Automated Detection of Tuberculosis Bacilli in Whole-Slide Images of Stained Sputum Smears

Ida Büschel * **Matteo Cadoni** * **Lorenzo Capobianco** * **Marina Domínguez** * **Maximilian Buser** **Carsten Marr**

Abstract

Tuberculosis (TB), an infectious disease caused by Mycobacterium tuberculosis bacilli, remains one of the top ten causes of death worldwide. As of today, the standard diagnostic tool in low and middle income countries (WHO, 2006) is microscopic sputum smear examination, which is performed manually by human experts and, hence, is time-consuming and costly. In an effort to improve clinical workflows, we aim to develop a computational pipeline for the purpose of automatizing the detection and diagnosis of TB in whole-slide images (WSIs) of stained sputum smears. The computational methods that will be employed to preprocess, analyse and classify the WSIs revolve around computer vision techniques for object detection as well as Convolutional Neural Network (CNN) architectures for classification.

1. Introduction: Problem Formulation

1.1. Background and Motivation

According to the World Health Organization (WHO), although curable and preventable, TB is one of the world's deadliest diseases. The primary diagnostic method, particularly in low-income regions, involves the examination of stained sputum smears under the microscope. Usually, this is done manually by qualified experts, making it expensive, time-consuming and error-prone. Generally, manual evaluation of TB severity has demonstrated low sensitivity (Hailemariam, 2018). The automated detection of TB bacteria in WSIs of stained sputum smears could enhance severity assessment and therapeutic decision-making. To this end, we implemented a computational pipeline explained in more detail in the following sections. The corresponding Python package can be found at `https://github.com/marinadominguez/TBProject/tree/module`.

1.2. Data

The dataset comprises 86 gray-scale WSIs of stained sputum smears. Each WSI has a resolution of 112534×55519 pixels at 20X magnification, where one pixel corresponds to $(0.34\ \mu\text{m})^2$. The WSIs are generated by assembling approximately 1200 microscopic image tiles of 2048×1504 pixels. Based on the estimated number of bacilli in a region of interest within these WSIs, disease severity was graded by a trained pathologist on an integer scale from 0 (very low risk) to 4 (very high risk).

1.3. Report Structure

The report is structured as follows: First, in Sect. 2, an overview of all steps required for object detection in the WSIs is provided. In Sect. 3 and 4, different classification models for distinguishing bacillus versus non-bacillus objects are described, followed by a detailed account of the training procedure for the best-performing model, the CNN. Lastly, the results of our experiments are presented and discussed in Sect. 5 and 6, respectively.

2. Data Preparation and Object Detection

The pipeline described in the following sections iterates over all image tiles of a WSI. First of all, the WSI is loaded into memory and the individual tiles are converted into numpy arrays. For the purpose of a more efficient downstream analysis, the tiles are rescaled to unsigned 8-bit images.

2.1. Image Binarization: Thresholding

To accurately detect stained, bacillus-sized objects, the gray-scale images are first binarised, i.e., regions assumed to belong to the background are set to black while foreground objects are set to white. This is achieved by applying thresholding techniques from the OpenCV library[1], assigning each pixel a binary value based on some threshold. We found that adaptive Gaussian thresholding and locally applying Otsu's thresholding worked best for our data.

For adaptive Gaussian thresholding[2], the specific threshold of a pixel is computed as the Gaussian-weighted sum

[1] `https://docs.opencv.org/4.x/d7/da8/tutorial_table_of_content_imgproc.html`

[2] `cv2.adaptiveThreshold(adaptiveMethod=cv2.ADAPTIVE_THRESH_GAUSSIAN_C)`

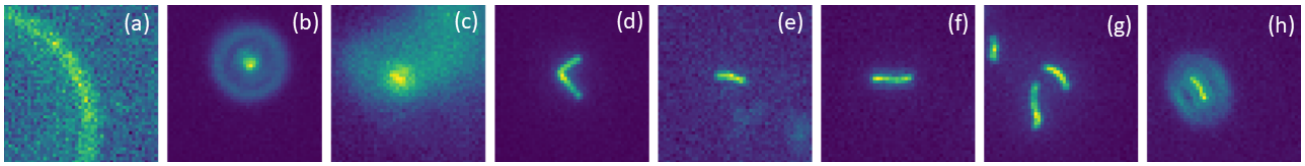

Figure 1. Sample patches obtained from object detection: (a) Diffraction halo, (b) spherical object of undefined nature, (c) possible TB bacilli behind background noise, (d) two entangled TB bacilli. (e)-(f) TB bacilli, (g) group of TB bacilli, (h) blurred TB bacillus.

of pixel intensities in a small neighborhood of fixed size, adjusted by a predefined constant. If the intensity of the pixel is below the calculated threshold, it is set to black (i.e., classified as background); otherwise, it is set to white. Due to high variability in background intensity seen in the WSIs, it proved particularly important to determine the threshold for each pixel individually, rather than predefining a global threshold for the whole tile to separate foreground objects from background.

Otsu's thresholding technique[3] is a classification-based method that seeks to identify the threshold value minimizing the intra-class variance, defined as the weighted sum of the variances of the pixel intensities of the two classes (i.e., background and foreground). Gaussian Otsu's method is an extension of Otsu's method, using the maximum intra-class variance as optimal threshold (Yousefi, 2015). We found that Otsu's thresholding does not perform well when applied to the entire tile due to high variability in background intensity. However, by subdividing the tiles into smaller windows of 16×16 pixels and applying Otsu's method to each window individually, it yielded good results. In windows containing foreground objects (i.e., with a clear difference in intensity between background and foreground), background pixels are set to black and pixels belonging to foreground objects are set to white. In windows of background only, the pixels are randomly set to white or black. Subsequently, all randomly coloured windows are set to black.

Eventually, we chose adaptive Gaussian over Otsu's thresholding due to better computational performance.

2.2. Image Binarization: Postprocessing

After thresholding, we perform morphological opening[4] and closing[5] on the binarized images. Morphological opening is another term for erosion followed by dilation and is used to remove small white background noise. For erosion, just as in 2D convolution, a kernel of predefined shape and size, often referred to as filter, is slid through the images. A pixel in the original image (either white or black) will be considered white only if all of the pixels under the kernel are white, otherwise, it is eroded (i.e., set to black). It is useful for removing small white background noise or detach two connected objects, but it also decreases the size of foreground objects in general. Dilation, basically the opposite of erosion, reverses this unwanted side effect. Morphological closing is reverse of opening, i.e., dilation followed by erosion. It is applied to close small black holes inside foreground objects or to connect two detached objects.

2.3. Object Detection

Subsequently, we run a connected component analysis on the binary images to extract all objects. From this, we obtain the coordinates of the centroids of all objects as well as geometric statistics such as, for instance, height, width and area. Based on these statistics, objects either too small or too big to be bacilli are discarded. Tuberculosis mycobacteria have a length range of 2-4 μm and a width range of 0.2-0.5 μm (Yamada, 2015). Considering the conversion rate of 0.35 μm for pixel, we were able to accurately filter objects by size. Additionally, overlapping bacilli pose a challenge in TB recognition tasks (Sheeba, 2015), therefore the upper bound for size filtering was applied with some degree of flexibility. Coordinates and centroids of the connected components are used to cut out 50×50 images around the remaining foreground objects from the original WSI, which are later fed to the classification models for training and inference. Fig. 1 shows some exemplary results.

2.4. Object Labelling

In order to train the classification models on the cropped images of the detected objects, the latter had to be labelled first, as only an overall severity grade for the WSI had been provided, but no labels for the individual objects in the WSI themselves. To facilitate object annotation, we implemented a simple interactive labelling tool which shows one cropped image at a time and provides two buttons, bacillus and non-bacillus, for annotation. For this study, given that expert annotation is time-consuming and costly, we labelled the objects ourselves. Eventually, a pathologist is to be consulted for critical cases. Around 5000 patches were labeled, with 40% identified as bacilli and 60% as non-bacilli.

[3] `cv2.threshold(type=cv2.THRESH_BINARY+cv2.THRESH_OTSU)`

[4] `cv2.morphologyEx(op=cv2.MORPH_OPEN)`

[5] `cv2.morphologyEx(op=cv2.MORPH_CLOSE)`

3. Classification: Model Description

Three models were implemented for the classification of objects detected in the WSIs into bacilli/non-bacilli, which are described in more detail below.

3.1. Baseline Model

First, since the prototypical shape of tuberculosis bacteria resembles a simple, elongated rod, we implemented a very straightforward base model to assess whether the use of more sophisticated models such as CNNs is worth the additional computational costs with regards to prediction accuracy. Based on the typically slightly elongated shape of tuberculosis bacilli, the following classification criterion was chosen for the base model: If the ratio of major to minor axis, of the ellipse enclosing the object, exceeds the threshold of 1.5, the object is classified as bacillus and as non-bacillus otherwise. Additionally the object has to be of reasonable size, see the reasoning in 2.3.

3.2. Classification I: Support Vector Machine (SVM)

Second, an SVM was implemented with statistics obtained from the connected component analysis. Training was attempted by passing the difference in height and width, to distinguish circular from elongated objects, and the area as parameters. As a last attempt, the lengths of minor and major axes of the enclosing ellipse were passed as parameters. However, the results were unsatisfactory, cf. Sect. 5.

3.3. Classification II: CNN

The proposed CNN model accepts a cropped 50×50 image as input and generates a probability value that is used to predict whether the given input patch displays a TB bacillus or not. The final architecture (cf. Fig. 6) consists of two convolutional layers with (3×3) filter maps activated by Rectified Linear Units (ReLUs) and followed by max-pooling layers with tiles of (2×2) pixels. The output from the series of convolutional layers is fed to a series of three fully connected layers and finally a sigmoid neuron generating a probability from 0 to 1. The object displayed in the input patch is classified as bacillus if the generated output is ≥ 0.5 and as non-bacillus otherwise.

4. Model Training

The proposed CNN model is trained by means of minimizing the binary cross entropy loss defined as

$$L := -\sum_{i=1}^{n} (y_i^{tr} \log(y_i^{pr}) + (1 - y_i^{tr}) \log(1 - y_i^{pr})),$$

with $\{y_i^{tr}\}_{i=1}^n$ denoting the true labels, $\{y_i^{pr}\}_{i=1}^n$ the predicted probabilities and n the number of training samples.

For loss minimization, we used the Adam optimizer, a stochastic gradient method based on adaptive estimation of 1^{st} and 2^{nd} order moments, and a learning rate of 0.001.

To avoid overfitting and improve the robustness of the model, we augment the training data by means of rotation and horizontal/vertical reflection and perform 5-fold cross validation. After splitting our labelled dataset into train and test set, the former is subdivided into five sets of roughly equal size. One of the subsets is used as validation data, whereas the other four subsets are used as training data. Then the model is fitted on the training data, evaluated on the validation data and discarded. This process is repeated five times, each time for 60 epochs with a batch size of 2.500, such that every subset serves as validation set exactly once and we obtain five trained models in total. The overall model accuracy as well as training and validation losses (and other prediction metrics such as precision, recall and f-score) are computed as averages over all five models.

5. Results

5.1. SVM

The SVM showed low accuracy, recall and precision, resulting in a large number of FNs and FPs: it failed to identify many bacilli and misclassified many non-bacillus objects. Although the idea of separating bacilli from non-bacillus objects based on their geometric properties seemed appealing in theory, the method proved unsuccessful. As can be seen in the scatter plot in Fig. 3, it appears that both bacilli and non-bacillus objects are located in the same region of space, making it difficult to separate them. This is true even when using a sigmoid kernel. The SVM performed poorly across all of the cases examined, including when using height, width and area as features, as well as when considering the lengths of the minor and major axes of the enclosing ellipse.

5.2. CNN Model: Training and Inference Results

The training results presented in Fig. 2 indicate that the CNN achieved high accuracy in the early epochs. This result is consistent with other literature on the subject, as reported in (Panicker, 2018), suggesting that identifying bacilli is not a particularly difficult task. However, our model did not achieve the same level of accuracy as other studies on TB bacilli detection that reported up to 97% accuracy.

There could be several reasons for this discrepancy, including the fact that other studies employed more complex model architectures than ours. Moreover, our model was trained on data labelled (possibly incorrectly) by amateurs rather than experts, as explained in more detail in Sect. 6.

As indicated by the average recall, the model is capable

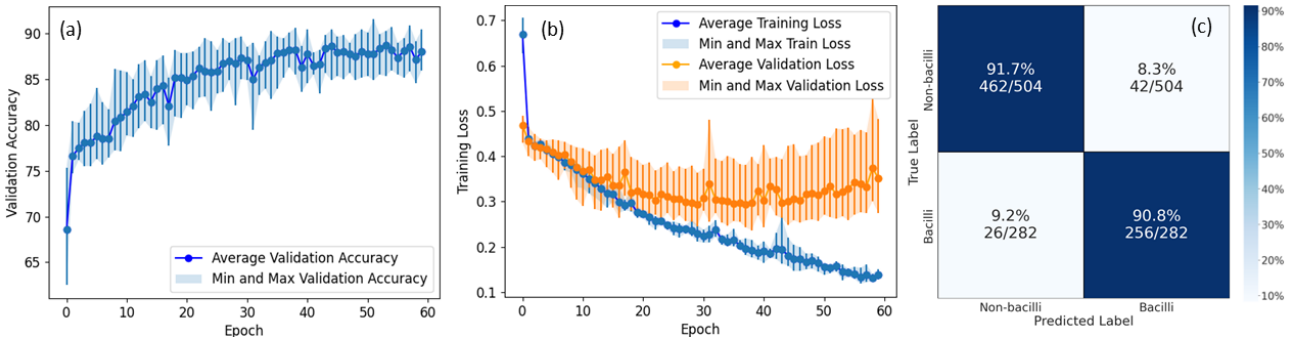

Figure 2. CNN training results averaged over all five models obtained from 5-fold cross-validation. (a) Averaged accuracy over all five validation sets, throughout the epochs. (b) Averaged train and validation losses over all five train and validation sets. (c) Confusion matrix with averaged values over the test set for true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN).

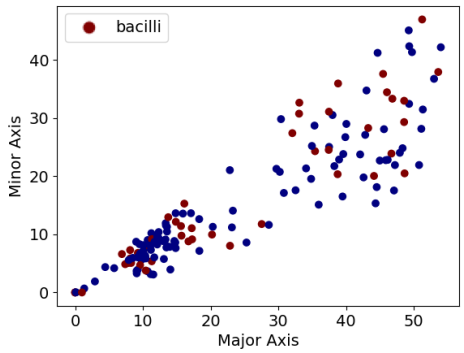

Figure 3. Scatter-plot representing bacilli and non-bacillus objects according to the lengths of the minor and major axes of their enclosing ellipses. Apparently, the two classes cannot be clearly separated, making it hard for an SVM to be a valuable classifier.

of identifying 85% of all bacilli. The average precision indicates that out of all objects classified as bacilli, 90% truly are bacilli. Refer to Tab. 1 for a statistical review of all five models.

	Accuracy	Recall	Precision	F1
Model 1	0.91	0.86	0.90	0.88
Model 2	0.91	0.83	0.94	0.88
Model 3	0.91	0.87	0.87	0.87
Model 4	0.89	0.79	0.95	0.86
Model 5	0.89	0.88	0.81	0.85
Mean	0.90	0.85	0.90	0.87

Table 1. Accuracy, recall, precision and F1 scores on the test set for each of the five models. Accuracy is the ratio of correct predictions to the total number of predictions. Recall calculates the ratio of TPs to the true number of positives, while precision is the ratio of TPs to the total number of positive predictions.

Finally, we run our pipeline on all WSIs to evaluate whether the predicted bacilli counts correlate to the expert severity grades. The results are tabulated in Tab. 2 and visualized in Fig. 4 as violin plots for each severity grade individually.

	Severity				
	0	**1**	**2**	**3**	**4**
Size	9	45	15	11	6
Mean	2.223	2.262	2.813	32.818	110.214
SD	2.249	4.689	2.556	26.550	52.947
Size	9	**43**	15	**13**	6
Mean	2.223	**1.783**	2.813	**31.215**	110.214
SD	2.249	**1.326**	2.556	**24.716**	52.947

Table 2. Initial mean and standard deviation (SD) of bacilli count vs. severity grade in the upper part of the table. Corrected mean and SD after moving outliers in the bottom part. The total number of WSIs per class is denoted as size.

6. Discussion and Conclusion

6.1. SVM

We have identified three challenges that may account for the poor performance of the SVM: First, many bacilli are often surrounded by round-like background noise (cf. Fig. 7 and Fig. 8), which results in the enclosing ellipses being more circular than expected for bacilli; second, bent bacilli (such as shown in Fig. 1d)) might also be enclosed in rather circular ellipses; lastly, our own inconsistent labeling of relatively short and chunky objects, cf. 6.2. Given these challenges, it is reasonable to assume that the geometric features alone are not enough to correctly classify bacilli and that pixel intensity plays an important role in this process. Overall, this suggests that a CNN may be necessary to achieve more accurate predictions.

6.2. CNN

It was found that the objects corresponding to FPs and FNs, respectively, look quite similar (cf. Fig. 7, 8): they do not show the prototypical, elongated shape of TB bacilli, but are rather short and chunky. The reason why the CNN model

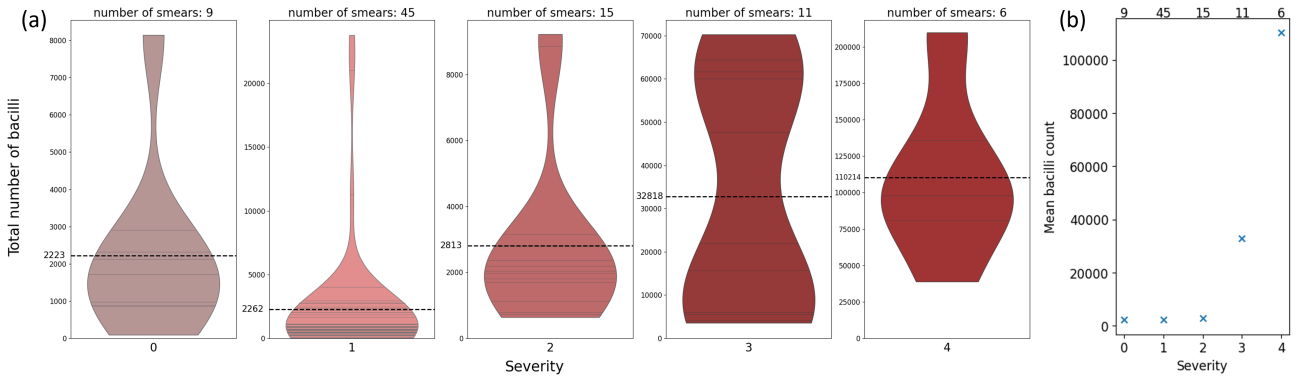

Figure 4. (a) Violin plots of bacilli counts for each severity class: each horizontal represents the bacilli count for a specific WSI, dashed lines correspond to mean bacilli counts. (b) Mean bacilli count per severity class with the number of smears per class indicated on top.

cannot classify all objects correctly could therefore lie in the inconsistent labeling of these (for non-experts) non clearly assignable objects. Some were labelled as bacillus, some as non-bacillus, such that it is no surprise that the model is just as indecisive when it comes to classifying these critical cases. In Fig. 7 and 8, all misclassified objects from the test set are depicted for a better understanding of this issue.

Regarding the bacilli counts of all 86 smears, we see that there is basically no difference in mean values between class 0 and 1 smears, respectively. At this point, more information on the accuracy of the annotations is needed to better evaluate this issue.

Moreover, we observe that the difference in mean values between class 1 and 2 smears, respectively, is quite small. However, we identified two outliers in class 1 (indicated by the long, narrow tail of the class 1 violin plot in Fig. 4), showing significantly higher bacilli counts than all other samples in class 1. Both the total bacilli counts and the mean bacilli counts per tile are comparable to class 3 smears. We suspect that these samples might have been wrongly graded by the expert and thus should be inspected again.

When correcting for these two outliers, that is, when moving the corresponding samples from severity 1 to severity 3, we see in Tab. 2 that the mean and standard deviation of the bacilli count for severity class 1 significantly decrease.

In an attempt to explain what seems to be a wrongful severity classification, we examined the spatial distribution of bacilli in the WSIs. As mentioned earlier, part of an expert's workflow in assessing the severity of a smear sample is to zoom in on regions of interest (RoI) and count the number of bacilli in these RoIs. However, when choosing these RoIs, the expert might be deceived by noisy signals from non-bacillus objects.

We conclude that a smear with numerous dense accumulations of bacilli is potentially more easily graded correctly by the pathologist as RoIs are more readily identified. In contrast, a sample might be wrongfully graded lower if the bacilli are distributed evenly across the smear, resulting in generally low bacilli counts, as exemplarily illustrated in Fig. 5 for one of the aforementioned outliers in class 1. We hypothesize that the distribution of bacilli across the smear plays a critical role in severity classification.

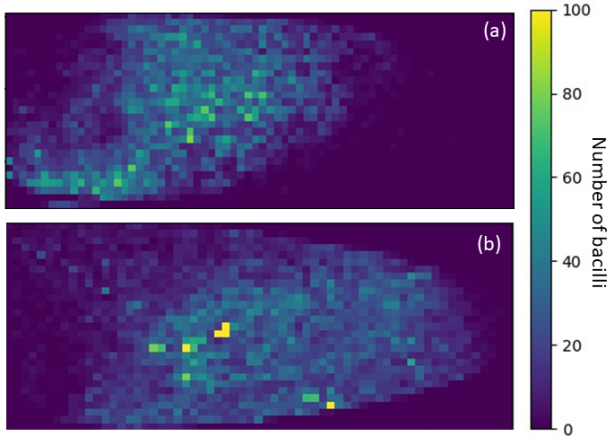

Figure 5. Heatmaps relating bacilli counts to spatial dependencies for smears (a) 2151_26_3 belonging to severity class 3 (high) and (b) 2162_82_1 belonging to class 1 (low), showing similar bacilli counts. The maps support our hypothesis that the visual impact of (a) is much greater than that of (b), with (b) having some high bacilli counts locally but low counts overall and (a) showing a more even distribution of higher bacilli counts across the smear.

6.3. Outlook

We are currently collaborating with a team of software engineers and data analysts developing an active learning platform. For object detection and segmentation, they employ a well-established R-CNN model. In an effort to enhance the efficiency of our own pipeline and contribute to the advancement of their platform, we seek to compare the results obtained by applying their model with those obtained by running our pipeline.

References

Hailemariam, M. Evaluation of laboratory professionals on afb smear reading at hawassa district health institutions, southern ethiopia. *International Journal of Research Studies in Microbiology and Biotechnology (IJRSMB)*, 2018.

Panicker. Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods. *biocybernetics and biomedical engineering 38 (2018) 691–699*, 2018.

Sheeba, F. Detection of overlapping tuberculosis bacilli in sputum smear images. In *7th WACBE World Congress on Bioengineering*, 2015.

WHO. International standards for tuberculosis care. In *(ISTC)*, 2006.

Yamada, H. Structome analysis of virulent mycobacterium tuberculosis, which survives with only 700 ribosomes per 0.1 fl of cytoplasm. In *Silvana Allodi, Academic Editor*, 2015.

Yousefi, J. Image binarization using otsu thresholding algorithm. In *Image Binarization using Otsu Thresholding Algorithm*, University of Guelph, Ontario, Canada, 2015.

A. Supplementary Figures

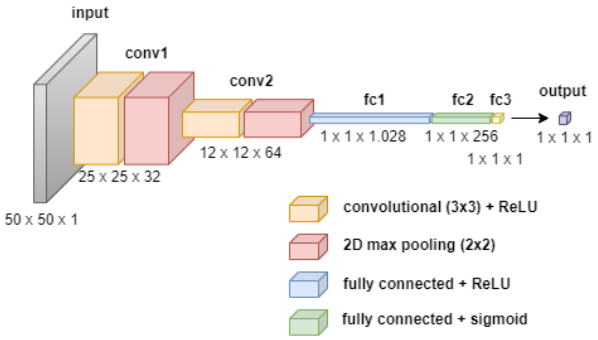

Figure 6. CNN model architecture.

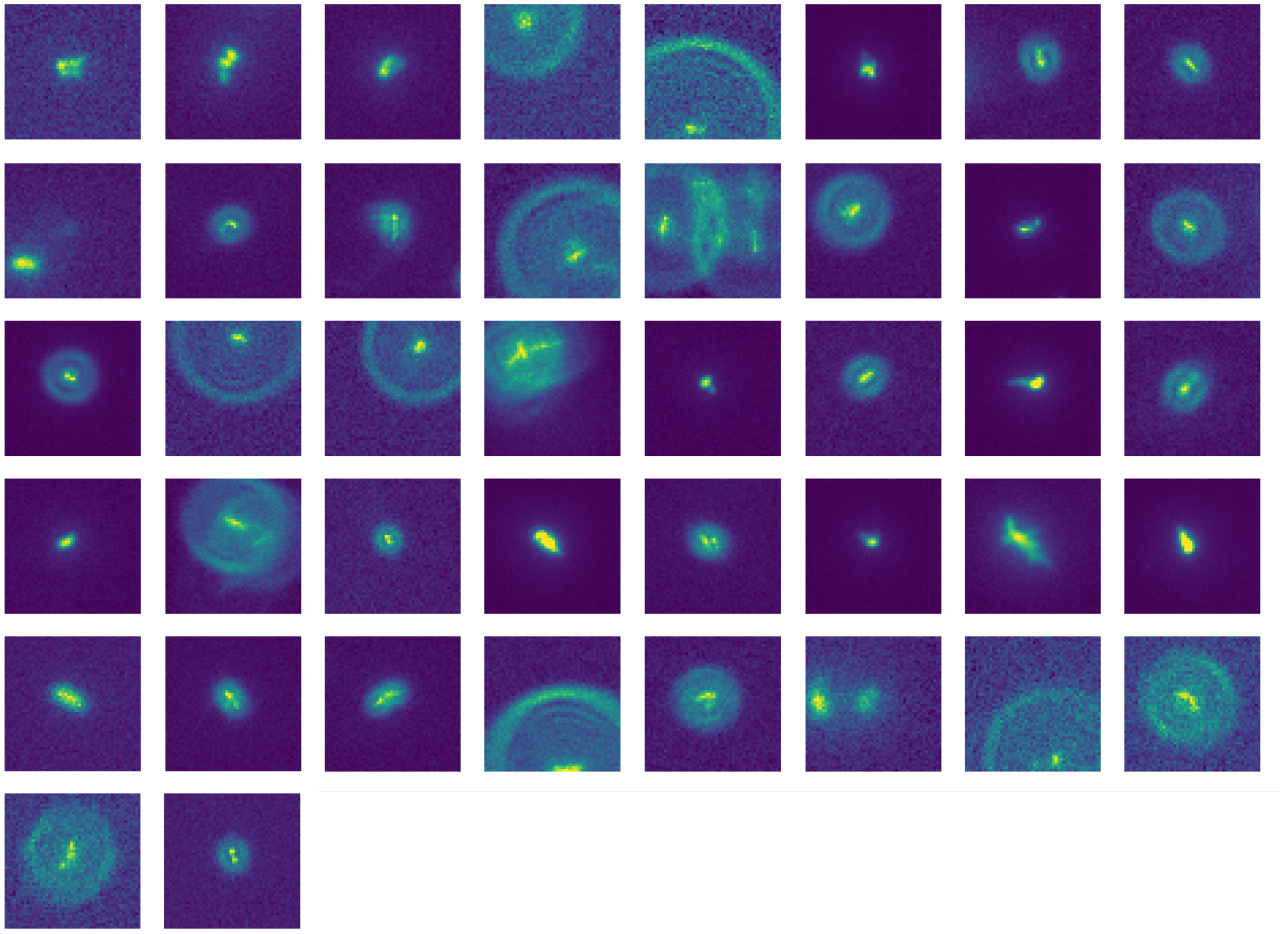

Figure 7. All 42 FPs, i.e., objects wrongly classified as bacilli and therefore associated with the upper right tile of the confusion matrix in Fig. 2c), belonging to the dataset used for testing after 5-fold cross-validation of the CNN.

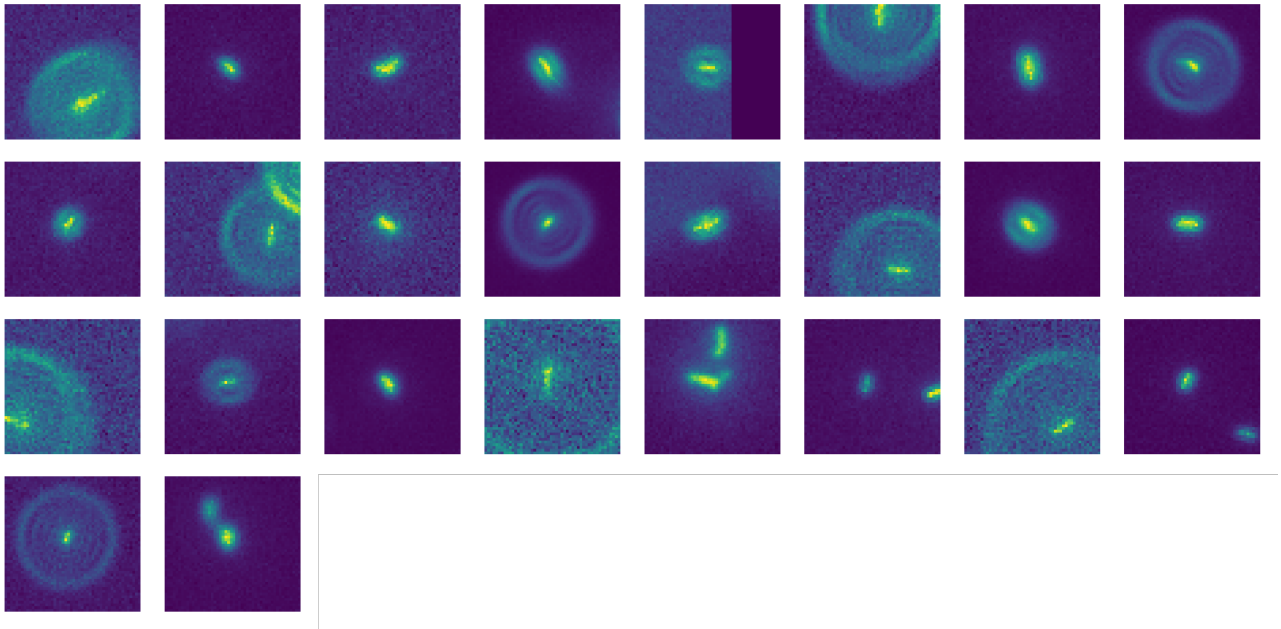

Figure 8. All 26 FNs, i.e., objects wrongly classified as non-bacillus and therefore associated with the bottom left tile of the confusion matrix in Fig. 2c), belonging to the dataset used for testing after 5-fold cross-validation of the CNN.