

# Supplementary information for "AGAR a microbial colony dataset for deep learning detection"

Sylvia Majchrowska<sup>1,2</sup>, Jarosław Pawłowski<sup>1,2</sup>, Grzegorz Guła<sup>1,3</sup>, Tomasz Bonus<sup>1</sup>, Agata Hanas<sup>1</sup>, Adam Loch<sup>1</sup>, Agnieszka Pawlak<sup>1</sup>, Justyna Roszkowiak<sup>3</sup>, Tomasz Golan<sup>1,\*</sup>, and Zuzanna Drulis-Kawa<sup>1,3,+</sup>

<sup>1</sup>NeuroSYS, Rybacka 7, 53-656 Wrocław, Poland

<sup>2</sup>Faculty of Fundamental Problems of Technology, Wrocław University of Science and Technology, Wybrzeże S. Wyspiańskiego 27, 50-372 Wrocław, Poland

<sup>3</sup>Department of Pathogen Biology and Immunology, Institute of Genetics and Microbiology, University of Wrocław, S. Przybyszewskiego 63, 51-148 Wrocław, Poland

\*t.golan@neurosys.com

+zuzanna.drulis-kawa@uw.edu.pl

## 1 The AGAR dataset in details

The AGAR is a color image dataset created for developing methods of microorganism colonies detection and counting. It contains 18 000 annotated photos of Petri dishes, which gave 336 442 annotated colonies in total. The images were taken under diverse lighting conditions with two cameras. More information about microbial colonies inoculation, and data acquisition setups can be found in Section 1.1, and Section 1.2. The image annotations procedure is described in detail in Section 1.3.

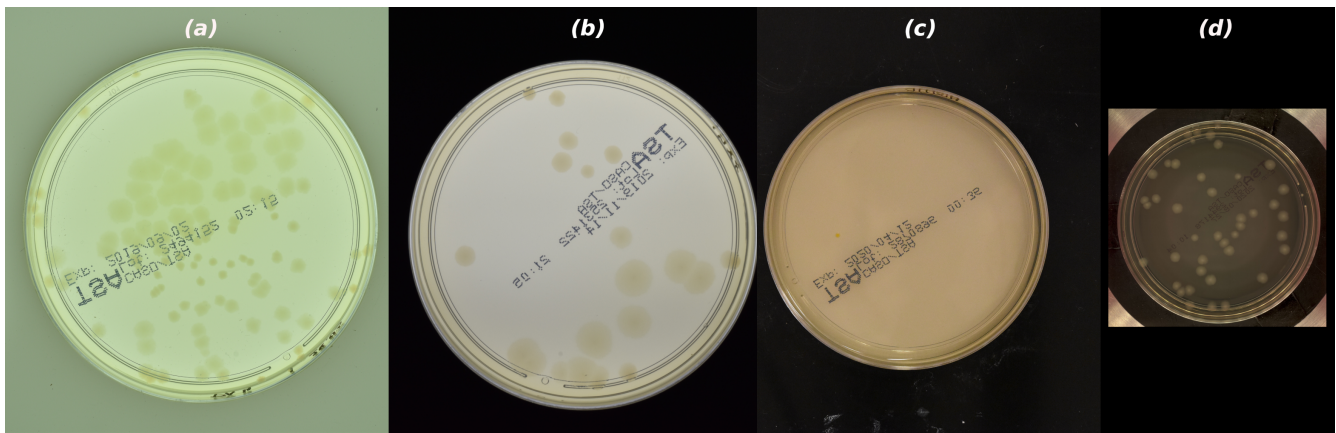
### 1.1 Microbial colonies growth conditions

Based on European, American, and Japanese Pharmacopoeia<sup>1-3</sup> regulations and ATCC guidelines<sup>4</sup>, *Staphylococcus aureus* subsp. *aureus* Rosenbach FDA 209 - ATCC 6538, *Bacillus subtilis* subsp. *spizizenii* NRS 231 - ATCC 6633, *Pseudomonas aeruginosa* R. Hugh 813 - ATCC 9027, *Escherichia coli* Crooks - ATCC 8739, and *Candida albicans* 3147 - ATCC 10231 strains were used for training and testing deep learning algorithms. Microorganisms have been purchased from American Type Culture Collection (LGC Standards, Poland) and were stored at -70°C in Trypticase Soy Broth (TSB, Becton Dickinson and Company, Cockeysville, MD, USA) supplemented with 20% glycerol (U.S. Merck Corporate Headquarters, Kenilworth, NJ, USA). For the experiment, strains were refreshed on Trypticase Soy Agar (TSA, Becton Dickinson and Company, Cockeysville, MD, USA) at 37°C for 18 h. A series of 10-fold dilutions were made in Ringer's fluid of the initial culture at a density equivalent to 0.5 McF (Densimat Densitometer, bioMérieux SA, France). From selected dilutions, 100 µl were cultured onto TSA medium in five technical replicates. Cultures were inoculated using sterile 6 mm diameter glass beads on a laboratory rocker (Multi Bio-3D Sunflower, Biogenet, Poland) for 40 s. The beads were then removed and the microbial cultures were incubated at 37°C for 18-24 hours.

### 1.2 Data acquisition

A simple image acquisition system consists of a Nikon D3500 camera with 60 mm lens and 24 Mpx CCD and a plexiglass stand for Petri dishes, with LED source light mounted below. Data collected with this setup is labeled as *higher-resolution* (4 000 x 6 000 px). In order to provide some diversity, small changes in the setup during the data acquisition process were introduced. Thus, three subgroups of *higher-resolution* photos are distinguished: *bright*, *dark*, and *vague* (see Fig. 1). The former two were made with the whole setup closed in a box to eliminate the influence of ambient light. The difference between them lies in a color of plexiglass used for a stand—white for *bright* subgroup, or black for *dark* one. The last subgroup, *vague*, was exposed to available light, which resulted in low contrast images, hard to annotate even by a professional microbiologist.

A different setup was used to collect the part of the dataset referred to as *lower-resolution* (2048 x 2048 px). The system includes a holder designed to mount a Petri dish on (visible on photos), a monochrome camera (IDS UI-5370CP-M-GL with 4.19 Mpx resolution), and a set of adjustable light sources below and above a holder (their parameters were changed during data acquisition process). RGB images are obtained as a composition of three monochromatic photos made with different optical filters, corresponding to red, green, and blue colors. Example photos of agar plate cultures taken under different conditions are presented in Fig. 1.



**Figure 1.** Example photos of agar plate cultures of AGAR dataset for different image acquisition setups: (a) *bright*, (b) *dark*, (c) *vague*, and (d) *lower-resolution*.

### 1.3 Image annotations

The design of a robust annotation pipeline was critical due to the desire to label 18 000 Petri dish photos, which led to over 330 000 labelled colonies. The image acquisition setup was equipped with a digital counter controlled by Raspberry Pi to display a sample identification (ID) number. This allowed us to verify if the appropriate photos were uploaded for the corresponding entry in the database. Please note that images are automatically cropped (based on their histograms) during the first stage of preprocessing to keep only a part with a Petri dish (and stored in this form in AGAR dataset), however, in some cases a part of the counter can be still visible.

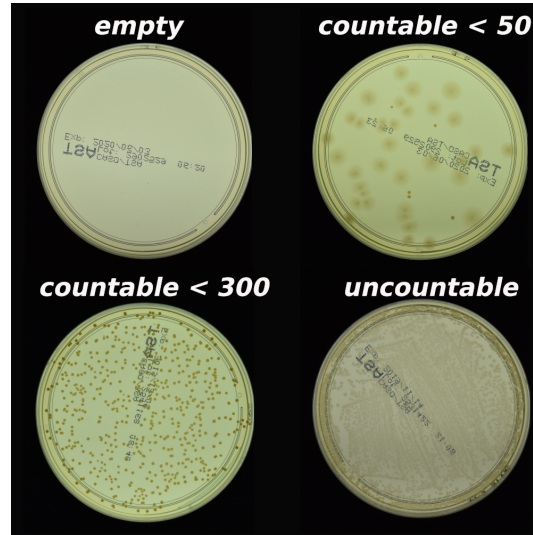
A web application was developed for microbiologists to upload and annotate agar plate culture photos. Each sample recorded in the database is marked as *countable*, *uncountable*, or *empty*. There are some indications<sup>5,6</sup> of the optimal number of colonies to perform manual counting. Based on them, samples with more than 300 colonies are treated as *uncountable*. Please note, however, that some samples may also be marked as *uncountable* if there were problems with clear identification of colonies (e.g. due to low contrast in the *lower-resolution* subset). Similarly, for samples with colonies number range between 50 and 300 instances, some colonies agglomerated, and it was difficult to identify their boundaries (see Fig. 2). In case of samples identified as *countable*, microbiologists annotated every colony with its location (by drawing a bounding box) and class. There are 7 possible classes: 5 microorganisms (*S. aureus*, *B. subtilis*, *P. aeruginosa*, *E. coli*, *C. albicans*), defects (crack marks on an agar surface), and contamination (microbial contamination like environmental microorganisms and sometimes fungi).

The annotations are stored in JSON files provided for every sample, with the same basename as the corresponding image file. They all share the same data structure presented in Listing 1. Each file contains a series of fields, including: the background category name (*bright*, *dark*, *vague*, or *lower-resolution*), the list of names of microbe species that were inoculated on a Petri dish (note that sometimes they have not grown), the total number of colonies (excluding defects and contamination), the list of bounding boxes, and the sample ID. Provided bounding boxes give information about pixel coordinates of the top-left corner, its width and height, microbe class, and annotation ID. Please note that in the case of *uncountable* samples *colonies\_number* key gets -1 value. The file structure is presented in the Listing 1.

```
{
  "background": str,
  "classes": [str],
  "colonies_number": int,
  "labels": [
    {
      "id": int,
      "class": str,
      "height": int,
      "width": int,
      "x": int,
      "y": int
    }, ...
  ]
  "sample_id": int }
```

**Listing 1.** The data structure of a single annotation JSON file.

For convenience, annotations are also provided in the well known COCO format<sup>7</sup>, with an additional key describing background category.



**Figure 2.** Examples of images of agar plate cultures with different number of colonies.

#### 1.4 Statistical data analysis

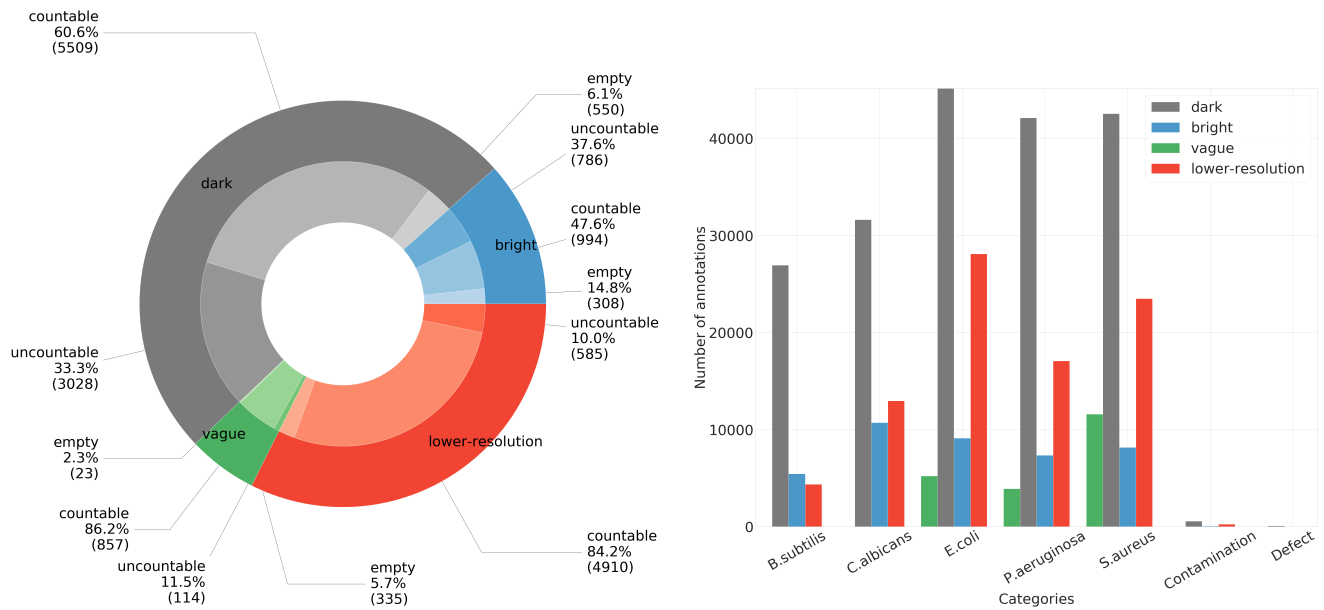
The summary of AGAR dataset statistics showing the distribution of images over different background categories is presented on a pie chart in Fig. 3(left). Each color represents different background category, while different shades indicate sample classes (*empty*, *countable*, or *uncountable*). The samples from *dark* subgroup constitute about half of the whole dataset and they include photos with the best quality in terms of contrast between grown microorganisms and an agar surface. At the very beginning of the data collection process, the full range of dilutions was used to collect the variety of samples, as in the case of *higher-resolution* subgroup. However, then a low dilution level was used less often to reduce the number of *uncountable* plates. That is why the *lower-resolution* group, acquired later, includes much less of them than the other subgroups.

In general the AGAR dataset achieves a good balance in the number of instances of different microbe species, which is important for building a robust deep learning model. However, given subgroups may be partially unbalanced as presented in Fig. 3(right). The *vague* subgroup does not include *B. subtilis* and *C. albicans* at all, but it has more than 80% of photos with two different microbe species, in contrast to 15% for *higher-resolution* (excluding *vague*) and 40% for *lower-resolution*. It is also not as numerous as other subgroups, because images from *vague* subset are difficult to annotate. *B. subtilis* is also underrepresented in the *lower-resolution* subset, though, it is still enough to train a detector to recognize this species.

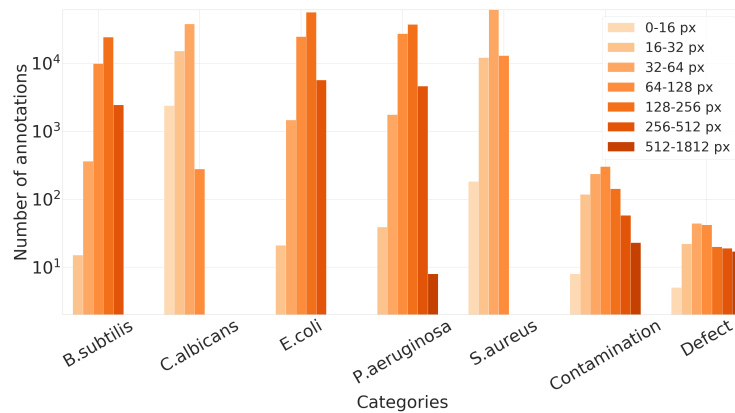
Fig. 4 shows the size variability of annotations per category for the whole dataset. Two basic box size distributions can be distinguished: 0-128 px for *C. albicans* and *S. aureus*, and 16-512 px for *P. aeruginosa*, *B. subtilis*, and *E. coli*. In total (excluding defects and contamination), there are 154 630 bounding boxes with squared root of area below 128 px, 180 173 within 128-512 px, and 1 639 above 512 px. This wide range of sizes makes the detection task more challenging because models have to be flexible enough to handle the variety of instances' dimensions.

Another interesting aspect is the distribution of growing colonies on the plate, as it is presented in Fig. 5 (only for *higher-resolution* background category, excluding *vague* samples). *C. albicans* and *S. aureus* form smaller colonies, which are more spatially separated on a Petri dish. The rest of considered species mainly cover a middle ring, which can be related to the method of microbes inoculation on an agar medium.

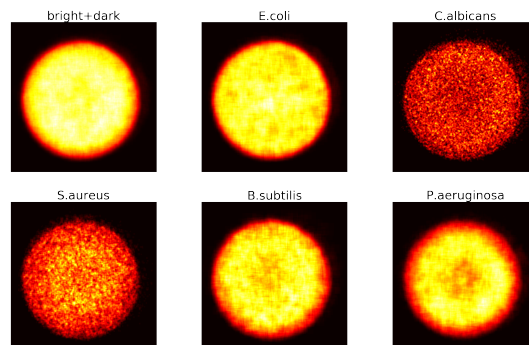
As mentioned before, the *countable* class includes samples with less than 300 colonies. The histograms shown in Fig. 6 present the distribution of *countable* samples in terms of the number of annotated colonies. The histograms of number of annotated colonies per image divided into microbial species are also included. In general, the interquartile range for the whole dataset is between 4 and 38 colonies per image. In the case of *B. subtilis*, higher number of instances can be observed outside this range. Also, the distribution of *vague* part of dataset shows slightly different behaviour where median is a bit higher.



**Figure 3.** (left) The distribution of samples over different subgroups. (right) The number of instances per category for all 5 microbe types and additional two classes – defects and contaminations.

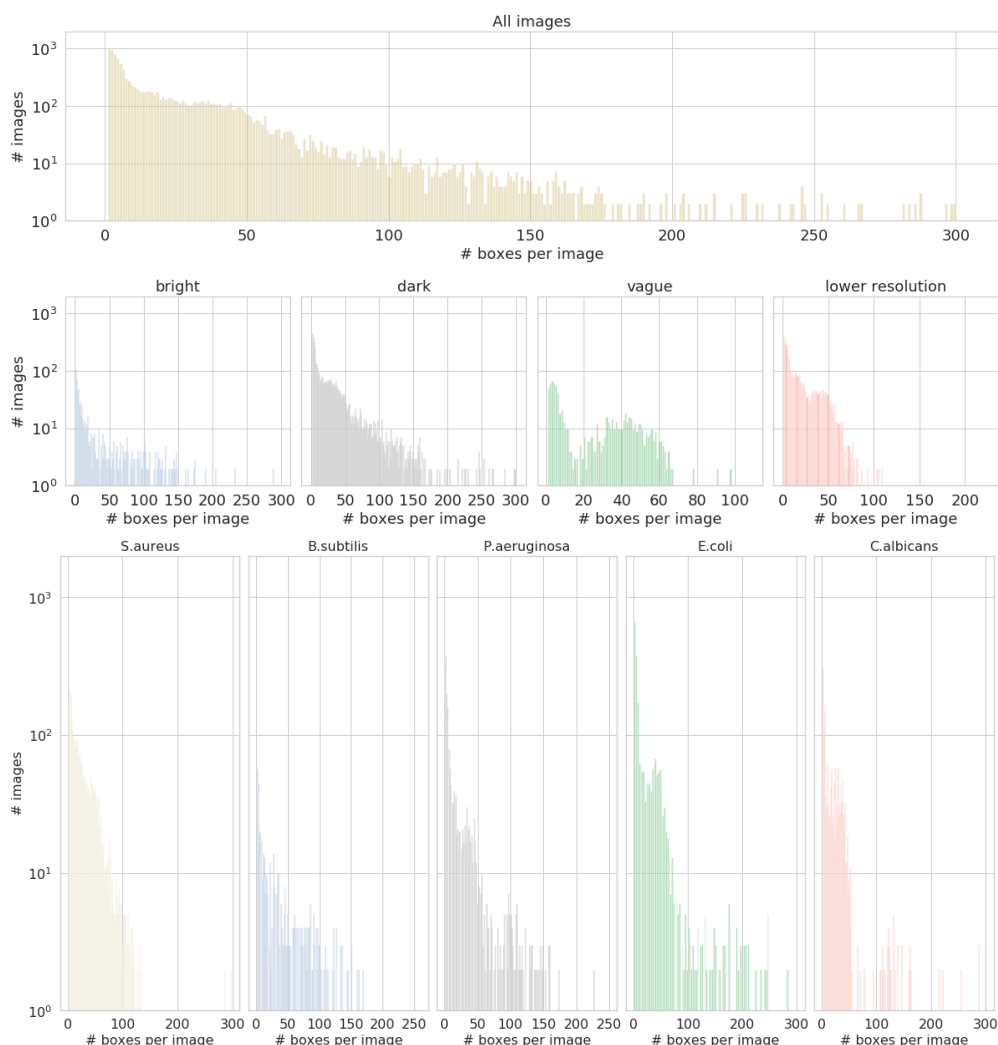


**Figure 4.** The number of instances per category for all annotated instances grouped by square root of bounding box area.



**Figure 5.** Heatmaps of all annotations at one picture for *higher-resolution* subgroup. Brighter colors indicates greater crowd in a given place in the picture. All results was normalized using maximum of a given subset (for each microbial species).





**Figure 6.** Histograms of the number of annotated instances per image, calculated for: the whole dataset, different subsets, and different microbial species.

## 2 Data processing

AGAR images are very large in size compared to regular datasets like COCO<sup>7</sup>. The original image resolution in the dataset ranges from about  $2048 \times 2048$  to about  $4000 \times 6000$  pixels, while most images in common datasets (e.g. COCO) are no more than  $1000 \times 1000$  pixels. The annotations prepared by microbiologists were made on resized images, so, during pre-processing, the pixel positions of bounding boxes have to be recalculated. Also, during pre-processing, the visible parts of electronic counter (which helped us to distinguish the samples from each other during the photo collection process) were cut out from the images. Nevertheless, such big images cannot in raw state go into neural network, thus pre-processing to split images into smaller parts should be applied first. Our pre-processing methods are described in Subsection 2.1. Post-processing methods that allowed us to count colonies from entire photos are presented in the Subsection 2.2.

### 2.1 Data pre-processing

In order to ensure that the training and test data distributions approximately match, 3/4 of the original images were randomly selected as the training sets and the rest were kept as validation sets for each background category separately. Please note that the same image subset is used for validation and test sets, however patches for the both sets are generated in different manner, as explained below. The list of image IDs of all original images used for training and validation set is publicly provided, but exact prepared patches are not.

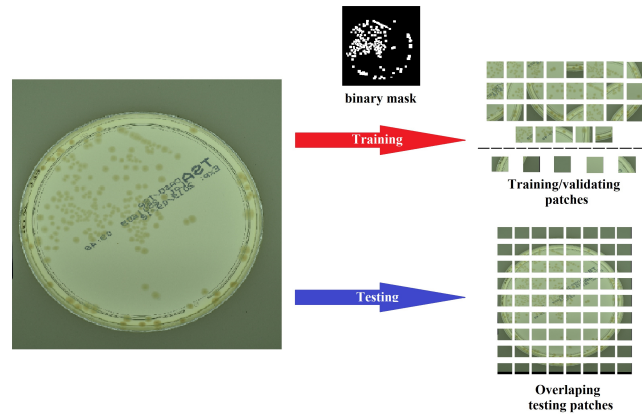
To control the balance between empty patches and the ones containing microbes, first only training patches with at least one colony are generated and then the appropriate number of empty ones (few percent of number of generated patches with

colonies) are added. The algorithm to divide images consists of a few subsequent steps. Firstly, a binary mask of all area of bounding boxes present on given image (white pixels make up the colony instances, black—the background) is generated. In the following steps, one box is chosen randomly, and a patch of size 512x512 containing the whole box (the left-top corner selection is also random) is cut out, and all other boxes (if any) on the patch are also marked as chosen. The above steps are repeated until every box at the image is marked. This procedure is illustrated as the red path in Fig 7.

On the other hand, extracting test data patches is done without exploiting annotations. A sliding window with an overlap being equal to 1/8 of the width of the patch is used to get adjacent images parts (blue path in Fig 7). The overlapping is necessary to get the final prediction for the whole image, and not to miss any colonies which accidentally lie in a division area. Removing redundant predictions is done in the post-processing of output given by a neural network.

## 2.2 Data post-processing

The post-processing consists of gathering together predicted bounding boxes on every patch belonging to one image and recalculating their exact position (rescale and use an appropriate offset taking into account given patch position relative to the whole image). After that, excessive boxes are filtered out using: (1) the probability of the detected microbe class and (2) the variation of the Soft Non-Maximum Suppression (NMS) algorithm<sup>8</sup>. Our modified version of soft NMS based not on the biggest bounding box confidence level, but on the biggest area of prediction. The thresholds for both filters are tuned on training data (using simple grid search for each model separately) in a way to minimize metrics for microbe counting, defined in Section 3.



**Figure 7.** Example random patches for training/validation pipeline and overlapping testing patches. A dashed line separates empty patches (containing no colonies) from those that are occupied for training/validating mode.

## 3 Models evaluation metrics

Two stages of the task are considered: detection – localization and classification (Section 3.1), and then counting the colonies of microorganisms (Section 3.2). As an evaluation metric for colonies detection, we rely on the mean Average Precision (mAP) established for the COCO competition<sup>7</sup>. The Average Precision (AP) is the precision integrated over the whole recall range for precision-recall (PR) curve at given Intersection over Union (IoU) threshold. IoU describes the level of overlapping between truth and predicted bounding box. The mean value of AP for different IoU thresholds and/or classes gives mAP. To measure the effectiveness of colony counting we provide three separate metrics, namely Mean Absolute Error (MAE), its cumulative extension called cMAE, and Symmetric Mean Absolute Percentage Error (sMAPE) evaluated for different models on every data subset.

### 3.1 Detection metrics

Two neural network architectures for two-stage object detection are evaluated to recognize and localize microbial colonies. Their performance is mainly measured using AP@[0.50:0.95] (mAP over different IoU thresholds, from 0.5 to 0.95, step 0.05).

IoU measures how well predicted bounding boxes fit the real location of an object. It is defined as

$$\text{IoU} = \frac{\text{area of intersection}}{\text{area of union}}, \quad (1)$$

where the intersection and union refer to true and predicted bounding boxes. To decide if a bounding box proposed by a model is valid or not, one should choose appropriate threshold value. For the value of selected threshold we can distinguish:

- True Positive (TP) detection if IoU is greater than or equals the threshold value,
- False Positive (FP) detection if IoU is less than the threshold value,
- False Negative (FN) detection if a model fails to detect the proper object,
- and True Negative (TN) in case when model correctly did not predict any object.

With the above definitions, precision and recall are given by expressions

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}.\end{aligned}\tag{2}$$

Precision shows how accurate predictions are, while recall measures how good a model is at finding objects. The precision-recall curves used in this study show the trade-off between precision and recall. Usually the increase of one metric causes the decrease of the other. High recall but low precision reflects the increase of the number of predicted bounding boxes, but classified incorrectly in most cases. On the other hand, high precision but low recall is just the opposite situation, a model returns very few results, but most of its predicted labels are correct.

Finally, the Average Precision for a given microbe class is calculated by integrating the area under the precision-recall curve (using e.g. the trapezoidal rule). It is usually done for different IoU thresholds. The mAP score is the mean over all thresholds, typically AP@[0.50:0.95] is considered<sup>7</sup>, and/or all microbe classes.

### 3.2 Counting metrics

Counting by detection is supposed to be the most natural approach to get the number of objects on an image. Even if localizing colonies itself is not necessary for this task, it is very useful to investigate the achieved results (especially to understand the cases in which model makes mistakes). To evaluate counting task the MAE, cMAE, and sMAPE were used.

MAE measures an average error between paired observations: ground truth count of all annotations and the number of predicted bounding boxes, for the whole image of a Petri dish. As the metric do not include the error coming from the misclassification of microorganisms, cMAE being a sum of MAE calculated for every microbe separately is also provided (given in brackets in altogether row in Tables 4-5). For cMAE only properly recognized microbes are taken as a correct count. sMAPE is similar to general MAE, but it weights particular samples errors inversely with the number of colonies on a plate. The intuition behind this metric is as follows. In a real world scenario, humans tend to perceive counts in the logarithmic scale. That means, that a mistake of 1 for an *empty* dish might seem intolerable but the same mistake for a ground truth count of 50 might seem acceptable<sup>9,10</sup>.

For this reason, the thresholds for the microbe class probability and NMS are selected for each model to minimize first the sMAPE metric, which is defined as

$$\text{sMAPE} = \frac{100\%}{N} \sum_{i=1}^N \frac{|x_i - \tilde{x}_i|}{|x_i + \tilde{x}_i|},\tag{3}$$

where  $N$  is a number of all *countable* samples,  $x_i$  is total count of instances present on  $i$ -th image, and  $\tilde{x}_i$  is predicted number of microbe instances. In the above formula, if  $x_i = \tilde{x}_i = 0$ , then the  $i$ -th term in the summation is 0 (calculated error is zero, because the number of predicted colonies is correct). Please note that the denominator favors counting errors for samples with a small number of colonies, ensuring that they are equally important as those with more microbe instances. Moreover, the applied symmetrization favors neither underestimated (small recall) nor overestimated (low precision) counts by a model.

If the value of a few lowest sMAPE values (for different thresholds) slightly differs (less than 0.1%), we compare achieved MAE value calculated using the formula

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |x_i - \tilde{x}_i|,\tag{4}$$

and choose such thresholds pair that gives the lowest MAE. Additionally, sum of MAE calculated for every microbe separately, called cumulative MAE, is given as

$$\text{cMAE} = \sum_{m=1}^K \text{MAE}_m,\tag{5}$$

where  $K = 5$  is a number of all microbe species, and  $\text{MAE}_m$  indicates MAE calculated for  $m$ -th microbial class solely.

## 4 Deep learning models evaluation and comparison

In this section detailed analysis is presented to verify the suitability of the AGAR dataset for building deep learning models for image-based microorganisms recognition. To set a benchmark for the task, the performance of two neural networks architectures for object detection: Faster R-CNN<sup>11</sup> and Cascade R-CNN<sup>12</sup>, with four different backbones: ResNet-50<sup>13</sup>, ResNet-101<sup>13</sup>, ResNeXt-101<sup>14</sup>, and HRNet<sup>15</sup> is evaluated. The implementation from the MMDetection<sup>16</sup> toolbox was used. Backbones' weights were initialized using models pre-trained on ImageNet<sup>17</sup>, available in the *torchvision* package of the PyTorch library.

The quantitative results for each subset of AGAR analyzed separately, as well as the results for models trained on the whole dataset, are reported. Each subset was split randomly into approximately 75% samples constituting the training set, and 25% for the validation/testing set. Due to the relatively high resolution of all images, they were divided into patches of 512x512 px to avoid scaling them down, which could make it difficult to recognize small colonies. If necessary, zero padding was applied to get a proper aspect ratio. As the goal of this study is to establish a baseline for the microbe detection task, neural network architectures were not adjusted, so the patches were resized to match default input layer size.

Several approaches to data augmentation were tested. During training, Gaussian blur and salt and pepper noise were added, image color space was changed to LAB and HSV, histogram equalization was applied, images were rotated in the  $[-45, 45]$  degrees range, and cropped around the annotated bounding boxes, such that there is always a visible microbe object. However, the best results were obtained through splitting the images into patches, and normalizing them using means and standard deviation values per channel as for the COCO dataset<sup>7</sup>.

### 4.1 Training and validation details

As it was mentioned earlier, patches for validation and testing are prepared from the same photos (1/4 of the given subset), albeit in a different manner: patches for testing are cut out from images evenly with some padding applied – see the blue path in Fig. 7. It turns out that to filter out unwanted bounding boxes during the post-processing stage, two parameters need to be adjusted: detected colony classification probability and NMS threshold. Both of them are fitted to each subset separately. However, to ensure methodological correctness during tests, they are optimized on the subset of training data, and the values which minimize sMAPE is chosen.

The AGAR dataset is relatively large and the top deep learning models used in this studies are of complex structure (e.g. the Cascade R-CNN model with the HRNet has got 956 different layers in total). Therefore, the training process lasts from about 2 days for the simplest (and smallest) Faster R-CNN with ResNet-50 backbone to about 5 days for the biggest one, Cascade R-CNN with the HRNetV2W48 backbone. Calculations were performed on NVIDIA Tesla V100 GPUs.

During training, the Stochastic Gradient Descent (SGD) method was used for optimizing the network weights, and models were trained over 20 epochs with a batch size of 3-10 patches (depending on the model complexity, i.e. GPU memory consumption), and an adaptive learning rate was initialized with 0.0001. After about 20 epochs loss values saturate, and longer training reduces neither the training nor the validation error.

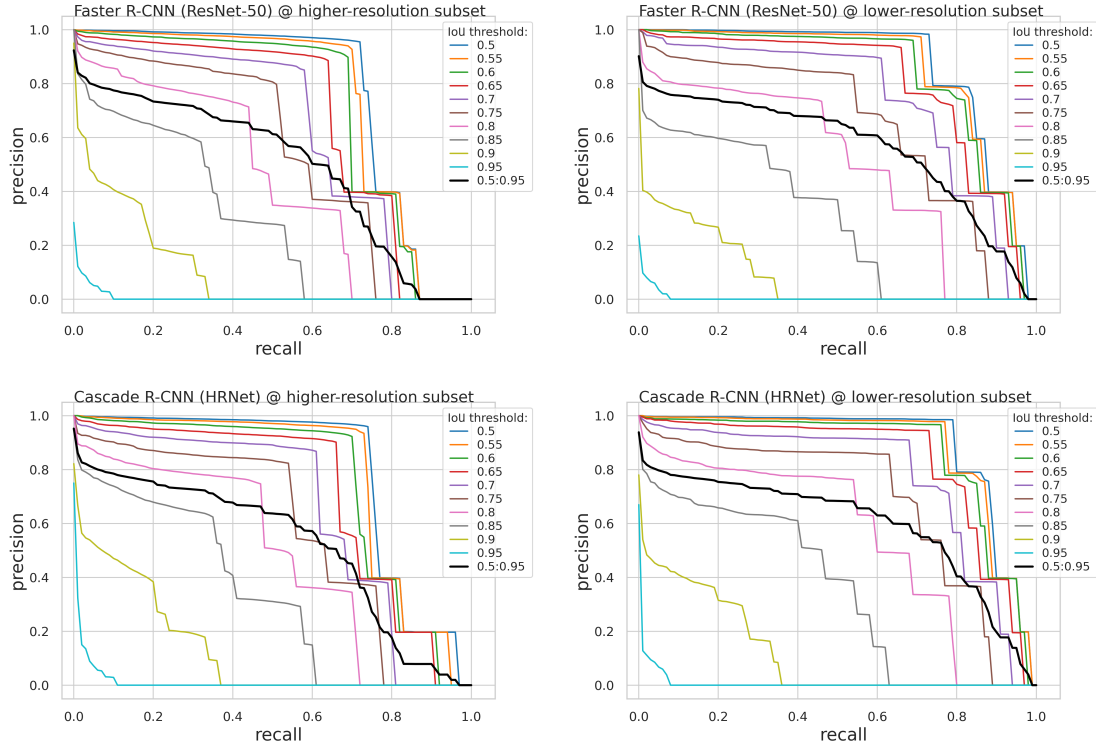
### 4.2 Precision of detection

Generally speaking, the precision of detection relates to overlapping between predicted and ground truth bounding boxes, which is measured by the IoU. The prediction is valid if IoU is above given threshold (and a microbe is properly classified), as described in Section 3.1. Precision and recall are then calculated (as defined by Eq. 2) for every colony in a subset giving a precision-recall curve.

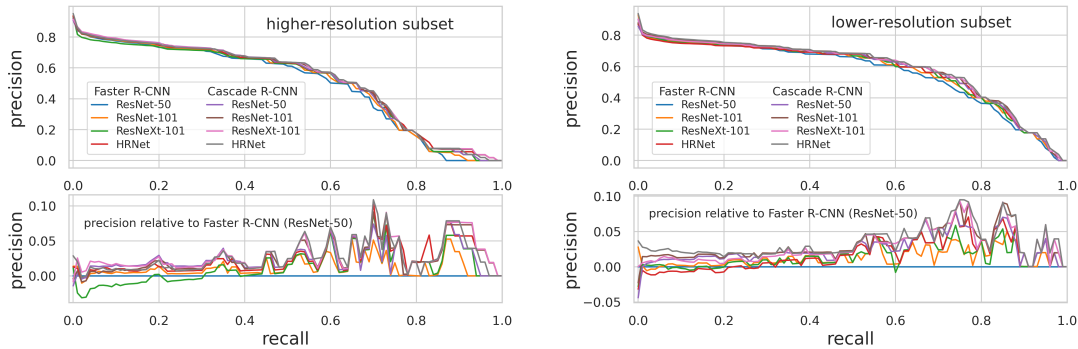
Fig. 8 presents the PR curves at different IoU thresholds levels, ranging from 0.5 to 0.95, for two selected models on *lower-* and *higher-resolution* data subset separately. The black curve is the average over all thresholds. These curves shows two characteristic features. Firstly, precision competes with recall in a sense that requirement of high detection precision results in low recall, and vice versa. Moreover, the higher IoU threshold (so higher overlapping requirements), the lower precision values. In the ideal case, precision equals one for all the values of recall. Secondly, for high recall values, precision is greater for the *lower-resolution* background category.

The averaged PR curves (over IoU thresholds in the range  $[0.5 : 0.95]$ ) for different models are presented in Fig. 9. Altogether, eight models are considered as the all possible combinations of two detector architectures (Faster R-CNN and Cascade R-CNN) with four different backbones (ResNet-50, ResNet-101, ResNeXt-101, and HRNetV2W48). The smallest model (Faster R-CNN with ResNet-50) is chosen as a baseline and the relative performance of all other models is also show in Fig. 9. The best result are obtained for Cascade R-CNN with ResNeXt-101 in the case of *higher-resolution* background category, and with HRNet in the case of *lower-resolution* subset. However, the small difference between the models performance suggests that the models capability (at least in this class of architectures) has been reached, and further models complicating (e.g. by adding more layers in the backbones) may not lead to much better performance.

The average precision values at different IoU thresholds are presented in Table 1 for the *higher-resolution* subset, and in Table 2 for the *lower-resolution* one. The last row is mean AP (mAP) over the whole range of thresholds, i.e.  $[0.5 : 0.95]$ . In



**Figure 8.** The precision-recall curves for different IoU thresholds obtained for the two major subsets of the AGAR dataset using Faster R-CNN with ResNet-50, and Cascade R-CNN with HRNet models.



**Figure 9.** The precision-recall curves for various models: Faster R-CNN and Cascade R-CNNs with four different backbones, trained on *higher-resolution* and *lower-resolution* subsets. For bottom charts we establish a baseline based on a the Faster R-CNN with ResNet-50, i.e. precision for the all other models are presented relative to this model.

general, Cascade R-CNN performs slightly better on the AGAR dataset than Faster R-CNN. Although increasing the complexity of a backbones gives slightly better results, most of them are quite close to each other.

The results presented in Figures 8-9 and Tables 1-2 are averaged over the all microbe classes. Per-microbe precision is shown in Tab. 3 for the baseline model compared to Cascade R-CNN with the HRNet backbone, on the *higher-resolution* subset. In general, detector performs better for microbes that form smaller colonies, i.e. *S. aureus* and *C. albicans*. The lower precision values for bigger colonies are mainly due to their tendency to aggregate and overlap, especially in higher populated samples. Such overlapped colonies make it harder to detect individual colonies. Moreover, larger colonies tend to have blurred edges (with the lower contrast relative to agar substrate), which affects IoU.



**Table 1.** Average precision for different models averaged over different IoU-threshold levels obtained for the *higher-resolution* subset. The Cascade R-CNN model with ResNeXt-101 backbone gives the highest mAP.

| IoU      | Faster R-CNN |            |             |       | Cascade R-CNN |            |              |       |
|----------|--------------|------------|-------------|-------|---------------|------------|--------------|-------|
|          | ResNet-50    | ResNet-101 | ResNeXt-101 | HRNet | ResNet-50     | ResNet-101 | ResNeXt-101  | HRNet |
| 0.5      | 76.7%        | 77.9%      | 78.5%       | 79.9% | 79.2%         | 79.9%      | 80.1%        | 79.5% |
| 0.55     | 74.7%        | 76.1%      | 76.6%       | 78.0% | 77.3%         | 78.0%      | 78.2%        | 77.8% |
| 0.6      | 71.9%        | 73.0%      | 73.1%       | 74.8% | 74.2%         | 74.9%      | 75.2%        | 74.6% |
| 0.65     | 67.3%        | 68.6%      | 68.4%       | 70.3% | 70.4%         | 71.3%      | 71.5%        | 70.7% |
| 0.7      | 61.9%        | 63.1%      | 63.1%       | 64.0% | 64.2%         | 64.2%      | 64.5%        | 64.3% |
| 0.75     | 54.8%        | 55.9%      | 55.7%       | 57.1% | 57.0%         | 56.9%      | 57.6%        | 57.3% |
| 0.8      | 44.4%        | 45.6%      | 46.0%       | 47.4% | 47.3%         | 47.2%      | 48.2%        | 47.9% |
| 0.85     | 29.4%        | 30.8%      | 30.9%       | 32.2% | 32.7%         | 32.5%      | 33.2%        | 33.2% |
| 0.9      | 11.1%        | 11.9%      | 11.8%       | 12.5% | 12.8%         | 13.0%      | 12.9%        | 13.2% |
| 0.95     | 0.8%         | 1.4%       | 1.0%        | 1.3%  | 1.1%          | 1.6%       | 1.5%         | 1.7%  |
| 0.5:0.95 | 49.3%        | 50.4%      | 50.5%       | 51.8% | 51.6%         | 51.9%      | <b>52.3%</b> | 52.0% |

**Table 2.** Average precision for different models averaged over different IoU-threshold levels obtained for the *lower-resolution* subset. The Cascade R-CNN model with HRNet backbone gives the highest mAP.

| IoU      | Faster R-CNN |            |             |       | Cascade R-CNN |            |             |              |
|----------|--------------|------------|-------------|-------|---------------|------------|-------------|--------------|
|          | ResNet-50    | ResNet-101 | ResNeXt-101 | HRNet | ResNet-50     | ResNet-101 | ResNeXt-101 | HRNet        |
| 0.5      | 86.5%        | 87.6%      | 87.4%       | 88.2% | 88.6%         | 89.2%      | 88.7%       | 89.1%        |
| 0.55     | 85.0%        | 86.3%      | 86.7%       | 87.1% | 87.4%         | 87.9%      | 87.6%       | 87.9%        |
| 0.6      | 83.2%        | 84.4%      | 84.3%       | 84.8% | 85.5%         | 86.1%      | 86.2%       | 86.1%        |
| 0.65     | 79.5%        | 80.8%      | 80.7%       | 81.1% | 82.1%         | 82.6%      | 82.1%       | 82.6%        |
| 0.7      | 73.2%        | 75.0%      | 75.2%       | 75.2% | 76.6%         | 77.1%      | 76.7%       | 76.5%        |
| 0.75     | 63.6%        | 65.4%      | 66.1%       | 65.8% | 67.1%         | 68.3%      | 67.5%       | 68.0%        |
| 0.8      | 49.6%        | 51.3%      | 51.7%       | 52.2% | 53.4%         | 54.0%      | 53.7%       | 55.0%        |
| 0.85     | 29.5%        | 30.9%      | 30.9%       | 31.1% | 32.1%         | 32.9%      | 32.7%       | 35.1%        |
| 0.9      | 9.2%         | 9.8%       | 10.0%       | 10.0% | 10.7%         | 11.2%      | 11.0%       | 12.2%        |
| 0.95     | 0.6%         | 0.9%       | 0.7%        | 0.7%  | 0.7%          | 0.9%       | 0.9%        | 1.2%         |
| 0.5:0.95 | 56.0%        | 57.3%      | 57.4%       | 57.6% | 58.4%         | 59.0%      | 58.7%       | <b>59.4%</b> |

**Table 3.** Average precision calculated for each class of microbes separately at different IoU-threshold levels for (left) Faster R-CNN (ResNet-50), and (right) Cascade R-CNN (HRNet) on *higher-resolution* subset. We get overall better performance for smaller colonies.

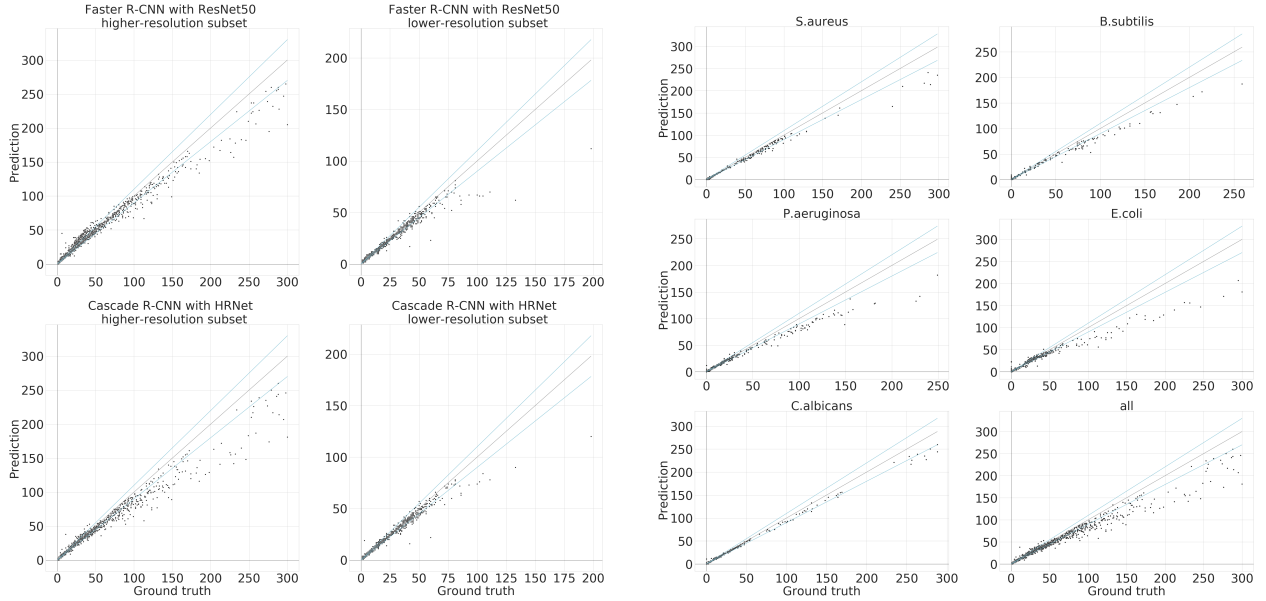
| IoU      | AP per class: Faster R-CNN (ResNet-50) |                    |                      |                |                    | AP per class: Cascade R-CNN (HRNet) |                    |                      |                |                    |
|----------|----------------------------------------|--------------------|----------------------|----------------|--------------------|-------------------------------------|--------------------|----------------------|----------------|--------------------|
|          | <i>S. aureus</i>                       | <i>B. subtilis</i> | <i>P. aeruginosa</i> | <i>E. coli</i> | <i>C. albicans</i> | <i>S. aureus</i>                    | <i>B. subtilis</i> | <i>P. aeruginosa</i> | <i>E. coli</i> | <i>C. albicans</i> |
| 0.5      | 86.0%                                  | 72.4%              | 69.9%                | 73.1%          | 82.0%              | 95.8%                               | 73.6%              | 71.7%                | 74.6%          | 82.1%              |
| 0.55     | 85.9%                                  | 69.5%              | 67.4%                | 68.9%          | 82.0%              | 93.8%                               | 70.8%              | 70.7%                | 71.5%          | 82.0%              |
| 0.6      | 84.8%                                  | 64.6%              | 65.6%                | 64.0%          | 80.8%              | 90.8%                               | 65.7%              | 68.9%                | 66.6%          | 80.9%              |
| 0.65     | 80.9%                                  | 56.9%              | 62.4%                | 56.6%          | 79.6%              | 89.7%                               | 59.1%              | 65.5%                | 59.3%          | 79.8%              |
| 0.7      | 77.9%                                  | 48.9%              | 57.9%                | 47.9%          | 77.1%              | 79.4%                               | 50.9%              | 61.5%                | 51.2%          | 78.4%              |
| 0.75     | 71.1%                                  | 39.8%              | 50.9%                | 39.0%          | 73.2%              | 74.4%                               | 41.6%              | 53.8%                | 41.7%          | 74.8%              |
| 0.8      | 58.4%                                  | 30.0%              | 38.0%                | 30.2%          | 65.5%              | 64.5%                               | 31.2%              | 42.7%                | 33.1%          | 68.1%              |
| 0.85     | 37.4%                                  | 17.5%              | 23.3%                | 19.4%          | 49.4%              | 46.5%                               | 19.2%              | 25.5%                | 22.3%          | 52.4%              |
| 0.9      | 14.8%                                  | 6.3%               | 7.9%                 | 7.4%           | 19.2%              | 18.0%                               | 7.6%               | 8.3%                 | 10.2%          | 22.1%              |
| 0.95     | 0.6%                                   | 0.4%               | 0.7%                 | 0.5%           | 1.8%               | 2.9%                                | 1.3%               | 0.8%                 | 0.8%           | 2.7%               |
| 0.5:0.95 | 59.8%                                  | 40.6%              | 44.4%                | 40.7%          | <b>61.1%</b>       | <b>65.6%</b>                        | 42.1%              | 46.9%                | 43.1%          | 62.3%              |

### 4.3 Microbe counting results

When it comes to counting, the recognition of microbes is crucial, but the exactness of their localization (i.e. true and predicted bounding boxes overlapping) is of secondary importance. In the post processing stage, the sMAPE and MAE metrics are used to tune the probability and NMS thresholds on the AGAR training set, as described in Section 3.2. The thresholds are used when the predictions for individual patches are merged into the whole test image prediction (to get rid of the excess of bounding boxes, e.g. appearing twice through the edges of neighbouring patches).

The performance of Faster R-CNN and Cascade R-CNN models with ResNet-50 and HRNet backbones for microbial counting is presented in Fig. 10(left) for both major background categories of the AGAR dataset, and Fig. 10(right) for all five bacteria species separately. The straight black line represents the identity function (meaning perfect prediction), while

cyan lines indicate 10% error to highlight the acceptable range. Starting from a around 50 colonies, the detectors clearly underestimate the their exact number. This is largely due to the nature of sMAPE, which give more weight to errors coming from less populated samples. Moreover, the AGAR dataset is dominated by the plates with less than 50 colonies. Therefore, the probability and NMS thresholds for post processing are tuned to favour these samples.



**Figure 10.** The performance of microbial colony counting on: (left) two main background categories of the AGAR dataset using Faster R-CNN and Cascade R-CNN with two backbones ResNet-50 and HRNet, and (right) results achieved using Cascade R-CNN with HRNet divided into 5 microbial species.

Microbe counting metrics for various models for the *higher-* and *lower-resolution* subsets are presented in Tables 4-5. While the best results are obtained with Cascade R-CNN with HRNet, it is worth noting that the most lightweight model (Faster R-CNN with ResNet-50) is not much worse. In general, the errors for the *lower-resolution* subset are smaller, which is likely caused by the presence of more difficult samples in the *higher-resolution* subset. While the *lower-resolution* subset is more homogeneous, therefore it is simpler. Moreover, the counting performance is better for the smallest microbial colonies (*C. albicans* and *S. aureus*), which is related to the highest mAP for these samples (as explained in Sec. 4.2).

**Table 4.** MAE, cMAE, and sMAPE calculated for the each class of microbes, and altogether, for Faster R-CNN and Cascade R-CNN with four different backbones on the *higher-resolution* subset. cMAE metrics for each model are written in brackets.

| Microbe counting metrics for various models @ <i>higher-resolution</i> subset |               |       |             |       |             |       |             |       |
|-------------------------------------------------------------------------------|---------------|-------|-------------|-------|-------------|-------|-------------|-------|
| microbe                                                                       | Faster R-CNN  |       |             |       |             |       |             |       |
|                                                                               | ResNet-50     |       | ResNet-101  |       | ResNeXt-101 |       | HRNet       |       |
|                                                                               | MAE           | sMAPE | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE |
| <i>S. aureus</i>                                                              | 0.42          | 2.78% | 0.66        | 2.43% | 0.39        | 2.35% | 0.67        | 2.55% |
| <i>B. subtilis</i>                                                            | 0.66          | 1.96% | 0.74        | 1.88% | 0.57        | 2.07% | 0.67        | 1.69% |
| <i>P. aeruginosa</i>                                                          | 1.21          | 5.53% | 1.35        | 4.64% | 1.50        | 5.24% | 1.28        | 4.37% |
| <i>E. coli</i>                                                                | 1.62          | 5.40% | 1.77        | 4.83% | 1.79        | 4.18% | 1.63        | 4.27% |
| <i>C. albicans</i>                                                            | 0.93          | 2.98% | 0.56        | 2.34% | 0.60        | 2.58% | 0.53        | 2.23% |
| altogether                                                                    | 4.75 (4.84)   | 5.32% | 4.44 (5.08) | 5.54% | 4.47 (4.85) | 5.14% | 4.35 (4.78) | 4.92% |
| microbe                                                                       | Cascade R-CNN |       |             |       |             |       |             |       |
|                                                                               | ResNet-50     |       | ResNet-101  |       | ResNeXt-101 |       | HRNet       |       |
|                                                                               | MAE           | sMAPE | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE |
| <i>S. aureus</i>                                                              | 0.67          | 2.51% | 0.65        | 2.40% | 0.35        | 2.48% | 0.68        | 2.27% |
| <i>B. subtilis</i>                                                            | 0.78          | 1.36% | 0.71        | 1.26% | 0.45        | 1.69% | 0.68        | 1.97% |
| <i>P. aeruginosa</i>                                                          | 1.39          | 5.04% | 1.30        | 4.59% | 1.69        | 4.31% | 1.26        | 4.48% |
| <i>E. coli</i>                                                                | 1.71          | 4.40% | 1.69        | 3.81% | 1.84        | 3.42% | 1.58        | 3.03% |
| <i>C. albicans</i>                                                            | 0.51          | 2.52% | 0.50        | 2.11% | 0.35        | 2.75% | 0.49        | 2.05% |
| altogether                                                                    | 4.67 (5.06)   | 5.15% | 4.41 (4.85) | 5.07% | 4.40 (4.68) | 4.99% | 4.31 (4.69) | 4.86% |

**Table 5.** Same as in Table 4 but for the *lower-resolution* subset.

| Microbe counting metrics for various models @ <i>lower-resolution</i> subset |             |       |             |       |             |       |             |       |
|------------------------------------------------------------------------------|-------------|-------|-------------|-------|-------------|-------|-------------|-------|
| Faster R-CNN                                                                 |             |       |             |       |             |       |             |       |
| microbe                                                                      | ResNet-50   |       | ResNet-101  |       | ResNeXt-101 |       | HRNet       |       |
|                                                                              | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE |
| <i>S. aureus</i>                                                             | 0.54        | 3.55% | 0.37        | 3.58% | 0.36        | 2.58% | 0.27        | 2.35% |
| <i>B. subtilis</i>                                                           | 0.23        | 0.63% | 0.25        | 0.54% | 0.37        | 0.79% | 0.16        | 0.40% |
| <i>P. aeruginosa</i>                                                         | 0.55        | 5.60% | 0.39        | 4.36% | 0.46        | 4.74% | 0.39        | 4.27% |
| <i>E. coli</i>                                                               | 0.75        | 4.71% | 0.65        | 3.82% | 0.57        | 3.62% | 0.59        | 3.09% |
| <i>C. albicans</i>                                                           | 0.37        | 1.49% | 0.33        | 1.08% | 0.25        | 1.22% | 0.22        | 1.15% |
| altogether                                                                   | 1.83 (2.24) | 4.68% | 1.69 (1.99) | 4.55% | 1.92 (2.01) | 4.21% | 1.54 (1.63) | 4.12% |

| Cascade R-CNN        |             |       |             |       |             |       |             |       |
|----------------------|-------------|-------|-------------|-------|-------------|-------|-------------|-------|
| microbe              | ResNet-50   |       | ResNet-101  |       | ResNeXt-101 |       | HRNet       |       |
|                      | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE | MAE         | sMAPE |
| <i>S. aureus</i>     | 0.33        | 3.45% | 0.33        | 3.94% | 0.30        | 3.29% | 0.32        | 2.20% |
| <i>B. subtilis</i>   | 0.35        | 0.60% | 0.31        | 0.57% | 0.32        | 0.58% | 0.31        | 0.49% |
| <i>P. aeruginosa</i> | 0.43        | 6.57% | 0.42        | 6.97% | 0.43        | 6.06% | 0.40        | 4.13% |
| <i>E. coli</i>       | 0.60        | 5.46% | 0.58        | 4.51% | 0.57        | 5.43% | 0.52        | 3.05% |
| <i>C. albicans</i>   | 0.25        | 1.56% | 0.24        | 2.17% | 0.22        | 1.18% | 0.21        | 1.32% |
| altogether           | 1.70 (1.96) | 4.15% | 1.62 (1.88) | 4.25% | 1.60 (1.84) | 4.31% | 1.57 (1.76) | 3.81% |

**Table 6.** Counting error metrics for various setups of anchors hyperparameters. Narrowing the range of anchors ratio lowers the sMAPE, while modifying anchors stride significantly improves MAE.

| Model (Cascade(HRNet) @ <i>higher-resolution</i> subset) hyperparameters optimization |                                  |       |                        |       |                          |       |                           |       |
|---------------------------------------------------------------------------------------|----------------------------------|-------|------------------------|-------|--------------------------|-------|---------------------------|-------|
| parameter modification                                                                | without modification (reference) |       | anchor ratio variation |       | anchor scaling variation |       | anchor stride list tuning |       |
|                                                                                       | MAE                              | sMAPE | MAE                    | sMAPE | MAE                      | sMAPE | MAE                       | sMAPE |
| metrics                                                                               | 4.31                             | 4.86% | 4.28                   | 4.68% | 4.35                     | 4.92% | 2.52                      | 5.14% |

#### 4.4 Anchor tuning

Hyperparameters are the variables which determines the network structure itself (e.g the number of hidden layers) or its training process (e.g. the learning rate or the number of epochs). The models tested in this study are already provided with the default (pre-optimized) learning hyperparameters values<sup>16</sup> and network architectures, which is specific for each model and backbone used. However, it may be interesting to verify if proper tuning of some parameters, to adjust them to our specific dataset, gives us some better results in terms of lower counting error metrics.

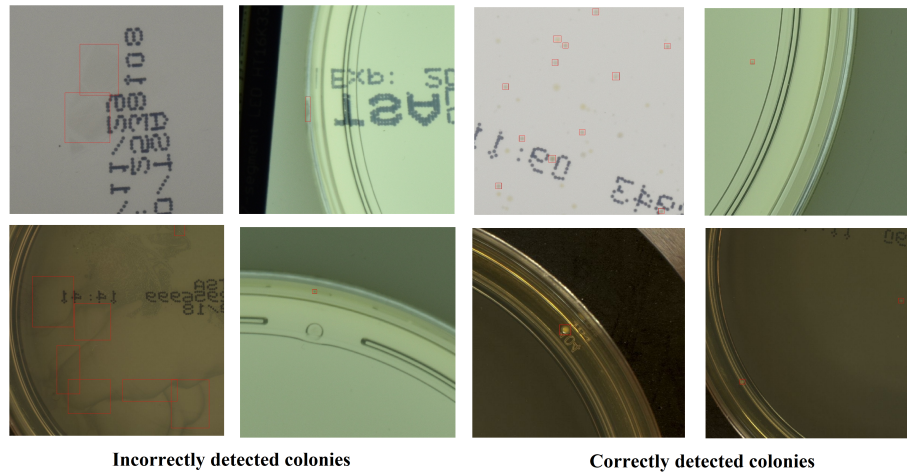
One of the crucial parameter that impact object detection performance is the anchor boxes setup. Anchors are some initial guessed bounding boxes that are assumed at the start of the detection procedure. The proper selection of anchors is a broad topic, currently investigated by many researchers. For example, automatic anchor selection techniques, i.e. which learn meta-anchors during network training, have been proposed recently<sup>18,19</sup>.

Here, basic approach is applied by simply varying the anchors' parameters and training the Cascade R-CNN with HRNet model on *higher-resolution* subset for each setup separately. In Table 6 there are presented results (counting errors) for the various anchors' parameters variations: (1) the aspect ratio (i.e.  $x$  vs  $y$  size), (2) the list of scales, and (3) the stride modification. While the anchors setup has an impact on the detection performance, it is hard to significantly overcome results for default settings. This suggests that chosen pre-tuned models are well adjusted to work on the AGAR dataset. The only significant impact comes from modifying anchors' strides, causing noteworthy decrease of MAE with slightly higher sMAPE. This means that the tuning of the stride improves the results for high populated samples. Moreover, narrowing the anchors' ratio range from  $[0.5, 2.0]$  to  $[0.75, 1.5]$  improves sMAPE from 4.86% to 4.68%. It is expected as a typical bounding box for microbe colony has a square-like shape.

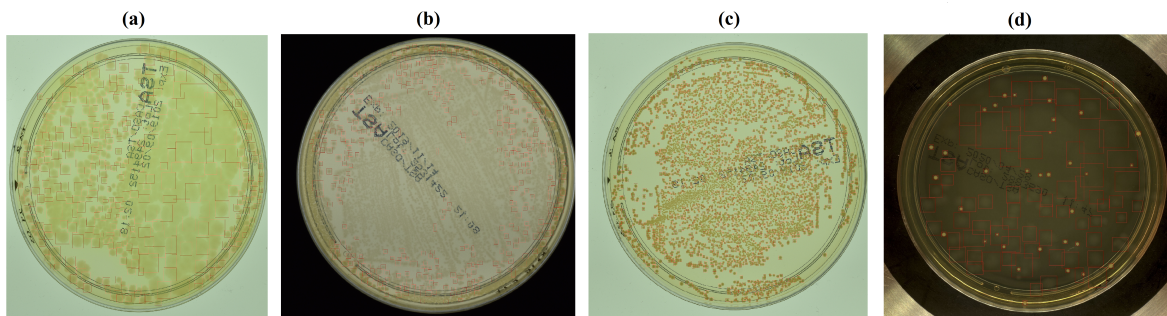
#### 4.5 Empty and uncountable plates

The performance of trained Cascade R-CNN with HRNet backbone was evaluated additionally on the *empty* and *uncountable* samples of the AGAR dataset, for both major background categories: *lower-* and *higher-resolution* (without the *vague* subset). In this case only qualitative results are provided, because ground truth annotations are available only for *countable* plates.

Testing *empty* samples allows us to investigate the false positive predictions (the model finds colonies on plates without microbes grown). The most common FPs are: discoloration and substrate loss, plate edge stains, formed blisters and water droplets. Also note that in some situations correct colonies are found, but they are hardly visible, so not marked by annotators. A few examples of such cases are presented in Fig. 11.



**Figure 11.** Examples of predictions for a few *empty* samples.



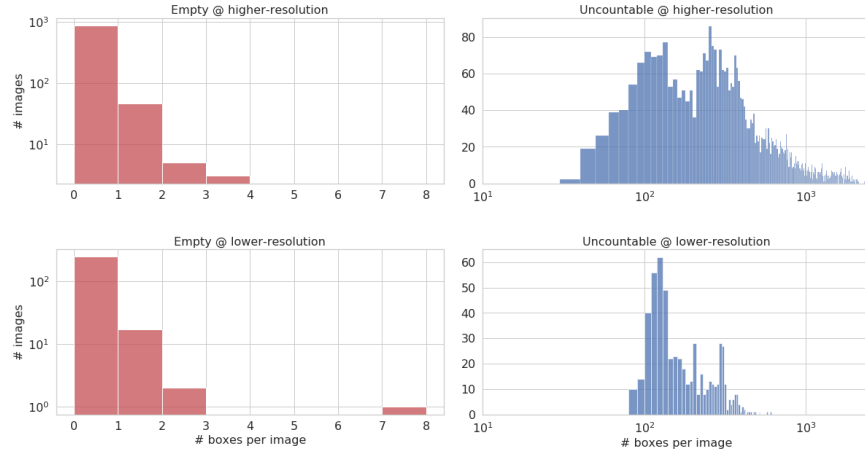
**Figure 12.** Examples of predictions for *uncountable* cases.

On the other hand, tests performed on *uncountable* samples enable us to check how the detector (trained on annotated data with max. 300 colonies) performs with very crowded plates. Here, the obvious problem for the detector is the so-called microbial lawn – meaning individual colonies fused together and spread evenly over the agar surface (Fig. 12(a)). Another similar challenge is recognizing (and consequently counting) individual colonies in growth bacterial streak, as presented in Fig. 12(b). Very small colonies (of a size of few pixels) are difficult to detect, however, due to their large separation, the model shows a great potential to roughly estimate the number of colonies even for very crowded plates – at best, it detects 2782 colonies (Fig. 12(c)). The last example (Fig. 12(d)) presents the sample with two different species grown on one agar plate. The model is able to detect even hardly visible colonies, although it also recognizes some discolorations as a microbe.

The model performs quite well for detecting colonies in both edge cases. However, it handles *empty* plates slightly better. The false positive predictions occur only for about 6-7% of all *empty* dishes. Note, however, that in some cases there are actually some formed colonies missed by the annotators (like in the examples in Fig. 11). In the case of *uncountable* subset, the number of colonies are very often underestimated. The model predicts less than 300 colonies for 41% and 85% samples for *higher-* and *lower-resolution*, respectively. However, only 7% and 4% are estimated to have less than 100 colonies. Note that it was harder to annotate the *lower-resolution* images (because of low contrast), so some samples labeled as *uncountable* may be actually within the defined countable range. Moreover, as discussed before, the sMAPE metric used for threshold optimization causes the model to be more accurate for samples with smaller number of colonies on a plate. Detailed distributions of counted colonies can be found in Fig. 13.

#### 4.6 Double thresholding approach

Looking at the obtained results for microbe counting (see Fig. 10), one can conclude that the trained models underestimate the number of colonies for more numerous samples. This is due to the distribution of our data, presented in Fig. 6, with more than 84% of samples consisting of photos of plates with less than 50 grown colonies. To remedy this, a double thresholding approach was investigated. Two sets of the probability and NMS thresholds are established based on training data: *general*



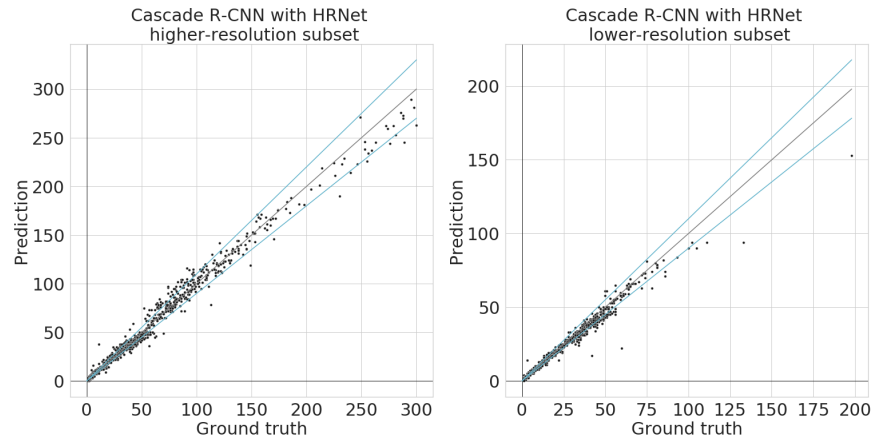
**Figure 13.** Distribution of number of predicted colonies for *higher-* and *lower-resolution* photos in both *empty* and *uncountable* edge cases.

(tuned to all samples) and *auxiliary* (tuned to samples with more than 50 colonies). The *mixed* approach uses *general* thresholds by default, however, if the model predicts more than 50 colonies, the *auxiliary* ones are applied for a given sample.

**Table 7.** Counting error metrics for Cascade with HRNet model in various setups of thresholds scenario.

| Parameters                                | Thresholds settings | MAE  | sMAPE |
|-------------------------------------------|---------------------|------|-------|
| Cascade HRNet<br><i>higher-resolution</i> | general             | 4.31 | 4.86% |
|                                           | auxiliary           | 3.49 | 7.64% |
|                                           | mixed               | 2.35 | 4.06% |
| Cascade HRNet<br><i>lower-resolution</i>  | general             | 1.57 | 3.81% |
|                                           | auxiliary           | 1.50 | 4.97% |
|                                           | mixed               | 1.49 | 3.67% |

The results for different thresholds settings are presented in Table 7. As one can expect, *auxiliary* thresholds taken individually increase sMAPE for test data, but they also decrease MAE. However, *mixed* approach gives a good balance, and works slightly better than single *general* thresholding for all samples in the AGAR dataset. Its performance for microbe counting is shown in Fig. 14.



**Figure 14.** Performance of microbial colony counting on two images subgroups using Cascade R-CNN with HRNet for double thresholding approach.



## 4.7 Transfer learning

Transfer learning is a technique used to improve learning a new task by transferring the knowledge from related problems<sup>20,21</sup>. It can be used to adjust the model trained on the AGAR dataset for new type of samples (different microbes, lighting condition etc.), unseen by the detector, if the new data is too small to perform efficient training solely on it. Here, the ability of a pre-trained model to learn new features from photos taken in different data acquisition setup is checked.

A transfer learning experiment is proposed by using *vague* (the smallest and the most difficult) subset to mimic unseen data, and assigning (much larger) *bright+dark* subsets as a primary dataset used to initially train the model. In the first approach, the Cascade R-CNN with HRNet model pre-trained on the primary data is taken and then additional learning of the whole network using solely the *vague* samples is performed. This leads to a new model, which (just after a few epochs) perfectly recognizes microbes on the *vague* subset. However, it loses the performance on the primary dataset.

To overcome this, i.e. make a model to learn new features but keep its ability to recognize the primary ones, one can update only a few last layers of the neural network, keeping the rest frozen<sup>22</sup>. The results of this technique is presented in Table 8. With the increasing number of unfrozen layers, the counting metrics (MAE and sMAPE) significantly improve for the *vague* subset, but get slightly worse for the primary data. Unfreezing more than 128 layers does not further improve the performance on the *vague* photos, but makes the model less capable of recognizing the samples from *bright* and *dark* subsets.

**Table 8.** Transfer learning for the model trained on the *bright+dark* subsets to detect microbes on the *vague* one, by updating only part of last layers i.e. all other weights are frozen. We observe that with increasing the number of unfrozen layers, the counting metrics for the *vague* subset improves, while for the *bright+dark* one—decreases.

| Fine-tuning of Cascade R-CNN (HRNet) model trained on <i>bright+dark</i> subset to capture <i>vague</i> subset |               |        |       |        |      |        |      |        |      |        |
|----------------------------------------------------------------------------------------------------------------|---------------|--------|-------|--------|------|--------|------|--------|------|--------|
| # unfrozen layers                                                                                              | 0 (reference) |        | 16    |        | 32   |        | 64   |        | 128  |        |
| metrics for:                                                                                                   | MAE           | sMAPE  | MAE   | sMAPE  | MAE  | sMAPE  | MAE  | sMAPE  | MAE  | sMAPE  |
| <i>bright+dark</i> subset                                                                                      | 4.31          | 4.86%  | 3.92  | 5.19%  | 7.36 | 10.73% | 7.39 | 10.77% | 7.21 | 11.38% |
| <i>vague</i> subset                                                                                            | 15.27         | 39.55% | 14.79 | 38.49% | 4.12 | 9.84%  | 3.32 | 8.59%  | 2.51 | 5.34%  |

## 4.8 Training on the whole dataset at once

Above, the results of transfer-learned version of Cascade R-CNN with HRNet on the AGAR dataset are reported. The conducted studies showed that the use of such transfer-learning methods to overcome some deficiencies introduced to our dataset (e.g. by removing the *vague* part) might be not sufficient to prepare a model capable for general predictions. To improve the model's ability to generalize, the model was trained on the entire dataset (combined training subsets of *lower-* and *higher-resolution* images). Metrics evaluated for this case (see Table 9) show that both Cascade R-CNN and Faster R-CNN models have great ability to capture different datasets *at once*. This was verified by testing such generalized model on test data for each subset separately, as presented in Table 9.

Let us compare the best result for the transfer technique presented in Subsection 4.7 (Table 8), i.e. sMAPE of 11.38% (*bright+dark*) and 5.34% (*vague*), with the results from the Table 9 for Cascade R-CNN (HRNet) giving about 4.98% for the *bright+dark*, and 1.96% for the *vague* subset. The models trained on all combined subsets at once performed better than these obtained using transfer of learning presented in the subsection 4.7. However, the transfer learning technique gives the possibility to quickly adopt a model for a new subset, without having access to the primary data the model was trained on.

**Table 9.** Results for the Faster and Cascade R-CNN models with ResNet-50 and HRNet backbones trained on the whole dataset (*higher-* and *lower-resolution*), and tested on different subsets respectively. Evaluated metrics show that the both models have great ability to capture various datasets simultaneously.

| Microbe counting metrics for various models trained on the all dataset |                    |       |              |       |                         |       |                    |       |              |       |                         |       |
|------------------------------------------------------------------------|--------------------|-------|--------------|-------|-------------------------|-------|--------------------|-------|--------------|-------|-------------------------|-------|
| model                                                                  | Faster R-CNN       |       |              |       |                         |       |                    |       |              |       |                         |       |
|                                                                        | ResNet-50          |       |              |       |                         |       | HRNet              |       |              |       |                         |       |
| tested on:                                                             | <i>bright+dark</i> |       | <i>vague</i> |       | <i>lower-resolution</i> |       | <i>bright+dark</i> |       | <i>vague</i> |       | <i>lower-resolution</i> |       |
| metrics                                                                | MAE                | sMAPE | MAE          | sMAPE | MAE                     | sMAPE | MAE                | sMAPE | MAE          | sMAPE | MAE                     | sMAPE |
|                                                                        | 4.87               | 5.70% | 1.10         | 2.60% | 1.96                    | 4.90% | 4.25               | 4.86% | 0.88         | 1.96% | 1.54                    | 3.90% |
| model                                                                  | Cascade R-CNN      |       |              |       |                         |       |                    |       |              |       |                         |       |
|                                                                        | ResNet-50          |       |              |       |                         |       | HRNet              |       |              |       |                         |       |
| tested on                                                              | <i>bright+dark</i> |       | <i>vague</i> |       | <i>lower-resolution</i> |       | <i>bright+dark</i> |       | <i>vague</i> |       | <i>lower-resolution</i> |       |
| metrics                                                                | MAE                | sMAPE | MAE          | sMAPE | MAE                     | sMAPE | MAE                | sMAPE | MAE          | sMAPE | MAE                     | sMAPE |
|                                                                        | 4.97               | 5.72% | 1.33         | 2.42% | 1.71                    | 4.48% | 4.29               | 4.98% | 0.83         | 1.96% | 1.66                    | 3.82% |

## References

1. European Pharmacopoeia, chapter 2.6.1 Sterility, page 155–158. Council of Europe, 6 edition, 2007. Ph. Eur. 6.0, 01/2008:20601.
2. <62> microbiological examination of non-sterile products: Tests for specified microorganisms. In United States Pharmacopeia and National Formulary (USP 31-NF 26), volume 31. United States Pharmacopeia, Washington, USA, 2007.
3. The Pharmacopoeia of Japan. Society of Japanese Pharmacopoeia, Tokyo, Japan, 17 edition, 2016.
4. Strains specified by official microbial assays, American Type Culture Collection [ATCC]. <https://www.atcc.org/~media/PDFs/Marketing%20Material/Microbiology/Pharmaceutical%20Microbiology.ashx>, 2017.
5. Diane M Tomasiewicz, Donald K Hotchkiss, Ralston B Read Jr George W Reinbold, and Paul A Hartman. The most suitable number of colonies on plates for counting. Journal of Food Protection, 43(4), 1980.
6. Scott Sutton. Accuracy of plate counts. Journal of validation technology, 17(3):42–46, 2011.
7. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740–755, Cham, 2014. Springer International Publishing.
8. Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Improving object detection with one line of code. CoRR, abs/1704.04503, 2017.
9. Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. Journal of the Operational Research Society, 66:1352–1362, 03 2015.
10. Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. CoRR, abs/1604.03505, 2016.
11. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15, page 91–99, Cambridge, MA, USA, 2015. MIT Press.
12. Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6154–6162, 2018.
13. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.
14. S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5987–5995, 2017.
15. Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, and Yadong Mu. Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, April 2020.
16. Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155, 2019.
17. J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.
18. Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10206–10215, 2020.
19. Yuanyi Zhong, Jianfeng Wang, Jian Peng, and Lei Zhang. Anchor box optimization for object detection. In The IEEE Winter Conference on Applications of Computer Vision, pages 1286–1294, 2020.
20. Lisa Torrey and Jude Shavlik. Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, pages 242–264. IGI global, 2010.
21. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22:1345–1359, 2010.

- 22.** Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328, 2014.