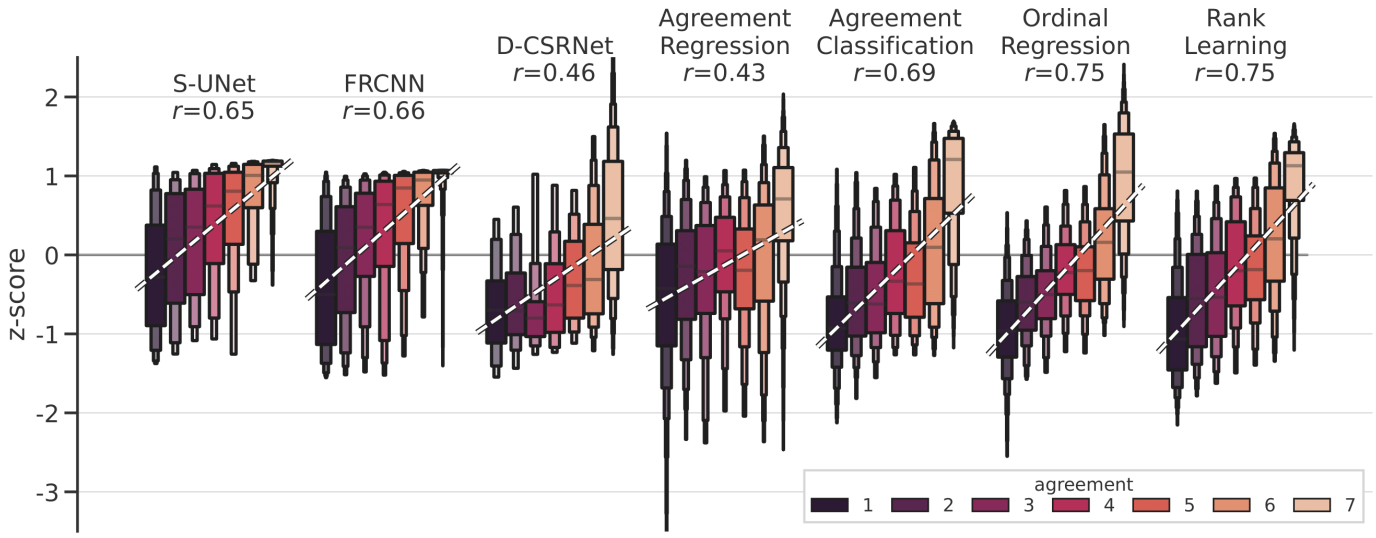


Fig. 5: **PNN-MR: Correlation between scores and raters' agreement.** We show the distribution of (z-normalized) scores per agreement level, the regression lines, and the Pearson's correlation coefficient r between scores and raters' agreement for each tested method.

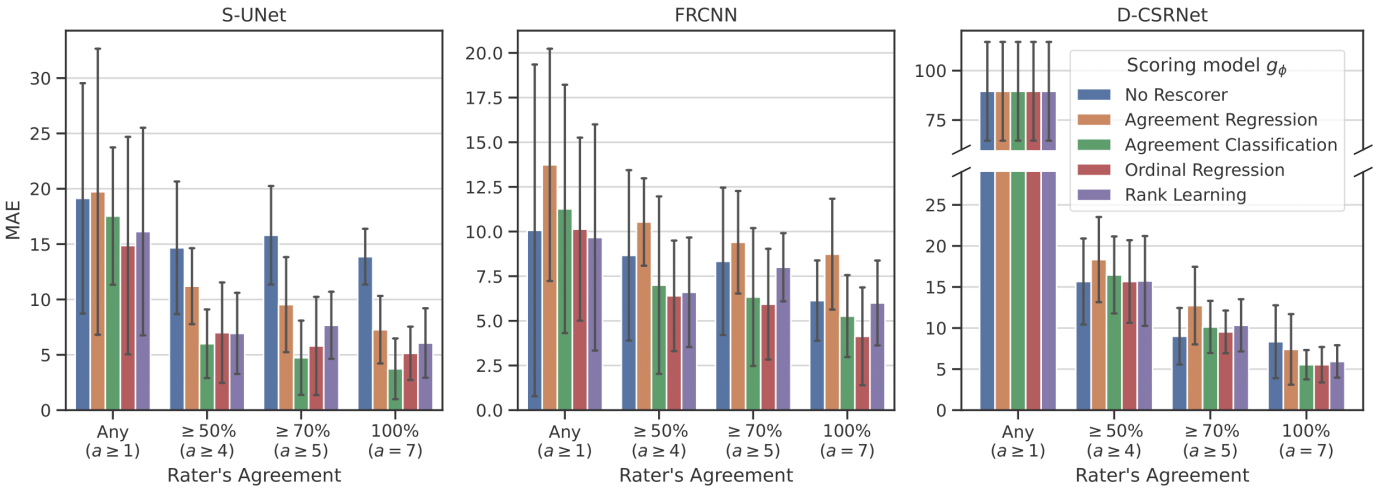


Fig. 6: **PNN-MR (Test Subset): Impact of the rescoring stage g_ϕ on counting performance in terms of MAE.** We show mean values ± standard deviation over five runs with randomized train/val/test splits of the PNN-MR subset. Numerical values in tabular format can be found in Table A.5 in Appendix A.

exploited large single-rater data to bootstrap state-of-the-art counting methodologies; we evaluated three CNN-based methods i.e., segmentation-based *S-UNet*, detection-based *FRCNN* and density-based approaches *D-CSRNet*. We showed that this step alone leads to sub-optimal results due to the underlying noisy nature of the employed single-rater data. Thus, we introduced a second rescoring stage that harnesses a small multi-rater subset and refines the previously computed predictions.

We performed an extensive experimental evaluation of our pipeline on a novel weakly-labeled dot-annotated dataset introduced on purpose, consisting of a collection of fluorescence microscopy images of mice brains containing biological structures. Results showed that rescoring strategies can improve the correlation between the scores and the raters' agreement. Using the proposed pipeline, we enhanced counting performance, in some cases significantly reducing the MAE. Whereas even simple rescoring methods such as Agreement Classification is beneficial, we deem the rank-based ones, like Agreement Ordinal Regression and Agreement Rank Learning, to be also data-efficient and robust to data unbalance, operating on pairs or tuples of samples. However, we leave a rigorous evaluation of those aspects to future work.

The proposed methodology still has some limitations. As future work, one direction could be to reduce computational costs

by using a unique model, still trained in two distinct stages, that could deliver the same counting performance while reducing the overall computation by sharing the parameters. Moreover, in the current two-stage solution, structures that are not localized in the first stage are excluded from the counting without the possibility of being filtered by score. We noted that this occurs on low-agreement structures that usually are not considered in the final count and thus do not affect performance significantly in practice. However, unifying the two stages in a unique model could help mitigate this problem and improve the applicability of the proposed method. In light of our results, *FRCNN* and *S-UNet* are the most promising solutions to be extended in future work for integrating the rescoring stage.

Acknowledgments

This work was partially funded by: AI4Media - A European Excellence Centre for Media, Society and Democracy (EC, H2020 n. 951911); Extension (ESA, n. 4000132621/20/NL/AF); AI-MAP (CNR4C program) - Tuscany POR FSE 2014-2020 (CUP B15J19001040004); LARSyS - FCT Project UIDB/50009/2020; funding from the Italian Ministry for university and research Grant MIUR-PRIN (2017HMH8FA); Orphan Disease Center University of Pennsylvania - (Grant MDBR-19-103-CDKL5).

The experiments were carried out in accordance with the directives of the European Community Council (2011/63/EU) and approved by the Italian Ministry of Health (Authorization # 621/2020-PR).

References

Aich, S., Stavness, I., 2018. Improving object counting with heatmap regulation. CoRR abs/1803.05494. arXiv:1803.05494.

Amato, G., Bolettieri, P., Moroni, D., Carrara, F., Ciampi, L., Pieri, G., Gennaro, C., Leone, G.R., Vairo, C., 2018. A wireless smart camera network for parking monitoring, in: 2018 IEEE Globecom Workshops (GC Wkshps), IEEE. pp. 1–6. doi:10.1109/glocomw.2018.8644226.

Amato, G., Ciampi, L., Falchi, F., Gennaro, C., 2019. Counting vehicles with deep learning in onboard UAV imagery, in: 2019 IEEE Symposium on Computers and Communications (ISCC), IEEE. pp. 1–6. doi:10.1109/iscc47284.2019.8969620.

Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A., 2016a. Detecting overlapping instances in microscopy images using extremal region trees. Medical Image Analysis 27, 3–16. doi:10.1016/j.media.2015.03.002.

Arteta, C., Lempitsky, V.S., Zisserman, A., 2016b. Counting in the wild, in: Computer Vision - ECCV 2016, Springer. pp. 483–498. doi:10.1007/978-3-319-46478-7_30.

Berretta, S., Pantazopoulos, H., Markota, M., Brown, C., Batzianouli, E.T., 2015. Losing the sugar coating: Potential impact of perineuronal net abnormalities on interneurons in schizophrenia. Schizophr. Res. 167, 18–27. doi:10.1016/j.schres.2014.12.040.

Boggio, E.M., Ehlert, E.M., Lupori, L., Moloney, E.B., Winter, F.D., Kooi, C.W.V., Baroncelli, L., Mecollari, V., Blits, B., Fawcett, J.W., Verhaagen, J., Pizzorusso, T., 2019. Inhibition of semaphorin3a promotes ocular dominance plasticity in the adult rat visual cortex. Mol. Neurobiol. 56, 5987–5997. doi:10.1007/s12035-019-1499-0.

Boominathan, L., Kruthiventi, S.S.S., Babu, R.V., 2016. Crowdnet: A deep convolutional network for dense crowd counting, in: Proceedings of the 24th ACM international conference on Multimedia, ACM. pp. 640–644. doi:10.1145/2964284.2967300.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent, in: Proceedings of the 22nd international conference on Machine learning - ICML '05, ACM Press. pp. 89–96. doi:10.1145/1102351.1102363.

Campagner, A., Ciucci, D., Svensson, C.M., Figge, M.T., Cabitza, F., 2021. Ground truthing from multi-rater labeling with three-way decision and possibility theory. Information Sciences 545, 771–790. doi:10.1016/j.ins.2020.09.049.

Ciampi, L., Carrara, F., Totaro, V., Mazziotti, R., Lupori, L., Santiago, C., Amato, G., Pizzorusso, T., Gennaro, C., 2021a. A Multi-Rater Benchmark for Perineuronal Nets Detection and Counting in Fluorescence Microscopy Images. URL: https://doi.org/10.5281/zenodo.5567032, doi:10.5281/zenodo.5567032.

Ciampi, L., Santiago, C., Costeira, J., Gennaro, C., Amato, G., 2021b. Domain adaptation for traffic density estimation, in: Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, SCITEPRESS - Science and Technology Publications. pp. 185–195. doi:10.5220/0010303401850195.

Ciampi, L., Santiago, C., Costeira, J.P., Gennaro, C., Amato, G., 2020. Unsupervised vehicle counting via multiple camera domain adaptation, in: Proceedings of the First International Workshop on New Foundations for Human-Centered AI (NeHuAI) co-located with 24th European Conference on Artificial Intelligence (ECAI 2020), Santiago de Compostella, Spain, September 4, 2020, CEUR-WS.org. pp. 82–85.

Cohen, J.P., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y., 2017. Count-ception: Counting by fully convolutional redundant counting, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), IEEE. pp. 18–26. doi:10.1109/iccvw.2017.9.

Dawid, A.P., Skene, A.M., 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. Journal of the Royal Statistical Society: Series C (Applied Statistics) 28, 20–28.

Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J., Heng, P.A., 2017. 3d deeply supervised network for automated segmentation of volumetric medical images. Medical Image Analysis 41, 40–54. doi:10.1016/j.media.2017.05.001.

Falk, T., Mai, D., Bensch, R., Özgün Çiçek, Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Bosco, C.D., Walsh, S., Saltukoglu, D., Tay, T.L., Prinz, M., Palme, K., Simons, M., Diester, I., Brox, T., Ronneberger, O., 2018. U-net: deep learning for cell counting, detection, and morphometry. Nat. Methods 16, 67–70. doi:10.1038/s41592-018-0261-2.

Fawcett, J.W., Oohashi, T., Pizzorusso, T., 2019. The roles of perineuronal nets and the perinodal extracellular matrix in neuronal function. Nat. Rev. Neurosci. 20, 451–465. doi:10.1038/s41583-019-0196-3.

Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S., Oñoro-Rubio, D., 2015. Extremely overlapping vehicle counting, in: Pattern Recognit. Image Anal.. Springer International Publishing, pp. 423–431. doi:10.1007/978-3-319-19390-8_48.

Guo, Y., Krupa, O., Stein, J., Wu, G., Krishnamurthy, A., 2021. SAU-net: A unified network for cell counting in 2d and 3d microscopy images. IEEE/ACM Trans. Comput. Biol. Bioinform. , 1–1doi:10.1109/tcbb.2021.3089608.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2020. Mask r-cnn. IEEE Trans. Pattern Anal. Mach. Intell. 42, 386–397. doi:10.1109/TPAMI.2018.2844175.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 770–778. doi:10.1109/cvpr.2016.90.

He, S., Minn, K.T., Solnica-Krezel, L., Anastasio, M.A., Li, H., 2021. Deeply-

supervised density regression for automatic cell counting in microscopy images. Medical Image Analysis 68, 101892. doi:10.1016/j.media.2020.101892.

Huang, Z., Ding, Y., Song, G., Wang, L., Geng, R., He, H., Du, S., Liu, X., Tian, Y., Liang, Y., Zhou, S.K., Chen, J., 2020. BCData: A large-scale dataset and benchmark for cell detection and counting, in: Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. Springer International Publishing, pp. 289–298. doi:10.1007/978-3-030-59722-1_28.

Jiang, N., Yu, F., 2020a. A cell counting framework based on random forest and density map. Appl. Sci. 10, 8346. doi:10.3390/app10238346.

Jiang, N., Yu, F., 2020b. A foreground mask network for cell counting, in: 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), IEEE. doi:10.1109/icivc50857.2020.9177433.

Jiang, N., Yu, F., 2021. A two-path network for cell counting. IEEE Access 9, 70806–70815. doi:10.1109/access.2021.3078481.

Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V., 2015. You should use regression to detect cells, in: Lecture Notes in Computer Science. Springer International Publishing, pp. 276–283. doi:10.1007/978-3-319-24574-4_33.

Koyuncu, C.F., Gunesli, G.N., Cetin-Atalay, R., Gunduz-Demir, C., 2020. DeepDistance: A multi-task deep regression model for cell detection in inverted microscopy images. Medical Image Analysis 63, 101720. doi:10.1016/j.media.2020.101720.

Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M., 2018. Where are the blobs: Counting by localization with point supervision, in: Computer Vision – ECCV 2018. Springer International Publishing. volume 11206, pp. 560–576. doi:10.1007/978-3-030-01216-8_34.

Lempitsky, V.S., Zisserman, A., 2010. Learning to count objects in images, in: Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, Curran Associates, Inc.. pp. 1324–1332.

Li, Y., Zhang, X., Chen, D., 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1091–1100. doi:10.1109/cvpr.2018.00120.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical Image Analysis 42, 60–88. doi:10.1016/j.media.2017.07.005.

Lu, E., Xie, W., Zisserman, A., 2019. Class-agnostic counting, in: Computer Vision – ACCV 2018. Springer International Publishing, pp. 669–684. doi:10.1007/978-3-030-20893-6_42.

Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O'Connor, N.E., 2018. People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE. pp. 8070–8079. doi:10.1109/cvpr.2018.00842.

Mirikharaji, Z., Abhishek, K., Izadi, S., Hamarneh, G., 2021. D-LEMA: Deep learning ensembles from multiple annotations - application to skin lesion segmentation, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE. pp. 1837–1846. doi:10.1109/cvprw53098.2021.00203.

Mueller, A.L., Davis, A., Sovich, S., Carlson, S.S., Robinson, F.R., 2016. Distribution of n-acetylgalactosamine-positive perineuronal nets in the macaque brain: Anatomy and implications. Neural Plast. 2016, 1–19. doi:10.1155/2016/6021428.

Napoli, D., Lupori, L., Mazziotti, R., Sagona, G., Bagnoli, S., Samad, M., Sacramento, E.K., Kirkpartick, J., Putignano, E., Chen, S., Tozzini, E.T., Tognini, P., Baldi, P., Kwok, J.C., Cellerino, A., Pizzorusso, T., 2020. MiR-29 coordinates age-dependent plasticity brakes in the adult visual cortex. EMBO reports 22. doi:10.15252/embr.202052108.

Oñoro-Rubio, D., López-Sastre, R.J., 2016. Towards perspective-free object counting with deep learning, in: Computer Vision – ECCV 2016. Springer International Publishing. volume 9911, pp. 615–629. doi:10.1007/978-3-319-46478-7_38.

Paulauskaite-Taraseviciene, A., Sutiene, K., Valotka, J., Raudonis, V., Iesmantas, T., 2019. Deep learning-based detection of overlapping cells, in: Proceedings of the 2019 3rd International Conference on Advances in Artificial Intelligence, ACM. pp. 217–220. doi:10.1145/3369114.3369120.

Pedregosa, F., Bach, F.R., Gramfort, A., 2017. On the consistency of ordinal regression methods. J. Mach. Learn. Res. 18, 55:1–55:35.

Pizzorusso, T., Medini, P., Berardi, N., Chierzi, S., Fawcett, J.W., Maffei, L., 2002. Reactivation of ocular dominance plasticity in the adult visual cortex. Science 298, 1248–1251. doi:10.1126/science.1072699.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster r-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149. doi:10.1109/tpami.2016.2577031.

Rodrigues, F., Pereira, F., Ribeiro, B., 2013. Learning from multiple annotators: Distinguishing good from random labelers. Pattern Recognit. Lett. 34, 1428–1436. doi:10.1016/j.patrec.2013.05.012.

Rogers, S.L., Rankin-Gee, E., Risbud, R.M., Porter, B.E., Marsh, E.D., 2018. Normal development of the perineuronal net in humans; in patients with and without epilepsy. Neuroscience 384, 350–360. doi:10.1016/j.neuroscience.2018.05.039.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, Springer. pp. 234–241. doi:10.1007/978-3-319-24574-4_28.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: 3rd International Conference on Learning Representations, ICLR 2015.

Sirinukunwattana, K., Raza, S.E.A., Tsang, Y.W., Snead, D.R.J., Cree, I.A., Rajpoot, N.M., 2016. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Transactions on Medical Imaging 35, 1196–1206. doi:10.1109/tmi.2016.2525803.

Slaker, M.L., Harkness, J.H., Sorg, B.A., 2016. A standardized and automated method of perineuronal net analysis using wisteria floribunda agglutinin staining intensity. IBRO Reports 1, 54–60. doi:10.1016/j.ibror.2016.10.001.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 1–9. doi:10.1109/cvpr.2015.7298594.

Tofighi, M., Guo, T., Vanamala, J.K.P., Monga, V., 2019. Prior information guided regularized deep learning for cell nucleus detection. IEEE Trans. Medical Imaging 38, 2047–2058. doi:10.1109/tmi.2019.2895318.

Venkatalakshmi, B., Thilagavathi, K., 2013. Automatic red blood cell counting using hough transform, in: 2013 IEEE Conference on Information and Communication Technologies, IEEE. pp. 267–271. doi:10.1109/cict.2013.6558103.

Wei, J., Suriawinata, A., Ren, B., Liu, X., Lisovsky, M., Vaickus, L., Brown, C., Baker, M., Nasir-Moin, M., Tomita, N., Torresani, L., Wei, J., Hassanpour, S., 2021. Learn like a pathologist: Curriculum learning by annotator agreement for histopathology image classification, in: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE. pp. 2472–2482. doi:10.1109/wacv48630.2021.00252.

Xie, W., Noble, J.A., Zisserman, A., 2016. Microscopy cell counting and detection with fully convolutional regression networks. Comput. methods Biomech. Biomed. Eng. Imaging Vis. 6, 283–292. doi:10.1080/21681163.2016.1149104.

Xie, Y., Xing, F., Shi, X., Kong, X., Su, H., Yang, L., 2018. Efficient and robust cell detection: A structured regression approach. Medical Image Analysis 44, 245–254.

Yu, F., Koltun, V., 2016. Multi-scale context aggregation by dilated convolutions, in: 4th International Conference on Learning Representations, ICLR 2016.

Zhang, P., Obradovic, Z., 2012. Integration of multiple annotators by aggregating experts and filtering novices, in: 2012 IEEE International Conference on Bioinformatics and Biomedicine, IEEE. pp. 1–6. doi:10.1109/bibm.2012.6392657.

Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y., 2016. Single-image crowd counting via multi-column convolutional neural network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. pp. 589–597. doi:10.1109/cvpr.2016.70.

Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R., 2017. Truth inference in crowdsourcing: Is the problem solved? Proceedings of the VLDB Endowment 10, 541–552.

Appendix A. Additional Results

In this section, we provide additional results obtained in our experimental evaluation.

First, we consider the experiments to validate the localization stage of our counting pipeline, i.e., considering the large single-rater weakly-labeled PNN-SR data to train the models and the multi-rater subset PNN-MR for their evaluation. In addition to MAE, we also report here the *Mean Absolute Relative Error* (MARE) and the *Grid Average Mean absolute Error* (GAME) (Guerrero-Gómez-Olmedo et al., 2015) as counting metrics. The MARE provides a percentage relating the absolute counting error to the number of objects to be counted. Following the notation already exploited for the MAE, it is defined as

$$\text{MARE} = \frac{1}{N}\sum_{n=1}^{N}\frac{\left|c_{\text{gt}}^{n} - c_{\text{pred}}^{n}\right|}{c_{\text{gt}}^{n}}. \tag{A.1}$$

MAE and MARE are fair metrics for comparing counting performance, but they do not capture localization errors; models might achieve low values on these metrics while providing wrong predictions (e.g., a high numbers of false positive and false negatives in detection-based methods, or a bad allocation of density values in predicted maps of density-based approaches). The GAME metric accounts for localization errors, as it is computed by sub-dividing the image in 4^L non-overlapping regions and computing the MAE in each of these sub-regions

$$\text{GAME}(L) = \frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{4^L}\left|c_{\text{gt}}^{n,l} - c_{\text{pred}}^{n,l}\right|, \tag{A.2}$$

where N is the number of test images, $c_{\text{pred}}^{n,l}$ is the estimated count in the l-th region of the n-th image, and $c_{\text{gt}}^{n,l}$ is the respective ground truth count. The higher the value of L, the more restrictive the GAME metric will be. Note that the MAE can be obtained as a particular case of the GAME when $L = 0$.

In the firs three rows of Fig. A.7, we show the results in terms of MARE, GAME(3), and F1-score obtained by the three localization models, i.e., *S-UNet, FRCNN, D-CSRNet*, on the entire multi-rater PNN-MR subset when varying the patch size and the threshold on the scalar score s. As already seen in the results concerning the MAE, patch size does not significantly influence the performance of *FRCNN* and *D-CSRNet*, hinting at the use of bigger patch sizes for these models to reduce processing overhead. On the other hand, *S-UNet* is more susceptible to this aspect, and different patch sizes induce different score distributions that respond differently to score thresholding. In general, for this latter model, we obtain better performance exploiting small patch sizes. Again, as already shown for the MAE, also using these three metrics, the density-based solution *D-CSRNet* achieves the worst overall performance when trained with weakly-labeled training data, and score thresholding is not effective. Finally, in the last row of Fig. A.7, we show Precision-Recall (PR) curves, with the goal to better highlight the influence of the considered threshold. Precision is computed as the percentage of model predictions that correspond to a ground-truth object, whereas Recall is the percentage of ground-truth objects identified by the model. Predictions and ground-truth objects are matched using the Hungarian algorithm based on the 2D Euclidean distance between pixel coordinates; we assign an infinite cost to pairs separated by a distance greater than 1.25 times the typical radius of objects. As depicted in the figure, the PR curves concerning the *FRCNN* and the *D-CSRNet* are monotonic, while the *S-UNet* does not have this property, again due to artifacts in the segmentation maps and on merging/separating components.

In Table A.4, we report the results obtained on the four sets of ground truth labels for PNN-MR comprising PNNs labeled by at least 1, 4, 5, and 7 out of 7 raters, respectively, by the three tested models in their most effective combination of patch size and the threshold value. Here we show the results in terms of MARE, GAME(3), and F1-score. As already inferred from the results regarding the MAE, also these metrics highlight the limitations of *D-CSRNet* to achieve a high recall on low-agreement samples. On the other hand, again, *FRCNN* is in general able to reach the best overall performance on all sets.

Regarding the experiments performed in the score calibration stage, we report additional results in Fig. A.8, showing the predictions ordered by score. In particular, the firsts three plots show the uncalibrated scores obtained using the localization models without the refinement provided by our proposed second calibration stage, while the remaining plots concern the scores obtained exploiting the calibration tested methodologies.

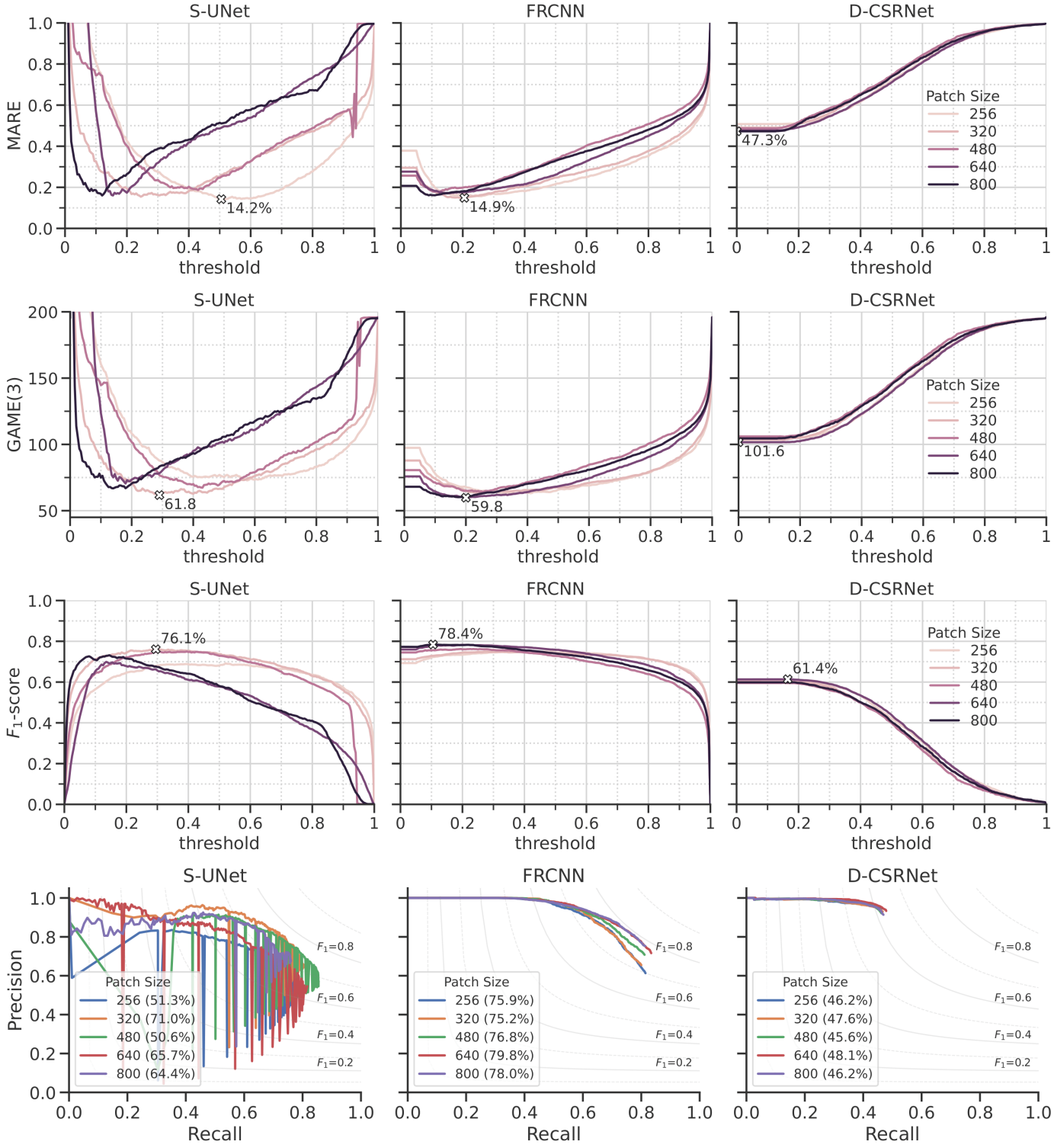


Fig. A.7: **PNN-MR: Impact of patch size and threshold to counting and localization performance** on standard segmentation-based (S-UNet), detection-based (FRCNN), and density-based (D-CSRNet) approaches (without scoring stage). Additional results in terms of MARE, GAME(3), F1-score and Precision-Recall.

Table A.4: **PNN-MR: Performance on different agreement levels.** We show the counting (MARE and GAME(3)) and detection (F_1-score) performance of all tested models. Models are trained on the weakly-labeled PNN-SR dataset. We define four sets of ground-truth labels for PNN-MR composed of objects labeled by any number of raters (Any), at least 50%, at least 70%, or all raters, respectively, to simulate different counting policies, from liberal to conservative ones.

	Raters' Agreement			
	Any ($a \geq 1$)	$\geq 50\%$ ($a \geq 4$)	$\geq 70\%$ ($a \geq 5$)	100% ($a = 7$)
MARE (%)				
S-UNet	14.2 ± 11.3	12.6 ± 12.9	14.5 ± 9.6	19.0 ± 13.3
FRCNN	14.9 ± 11.0	13.6 ± 10.9	12.7 ± 12.5	11.8 ± 13.6
D-CSRNet	47.8 ± 20.3	18.9 ± 18.0	14.3 ± 14.8	15.7 ± 11.7
GAME(3)				
S-UNet	61.8 ± 18.7	36.8 ± 15.2	34.9 ± 15.4	29.0 ± 13.3
FRCNN	60.2 ± 12.9	31.7 ± 13.5	28.5 ± 12.6	23.2 ± 7.8
D-CSRNet	99.3 ± 38.5	41.2 ± 18.4	35.0 ± 14.8	28.1 ± 11.8
F_1-score (%)				
S-UNet	76.1 ± 9.3	79.1 ± 10.4	78.7 ± 9.2	75.5 ± 8.4
FRCNN	78.4 ± 5.5	82.3 ± 6.6	82.5 ± 8.5	80.6 ± 9.5
D-CSRNet	61.7 ± 14.6	73.8 ± 12.2	74.7 ± 11.5	72.1 ± 14.7

As already demonstrated, the scores obtained by Agreement Ordinal Regression (OR) and Agreement Rank Learning (RL) strategies correlate better with the raters' agreement, thus confirming that their use is suggested.

Appendix B. Training Details

Table B.6 reports the average number of objects per agreement level in the PNN-MR train/validation/test splits.

We report in Tables B.7, B.8, and B.9 the hyperparameters adopted to train the three considered counting architectures, i.e., *S-UNet*, *FRCNN*, and *D-CSRNet*, exploited in the experiments described in Section 5.1 and Section 5.2. We adopt early stopping for each configuration. In particular, we select the snapshot achieving the lowest GAME(3) value on the validation set. Additionally, we report in Table B.10 the hyperparameters used for our re-implementation of FCRN-A (Xie et al., 2016), exploited in Section 5.1 for an exact head-to-head comparison against the three adopted counting architectures on standard single-rater counting benchmarks. In this case, we pick up the snapshot achieving the lowest MAE value on the validation set.

For the scoring stage, we report in Table B.11 the training hyper-parameters concerning the scoring model g_ϕ for each adopted learning methodology, i.e., AR, AC, OR, and RL.

Table A.5: **PNN-MR (Test Subset): Impact of the rescoring stage g_ϕ on counting performance in terms of MAE.** We report mean and standard deviation over five runs with randomized train/val/test splits of the PNN-MR subset. AR = Agreement Regression. AC = Agreement Classification. OR = Agreement Ordinal Regression. RL = Agreement Rank Learning. 'w/o' indicates results without the rescoring stage, i.e., filtering is performed on method-specific scores extracted in the first localization stage.

		Raters' Agreement			
	g_ϕ	Any ($a \geq 1$)	$\geq 50\%$ ($a \geq 4$)	$\geq 70\%$ ($a \geq 5$)	100% ($a = 7$)
S-UNet	w/o	19.13 ± 11.63	14.67 ± 6.70	15.80 ± 4.98	13.87 ± 2.81
	AR	19.73 ± 14.44	11.20 ± 3.83	9.53 ± 4.80	7.27 ± 3.41
	AC	17.53 ± 6.93	6.00 ± 3.46	4.73 ± 3.75	3.73 ± 3.07
	OR	14.87 ± 10.98	7.00 ± 5.07	5.80 ± 4.96	5.13 ± 2.69
	RL	16.13 ± 10.48	6.93 ± 4.10	7.67 ± 3.39	6.07 ± 3.51
FRCNN	w/o	10.07 ± 10.38	8.67 ± 5.33	8.33 ± 4.61	6.13 ± 2.51
	AR	13.73 ± 7.27	10.53 ± 2.73	9.40 ± 3.21	8.73 ± 3.47
	AC	11.27 ± 7.77	7.00 ± 5.55	6.33 ± 4.31	5.27 ± 2.56
	OR	10.13 ± 5.73	6.40 ± 3.46	5.93 ± 3.47	4.13 ± 3.06
	RL	9.67 ± 7.08	6.60 ± 3.43	8.00 ± 2.13	6.00 ± 2.66
D-CSRNet	w/o	89.53 ± 27.92	15.67 ± 5.85	9.00 ± 3.85	8.33 ± 4.96
	AR	89.53 ± 27.92	18.33 ± 5.80	12.73 ± 5.28	7.40 ± 4.80
	AC	89.53 ± 27.92	16.47 ± 5.24	10.13 ± 3.55	5.53 ± 1.99
	OR	89.53 ± 27.92	15.67 ± 5.62	9.53 ± 2.90	5.53 ± 2.41
	RL	89.53 ± 27.92	15.73 ± 6.12	10.33 ± 3.55	5.93 ± 2.20

Table B.6: **PNN-MR Splits Statistics**. Mean and standard deviation of number of objects in the five PNN-MR 70/15/15 random splits.

	Raters' Agreement			
	Any ($a \geq 1$)	$\geq 50\%$ ($a \geq 4$)	$\geq 70\%$ ($a \geq 5$)	100% ($a = 7$)
Total	2351	1384	1234	880
Train	1167 ± 70	678 ± 56	606 ± 50	428 ± 33
Validation	569 ± 60	351 ± 38	314 ± 33	232 ± 28
Test	615 ± 58	356 ± 27	315 ± 24	220 ± 25

Table B.7: **Training hyperparameters of D-CSRNet.** σ is the standard deviation of the Gaussian kernel over-imposed on each object location to obtain the target density map. Density values outside the image borders are reflected by the image limits.

	VGG	MBM	ADI	BCData	PNN
σ	5	10	5	15	15
Patch Overlap	-	-	-	-	120
Optimizer	Adam				
LR	10^{-5}				
Batch size	8	5	4	12	64
Epochs	1000	1000	1000	250	100
LR Steps	800/900	800/900	800/900	180/230	50/75
LR Step Factor	0.1				

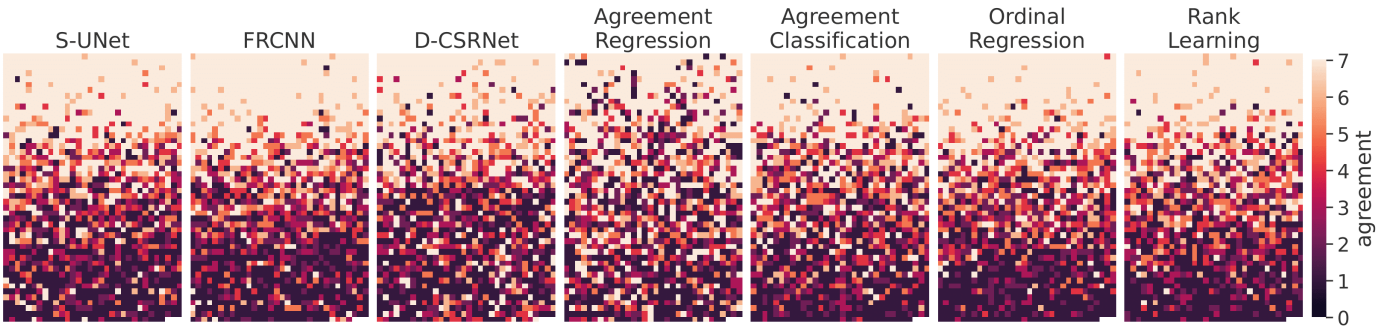


Fig. A.8: **PNN-MR: Predictions ordered by score** (left to right and top to bottom). The raters' agreement of each PNN is color-coded. The first three plots show the uncalibrated scores derived from the localization models. Whereas most methods correctly rank high-agrement samples, scoring methods rank better the low-agreement ones.

Table B.8: **Training hyperparameters of S-UNet**. r_{disk}: radius in px of an object. r_{ignore}: radius in px of the 'ignore' zones. l_{sep}: width in px of the background ridge separating two nearby objects. All other parameters are described in Falk et al. (2018) and control the loss weighting of pixels around foreground objects.

	VGG	MBM	ADI	BCData	PNN
r_{disk}	5	12	5	15	20
r_{ignore}	6	15	6	18	25
v_{bal}			0.1		
σ_{bal}	3	5	3	7	10
l_{sep}	1	1	1	2	1
σ_{sep}	3	4	3	8	6
λ_{sep}			50		
Patch Overlap	-	-	-	-	100
Optimizer			Adam		
LR			0.01		
Batch size	8	5	5	5	8
Epochs	500	1000	1000	50	100
LR Steps	200	750/900	750/900	30/45	50/75
LR Step Factor			0.1		

Table B.9: **Training hyperparameters of FRCNN**. Target bounding boxes are squares with given side length.

	VGG	MBM	ADI	BCData	PNN
Bouding Box Side	12	20	12	30	60
Max Detections			300		
NMS Threshold			0.6		
Patch Overlap	-	-	-	-	120
Optimizer			SGD		
LR			0.005		
Momentum			0.9		
Weight Decay			0.0005		
Batch size	8	4	4	8	64
Epochs	150	150	100	150	100
LR Steps	100	100	50/75	100	50/75
LR Step Factor			0.1		

Table B.10: **Training hyperparameters of FCRN-A. σ is the standard deviation of the Gaussian kernel over-imposed on each object location to obtain the target density map. Density values outside the image borders are reflected by the image limits.**

	VGG	MBM	ADI	BCData
σ	1	1	1	1
Optimizer		SGD		
LR	0.01	0.01	0.01	0.001
Momentum		0.9		
Weight Decay		10^{-5}		
Batch size	8	4	4	4
Epochs	150	300	150	150

Table B.11: **Training hyperparameters of Scoring ConvNet (g_ϕ)**. Parameters for each adopted learning methodology.

	AR	AC	OR	RL
Input Size		64×64		
Optimizer	SGD	SGD	Adam	SGD
LR	0.001	0.001	0.001	0.01
Momentum	0.9	0.9	-	0.9
Batch size		32		
Epochs	1000	200	1000	300
LR Steps	750	60	750	100
LR Step Factor		0.1		
#Tuples/Epoch	-	-	-	350