Environmental Microorganism Classification Using Conditional Random Fields and Deep Convolutional Neural Networks

Sergey Kosov[a], Kimiaki Shirahama[a], Chen Li[b], Marcin Grzegorzek[a,c]

[a] *Research Group for Pattern Recognition, University of Siegen, Germany*
[b] *Sino-Dutch Biomedical and Information Engineering School, Northeastern University, China*
[c] *Faculty of Informatics and Communication, University of Economics in Katowice, Poland*

Abstract

The labeling of Environmental Microorganisms (EM) which help decomposing pollutants, plays a fundamental role for establishing sustainable ecosystem. We propose an environmental microorganism classification engine that can automatically analyze microscopic images using *Conditional Random Fields* (CRF) and *Deep Convolutional Neural Networks* (DCNN). First, to effectively represent scarce training images, a DCNN pre-trained for image classification using a large amount of data is re-purposed to our feature extractor that distils pixel-level features in microscopic images. In addition, pixel-level classification results by such features can be refined using global features that describe the whole image *in toto*. Finally, our CRF model localizes and classifies EMs by considering the spatial relations among DCNN-based features, and their relations to global features. The experimental results have shown 94.2% of overall segmentation accuracy and up to 91.4% mean average precision of the results.

Keywords: Environmental Microorganism; Conditional Random Fields; Global Feature Extraction; Image Classification; Image Segmentation

1. Introduction

Recent decades, due to industrialization, we can observe a growing number of pollutants like waste water entering the human environment. This increases

the risks of serious diseases, such as cancer. Instead of using chemicals to eliminate such pollutants, a more harmless approach would be taking advantage of the natural consumption of *Environmental Microorganisms* (EM) [1]. EMs are microscopic organisms living in natural and artificial environments (*e.g.*, forests and farmlands), which are useful for cleaning environments. For example, *Actinophrys* can digest the organic waste in sludge and increase the quality of fresh water, whereas *Rotifera* can decompose rubbish in water and reduce the level of eutrophication. To achieve the environmental treatments, EM classification is necessary.

There are two traditional approaches for EM classification. The first is the molecular biological method which distinguishes an EM by its DNA or RNA [2, 3]. This approach requires a long time and an expensive equipment. The second approach adopts a morphological approach, in which an EM is observed under a microscope and classified manually based on its shape [1]. This approach requires huge manual effort.

In this paper, we develop a system which conducts EM classification by directly analyzing microscopic images. We consider EM classification as a pixel-based labeling and address the following three problems: First, one important factor for this is the representation of each pixel, that is, feature. A good feature makes it easier to assign the appropriate label to the pixel. With respect to this, we focus on a *Deep Convolutional Neural Network* (DCNN) considering its impressive performance in many computer vision problems, including classification, segmentation and captioning of images/videos, object detection and action recognition [4]. However, environmental investigations are always operated in outdoor environments, where conditions like temperature and salinity are changing continuously. Because EMs are very sensitive to these conditions, their quantity is easily influenced. So, we face with a small training dataset problem [5], where it is difficult to collect sufficient EM images for training a DCNN with numerous parameters to be optimised.

Our solution for this is inspired by the "pre-training and fine-tuning" approach that pre-trains a DCNN on a large auxiliary dataset, followed by domain-

specific fine-tuning on a small dataset [6, 7]. However, DCNNs usually used in this approach target at extracting image-level or region-level features, and cannot be used for our pixel-level feature extraction. Hence, we re-purpose a DCNN pre-trained on a large image dataset by replacing fully connected layers with convolutional layers with upsampled (dilated) filters [8]. Then, EM images are used to fine-tune this DCNN to produce dense pixel-level feature maps for an image. Here, each pixel is represented as a feature vector consisting of values in these maps. It should be noted that the feature implicitly includes spatial relations between the pixel and surrounding ones, because field of views of units in the DCNN are gradually enlarged by passing layers. But, on top of such pixel-level features, we implement the two extensions described below in order to explicitly handle spatial characteristics of EMs.

Second, global features are useful in applications where a rough segmentation of the object of interest is available. Whereas pixel-level features are extracted in a 'bottom-up' fashion that only exploits physical pixel values in an image, we utilise global features as a 'top-down' prior knowledge about contours, shapes and textures of EMs. Global features provide such different kinds of information and we expect classifiers that use both of them will outperform classifiers based on pixel-level features only.

Third, the majority of EM samples are obtained from the complex environments, where a large amount of impurities like rubbish is present (see Fig. 5). This kind of noise degrades the performance of EM classification, leading to a noisy image problem. To overcome this, we apply *Conditional Random Fields* (CRF) [9] for pixel-based labeling. CRFs belong to a class of probabilistic models for including context in the classification process by considering the statistical dependencies between the class labels at neighboring image sites. Additionally CRFs offer great flexibility in modeling dependencies between random variables, providing a principled way to bind random variables not only for handling spatial relations among pixel-level features, but also for integrating them with global features.

To jointly solve the problems above, we propose an EM classification model

which incorporates pixel-level features and auxiliary global features into a CRF framework. As shown in Fig. 1, first we re-purpose one of the most successful DCNNs, called VGG-16 [10], which has been trained on 1.3 million images in ImageNet dataset [11], and fine-tune it using training EM images. For an image, feature maps output by the second last layer of the re-purposed VGG-16 are used to generate pixel-level features. We also extract global features from the image. Then extracted features together with the ground truth data are used to train *Random Forest* (RF) [12] classifiers by analyzing pixel-level and global features in training images. According to [13], RF are among the most accurate individual classification techniques. The trained RF classifiers are used as the unary potentials by the CRF model. Finally, together with the pairwise potentials, the CRF model is applied to localize and label into the classes the objects of interest in the test EM images.

There are three main contributions of our work: First, we develop a full-automatic EM classification system using a CRF framework. Second, we re-purpose a pre-trained DCNN to extract pixel-level features by handling the small dataset problem. Third, we significantly improve the classification rate by combining global and pixel-level features.

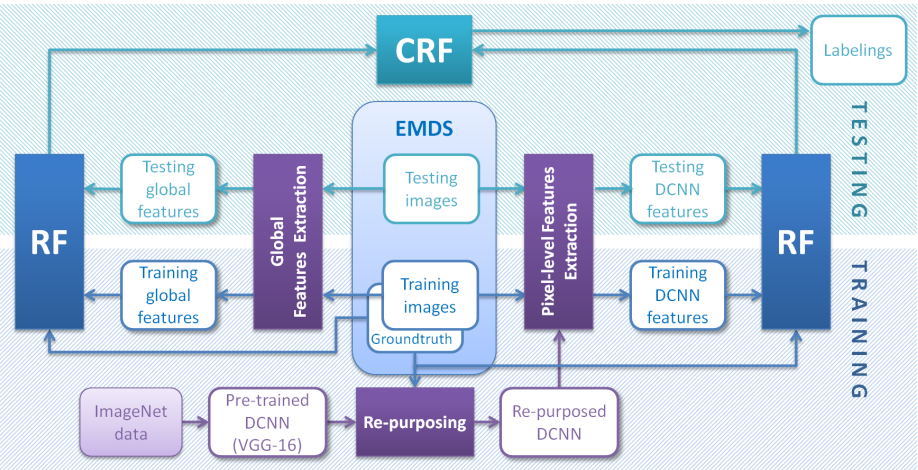

Figure 1: An overview of our CRF-based EM classification and segmentation framework.

2. Related Work

Because microorganisms are widely explored in many production and research activities, they are grouped based on their application domains [14], including agricultural microorganisms, food microorganisms, EMs, *etc.* In this section, selected works about microorganism classification using image analysis techniques are summarized. Please refer to [15, 16] for more detailed survey of each selected work.

Tab. 1 shows a comparison among features used in existing microorganism classification methods. First, there are two basic categories of feature extraction techniques: The one is "hand-crafted features", and the other is "feature learning" [17]. The former are manually designed based on prior knowledge and investigation, including global shape, local shape (including SIFT), texture, color, *etc.* as shown in the left side of Tab. 1. However, hand-crafted features are insufficient for representing diverse appearances of EMs, because all of those appearances cannot be assumed in advance. Compared to this, feature learning aims to extract useful features from a large amount of images. For example, Bag of Visual Words (BoVW) performs clustering of numerous local features (*e.g.*, SIFT) to find statistically characteristic ones called visual words [18]. Sparse Coding (SC) analyses a large number of image patches to learn a set of bases, each of which is a feature expressing a characteristic patch pattern [19]. Deep learning builds a DCNN representing feature hierarchies with higher-level features formed by the composition of lower-level ones [20]. In this paper, we adopt a deep learning approach because of its superior expressive power over BoVW and SC [17, 21]. The combination of features at each layer can define the exponential order of higher-level features, and this order is further exponentially increased by passing through layers. In addition, a DCNN is a model that mimics the process of the visual cortex in a human brain. The effectiveness of features extracted from such a biologically inspired model has been validated in many works [22].

In [20], deep learning has been used for EM classification and segmentation

Hand-crafted features		Feature learning	
Global shape	[23, 24, 25, 26]	BoVW	[18]
Local shape	[27, 15]	SC	[19]
Texture	[28, 29, 30]	Deep learning	[20]
Color	[31, 32, 33]		

Table 1: Categorization of microorganism classification methods in terms of features.

tasks. First, a Convolutional Deep Belief Network (CDBN) and an SVM classifier are used to segment possible object regions, then a DCNN consisting of six layers is applied to predict the class of each possible region. In contrast, we adopt deep learning to jointly address EM classification and segmentation problems in a CRF framework. Because of this single integrated framework, our method is more compositive and effective. Furthermore, in [20], they train the DCNN for the EM classification task from scratch, so a data augmentation approach is applied to solve the small dataset problem. In contrast, we transfer an existing pre-trained DCNN to manage the small dataset problem.

Recently researchers have proposed several DCNN-based image segmentation approaches that produce dense (high resolution) feature maps and can be extended to our pixel-level feature extraction [8, 34, 35]. One approach for obtaining dense feature maps is to upsample feature maps using a deconvolution layer [35] or using a decoder based on the downsampling record that represents value locations selected by a max-pooling layer [34]. However, this requires to learn filter weights used in the deconvolution or decoder, and causes a significant increase of parameters. Hence, upsampling feature maps is not suitable for EM classification involving the small dataset problem. Thus, we adopt another approach that upsamples 'filters' by inserting zeros (holes) between filter weights [8]. By utilizing these upsampled filters with the stride of size 1, the resolution of feature maps can be efficiently maintained by suppressing the increase of parameters.

Furthermore, microorganism classification algorithms are compared in Tab. 2.

We find that the most popular classifier is *Support Vector Machines* (SVM). Then, other classifiers, *k-Nearest Neighbors* (k-NN), *Artificial Neural Network* (ANN), *Random Forest* (RF) and *Convolutional Neural Network* (CNN), are also used. Besides these general classification methods, many algorithms are specially designed to solve the microorganism classification problem. However, it is difficult to represent the spatial relations among local image regions (pixels) using the classifiers described above. In contrast, we use CRF to explicitly model such spatial relations as well as the relations between pixel-level and global features.

Classifier	Related work	Classifier	Related work
k-NN	[28, 36]	CNN	[20]
ANN	[33, 37]	Other methods	[38, 39, 30]
SVM	[40, 41, 27]	CRF	Our method
RF	[31]		

Table 2: Overview of microorganism classification methods grouped by utilized classifiers. Nearest Neighbor (NN), k-nearest Neighbor (k-NN), Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest (RF), Convolutional Neural Network (CNN).

Stochastic approaches based on graphical models such as Conditional Random Fields [9] are popular in computer vision and pattern recognition because of a great flexibility in modeling dependencies between random variables. Particularly, pairwise CRFs have been applied successfully to many labeling tasks [42, 43, 44, 45]. Particularly, the CRF model in [46] can handle a variety of dependencies for all possible pairs of random variables (pixels). Although the exact probabilistic inference of such a model is infeasible, the researchers use a mean field approximation and high-dimensional filtering to make the inference sublinear in the number of pairwise dependencies. Also, instead of popular l_2 norm, the CRF model in [47] leverages is l_1 norm to regularise model parameters, so as to enhance the robustness to outliers and the effectiveness for high-dimensional features. Among existing CRF models, we focus on the one that combines local and global features [48]. In our CRF, the former characterise

EMs in a bottom-up fashion, while the latter reflect top-down prior knowledge about their overall characteristics. The usefulness of CRFs for incorporating local and global random variables within a solid graphical model can scarcely be overestimated. Moreover, to our best knowledge, CRF model has not been applied to the problem of microorganism classification yet.

3. Method

We start this section with a short overview of our method. First of all, we describe an individual observation $y \in \mathbb{Y}$ with a feature vector $\mathbf{f}(y)$ of distinct, measurable properties of the observation. Within this work, we consider two types of observations: *local* (y) - a single pixel and *global* ($\mathbf{y}$)- the whole image *in toto*. In order to describe the local observations (*i.e.*, pixel-level features), we use feature maps that are produced by the second last layer of a DCNN, which is pre-trained on large-scale image data and fine-tuned to EM images. Sec. 3.1 presents our pixel-level feature extraction approach by explaining the construction of the re-purposed DCNN.

We chose the nodes of the graphical model to correspond to single pixels and the neighborhood of a node to consist of four direct neighbors in the data grid; the neighboring nodes are connected with edges. Thus, our graphical model comprises the number of nodes equal to the number of pixels in an EM image. Every node is initialized with the node potentials, trained with RF on the pixel-level features. The graph edges are initialized with the interaction (pairwise) potentials, obtained using the contrast-sensitive Potts model [49].

In order to incorporate the global features to our model, we add one more extra graph node, initialized with the potentials, trained with RF on the global features. This 'global' node is connected with all other 'local' nodes via graph edges, initialized with the interaction potentials, represented by data-independent Potts model. We describe the global features and our CRF model with its potential functions in more detail in Sec. 3.2 and 3.3, respectively.

3.1. Pixel-level Feature Extraction by Re-purposing a DCNN

Fig. 2 illustrates an overview of our pixel-level feature extractor that is an extension of the DCNN-based image segmentation approach, called *DeepLab*, introduced in [8]. First, a pre-trained DCNN *VGG-16* for image classification is adapted to a segmentation model *DeepLab-VGG-16* by re-using/modifying the bottom and middle layers, as depicted by the dashed arrows in Fig. 2. In addition, three fully connected layers at the top of VGG-16 are replaced with one average pooling, three convolution and one (bilinear) interpolation layers. The right side of Fig. 2 provides a more detailed view of these three convolution layers. While the bottom one computes region-wise convolution in 512 feature maps obtained at the average pooling layer, the other two layers perform 1×1 pixel-wise convolution to enhance the non-linearity of pixel classification. Finally, letting C be the number of classes assigned to pixels, C feature maps at the top convolution layer are resized to the original image size using the interpolation layer. These resized feature maps represent a dense segmentation result where every pixel is associated with C scores, expressing how likely the pixel belongs to each class. We consider that, rather than pixel-wise convolution at the top layer, a better segmentation could be accomplished if 1024 feature maps at the penultimate layer would be resized to the original image size and used as pixel-level features for a more sophisticated classifier (*e.g.*, CRF), as shown in the rightmost part of Fig. 2. This feature extraction is detailed below.

VGG-16 is a DCNN consisting of 16 weight layers (*i.e.*, convolution and fully connected layers) [10]. While a depth is one very important factor for accurate recognition, a deep architecture involves a huge number of parameters, and its appropriate optimisation is difficult even using large-scale training data. Compared to this, VGG-16 adopts a very small field of view (3×3) for each convolution filter (see Fig. 2), so that its deep architecture contains a much smaller number of parameters. This property of VGG-16 is suitable for the small training dataset problem in EM classification. Actually VGG-16 trained on 1.3 million images in ImageNet dataset [11] demonstrated excellent performances in many tasks [10, 8, 35, 50].

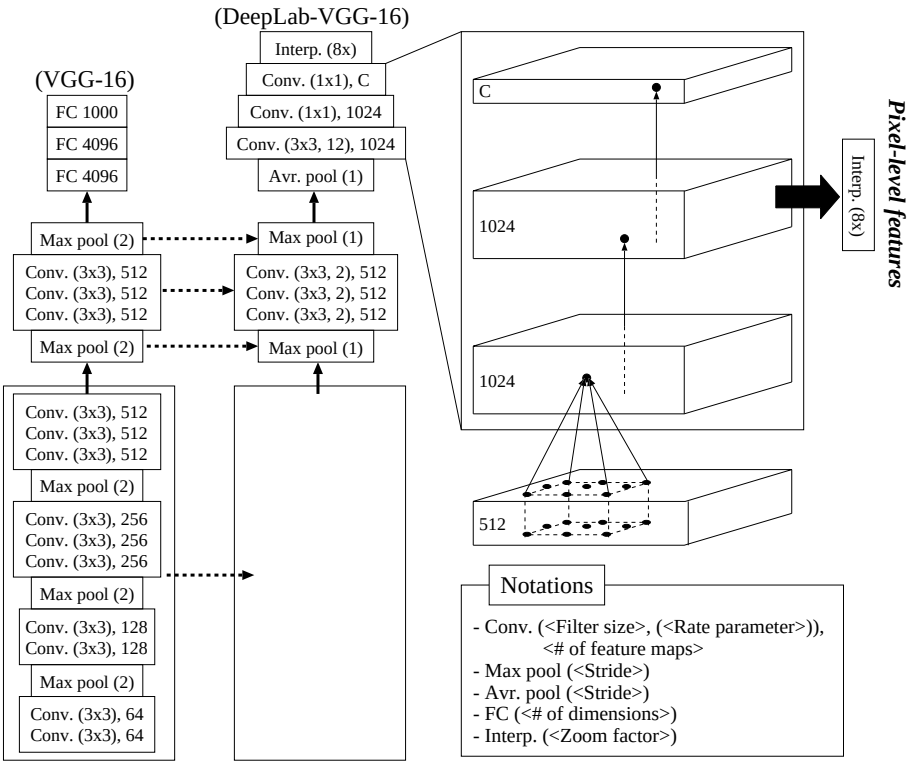

Figure 2: An overview of our pixel-level feature extraction where the pre-trained VGG-16 is re-purposed to DeepLab-VGG-16, and feature maps at the penultimate convolution layer are drawn out as pixel-level features.

VGG-16 is re-purposed to DeepLab-VGG-16 that aims to effectively maintain the spatial resolution of feature maps. In VGG-16, five max-pooling layers with stride 2 reduce the resolution of feature maps by a factor of 32 compared to the original image (see Fig. 2), so a lot of detailed information is lost. To take a good trade-off between the accuracy and efficiency, in DeepLab-VGG-16, the stride of the top two max pooling layers is set to 1, and feature maps with one-eighth of the original resolution are processed. In addition, the following *atrous convolution* is utilised to efficiently widen the field of view of a convolution filter:

$$f_l(x,y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f_{l-1}(x + r \cdot i,\ y + r \cdot j)\ w(i,j). \tag{1}$$

For simplicity, we assume that the lth and $(l-1)$th layers have single feature maps f_l and f_{l-1}, respectively (it is straightforward to extend this to multiple feature maps). In Eq. 1, a convolution filter of size $(2k+1)^2$ is represented by $w(i,j)$. But, based on the 'rate' parameter r, the convolution is done for every r values in f_{l-1} as depicted by the set of small dots in the right side of Fig. 2. In other words, the filter is dilated by introducing zeros (*i.e.*, holes) for values in f_{l-1} that are excluded from the convolution. This way, the field of view of the convolution filter is enlarged without requiring any extra parameters.

We train DeepLab-VGG-16 using 200 training EM images containing $C = 21$ classes (including the background class). Then, each EM image is fed into the trained DeepLab-VGG-16, and 1024 feature maps at the penultimate convolution layer are extracted and bilinearly interpolated to the original image size. As a result, each pixel is now represented by a 1024-dimensional feature vector. Fig. 3 visualises pixel-level features extracted for three example images in a very simple way, where each pixel is characterised by the index of the dimension having the highest value among 1024 dimensions. Such indexes are then scaled and visualised as an image. As can be seen from Fig. 3, even with this simple visualisation, the region of each EM is outlined, which implies the effectiveness of extracted pixel-level features. It is worth noting that we tested to train DeepLab-VGG-16 using natural images in PASCAL VOC 2012 dataset [51], but pixel-level features extracted from it were not so useful. It is considered that a general-purpose feature extractor trained on natural images is not suitable for a special type of EM images. Finally, it could be possible to extract further useful features by re-purposing a more advanced DCNN than VGG-16 like ResNet [52]. We leave this as a future work because its re-purposing needs much RAM and cannot be performed on our current GPU with 8GB RAM.

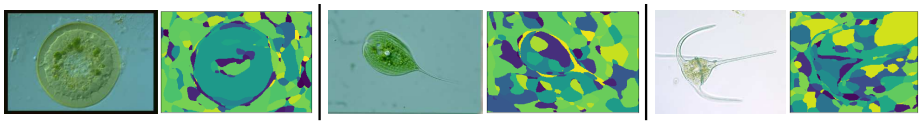

Figure 3: Simple visualisation of pixel-level features for three example images.

3.2. Global Features

We use 7 global features: three shape-driven features *perimeter*, *area* and *compactness* (perimeter squared over area); two Hough-transform-driven features *number of lines* and *number of circles*; and also we make use of *variance* and *opacity*. In order to extract the shape-driven features we first separate the EM from the background by applying a simple global bimodal segmentation. We use expectation maximization to fit a mixture of two Gaussian functions to the histogram of gray values for a given image [53]. The Bayesian decision boundary defines the cut point between the foreground and background. After that, morphological hole filling [54] is used to capture the stray bright pixels inside the object. Next, we apply a Hough transform [55] to detect lines and circles in the original images and use the number of corresponding detections as two more features. Finally, we calculate the variance of image gray-values, and EM opacity that is evaluated as

$$\frac{1}{m}\sum_{i=1}^{m}(1-\Delta_i^E)\cdot|\mu-y_i|, \tag{2}$$

where Δ_i^E is the normalized Euclidean distance between a spacial position of pixel y_i and the image center and μ is the image mean value.

3.3. Conditional Random Fields

We address the general problem of learning a mapping from input observations $y \in \mathbb{Y}$ to discrete response variables $x \in \mathbb{X}$, based on a training sample of input-output pairs (x_1, y_1), $(x_2, y_2), \ldots, \in \mathbb{X} \times \mathbb{Y}$ drawn from some fixed but unknown probability distribution. We assume that an image $\mathbf{y}$ consists of m image sites (pixels or segments) $i \in \mathcal{V}$ with observed data y_i, *i.e.*, $\mathbf{y} = (y_1, y_2, \ldots, y_m)^\top$, where $\mathcal{V}$ is the array of all sites, corresponding to the nodes of an associated undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose edges $\mathcal{E}$ model interactions between adjacent sites. Each site i is also associated with a discrete class variable $x_i \in \mathbb{X}$ which takes values from a given set of classes $\mathbb{L}$. According to [9] *conditional random fields* are probabilistic models for computing the posterior probability $p(\mathbf{x} \,|\, \mathbf{y})$ of a possible output $\mathbf{x} \in \mathbb{X}^m$ given the input $\mathbf{y} \in \mathbb{Y}^m$. Restricting

ourselves to pairwise interactions, CRF can be modeled by:

$$p(\mathbf{x}\,|\,\mathbf{y}) = \frac{1}{Z}\prod_{i\in\mathcal{V}}\varphi_i(x_i;\,\mathbf{y})\prod_{(i,j)\in\mathcal{E}}\psi_{ij}(x_i,x_j;\,\mathbf{y}), \tag{3}$$

where φ_i are the *unary potentials* which associate the observations with the label variables at site i; ψ_{ij} are the *pairwise potentials* which model the interaction of the label variables at two adjacent sites i and j; and Z is a normalizer (partition function) defined by:

$$Z = \sum_{\mathbf{x},\mathbf{y}}\prod_{i\in\mathcal{V}}\varphi_i(x_i;\,\mathbf{y})\prod_{(i,j)\in\mathcal{E}}\psi_{ij}(x_i,x_j;\,\mathbf{y}). \tag{4}$$

Finally, we can formulate the problem of image classification as finding the maximum a posteriori labelings $\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}\,|\,\mathbf{y})$.

250 *3.3.1. Unary Potential Functions*

The unary potentials $\varphi_i(x_i;\,\mathbf{y})$ in Eq. 3 are related to the probability of a label x_i taking a value $c \in \mathbb{L}$ given the data $\mathbf{y}$ by $\varphi_i(x_i;\,\mathbf{y}) \propto p(x_i{=}c\,|\,\mathbf{f}_i(\mathbf{y}))$ [56], where the image data are represented by site-wise feature vectors $\mathbf{f}_i(\mathbf{y})$ that may depend on all the data $\mathbf{y}$. The local observation describes a pixel belonging to one of EM classes or to the background. For $\mathbf{f}_i(\mathbf{y})$ we use 1024-dimensional pixel-level features $\mathbf{w}_i$, obtained based on feature maps at the penultimate convolution layer of DeepLab-VGG-16 in Sec. 3.1:

$$\varphi_i(x_i;\,\mathbf{y}) = \varphi_i(x_i;\,\mathbf{w}_i). \tag{5}$$

In order to describe the global observation, we use a feature vector $\mathbf{f}_g(\mathbf{y})$ consisting of seven global features described in Sec. 3.2. We add to the graphical model one special graph node x_g with its own unary potential

$$\varphi_g(x_g;\,\mathbf{f}_g(\mathbf{y})). \tag{6}$$

Since the current work is aiming to distinguish between EMs, the global observation only describes an EM and no *background*. Thus, the potential $p(x_g = \textit{background}\,|\,\mathbf{f}_g(\mathbf{y})) = 0$. In order to conform the global observations to

the local ones, where the potentials for the *background* class are non-zero[1], we set the potential $p(x_g{=}background \,|\, \mathbf{f}_g(\mathbf{y})) = \max_{c \in \mathbb{L}} p(x_g{=}c \,|\, \mathbf{f}_g(\mathbf{y}))$.

For the both kinds of the unary potentials we use a *Random Forest* (RF) [12]. Our RF consists of N_T decision trees that are generated in the training phase. In the classification, each tree casts a vote for the most likely class. If the number of votes for a class c is N_c, the probability underlying our definition of the association potentials is $p(x_i{=}c \,|\, \mathbf{f}_i(\mathbf{y})) = N_c/N_T$.

3.3.2. Pairwise Potential Functions

The pairwise potentials $\psi_{ij}(x_i, x_j; \mathbf{y})$ in Eq. 3 describe how likely the pair of neighboring sites i and j is to take the labels $(x_i, x_j) = (c, c')$ given the data: $\psi_{ij}(x_i, x_j; \mathbf{y}) = p(x_i{=}c; x_j{=}c' \,|\, \mathbf{f}_i(\mathbf{y}), \mathbf{f}_j(\mathbf{y}))$ [56]. Since we had introduced the additional *global* graph node x_g, we make use of two different types of pairwise potentials: The first corresponds to the edges connecting the global node x_g with all other graph nodes $x_i, i \in \mathcal{V}$, and the second corresponds to the edges connecting every node x_i with its neighbors x_j in the data grid.

In order to model the first type of pairwise potentials $\psi(x_g, x_i; \mathbf{y})$ we use the Potts model [49] that guarantees the conformity between global and local graph nodes *i.e.,*

$$\psi(x_g, x_i; \mathbf{y}) \equiv \psi(x_g, x_i) \propto p(x_i{=}c, x_j{=}c') = \begin{cases} \theta & \text{if } c = c' \\ 1 & \text{otherwise,} \end{cases} \tag{7}$$

where parameter θ modulates the degree to which the interaction potential favors identical classes between the global node and all local image sites.

For modeling the second type of pairwise potentials $\psi_{ij}(x_i, x_i; \mathbf{w}_i, \mathbf{w}_j)$ we use the contrast-sensitive Potts model [57], which can be considered as an extension of the Potts model in Eq. 7. It takes into account the difference (contrast) between observations on adjacent sites. This difference is expressed in terms of Euclidean distance between corresponding feature vectors [58]:

$$\Delta_{ij}^{E} = \mathbb{E}(\mathbf{w}_i, \mathbf{w}_j). \tag{8}$$

[1] In practical applications, the *background* class may cover up to 90% of EM image area.

Having defined the difference measure Δ_{ij}^{E}, that is supposed to be high on segments' borders and low at homogenous regions of classifying images, we add to the Potts model in Eq. 7 a contrast-sensitive regularization function, which ought to penalize the smoothness term on regions, corresponding to abrupt changes of observed data:

$$\psi_{ij}(x_i, x_j;\, \mathbf{w}_i, \mathbf{w}_j) \propto p(x_i{=}c, x_j{=}c' \,|\, \Delta_{ij}) = \begin{cases} \theta \cdot e^{-\rho \cdot \Delta_{ij}^2} & \text{if } c = c' \\ 1 & \text{otherwise} \end{cases} \tag{9}$$

where parameter ρ modulates the contrast-sensitive term.

The contrast-sensitive Potts model replicates the original Potts model behavior if the feature vectors $\mathbf{w}_i$, $\mathbf{w}_j$ are identical, but large differences between the features will reduce the impact of this smoothness assumption and make a class change between neighboring image sites more likely. This results in smooth label maps covering homogenous image regions, while preserving edges, where the objects' borders are more probable.

3.4. Local-Global CRF

Having Eq. 5, 6, 7 and 9 we can rewrite Eq. 3 in form:

$$\begin{aligned} p(\mathbf{x} \,|\, \mathbf{y}) = \frac{1}{Z}\, \varphi_g(x_g;\, \mathbf{f}_g(\mathbf{y})) \prod_{i \in \mathcal{V}} \varphi_i(x_i;\, \mathbf{w}_i) \cdot \\ \psi(x_g, x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j;\, \mathbf{w}_i, \mathbf{w}_j). \end{aligned} \tag{10}$$

The structure of our graphical model is depicted in Fig. 4. The first unary potential $\varphi_g(x_g;\, \mathbf{f}_g(\mathbf{y}))$ in Eq. 10 corresponds to the global node x_g (shown with magenta color in Fig. 4) and provides the CRF model with a single prediction about the EM depicted at the image $\mathbf{y}$. The second term $\varphi_i(x_i;\, \mathbf{w}_i)$ corresponds to the 'local' nodes x_i (red nodes in Fig. 4), providing per-pixel predictions for every image site $i \in \mathcal{V}$. This term corresponds to the unary potentials in Eq. 3. The pairwise potentials $\psi(x_g, x_i)$ correspond to connections between the global node x_g and all local nodes x_i, thus every two local nodes x_i and x_j are also bound through the global node x_g: $x_i \leftrightarrow x_g \leftrightarrow x_j,\ \forall i, j \in \mathcal{V}$. Finally, the last

term $\psi_{ij}(x_i, x_j; \mathbf{w}_i, \mathbf{w}_j)$ corresponds to the pairwise potentials in Eq. 3. The partition function Z in Eq. 10 is represented in the similar way to Eq. 4.

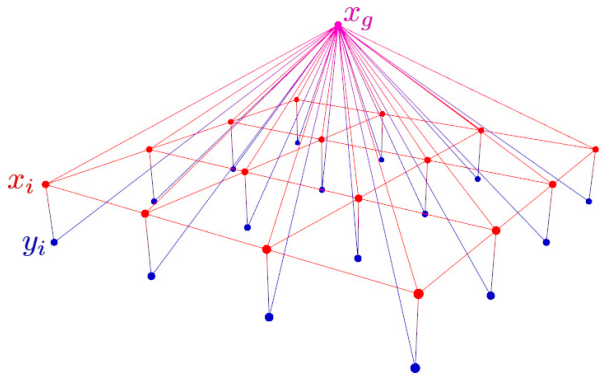

Figure 4: The structure of our graphical model. Blue, red and the magenta nodes correspond to the observation $y_i \in \mathbf{y}$, the labels $x_i \in \mathbf{x}$ and the global node x_g, respectively. Each label x_i is connected with the corresponding observation y_i (unary potentials); and also with four nearest neighbors and with the global node x_g (pairwise potentials). Please note that the global node x_g is also connected with every observation, building the whole image $\mathbf{y}$.

4. Evaluation

4.1. Experiment Setup

We use the *Environmental Microorganism Data Set* (EMDS) [38], containing 20 classes of EMs $\{\omega_1, \ldots, \omega_{20}\}$ as shown in Fig. 5. Each EM class is represented by 20 microscopic images, thus the dataset includes altogether 400 scenes. In the following discussion, for simplicity, an EM name is sometimes represented by the symbol ω_i $(1 \leq i \leq 20)$, as depicted in Fig. 5.

We use 50% of the dataset images to train DeepLab-VGG-16 for pixel-level feature extraction, RFs for unary potentials φ_i, and a local-global CRF for pixel labeling. DeepLab-VGG-16 is trained by following the network structure and hyper parameters provided as DeepLab-LargeFOV [2], except that the mini-batch

[2] `http://liangchiehchen.com/projects/DeepLab-LargeFOV.html`

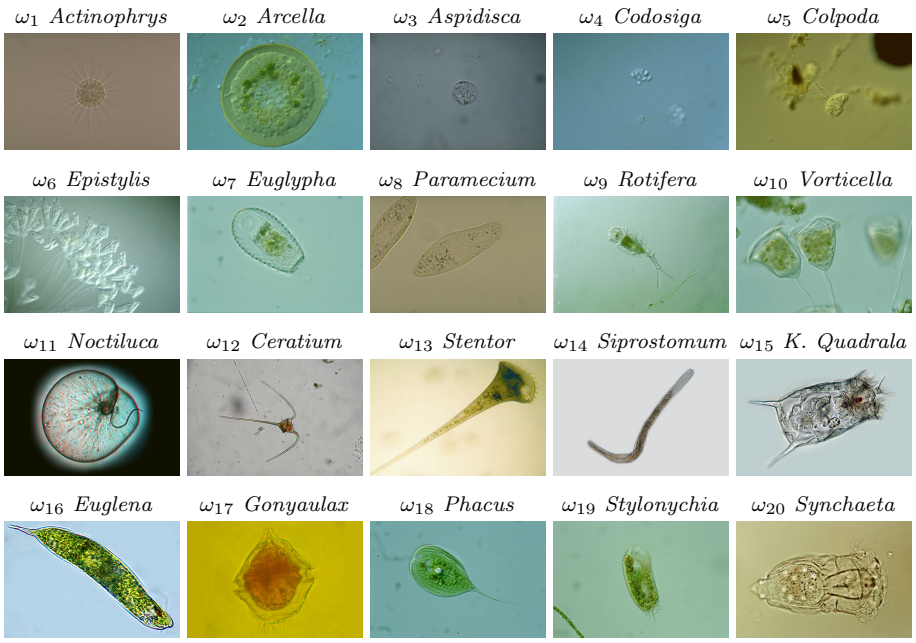

Figure 5: Examples of images on EMDS.

size is changed from 30 to 20 due to the RAM size of our GPU. With respect to RF training on 1024-dimensional pixel-level features, most regions in EM images are backgrounds. This causes the imbalanced problem that pixel-level features for backgrounds (majority class) significantly outnumbers features for 20 EM classes (minority classes) [59]. As a result, a meaningless RF that classifies almost all pixels into the background class is favored, because its classification accuracy on training images is high. To overcome this, for each of 21 classes (20 EM classes and the *background* class), an RF is trained by randomly sampling the same number of pixels. Here, this number is chosen as the minimum number of pixels among 21 classes (specifically, $19,063$ pixels for *Epistilis* (ω_6)). The RF consists of $N_T = 100$ trees of maximal depth 15. For the global node, we train a RF with the same kind and amount of trees using global features in the 200 training images. Finally, a local-global CRF is implemented in the framework of *Direct Graphical Models* C++ library [60].

Finally, the classification results are compared with the reference. For eval-

uation of pixel-level segmentation we report the recall and the precision of pixel classification (labeling) as well as the overall accuracy. Additionally, we use *Average Precision* (AP) [61] as an evaluation measure of pixel-level classification. An AP is calculated by considering pixel classification results in each image separately. For each EM class, pixels in the image are sorted based on the potentials for being that class. Then, an AP is computed as the average of precisions each of which is computed at the position of a pixel belonging to the class. A larger AP means a better result where pixels for the class are ranked at higher positions. Such APs are computed for test images containing EMs for the class, and averaged to indicate an abstracted pixel-level classification performance. Finally, we take the 'Mean of such averaged APs' (MAP) over all the 20 classes to obtain an overall performance.

4.2. Evaluation of Pixel-level Features

In this section we demonstrate the effectiveness of pixel-level features extracted from DeepLab-VGG-16. For short, we call these 'DCNN' features and compare them to the following two sets of features:

SIFT: A *Scale-Invariant Feature Transform* (SIFT) feature is one of the most popular local feature, and represents the shape in a local region, reasonably irrespective of changes in illumination, rotation, scaling and viewpoint [62]. We densely extract SIFT features by locating interesting points at all pixels. As a result, y_i for each pixel is characterized by a 128-dimensional SIFT descriptor.

Simple: Here we have gathered some common features, which are usually used in image classification: the intensity, calculated as the average of the red, blue and green channels; the saturation component in HSL color space. These two features are derived at 3 different scales: for the individual pixels and as the average in a local neighborhood of 15×15 and 25×25 pixels. Next we determine the variances of intensity, saturation and the gradient determined in local neighborhoods of 7×7, 15×15 and 25×25 pixels of each site. The last feature is the spacial coordinate feature, which describes the normalized distance of every pixel from the image center. This results in 16 simple features.

We carry out two sets of experiments for every feature. In the first experiment (*noEdge*) each node is classified solely based on the unary potentials, thus setting the pairwise potential $\psi_{ij}(x_i, x_j) \equiv \mathbb{1}$. In the second experiment (*Potts*) we use our CRF model with the Potts pairwise potentials defined in Eq. 7. This comparison should show the impact of different features on the classification with RF and CRF.

	SIFT		Simple		DCNN	
	local	+global	local	+global	local	+global
noEdge	2.21 %	10.77 %	18.04 %	35.03 %	54.78 %	62.19 %
Potts	3.80 %	10.99 %	19.60 %	31.40 %	53.69 %	61.77 %

Table 3: MAPs of the results for 3 sets of local features: SIFT, Simple, DCNN; and 2 types of CRFs: local and local-global. The impact of the local features and addition the global features is compared for two classification models: per-pixel RF (*noEdges*) and CRF with Potts pairwise potentials (*Potts*).

The MAP for local CRF over 20 EM classes in these two experiments are shown in Tab. 3. SIFT features show very poor results, and in spite of they might be useful for solving correspondence problems, they lead to a low classification rate, when used to support CRFs. Finally, we can observe that DCNN features deliver us more than 'twice-as-better' results than Simple features: 54.78%. This validates that DCNN features work more robustly than SIFT and Simple features for EM classification.

4.3. Evaluation of Global Features

In this section we demonstrate the effectiveness of the additional global features. As in the previous experiment, we compare three sets of features: SIFT, Simple and DCNN features. Again we carry out two different experiments for every feature set: *noEdge* and *Potts* (see Sec. 4.2 for more details).

The MAP over 20 EM classes in the two experiments are shown in Tab. 3. As we can see from Tab. 3, the combination of Simple features with the global features increases the classification rate for them for all experiments near by factor of two. This is done because the additional global node connected to

all the local nodes provides a long-range interaction between the local nodes and thus reduces the number of different EM classes present in the image. In other words, the global node supports the decision on the correct EM class and helps to reduce the segments, labeled as wrong EM classes. We can observe it in Fig. 6, where the incorporation of the global features leads to significant clearance of the resulting label maps from the allogenic segments. We consider this as the consequence of the global smoothness effect infused by the addition of the global node. This validates that the combination of pixel-level features with the global features works more robustly than pixel-level features alone.

The impact of the global features to the classification with the local DCNN features is not so huge: in average they increase already high MAP values by additional 8%. We explain that by the fact, that the DCNN features in comparison to the Simple features are very powerful for EM classification itself, and thus the introduction of the additional global constraints make less effect as for other more weak features.

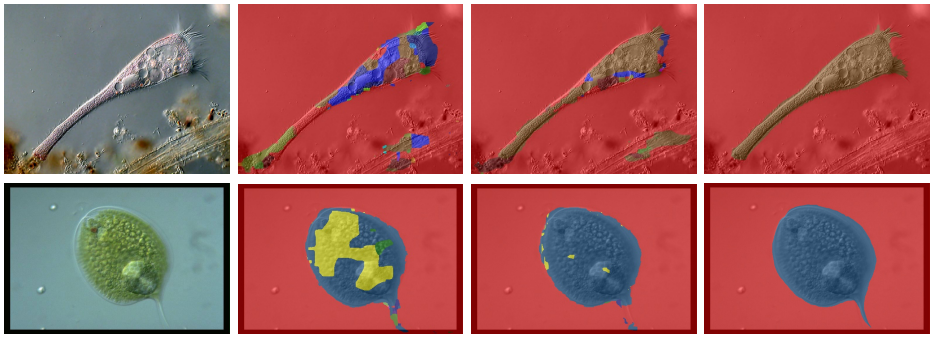

Figure 6: Segmentation results for two EMs: *Stentor* (top row) and *Stylonychia* (bottom row), achieved in *Potts* experiment. Red-colored pixels indicate that they are classified (labeled) as *background* while other colors represent pixels classified into different EM classes.

4.4. Evaluation of Segmentation Quality

In Sec. 4.2 and Sec. 4.3 we concentrated on evaluation of pixel-level EM classification results. In addition to the experiments *noEdge* and *Potts* (Sec. 4.2)

we carry out an experiment (*Potts CS*) - CRF model with the contrast-sensitive Potts pairwise potentials defined in Eq. 9. Furthermore, to examine the effectiveness of our local-global CRF, it is compared to an advanced CRF model, *denseCRF*, where pairwise potentials are defined by fully connecting all possible pairs of pixels, in order to capture a variety of spatial relations and especially recover detailed local structures [46] (see also Sec. 2). In particular, the original DeepLab approach uses *denseCRF* in the post-processing phrase [8]. Here, DeepLab-VGG-16 is not used as a feature extractor, but used to obtain an initial segmentation result that is then refined by *denseCRF*. We name this approach as $denseCRF_{org}$ and examine its performance, in order to check the effectiveness of our approach that distills outputs at the penultimate layer of DeepLab-VGG-16 as pixel-level features and uses them in other CRFs.

Moreover, our approach is compared to a currently popular DCNN-based image segmentation model, *Fully Convolutional Network* (FCN) [35]. In FCN, a pre-trained DCNN for image classification is re-purposed to segmentation using deconvolution layers, which upsample low-resolution feature maps into the ones that has the original image size and represent pixel-level classification. A version of FCN based on VGG-16 is selected for fair comparison to our approach. In particular, we choose FCN-32s where feature maps after (two customised convolution layers following) the top max-pooling layer in VGG-16 (see the left side in Fig. 2) are enlarged into image-size feature maps based on a deconvolution layer with a stride size 32[3]. FCN-32s is trained on EM images by following the network structure and hyper parameters defined for voc-fcn32s[4].

Regarding evaluation measures for segmentation results, MAP served as a good measure for justifying our choice of features, but it left mainly unclear

[3]FCN-32s only analyzes coarse-level feature maps. Some versions of FCN (FCN-16s and FCN-8s) support a 'skip' architecture to fuse coarse- and fine-level feature maps. However, our preliminary experiments showed that this yields no performance improvement (FCN-16s and FCN-8s get (63.63% (average recall), 95.31% (OA)) and (64.34%, 95.48%), respectively), and often causes over-segmentation of EM regions into small meaningless ones.

[4]`https://github.com/shelhamer/fcn.berkeleyvision.org/tree/master/voc-fcn32s`

the impact of applying CRFs. Also, a precision is not considered suitable for the following reason: Fig. 7 show segmentation results by our local-global CRF (using Potts pairwise potentials) and $denseCRF_{org}$ for three EM images. As can be seen from ground truth images, each EM has a fine-grained structure such as radial lines of ω_1 or thin tails of ω_4 and ω_9. One method like our local-global CRF roughly covers the whole region of the EM, while another like $denseCRF_{org}$ only captures its main part. The latter essentially gets a higher precision than the former. However, in practice, the whole region of an EM is more meaningful for a user than its partial region, so as to avoid missing its appearance or misunderstanding its structural characteristic. Hence, a recall is used as our main evaluation measure for segmentation results, and an overall accuracy (OA) is used as an auxiliary measure to check how similar extracted EM regions are to the ground truth. OA is computed only for the case considering all classes. If one tries to compute an OA for each class, it is equal to a recall.

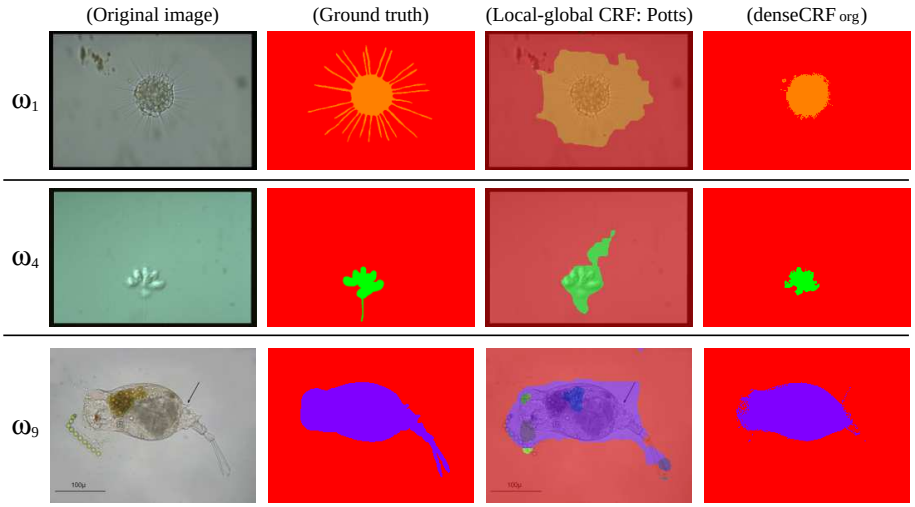

Figure 7: Examples of segmentation results for EMs that have fine-grained structures.

In order to evaluate the segmentation quality in more detail, let us consider the recall for 21 classes as well as the OA values in Tab. 4. We see that the

use of only unary potentials in *noEdge* experiment gives poor OAs, whereas the application of local edge potentials increases the segmentation accuracy more than by 10%. Such a great improvement is conditioned by the fact that the major part of the EM images is covered by the *background* class (which is not considered in MAP evaluation), and exactly the improvement on that class leads to the leap in OA values in Tab. 4. This may be more clear by looking at the recall values for the *background* class. In its turn, the average recall values for the local-global CRF models outperform both denseCRF models: 74.76% – 79.40% for local-global CRF versus 67.87% – 68.85% for denseCRF.

The example segmentations are shown in Fig. 8, which indicates some difficult cases with semi-transparent EMs, *e.g.*, *Arcella* (ω_2), *Codosiga* (ω_4), *Ceratium* (ω_{12}) and *Synchaeta* (ω_{20}), where it is hard for our method to find their invisible body parts. For example, ω_4 has class-specific antennas, which are labeled correctly, while the semi-transparent inner body is labeled wrongly. The same we observe for ω_{20} where transparent body parts have very similar features as the background and other EMs, so they are labeled wrongly. For the difficult cases RF labels only a particular part of an EM as the relevant object, where the classification result may be still correct, but segmentation is incomplete. Because this particular part contains very specific characteristics, only using this part is regarded as leading the most accurate classification. Tab 4 and Fig 8 show that our local-global CRF is better than FCN. The confusion matrix of the result achieved in the experiment *Potts* is shown in Fig. 9.

4.5. Comparison to Existing Methods

In this section we compare our CRF approach with the *Region-Based Support Vector Machine* (RBSVM) [19], which can also localize EMs with bounding boxes. The results for RBSVM are available only for the first 15 classes of EMDS dataset and only in terms of MAP. We compare the performance of 3 methods: **1**. *RBSVM*: RBSVM based on SIFT-BoVW features is trained on 15 classes. *Bag-of-Visual-Words* (BoVW) involves two steps: The first step is to obtain a set of visual words by clustering a large number of SIFT features. Each

Class	local-global CRF			denseCRF	denseCRF$_{org}$	FCN
	noEdge	*Potts*	*PottsCS*	*Gaussian*	*Gaussian*	
background	83.02 %	96.24 %	96.28 %	97.13 %	98.94 %	99.22 %
ω_1	91.43 %	81.47 %	79.22 %	75.00 %	53.86 %	45.78 %
ω_2	88.70 %	90.25 %	86.59 %	85.57 %	87.45 %	89.19 %
ω_3	94.13 %	88.86 %	93.11 %	84.49 %	83.50 %	85.10 %
ω_4	87.04 %	64.61 %	62.73 %	46.19 %	41.57 %	37.08 %
ω_5	75.48 %	72.75 %	72.17 %	68.30 %	74.59 %	80.88 %
ω_6	54.77 %	13.80 %	21.09 %	7.36 %	0.50 %	15.80 %
ω_7	79.93 %	81.07 %	76.12 %	71.36 %	80.22 %	83.11 %
ω_8	83.93 %	86.37 %	79.96 %	75.50 %	84.49 %	77.32 %
ω_9	45.82 %	42.93 %	45.74 %	33.08 %	35.64 %	44.22 %
ω_{10}	87.96 %	86.56 %	84.03 %	75.29 %	86.25 %	69.83 %
ω_{11}	92.57 %	93.57 %	91.92 %	90.63 %	90.19 %	83.44 %
ω_{12}	83.82 %	77.28 %	77.16 %	69.57 %	56.87 %	37.13 %
ω_{13}	75.19 %	81.62 %	74.05 %	62.78 %	68.10 %	55.76 %
ω_{14}	93.23 %	86.41 %	87.31 %	85.18 %	82.54 %	59.08 %
ω_{15}	79.20 %	81.39 %	76.07 %	67.46 %	68.45 %	68.06 %
ω_{16}	71.29 %	73.66 %	73.81 %	60.28 %	69.64 %	82.21 %
ω_{17}	65.35 %	64.52 %	65.20 %	60.95 %	65.12 %	65.87 %
ω_{18}	82.65 %	86.98 %	83.76 %	74.63 %	84.53 %	57.59 %
ω_{19}	65.77 %	64.39 %	61.05 %	53.63 %	62.82 %	27.88 %
ω_{20}	86.02 %	82.65 %	82.67 %	80.89 %	70.95 %	71.57 %
average:	79.40 %	76.07 %	74.76 %	67.87 %	68.85 %	63.62 %
OA:	82.63 %	94.19 %	93.98 %	94.05 %	95.85 %	95.48 %

Table 4: Recall and Overall Accuracy (OA) of the experimental results with DCNN features. Comparison between six classification models: per-pixel RF (*noEdges*), CRF with Potts pairwise potentials (*Potts*), CRF with contrast-sensitive Potts model (*PottsCS*), fully connected CRF with Gaussian pairwise potentials (*denseCRF*), fully connected CRF on segmentation results by the original DeepLab method [8] (*denseCRF*$_{org}$), fully convolutional network (*FCN*).

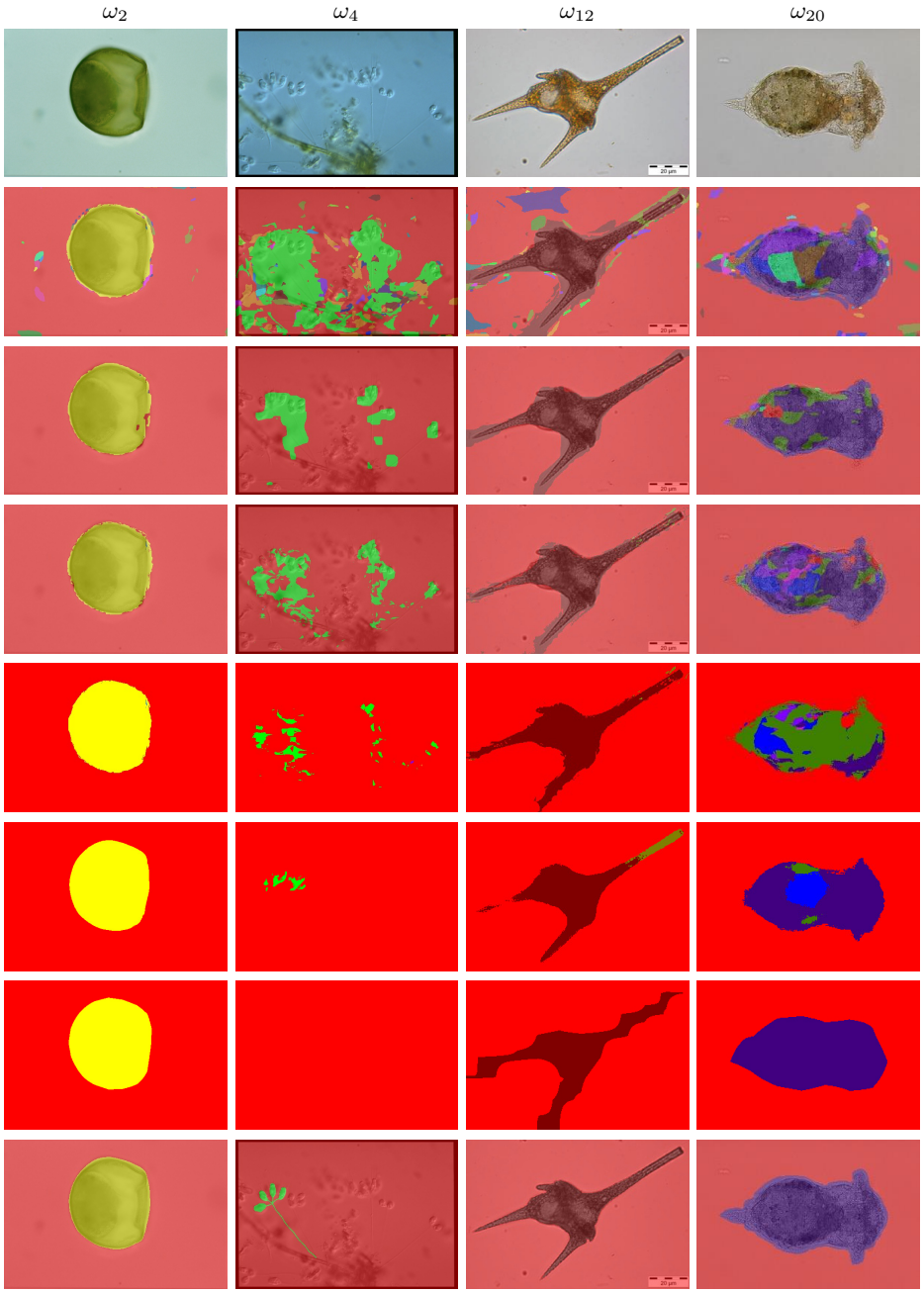

Figure 8: Segmentation results for 4 EMs (ω_2, ω_4, ω_{12}, ω_{20}) with DCNN features. 1^{st} row: input image; 2^{nd} row: local-global CRF *noEdge*; 3^{rd} row: local-global CRF *Potts*; 4^{th} row: local-global CRF *Potts CS*; 5^{th} row: denseCRF; 6^{th} row: denseCRF$_{org}$; 7^{th} row: FCN; 8^{th} row: groundtruth.

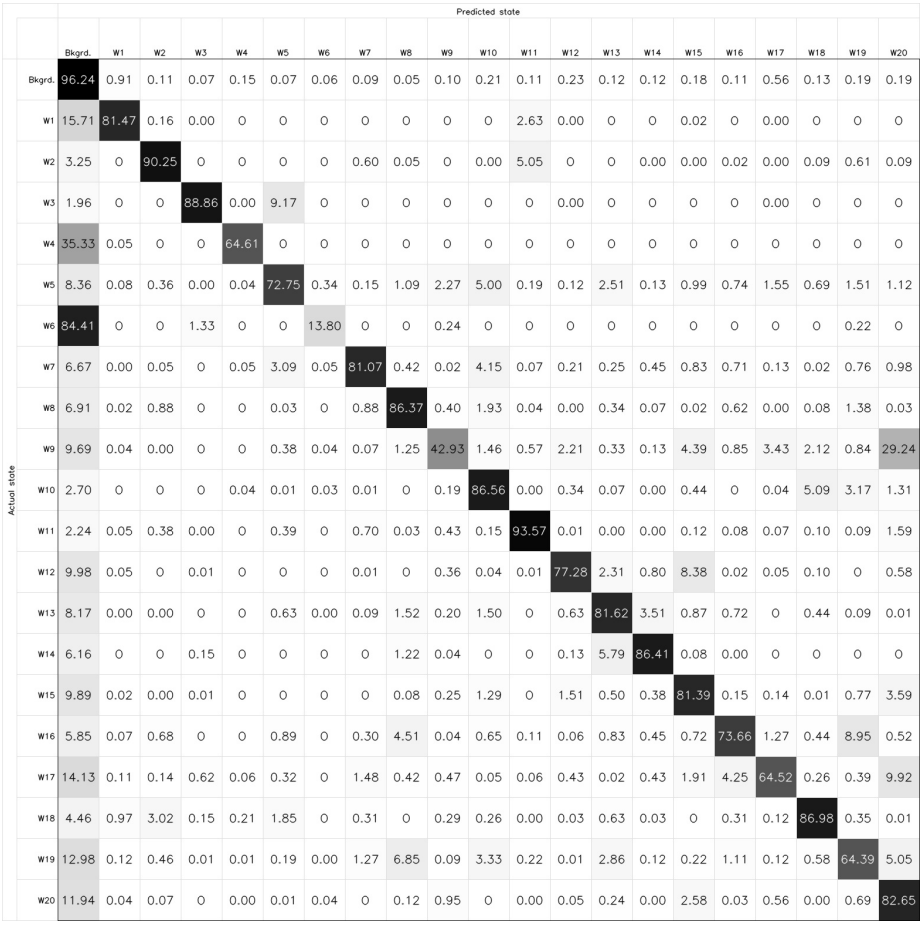

Figure 9: Confusion matrix for the experiment *Potts* with use of DCNN features. All values are given in [%]. Overall accuracy: 94.2%.

cluster center represents a characteristic SIFT feature and is regarded as a visual word. Given a region in an image, RBSVM creates a histogram of visual words by assigning each SIFT feature in this region to the most similar visual word. RBSVM localizes an EM to the region from which the histogram maximizing the SVM score is obtained [19]. **2**. *denseCRF*: denseCRF based on DCNN features is trained on 20 classes; **3**. *local-global CRF*: our CRF approach based on DCNN features which is also trained on 20 classes. Please note, that the CRF-based approaches are trained not on 15, but on 20 classes. Nevertheless, the

comparison is fair, because usually the additional classes lead to additional miss-classification, and thus to the reduction of the overall classification accuracy.

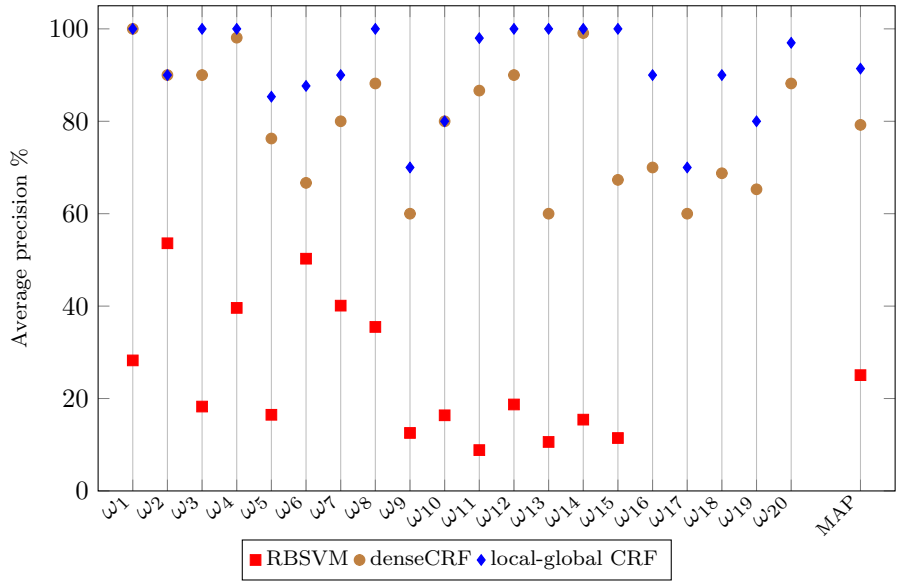

Figure 10: APs and MAPs of the EM classification results. Comparison between the RBSVM- and CRF-based approaches.

Fig. 10 represents the AP values for every EM class as well as the MAP value. First, we explain how to obtain image-level classification results based on pixel-level segmentation results by our CRF approach. We consider every label $x_i = c$, $c \in \mathbb{L} \setminus \{background\}$ of a pixel y_i of the segmented scene as a vote that the scene represents EM class c. Normalizing all the votes for a scene we achieve a probability of the scene to represent c. We sort all the test scenes in terms of these probabilities and calculate the AP value for c.

As seen from Fig. 10, the CRF-based classifiers (denseCRF and local-global CRF) significantly outperform the RBSVM classifier. The MAP value of RBSVM is 25.06%. Compared to these, the MAP of denseCRF reached 79.22% and MAP of local-global CRF – 91.40%. Such leap becomes possible because CRFs do not treat image patches independently as in RBSVM. The MAP value for local-global CRF is considerably higher than the MAP value for the denseCRF, this is due to the fact that the DCNN features in our model are supported by

the global ones.

4.6. Discussion about Computation Times

Finally, we briefly describe computational times of our EM classification method. Pixel-level feature extraction based on a DCNN were conducted on a workstation equipped with Intel® Core™ i7-7700 CPU with 3.60 GHz, 32GB RAM and GeForce GTX 1080 8GB. By following the original implementation of *DeepLab* [8], the feature extraction phase was run in the Caffe framework [63]. The re-purposing and fine-tuning step of a pre-trained DCNN (VGG-16) took 8417 seconds, and pixel-level features for all the 400 EM images were extracted in 3370 seconds by accessing the penultimate layer of the DCNN with the python interface (pycaffe).

Regarding the training our RF-based unary potentials on the DCNN features, building CRF and conducting inference we used another workstation equipped with Intel® Core™ i7-4820K CPU with 4.50 GHz, 64GB RAM and dual SLI GeForce GTX 780 3GB. Training and classification with our local-global CRF was implemented with the DGM library [60]. Training of our DGM-based classifier on 200 EM images (approx. 4.2×10^5 training samples) took 8.9 seconds, building and initializing the graphical model for one scene – 1.9 seconds and inference for one scene took in average 3.7 seconds.

5. Conclusion and Future Work

In this paper, we introduced an EM classification system. Considering the small training dataset problem, we adopt an approach where a DCNN pre-trained on large auxiliary image data is re-purposed and fine-tuned to a pixel-level feature extractor using EM images. The global features are used to support the classification and improve the segmentation quality by providing a long-range consistency between pixel labels. To overcome the noisy image problem, CRF is used to jointly localize and classify EMs by considering the spatial relations among pixel-level features, and their relations to global features. Ex-

perimental results validate the effectiveness of each of these three contributions (i.e., DCNN features, global features and CRF).

From the architectural perspective, our system consists of two main parts, feature extraction based on DCNN features, and pixel-level classification using the local-global CRF. Both parts benefit from the above-mentioned contributions. The experimental results in Tab. 3 have justified the usefulness of DCNN features over other common features. Although the performance improvement is not so dramatic as DCNN features, Tab. 4 validates the advantage of the CRF over denseCRF, with the support of global features that give about 8 % improvement, as seen from Tab. 3. But, our EM classification system provides a freedom in choosing feature extraction and classification methods, so the DCNN-based method and the local-global CRF can be replaced with more advanced ones in the future.

Regarding the technical improvement, we will use more advanced pairwise potentials, trained with a DCNN to connect the global node with the local nodes. This will allow to omit the assumption that only one EM present in the scene. Especially, rather than optimizing DCNN-based potentials and a CRF separately, we will learn 'message estimators' to efficiently perform their joint optimization [64]. Here, instead of explicitly compute potentials, message estimators only output messages required in a message passing algorithm (in our case Loopy Belief Propagation (LBP)).

References

[1] I. L. Pepper, C. P. Gerba, T. J. Gentry (Eds.), Environmental Microbiology, 3rd Edition, Academic Press, 2015.

[2] S. Greenwood, M. Sogin, D. Lynn, Phylogenetic relationships within the class oligohymenophorea, phylum cilophora, inferred from the complete small subunit rRNA gene sequences of colpidium campylum, glaucoma chattoni, and opisthonecta henneguyi, J. Mol. Evol. 33 (2) (1991) 163–174.

[3] D. Bernhard, D. Leipe, M. Sogin, M. Schlegel, Phylogenetic relationships of the nassulida within the phylum ciliophora inferred from the complete small subunit rRNA gene sequences of furgasonia blochmanni, obertrumia georgiana, and pseudomicrothorax dubius, J. Euk. Microbiol. 42 (2) (1995) 126–131.

[4] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, Recent Advances in Convolutional Neural Networks, Pattern Recognition (2017) Available online.

[5] G. Forman, I. Cohen, Learning from little: comparison of classifiers given little training, in: Proc. of ECML PKDD 2004, 2004, pp. 161–172.

[6] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. of CVPR 2014, 2014, pp. 580–587.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the Devil in the Details: Delving Deep into Convolutional Nets, arXiv:1405.3531.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell (2016) Accepted.

[9] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proc. of ICML 2001, 2001, pp. 282–289.

[10] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proc. of CVPR 2009, 2009, pp. 248–255.

[12] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[13] V. Y. Kulkarni, P. K. Sinha, Random forest classifiers: A survey and future research directions, Int. J. Adv. Comput. 36 (2013) 1144–1153.

[14] N. Orlov, J. Johnston, T. Macura, L. Shamir, I. Goldberg, Computer vision for microscopy applications, in: Vision Systems: Segmentation and Pattern Recognition, I-Tech, Austria, 2007, pp. 222–242.

[15] C. Li, Content-based Microscopic Image Analysis, Logos Verlag Berlin GmbH, Germany, 2016.

[16] C. Li, K. Wang, N. Xu, A Survey for the Applications of Content-based Microscopic Image Analysis in Microorganism Classification Domains, Artificial Intelligence Review (2017) Available online.

[17] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives, IEEE Trans. PAMI, special issue Learning Deep Architectures 35 (2013) 1798–1828.

[18] C. Li, K. Shirahama, M. Grzegorzek, Environmental microbiology aided by content-based image analysis, Pattern Anal. Appl. 19 (2) (2015) 531–547.

[19] C. Li, K. Shirahama, M. Grzegorzek, Environmental microorganism classification using sparse coding and weakly supervised learning, in: Proc. of EMR Workshop in ICMR, ACM, 2015, pp. 9–14.

[20] D. Nie, E. A. Shank, V. Jojic, A deep framework for bacterial image segmentation and classification, in: Proc. of ACM-BCB 2015, 2015, pp. 306–314.

[21] K. Shirahama, M. Grzegorzek, Towards large-scale multimedia retrieval enriched by knowledge about human interpretation, Multimed. Tools Appl. 75 (1) (2016) 297–331.

[22] D. Song, D. Tao, Biologically Inspired Feature Manifold for Scene Classification, IEEE Transctions on Image Processing 19 (1) (2010) 174–184.

[23] W. Nah, S. Hong, J. Baek, Feature extraction for classification of *Caenorhabditis Elegans* behavioural phenotypes, in: Developments in Applied Artificial Intelligence, Springer, Germany, 2003, pp. 287–295.

[24] K. Huang, P. Cosman, W. R. Schafer, Machine vision based detection of omega bends and reversals in *Caenorhabditis Elegans*, J. Neurosci. Method 158 (2) (2006) 323–336.

[25] C. Li, K. Shirahama, M. Grzegorzek, F. Ma, B. Zhou, Classification of environmental microorganisms in microscopic images using shape features and support vector machines, in: Proc. of ICIP 2013, 2013, pp. 2435–2439.

[26] C. Li, K. Shirahama, M. Grzegorzek, Application of content-based image analysis to environmental microorganism classification, Biocybern. Biomed. Eng. 35 (1) (2015) 10–21.

[27] A. Verikas, A. Gelzinis, M. Bacauskiene, I. Olenina, E. Vaiciukynas, An integrated approach to analysis of phytoplankton images, IEEE J. Ocean. Eng. 40 (2) (2015) 315–326.

[28] R. Tautenhahn, A. Ihlow, U. Seiffert, Adaptive feature selection for classification of microscope images, in: Fuzzy Logic and Applications, Springer, Germany, 2006, pp. 215–222.

[29] S. Kumar, G. S. Mittal, Textural characteristics of five microorganisms for rapid detection using image processing, J. Food Process Eng. 32 (1) (2009) 126–143.

[30] S. Promdaen, P. Wattuya, N. Sanevas, Automated microalgae image classification, Procedia Computer Science 29 (2014) 1982–1992.

[31] M. Kruk, R. Kozera, S. Osowski, P. Trzcinski, L. Sas-Paszt, B. Sumorok, B. Borkowski, Computerized Classification Systemfor the Identification of Soil Microorganisms, Applied Mathematics & Information Sciences 10 (1) (2016) 21–31.

[32] S. Kumar, G. S. Mittal, Geometric and optical characteristics of five microorganisms for rapid detection using image processing, Biosyst. Eng. 99 (1) (2008) 1–8.

[33] S. Kumar, G. S. Mittal, Rapid detection of microorganisms using image processing parameters and neural network, Food Bioproc. Tech. 3 (5) (2010) 741–751.

[34] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, arXiv:1511.00561.

[35] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. of CVPR 2015, 2015, pp. 3431–3440.

[36] B. Yu, C. Elbuken, C. L. Ren, J. P. Huissoon, Image processing and classification algorithm for yeast cell morphology in a microfluidic chip, J. Biomed. Opt. 16 (6) (2011) 1–9.

[37] S. Ayas, M. Ekinci, Random forest-based tuberculosis bacteria classification in images of ZN-stained sputum smear samples, Signal Image Video Process. 8 (1) (2014) 49–61.

[38] Y. Zou, C. Li, K. Shirahama, T. Jiang, M. Grzegorzek, Environmental microorganism image retrieval using multiple colour channels fusion and particle swarm optimisation, in: Proc. of ICIP 2016, 2016, pp. 2475–2479.

[39] E. Gutzeit, C. Scheel, T. Dolereit, M. Rust, Contour based split and merge segmentation and pre-classification of zooplankton in very large images, in: Proc. of VISAPP 2014, 2014, pp. 417–424.

[40] C. Li, K. Shirahama, J. Czajkowska, M. Grzegorzek, F. Ma, B. Zhou, A multi-stage approach for automatic classification of environmental microorganisms, in: Proc. of IPCV 2013, 2013, pp. 364–370.

[41] C. F. F. C. Filho, P. C. Levy, C. D. M. Xavier, L. B. M. Fujimoto, M. G. F. Costa, Automatic identification of tuberculosis mycobacterium, Res. Biomed. Eng. 31 (1) (2015) 33–43.

[42] J. Winn, J. Shotton, The layout consistent random field for recognizing and segmenting partially occluded objects, in: Proc. of CVPR 2006, 2006, pp. 37–44.

[43] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for markov random fields with smoothness-based priors, IEEE Trans. Pattern Anal. Mach. Intell. 30 (6) (2008) 1068–1080.

[44] P. Schnitzspan, M. Fritz, S. Roth, B. Schiele, Discriminative structure learning of hierarchical representations for object detection, in: Proc. of CVPR 2009, 2009, pp. 2238–2245.

[45] K. Schindler, An overview and comparison of smooth labeling methods for land-cover classification, IEEE Trans. Geosci. Remote Sens. 50 (2012) 4534–4545.

[46] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, in: Proc. of NIPS 2011, 2011, pp. 109–117.

[47] D. Song, W. Liu, T. Zhou, D. Tao, D. A. Meyer, Efficient Robust Conditional Random Fields, IEEE Transctions on Image Processing 24 (10) (2015) 3124–3136.

[48] D. A. Lisin, M. A. Mattar, M. B. Blaschko, M. C. Benfield, E. G. Learned-Miller, Combining local and global image features for object class recognition, in: Proc. of CVPR Workshop, 2005, pp. 47–55.

[49] M. Werlberger, M. Unger, T. Pock, H. Bischof, Efficient minimization of the non-local Potts model, in: Proc. of SSVM 2011, 2011, pp. 314–325.

[50] Y. Matsumoto, T. Shinozaki, K. Shirahama, M. Grzegorzek, K. Uehara, Kobe university, NICT and university of siegen on the TRECVID 2016 avs task, in: Proc. of TRECVID 2016, 2016, pp. 1–8.

[51] M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, Int. J. Comput. Vision 111 (1) (2015) 98–136.

[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of CVPR 2016, 2016, pp. 770–778.

[53] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. 39 (1) (1977) 1–38.

[54] P. Soille, Morphological Image Analysis: Principles and Applications, 2nd Edition, Springer-Verlag New York, 2003.

[55] R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1) (1972) 11–15.

[56] S. Kumar, M. Hebert, Discriminative random fields, Int. J. Comput. Vis. 68 (2) (2006) 179–201.

[57] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, in: Proc. of ICCV 2011, 2001, pp. 105–112.

[58] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.

[59] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Trans. on Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[60] S. Kosov, Direct graphical models C++ library, http://research.project-10.de/dgm/ (2015).

[61] J. A. Aslam, E. Yilmaz, Inferring document relevance via average precision, in: Proc. of SIGIR 2006, ACM, 2006, pp. 601–602.

[62] D. G. Lowe, Object recognition from local scale-invariant features, in: Proc. of ICCV 1999, 1999, pp. 1150–1158.

[63] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, in: Proc. of ACM MM, 2014, pp. 675–678.

[64] G. Lin, C. Shen, I. Reid, A. van den Hengel, Deeply Learning the Messages in Message Passing Inference, in: Proc. of NIPS, 2015, pp. 361–369.