Estimating Bacterial Load in FCFM Imaging

Citation for published version:
Seth, S, Akram, A, Dhaliwal, K & Williams, CKI 2017, Estimating Bacterial Load in FCFM Imaging. in *Medical Image Understanding and Analysis. MIUA 2017.*. Communications in Computer and Information Science, vol. 723, Springer, Cham, pp. 909-921, Medical Image Understanding and Analysis (MIUA 2017), Edinburgh, United Kingdom, 11/07/17. https://doi.org/10.1007/978-3-319-60964-5_79

Digital Object Identifier (DOI):
10.1007/978-3-319-60964-5_79

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Medical Image Understanding and Analysis. MIUA 2017.

Estimating Bacterial Load in FCFM Imaging

Sohan Seth[1], Ahsan R. Akram[2], Kevin Dhaliwal[2], and Christopher K. I. Williams[1] ⋆

[1] University of Edinburgh, School of Informatics, Edinburgh, EH8 9AB, UK
[2] University of Edinburgh, Queens Medical Research Institute, MRC Center for Inflammation Research, Pulmonary Molecular Imaging Group, Edinburgh, EH14 4TJ, UK

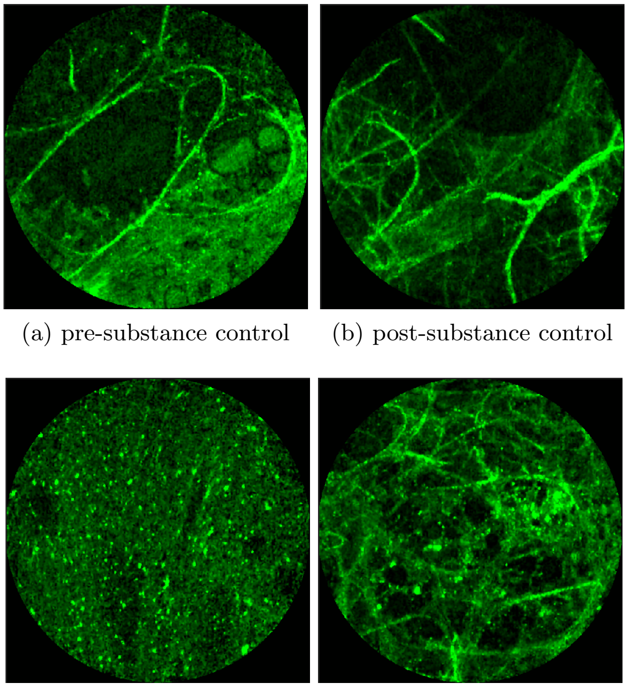
Abstract. We address the task of detecting bacteria and estimating bacterial load in the human distal lung with fibered confocal fluorescence microscopy (FCFM) and a targeted smartprobe. Bacteria appear as bright dots in the image when exposed to a smartprobe, but they are often difficult to detect due to the presence of background autofluorescence inherent to human lungs. In this study, we create a database of annotated image frames where a clinician has labelled bacteria, and use this database for supervised learning to build a suitable bacterial load estimation software.

1 Introduction

Fibered confocal fluorescence microscopy (FCFM) is a popular method for in vivo imaging of the distal lung, and has recently gained prominence in investigating the presence of bacteria using targeted *smartprobe* [1]. Smartprobes are specialized molecular agents introduced in the imaging area to make the bacteria fluoresce. FCFM imaging works by recording the number of emitted photons at each core of a optical fiber bundle. The photon counts are later translated into pixel intensities to achieve a 'smooth' image. Since the diameter of a bacterium is usually smaller than the width of the fibre core as well as the gap between two consecutive fiber cores, it appears as a high intensity dot in the image frame and, tends to 'blink' on and off in consecutive image frames due to movement of the apparatus.

Figure 1 shows examples of FCFM image frames without (top) and with (bottom) bacteria. In general, human lungs display a mesh-like structure due to autofluorescence of connective tissues (elastin) with or without the presence of smartprobe whereas bacteria appear as dots in the image frame when exposed to smartprobe (Figure 1c). However, one can observe bacteria-like dots in the absence of the smartprobe as well due to noise (Figure 1a). Additionally, if bacteria are present, they are usually easy to detect when the elastin structure

⋆ S.S., A.A., K.D., and C.W. would like to thank Engineering and Physical Sciences Research Council (EPSRC, United Kingdom) Interdisciplinary Research Collaboration grant EP/K03197X/1 for funding this work. A.A. is supported by Cancer Research UK.


(a) pre-substance control (b) post-substance control

(c) post-substance cases

Fig. 1: FCFM image frames w/ or w/o smartprobe in control or case group.

is not prominent (Figure 1c-left), but it becomes more difficult to discriminate them from the background in the presence of elastin structure (Figure 1c-right).

We address the task of estimating bacterial load in a FCFM image frame. We consider a formal approach to the bacteria detection problem by explicitly annotating bacteria in image frames with the help of a clinician, and use this knowledge in a supervised learning set-up to learn a classifier that assigns a probability value to each pixel of the image of a bacterium being present. Estimating bacterial load can generally be framed as a *learning-to-count* [2] problem where we need to count the bacteria. However, although the learning-to-count framework usually bypasses the problem of detection before counting, we suggest detecting the object to allow the clinicians to see where the bacteria are appearing, ideally while performing the bronchoscopy. That said, our method bears resemblance with the established learning-to-count approaches with the major difference being that we learn a classifier to predict a probability value at each pixel whereas the other approaches learn a regressor to predict the 'count density' (intensity value) at each pixel (see Section 5).

Our final goal is to build a real time system to assist clinicians in estimating bacterial load while performing bronchoscopy. We suggest using a multi-resolution spatio-temporal template matching scheme using radial basis func-

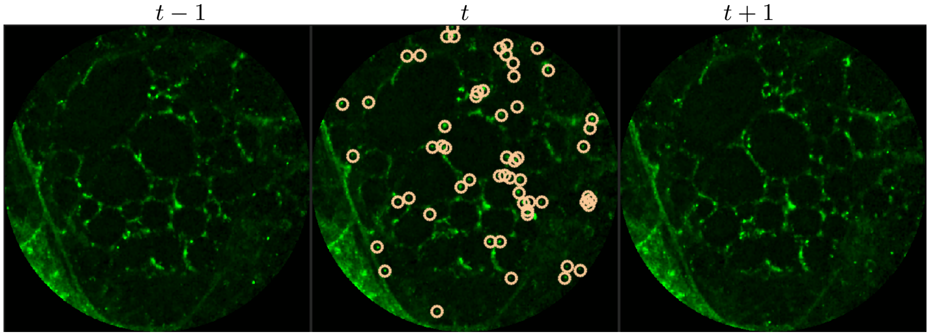


Fig. 2: FCFM image frame with bacteria annotated by a clinician in circles.

tions. Spatio-temporal analysis allows better capturing the ‘blinking’ effect, whereas multi-resolution analysis allows better discrimination between bacterial dots and elastin structure which may appear as series of dots. We use normalized intensity values around each pixel as features, which enables fast implementation of our method using 2D-convolution. We apply this method in estimating bacterial load in FCFM videos with and without bacteria (case and control), before and after applying the smartprobe, and show that we successfully infer low bacterial load in the control or the pre-substance videos and high bacterial load in the post-substance videos from the cases.

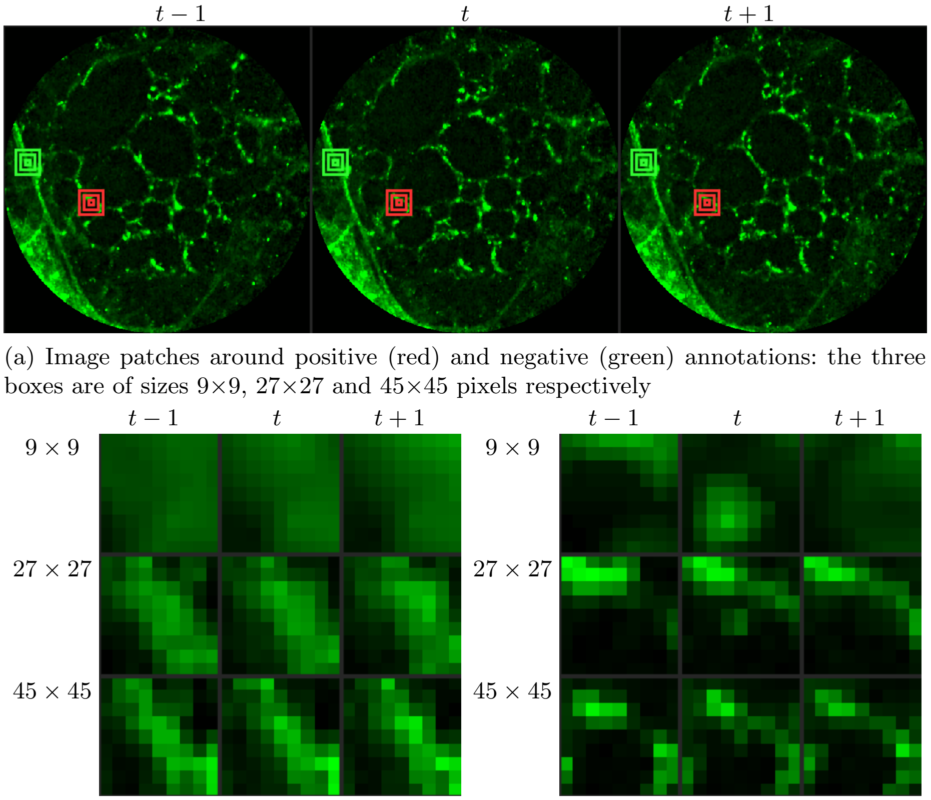
2 Dataset

2.1 Collection

In vivo imaging was performed in 6 patients (3 cases with gram-negative bacteria where a bacterial signal was detected and 3 cases with gram-positive bacteria (controls) where signal was not detected) where measurements were taken before and after administration of a gram-specific bacterial specific smartprobe (pre- and post-substance measurements respectively). Each measurement is a FCFM video (12 frames per second [4]) that were manually cleaned to ensure alveolar imaging by removing motion blur, air imaging, bronchi etc, and the remaining clean frames (∼500 in each video) were used for this study.

2.2 Annotation

The learning-to-count problem can be addressed in a variety of different annotation scenarios among which two widely used ones are the dot-annotation and the count-annotation [5]. While dot-annotation provides the location of where an object appear in the image, count-annotation only provides the number of objects in an image without explicitly revealing the locations. We use dot-annotations since they are more informative given the small size of the objects we are trying to count.



(a) Image patches around positive (red) and negative (green) annotations: the three boxes are of sizes 9×9, 27×27 and 45×45 pixels respectively

(c) Features for a negative sample

(c) Features for a positive sample

Fig. 3: Illustration of spatio-temporal and multi-resolution feature extraction.

We chose 144 image frames from 12 videos such that 72 of them come from videos without bacteria (8 frames from 9 videos that are either in the control group or in the pre-substance group), and 72 of them come from videos with bacteria (24 frames from 3 videos that are in the post-substance case group). Along with the 144 image frames, the previous and next frames corresponding to those frames were extracted as well, and the clinician was allowed to toggle between the previous and next frame to annotate a bacterium in the current frame. Thus, a bacterium was identified in a spatio-temporal context. Figure 2 shows an example of annotated frame along with respective previous and next frames to demonstrate the blinking effect.

We observe that although we might encounter false positives, i.e., bacteria are annotated in either control group or pre-substance group, the clinician successfully annotates more bacteria in the frames where bacteria should exist, i.e., post-substance case group. For training purposes we only considered positive annotations from the post-substance case group.

3 Method

3.1 Preprocessing

To reduce the effect of noise and spurious intensity values we adjust the lowest and highest values of each image frame individually as follows: for each image frame, first, we set any intensity values below 1% quantile to 0, and next, we set any intensity values above 99% quantile to the respective 99% quantile values[3].

To allow supervised learning, we associate a feature vector $\mathbf{x}_p$ and label y_p to each pixel in the image and group the pixels in positive ($y_p = 1$) and negative samples ($y_p = 0$). We use the intensity values over a patch around a pixel as feature vector. However, we observe that image patches can vary significantly in contrast and therefore, we normalize them by the total intensity of the patch as follows: given an image patch $\{\tilde{x}^p_{ij}\}^w_{i,j=1}$ of size $w \times w$ around pixel p, we normalize the patch as $x^p_{ij} = \tilde{x}^p_{ij} / \sum_{ij}(\tilde{x}^p_{ij} + \epsilon)$ where $\epsilon = 1$ is added to suppress noisy image patch, i.e., $\tilde{x}^p_{ij} \approx 0$. Thus, our basic feature vector is $\mathbf{x}_p = (x^p_{11}, \ldots, x^p_{ij}, \ldots, x^p_{ww})$. $y_p = 1$ if p has been annotated by the clinician and 0 otherwise.

For positive labels ($y_p = 1$), we pool all image patches around the pixels annotated by the clinician. For negative labels ($y_p = 0$), we extract equispaced image patches over a grid (15×15 pixels apart). If any of the 'negative' image patches have a bacterium, they were assigned to the positive samples. Along with the original image patches, we performed data augmentation by rotating each image patch by 90, 180 and 270 degrees and adding them to the pool of samples. This results in about 18,000 positive samples and 540,000 negative samples. Notice that our classes are severely imbalanced: we use undersampling of the negative class to maintain a class balance while training a classifier.

For temporal analysis, we extract patches around the same pixel from the previous and the next image frame, normalize them individually and concatenate them to the feature vector from the current frame. For multi-resolution analysis, we extract larger image patches and 'downsample' them to the size of the smallest patch. This is done to allow equal importance over each resolution. These patches are normalized individually and concatenated to the feature vector. We 'downsample' the larger image patch, usually 3 or 5 times larger than the smallest image patch, by averaging over a 3×3 or 5×5 window around the pixel (in the patch) being downsampled. Figure 3 shows examples of positive and negative image patches. We also observe that i) dot annotations can be noisy in the sense that the pixel with a bacterium might not be centered, and ii) larger patch captures the context whereas smaller patch captures the object.

3.2 Supervised learning

Our approach is to assign a probability at each pixel of a bacterium being present. Therefore, we essentially solve a classification problem from $\mathbf{x}_p$ to y_p. For simplicity and ease of implementation, we suggest *logistic regression* as a baseline

[3] This also increases the dynamic range, and thus helps the clinicians in annotating.

method. However, we observe that it performs rather poorly. We then suggest *radial basis functions network* as an alternative, and show that it improves the performance significantly.

Logistic regression: We solve a L_2 regularized logistic regression to learn a linear classifier, i.e.,

$$\min_{\mathbf{w},b} -\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\big(y_p\log\sigma(\mathbf{w}^\top\mathbf{x}_p+b)+(1-y_p)\log(1-\sigma(\mathbf{w}^\top\mathbf{x}_p+b))\big)+\frac{\lambda}{2|\mathcal{P}|}||\mathbf{w}||^2$$

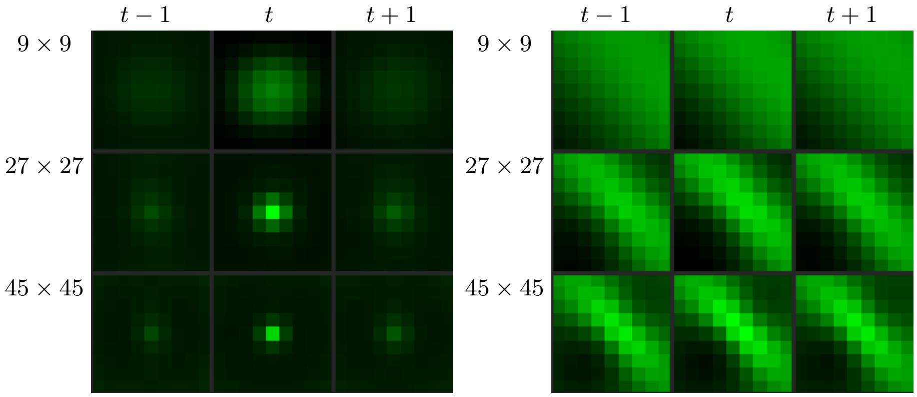
where σ is the sigmoid function and $\mathcal{P}$ is the set of all pixels in either positive or negative samples, and λ is the regularization parameter. We set $\lambda = 0.01$. Since we have class imbalance, we resample the negative samples to maintain class balance.

We test the performance of the baseline method with three different choices of features, i) with 9×9 image patches extracted from the current frame, i.e., $\mathbf{x}_p \in \mathbb{R}^{81}$, ii) with 9×9 image patches extracted from the current as well as previous and next frame, i.e., $\mathbf{x}_p \in \mathbb{R}^{81\times 3}$, and iii) with 9×9 and 45×45 image patches extracted from the previous, current and future frames, i.e., $\mathbf{x}_p \in \mathbb{R}^{81\times 6}$. We expect (ii) to perform better than (i) since it captures the blinking effect, whereas (iii) to perform better than (ii) since it provides more contextual information around a bacterial dot.

Radial basis function network: We learn a radial basis function network (RBF) [6] as a nonlinear classifier where the input vector $\mathbf{x}_p$ is first transformed through a set of nonlinear transformations, or radial basis functions, $\phi_i(\mathbf{x}_p)$ before being used as the input to the classifier, i.e., we solve the same problem as in logistic regression but replace the original feature vector $\mathbf{x}_p$ with the output of the radial basis functions $\{\phi(\mathbf{x}_p - \mathbf{c}_i)\}_{i=1}^{64}$ where $\mathbf{c}_i$ are centers of the radial basis functions. For ϕ we used Gaussian kernel with bandwidth set to median intersample distance. We chose the centers of the radial basis functions using k-means [4]. Since we have many fewer positive samples than negative, we chose 16 centers from the positive samples and 48 centers from the negative samples. Figure 4 shows examples of centers chosen from positive and negative samples. After choosing the centers, we perform logistic regression from 64 dimensional feature vector to the class label to learn the weight vector of the network. Notice that for selecting the centers we used the entire[4] negative set rather than undersampled set that we use for learning the weight vector.

We test the performance of the RBF network with two different choices of features, i) with 9×9 image patches extracted from the current as well as previous and next frame, i.e., $\mathbf{x}_p \in \mathbb{R}^{81\times 3}$, and iii) with 9×9, 27×27 and 45×45 image patches extracted from the previous, current and future frames, i.e., $\mathbf{x}_p \in \mathbb{R}^{81\times 9}$. We expect (ii) to perform better than (i) since it captures more contextual information around a bacterial dot. Additionally, we expect RBF network to perform better than linear logistic regression since it effectively uses 64 'templates' than one (weight vector in the linear classifier).

[4] after cross-validation split.



(a) Positive center

(b) Negative center

Fig. 4: Examples of centers of RBF network.

3.3 Postprocessing

The classifier returns a probability value at each pixel. These probability values are then thresholded, and pixels that exceed this threshold value are counted after non-maximum suppression [8] to estimate bacterial load.

3.4 Evaluation method

Performance metric: We compare different methods in terms of the precision-recall curve for varying thresholding of the probability map before non-maximum suppression. Given a contingency table of false positives, true positives, and false negatives, precision P and recall R are defined as follows.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

Given the locations of pixels where a bacterium has been detected and the annotations by the clinician, we draw a disk of radius r around the annotations, and if a detection exists within the disk then it is declared to be a match as in [7].

- If a detection does not match any ground truth annotation, then it is declared to be a false positive.
- If a ground truth annotation does not match any detection then it is declared to be a false negative.
- If a detection exclusively matches a ground truth annotation and vice versa, then the detection is declared to be a true positive.
- If multiple detections exclusively match a ground truth annotation, then one of them is declared to be a true positive while the rest are declared to be false positives.

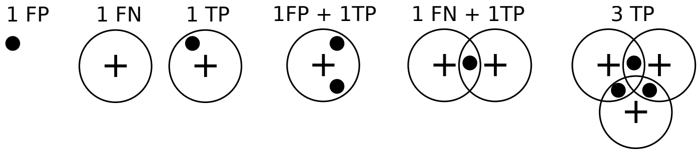


Fig. 5: Illustration of true positives, false negatives and false positives. + are ground truth annotations, • are detections and ∘ are disks of radius r around ground truth annotations.

- If multiple ground truth annotations exclusively match a detection, then the detection is declared to be a true positive while the rest of the ground truth annotations are declared to be false negatives.
- If multiple detections and multiple ground truth match each other non-exclusively then their assignments are resolved greedily. Notice that the assignment might not be optimal.

Figure 5 illustrates these different situations. We set $r = 4$ pixels.

Precision recall curves: We utilize cross-validation to test the performance of the methods. To elaborate, we divide 144 image frames in 5 groups $\mathcal{C}_i, i = 1, 2, 3, 4, 5$. To learn the probability map for image frames in group $\mathcal{C}_i$ we train a classifier with positive and negative samples extracted from the remaining four groups $\mathcal{C}_j, j \neq i$. After repeating this process for each group, these probability maps are treated as the output of the classifiers, and the precision-recall curve is estimated by adding the false positive, false negative and true positive values over all image frames.

4 Result

4.1 Cross-validation

Figure 6 shows the precision-recall curves from all learning methods and feature extraction strategies. We observe the following,

- RBF network performs better than linear logistic regression
- Using temporal information enhances performance. This can be seen from the performance of logistic regression with and without temporal information.
- Using multiple resolution enhances performance. This can be seen from the performance of RBF with and without multiple resolutions.

This follows the intuition behind the use of spatio-temporal multi-resolution analysis. Figure 7 compares annotations by the clinician and detections made by the learning algorithms on one of the validation image frame. We observe more true positives for RBF with multi-resolution spatio-temporal analysis.

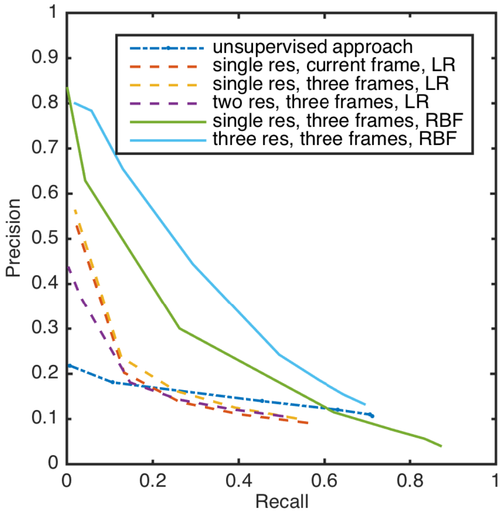


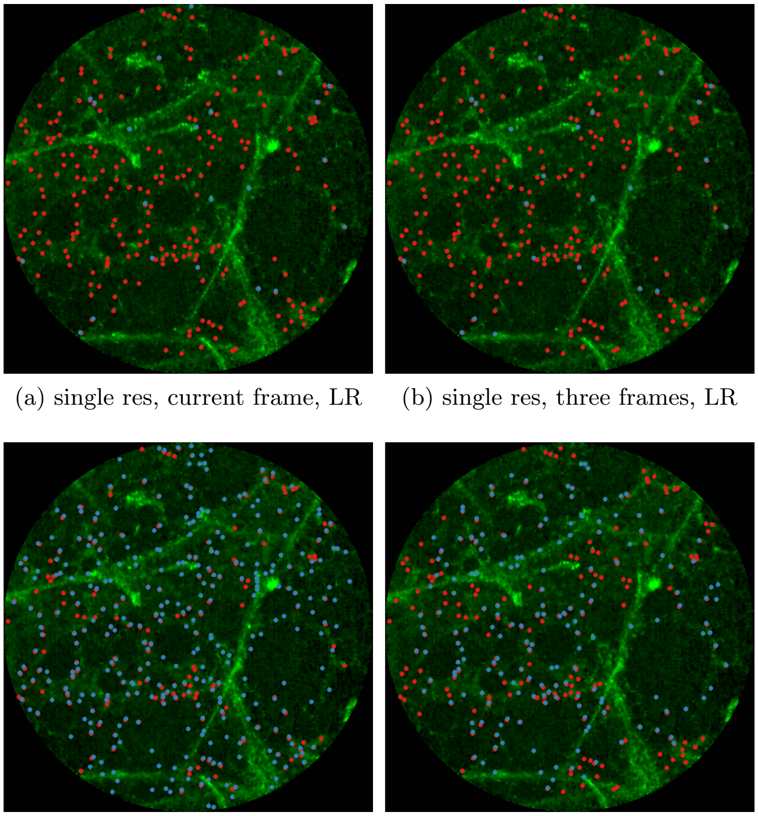
Fig. 6: Precision recall curves for different methods in detecting bacteria.

4.2 Case vs. control study

We use the RBF network learned using spatio-temporal multi-resolution analysis from one of the cross-validation set to estimate bacterial load in each image frame of the entire dataset (12 videos). Figure 8 shows the estimated bacteria load. As expected, we observe that the estimated bacterial load shows significant change in the case group as opposed to control group. Notice that there might exist image frames in post-substance case videos which do not image a region with bacteria, thus resulting in low bacteria count.

5 Related work

Estimating bacterial load has previously been addressed in an unsupervised learning set-up in our group where difference of Gaussians features were used to enhance the dots in the image. We assessed the precision-recall curve for the unsupervised approach in the context of the acquired ground truth annotations. This is presented in Figure 6. We observed that the proposed supervised approach outperforms the unsupervised approach significantly. We also demonstrated the performance of the proposed method using fold change of bacterial load in control and case groups before and after the application of the smartprobe. In the control group, the estimated bacterial load did not change much when exposed to the smartprobe whereas in the case group the estimated bacterial load showed a significant change. Although the results showed the desired performance in case vs. control group in pre- vs. post-substance measurements, the method tended to overestimate bacterial load in an image frame.



(a) single res, current frame, LR (b) single res, three frames, LR

(c) single res, three frames, RBF (d) multi res, three frames, RBF

Fig. 7: Ground truth annotations (red) and detected bacteria (blue)

Arteta *et al.* tackle the problem of object counting as density estimation, known as *density counting*, where integrating the resulting density gives an estimate of the object count [2]. The authors extract image patch based feature vector[5] $\mathbf{x}_p$ at each pixel p and construct a dictionary (512 elements) via k-means with l_2 distance. Each feature vector is then mapped to a binary vector $\mathbf{z}_p$ via one-hot encoding based on its smallest l_2 distance to the dictionary elements. The authors suggest learning a regression model from $\mathbf{z}_p$ to y_p. This, however, may lead to overfitting due to matching the density value at each pixel. Instead the authors suggest matching the densities such that they should match when integrated over an extended region which leads to a smoothed objective function, and is equivalent to a spatial Gaussian smoothing[6] applied to each feature vector

[5] Contrast-normalized intensity values at each pixel in the patch after rotating the patch by the dominant gradient.

[6] The authors suggest using the width of the kernel to be greater than half of the typical object diameter.

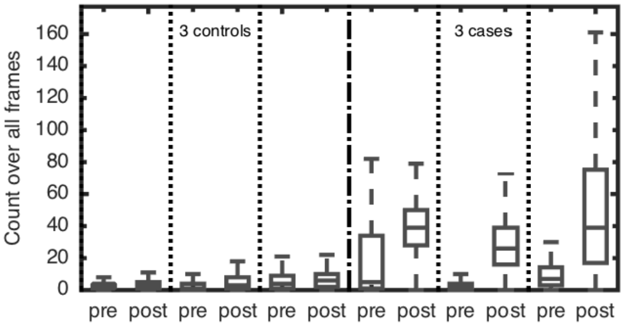


Fig. 8: Change of bacterial load in 3 controls and 3 cases, pre- and post-substance as assessed by RBF network with spatio-temporal and multi-resolution features.

and response vector before applying ridge regression. Essentially, the algorithm constructs a set of templates, and each template is assigned a probability value corresponding to the learned weight vector. Given a test image patch first the closest dictionary element is found, and the related probability value is assigned to the patch. Our approach is similar to [2] in the sense that we work with dot annotations. However, we learn a classifier instead of a regressor, and explicitly detect where each bacterium appears.

Following the work from [2], Arteta *et al.* address learning-to-count penguins in natural images [3]. The authors have access to around 500 thousands images, and each image has been dot-annotated by a maximum of 20 annotators. The authors use multi-task learning through convolutional neural network to, 1) separate foreground containing penguins from background, 2) estimate count density within foreground region, and 3) estimate variability in annotations in foreground region [3, Figure 2]. We do not use a convolutional neural network due to lack of training images frames.

6 Discussion

We address the task of estimating bacterial load in FCFM images using targeted smartprobe. We create a database of annotated image frames where a clinician has dot-annotated bacteria, and use this database to train a radial basis function network for estimating bacterial load. We show that spatio-temporal features along with multi-resolution analysis can better predict the bacterial load since they capture the ‘blinking’ effect of the bacteria, and provide better contextual information about the bacterial dot. An attractive aspect of the suggested method is that it can be implemented efficiently using convolution since it estimates the inner product between image patches to compute the outcome of the radial basis functions. We apply the suggested method in estimating bacterial load at each image frame of FCFM videos from control and case group. We observe significant fold change in the case videos before and after introducing smartprobe, which is not observed in the control group.

While annotating ground truth, it is highly likely that the annotator makes mistakes: (s)he can either falsely annotate a bacterium when it is noise, or simply misses annotating a bacteria due to their overwhelming numbers in each frame. These types of error are common in any annotation process, but it might have a more severe impact on learning since our objects are ‘dots’: while mis-annotation in other datasets such as penguin or crowd is mostly due to occlusion or objects being far away from the camera (for both cases the features of the object pixels are different from positively annotated objects), for bacteria datasets this is not true. Therefore, wrongly annotated bacterium can assign different labels to same feature vector.

We observe that even with more contextual information, elastin structure is often misinterpreted as bacterial dots. The proposed method can be extended further using *hard negative mining*, i.e., explicitly annotating dots which are misclassified as bacteria but actually part of the elastin structure. This definitely requires extra effort from the annotator, but should improve the performance of the classifier. A potential problem with this approach is that it needs to be revised when more diverse elastin structure becomes available, e.g., we do not have patients with granular structure (which arises from smoking). We would need case and control data for this situation in order to cover this scenario. We plan to explore these extensions and limitations as more clinical data becomes available, with the goal of building a robust clinical system.

References

1. Akram, A.R., Avlonitis, N., Lilienkampf, A., Perez-Lopez, A.M., McDonald, N., Chankeshwara, S.V., Scholefield, E., Haslett, C., Bradley, M., Dhaliwal, K.: A labelled-ubiquicidin antimicrobial peptide for immediate in situ optical detection of live bacteria in human alveolar lung tissue. Chem Sci 6, 6971–6979 (2015)
2. Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Interactive object counting. In: European Conference on Computer Vision (2014)
3. Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the wild. In: European Conference on Computer Vision (2016)
4. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (2007)
5. von Borstel, M., Kandemir, M., Schmidt, P., Rao, M.K., Rajamani, K.T., Hamprecht, F.A.: Gaussian process density counting from weak supervision. In: European Conference on Computer Vision I (2016)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edn. (1998)
7. Mandula, O., Šumanovac Šestak, I., Heintzmann, R., Williams, C.K.I.: Localisation microscopy with quantum dots using non-negative matrix factorisation. Opt. Express 22(20), 24594–24605 (2014)
8. Neubeck, A., Gool, L.V.: Efficient non-maximum suppression. In: International Conference on Pattern Recognition. vol. 3 (2006)