Iterative image segmentation for small-sample and unlabeled data

Lisa Tostrams L.TOSTRAMS@SCIENCE.RU.NL
Rajat Thomas RAJAT@MACHINE2LEARN.NL
Tom Heskes TOM.HESKES@RU.NL

Editor:

Abstract

We propose a novel unsupervised segmentation algorithm. We show it is possible to learn the segmentation task for an unlabelled set of images. The method learns to segment images by only looking at the raw samples and applying some basic thresholding, then starting an iteratively updated learning process with these weak labels, and the careful selection of a loss function.

Keywords: image segmentation, jaccard coefficient, iterative learning, U-net

1. Introduction

In the supervised scenario of semantic image segmentation, a set of pairs of images and pixel-level semantic labels (such as 'sky', 'ground', 'road', and 'car') is used for training. The goal is to train a system that classifies the labels of known categories for image pixels. In the unsupervised scenario, segmentation is used to predict 'foreground' and 'background'. The unsupervised problem is considered to be more challenging than the supervised [10]. In recent years, this and similar tasks have made tremendous progress due to the widespread adoption of Convolutional Neural Networks.

Generally, the successful training of deep networks requires many thousands training samples. While deep convolutional networks have been outperforming the state of the art in many visual recognition tasks, their widespread success is limited by the requirements on the size of the labelled data. One method that handles a lack of large sets of training samples for image segmentation is the elegant architecture U-Net [20], which relies on the strong use of data augmentation and thus uses the available annotated samples more efficiently.

Besides size requirements, the learning process of deep networks is also limited by the presence and quality of the annotation of the training samples. For structured prediction problems fully supervised, i.e. pixel-level, labels are both expensive and time consuming to obtain. Summarization of the semantic labels in terms of weak supervision, e.g., image tags or bounding box annotations, is often less costly [19]. Recent advances have sparked significant interest in avoiding the traditional, explicit *a priori* modelling which is common for semantic segments [26] and pixel-wise segmentation masks [10].

We propose a pipeline that includes U-Net to learn the unsupervised segmentation task on a dataset that is both small ($\leq$10 samples) and unlabelled, by first applying simple pixel-level

		Not model based	Fully unsupervised[1]	No data size requirement	No additional parameter settings	Continuous[2]	Preserves locality	Computationally effective	Quick[3]	X	X
Graph based	Felzenzwalb and Huttenlocher [6]	✓	✓	✓	✓	✗	✓	✓	✓		
Cluster based	Achanta et al. [1]	✓	✓	✓	✗	✓	✓	✗	✗		
Probability based	Shi et al. [22]	✗	✗	✓	✗		✓				
	Zheng et al. [27]	✗	✓	✓		✓	✓				
	Chen et al. [4]	✗	✓	✓	✗						
CNN/ANN based	Joyce et al. [9]	✓	✗	✗	✗	✓	✓	✗			
	Xia et al. [26]	✓	✗	✗	✗	✓	✓	✗			
	Pathak et al. [19]	✓	✗	✗	✗	✓	✓	✗			
	Huo et al. [8]	✓	✗	✗	✗	✓	✓	✗			
	Souly et al. [23]	✓	✗	✗	✗	✓	✓	✗			
	Paiva and Tasdizen [18]	✓	✗	✗	✗	✓	✓	✗			
	Kanezaki [10]	✓	✗	✗	✗	✓	✓	✗			
	Chen et al. [5]	✓	✗	✗	✗	✓	✓	✗			
	Maggiori et al. [12]	✓	✗	✗	✗	✓	✓	✗			

Table 1: A comparison of the advantages and disadvantages of segmentation methods.

[1] *does not require bounding boxes, image tags, partially labelled data*

[2] *does not lead to sparse pixels*

[3] *takes less than a minute per large (≥(1024,1024) pixels) image on a single GPU*

thresholds to create weak labels, and then iteratively updating the model with a specific cost function that allows it to learn strong labels.

1.1 Previous work in Unsupervised and Weakly Supervised Segmentation

Convolutional neural networks (CNNs) have been successfully applied to semantic image segmentation in several unsupervised or weakly supervised learning scenarios.

Constrained Convolutional Neural Networks can learn pixel-wise labelling from image-level tags, such as 'car' or 'human'. Pathak et al. [19] introduce a loss function that optimizes for any set of linear constraints on the output space of a CNN. The key idea is to phrase the training objective as a relaxed biconvex optimization.

A semi-supervised framework based on Generative Adverserial Networks (GANs) [23] leverages a massive amount of available unlabeled or weakly labeled data in addition to the generated images. The idea is that adding fake samples forces real samples to be close in the feature space, enabling a bottom-up clustering process which, in turn, improves multiclass pixel classification.

Since CNNs have great potential for extracting detail features from image pixels, a method has been proposed that joins learning approaches to predict for an arbitrary image input unknown cluster labels and learns optimal CNN parameters for the image pixel clustering [10]. The method optimizes constraints on pixel feature similarities, spatial continuity, and number of unique clusters over $\sim$500 iterations.

Chen et al. [5] present ReDO, a model able to extract objects from images without any annotation in an unsupervised way . ReDO builds a dataset of images and masks from an unlabelled set. Then an adversarial architecture, given an input image, extracts the object mask and redraws the object at the same location. The generator is controlled by an discriminator that ensures that the distribution of generated images is aligned to the original one.

The nature of most common CNN architectures makes them good at recognizing but poor at localizing objects precisely [12]. When the goal is to label images at the pixel level, the output is usually too coarse and the boundaries rarely coincide with the true boundaries. One reason for this, as claimed by Maggiori et al., is the structural limitation of CNNs to carry out fine-grained classification. With a limited amount of learnable parameters, the ability to learn long-range contextual features comes at the cost of losing spatial accuracy. Different enhancement algorithms have been presented to improve coarse CNN outputs, seeking to sharpen object boundaries around real images edges.

Maggiori et al. [12] propose a generic iterative enhancement process inspired from partial differential equations and expressed as a recurrent neural network (RNN). They assume that they can afford to manually label small amounts of data. A large amount of inaccurately labelled data is used to train a large CNN to learn the generalities of the object classes, and a small amount of manually labelled data is used to tune and validate the algorithm that enhances the coarse classification maps.

Zheng et al. [27] formulate fully connected conditional random fields (CRFs) as RNNs. Traditional CNNs lack smoothness constrains that encourage label agreement between similar pixels, and spatial and appearance consistency of the labelling output. Lack of such smoothness constraints can result in poor object delineation and small spurious regions in the segmentation output. Probabilistic graphical models have been developed as effective methods to enhance the accuracy of pixel-level labelling tasks. In particular Markov Random Fields (MRFs) and CRFs are used in computer vision. The idea of CRF inference for semantic labelling is to formulate the label assignment problem as a probabilistic inference problem the incorporates assumptions such as the label agreement between similar pixels.

2. Method

The literature can be divided into two possible strategies to improve the coarse labelling resulting from CNN's: 1) adapt the structure of CNNs to produce more detailed output, or 2) take the base classification from the CNN and improve upon it with another learning step. In this article, we are focusing on the second strategy. We can first use the base CNN as a rough classifier of the objects locations, and then process this classification using the original image as guidance, so that the output objects better align to real image edges.

2.1 Training process

Here we introduce the iterative segmentation process with a weak labelling step, a learning step, and a repeated improvement, see also figure 1.

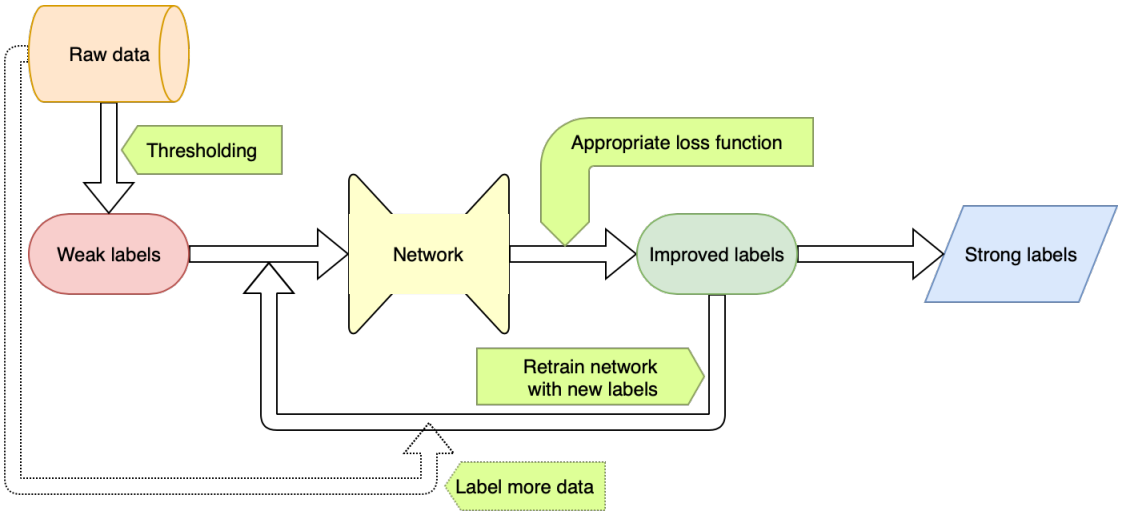


Figure 1: Method pipeline.

2.1.1 Coarse labelling

We create a very coarse segmentation mask by applying a simple filter, which selects a threshold in the gray values of the image and only keeps the values above that threshold. Examples of such simple filters are the Otsu filter and local threshold filter.

2.1.2 Convolutional Network

The input images and their corresponding coarse masks are used to train U-Net [20], with respect to the Jaccard loss, using Adam stochastic gradient descent as implemented by Keras [11]. Images are rescaled to 256x256 or 512x512 pixels for training. For the predictions, the images are rescaled to their original size in the final layer of the network. For data augmentations, unless mentioned otherwise in the results section, the original suggestions from the U-Net paper are applied [20]. Because of the information propagation through the bottleneck of the network, the learnable information is limited. This propagation is

optimized with regards to the loss function. By selecting an appropriate loss function that will 1) not punish the network for forgetting information that cant be extracted from the original image, and 2) reward the network for sharp edges and fully connected areas, we can learn segmentation that is better than the original.

2.1.3 Iterative improvement

The output from the network is used to set new masks for the network to train on. This results in a sharp decrease in the loss function. This step is repeated until convergence, usually within 2-4 repetitions.

2.2 Problem formulation

Semantic segmentation is formulated as a discrete labelling problem that assigns each pixel $x \in \omega$ with $\omega \subset \mathbb{Z}^2$ of an image to a label l from a fixed set of labels. Given the N observations, the task is to predict the set of labels that minimizes some error w.r.t. the true labels. When viewed as a statistical inference problem, you could say that every pixel has a probability of belonging to class 'background', having value 0, or belonging to class 'foreground', having value 1. When we view the output of our model as probabilities of each pixel belonging to either of the classes, we can maximize the expectation of the data under our model. This means we can formulate the segmentation task as a soft-max (structural) clustering task. In other words, the extraction of relevant information from the image is done by means of clustering the N pixels into two clusters, in such a way that the structure of those clusters is relevant for the prediction of the true label. This formulation is inspired by the Information Bottleneck principle [24] that considers clustering in the context of a higher level task.

Seldin and Tishby [21] show that for a zero loss function L they can define a bound on the generalization error in binary classification. Maurer [14] has shown that due to convexity of the KL-divergence, that bound generalizes to all loss functions bounded in the $[0, 1]$ interval.

2.3 Loss functions

In our segmentation task, we need to solve the following problems:

- Since we start from an unlabelled set of samples, there is some uncertainty around the true labelling $l(x)$ to which we compare the predictions of the model.
- We need to find an error function that allows the model to learn the clustering task for this context, and to improve upon coarse labellings.

We will use the CNN model by treating it as a clustering algorithm. We already have the soft-max cluster assignment since the energy function over the final feature map is defined as a pixel-wise soft-max. The soft-max function is defined as

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{k'=0}^{K-1} \exp(a_{k'}(x))}$$

where $a_k(x)$ denotes the activation in feature channel k at the pixel position $x \in \omega$ with $\omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(x)$ is the approximated soft-max function. I.e.

$p_k(x) \approx 1$ for the k that has $a_k(x) \geq a_{k'}(x)\ \forall k' \neq k$ and $p_k(x) \approx 0$ for all other k. The predicted class for a given pixel is $\arg\max_k p_k(x)$.

In the next sections, we will discuss why the loss function normally associated with CNN's used for image segmentation, binary cross-entropy, is not as effective in the current unsupervised problem formulation. In stead, we will show why a statistic used to describe the similarity between sets, the Jaccard index, provides a better loss metric.

2.3.1 Binary Cross-entropy loss

Most of the deep network based segmentation methods rely on logistic regression, optimizing the cross entropy loss. The cross entropy penalizes at each position the deviation of $p_{l(x)}(x)$ from 1 using

$$E(p_{l(x)}(x)) = \log(p_{l(x)}(x))$$

where $l : \omega \rightarrow \{0, ..., K-1\}$ is the true label of each pixel. This loss generalizes the logistic loss and leads to smooth optimization. The cross entropy loss enforces a similarity of the soft-max distribution at pixel-level to the 'distribution' of the true pixel labels. In the context of viewing the segmentation task as a clustering problem, this purpose of this loss is less intuitive. When the sizes classes are unbalanced and the feature space is large, binary cross entropy results in a model that does not enforce sharp boundaries around objects (see figure 2), with less than optimal localization. Since we do not know the quality of the true labels, it might be counter productive to get close to its distribution. Figure 3 shows an example of how Binary cross-entropy handles a 'false negative' in a feature space with a lot of white pixels. Figure 5 shows an example of how it handles a 'false positive'. In both cases, the occurrence of a false assignment in the coarse mask results in lower probabilities across all pixels. Figure 6 shows the behavior of binary cross-entropy as a loss function for pixels that are close in the feature space, but have different ground truth.

2.3.2 Jaccard Distance loss

The other performance measure we consider for evaluating segmentation masks is the zero-one Jaccard index [2]. In order to optimize the Jaccard index in a continuous optimization task, the smooth extension of the Jaccard index $JI = \frac{|X \cap Y|}{|X \cup Y|}$ should be considered. This extension is based on submodular analysis of set functions, where the set function maps from a set of mispredictions to a set of real numbers [13]. A candidate for such a loss is the convex closure of JI in $\mathbb{R}$. The Jaccard index has been shown to be submodular [2], and the convex closure of submodular set functions is tight and computable in polynomial time. Berman et al. [2] show that based on the result that the Jaccard index is submodular, a segmentation task can be implemented using a soft-max activation layer.

The Jaccard distance loss penalizes at each position the Jaccard index between the activation in the label feature channel $p_{l(x)=1}(x)$ and $l(x)$

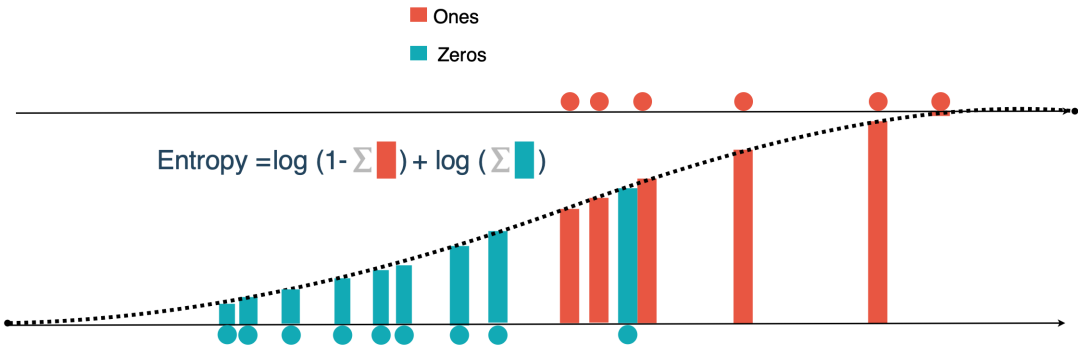


Figure 2: Binary cross entropy loss in a soft-max clustering context. The dotted line represents the shape of the soft-max function. The blue and red bars represent the probability of each pixel being assigned a 1. In the network output, this probability will be converted to gray values, i.e. black for low probabilities and white for high probabilities of being assigned a 1. The x-axis is the feature space, for example, it could represent the gray value of each pixel in the original image. For a large feature space that has a few pixels of different classes close together, the resulting soft-max will have a shallow slope that does not result in sharp borders around objects.

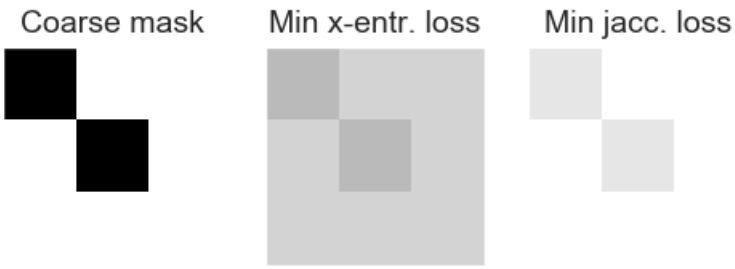


Figure 3: The left image is the 'coarse mask', which assigns labels 1 (white pixel) and 0 (black pixel) to the pixels in the original image. For this example, it is assumed that at least one of the black pixels is a false negative, and was assigned a 0 by the coarse mask despite having the same features as the white pixels. The right two images are produced by the network using the Binary cross-entropy loss and the Jaccard loss, respectively. The gray values are determined by the soft-max assignment to each pixel. Because of the false negative, the Binary cross-entropy is minimized by 'averaging' the probability assignments to the white and black pixels, while the Jaccard loss puts much more importance on keeping the white pixels.

$$
\begin{aligned}
JL(p_{l(x)=1}(x), l(x)) &= 1 - \frac{p_{l(x)=1}(x) \times l(x)}{p_{l(x)=1}(x) + l(x) - \left[p_{l(x)=1}(x) \times l(x)\right]} \\
&= 1 - \frac{p_{l(x)=1}(x) \times l(x)}{p_{l(x)=1}(x) \times l(x) + p_{l(x)=1}(x) \times (1 - l(x)) + l(x) - \left[p_{l(x)=1}(x) \times l(x)\right]} \\
&= 1 - \frac{p_{l(x)=1}(x) \times l(x)}{p_{l(x)=1}(x) \times (1 - l(x)) + l(x)}
\end{aligned}
$$

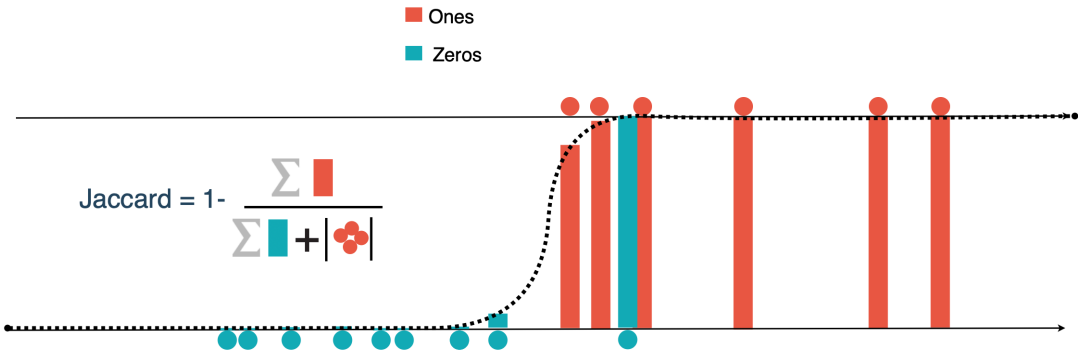


Figure 4: Jaccard distance loss in a soft-max clustering context. The dotted line represents the shape of the soft-max function. The blue and red bars represent the probability of each pixel being assigned a 1. In the network output, this probability will be converted to gray values, i.e. black for low probabilities and white for high probabilities of being assigned a 1. The x-axis is the feature space, for example, it could represent the gray value of each pixel in the original image. For a large feature space where some pixels belonging to different classes are close together, the Jaccard loss will favor sharp boundaries over avoiding false positives.

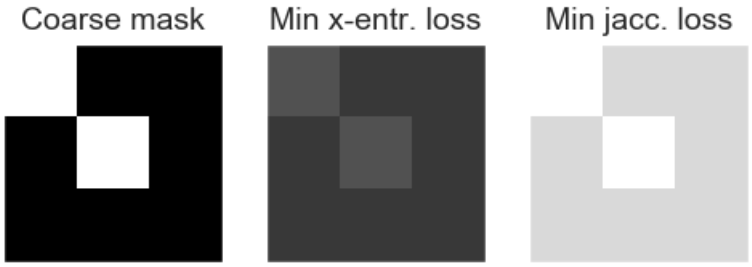


Figure 5: The left image is the 'coarse mask', which assigns labels 1 (white pixel) and 0 (black pixel) to the pixels in the original image. For this example, it is assumed that at least one of the black pixels is a false negative, and was assigned a 0 by the coarse mask despite having the same features as the white pixels. The right two images are produced by the network using the Binary cross-entropy loss and the Jaccard loss, respectively. The gray values are determined by the soft-max assignment to each pixel. Because of the false negative, the Binary cross-entropy is minimized by 'averaging' the probability assignments to the white and black pixels, while the Jaccard loss puts much more importance on keeping the white pixels.

where usually a smoothing operation is performed such that $\frac{0}{0} = 1$. The multiplication $p_{l(x)}(x) \times l(x)$ gives an approximation of the intersection in set theory by giving the probability $p_{l(x)}(x)$ when $l(x)$ is 1 and zero otherwise. This approximated intersection is highest when $p_{l(x)}(x)$ is 1 whenever $l(x)$ is 1.

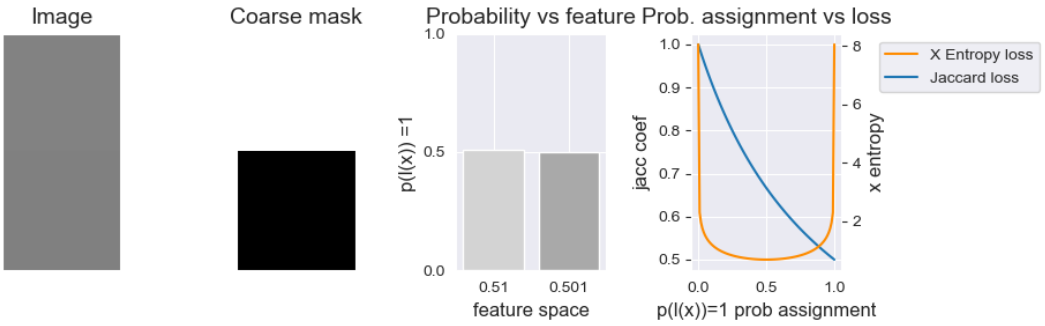


Figure 6: An example of the different ways the two loss functions balance the probability assignment in the soft-max layer for two pixels that are close in features but have different ground truth. The first plot are the pixel values in the image. The second plot is the coarse mask for these pixels. The third plot shows the pixels in the gray-value feature space, and an example of a probability assignment for both of these pixels. The final plot shows the error when one probability is assigned to both pixels. Here it is clear that Binary X-entropy tries to balance the errors by 'meeting in the middle', while Jaccard distance loss enforces the probability of the true positive to be as high as possible, even when this results in a false positive for the other pixel.

The Jaccard distance loss enforces that the overlap of the soft-max activations compared to the true labels is as large as possible. In terms of the clustering task, it will ensure that the predicted cluster of 'foreground' pixels is of similar size as and has maximum overlap with the true labels. At the same time, the total size of the mask (the sum of the probabilities of each pixel being assigned 1) is made as small as possible. This ensures sharper boundaries and better localization, as shown in figure 4.

Figure 3 shows an example of how Jaccard distance loss handles a 'false negative' in a feature space with a lot of white pixels. Figure 5 shows an example of how it handles a 'false positive'. In both cases, the occurrence of a false assignment in the coarse mask results in very high probabilities for the white pixels, and an overall increase of each pixel being assigned 1. Figure 6 shows the behavior of the Jaccard distance as a loss function for pixels that are close in the feature space, but have different ground truth.

3. Validation

We validated the method on two labelled datasets, where we checked how the results compared to the ground truth.

We also ran some simulations to better understand the limitations of this method. We hypothesised the working has to do with the feature space, not necessarily the quality of the masks, since we found it also learns reasonable results from blank masks. We analyzed the feature space for different contrasts and checked how those affected the results.

Furthermore, we show the results of the method on two new datasets, one set of images of plants collected at the Radboud University as part of an ecological study [25], and one set of images of faces generated by a neural network.

3.1 Data sets

We ran the method unsupervised on two datasets that did contain the ground truth, so we could compute the accuracy our method.

3.1.1 Tobacco plants

Plant phenotyping is the identification of effects on plant structure and function, resulting from genotypic differences and the environmental conditions a plant has been exposed to. Knowledge of plant phenotypes is a key ingredient in the evaluation of, for instance, biomass productivity [25].

While collection of phenotypic traits was previously manual, image-based methods are now increasingly utilized in non-invasive plant phenotyping and the resulting images need to be analyzed in a high-throughput, robust, and accurate manner [7]. There is a need for data at the plant feature level to be able to understand the effect of genomic variants on phenotypes that are needed to decipher the causes of plant health, crop yields, disease and evolutionary fitness. The practical importance of high-throughput automated phenotyping is widely recognized [15]. Large-scale experiments in plant phenotyping are a important factor in agriculture for meeting the future needs for food and biomass, while using less resources, under a constantly evolving environment due to climate change.

The acquisition of high-dimensional phenotypic data on an organism-wide scale differs from the usual tasks addressed by the computer vision community. A quick and cheap way to acquire data is taking two-dimensional images from photography. A benchmark data set is published by the Computer Vision Problems in Plant Phenotyping conference [15]. Here, raw and annotated images of the most frequently used rosette model plants (tobacco) is provided. 62 tobacco images were collected using a camera which contained in its field of view a single plant [16].

3.1.2 Road surface cracks

Cracks are the most common road pavement surface defects, and capturing imagery of pavement surface during road surveys, for crack detection and characterization, is considered an adequate procedure for the collection of data about the condition of the pavement surface. A set of 24 images of concrete was obtained that contained cracks in the surface [17]. The image database considered is composed by gray-level images of fixed size (1536 2048 pixels) with pixel intensities ranging from 0 (black) to 255 (white). They were captured by a digital camera at sunlight with its optical axis perpendicular to the road surface and its lateral edges parallel to the road axis. Each pixel corresponds to a road area of about 1 mm^2.

3.2 Simulations

Based on the data we simulated different feature spaces and analyzed how that affected the resulting segmentation. The parameter we adjusted is the contrast of the original image to be segmented. This contrast affects the entire pixel distribution, not just the contrast between the foreground and background.

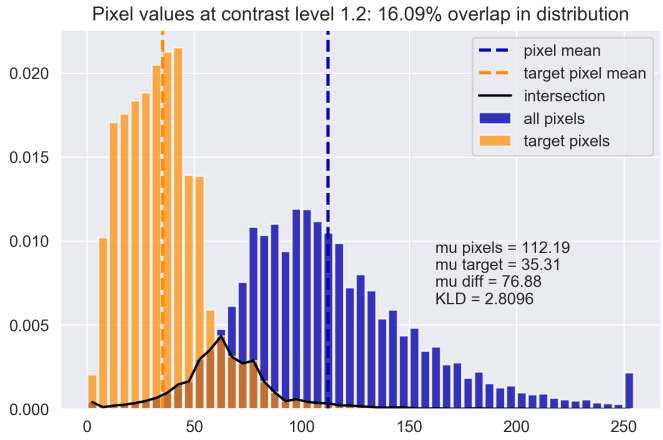


Figure 7: Example of the distributions of the target pixels and the pixels overall, at contrast level $\gamma = 1.2$.

By changing the contrast of the image, we generated different pixel value distributions. For each contrast level γ we calculate the entropy of the overall pixel distribution. See figure 7.

3.3 New application

We tried the method on two use cases where 8 images of plants, and 10 images of faces were obtained without labels.

4. Results

Overall, the method performs well. The performance is comparable to those that use supervised learning. The application to new data shows promising results for further applications.

4.1 Simulations

For the crack data, the correlation between the entropy of the feature space and the accuracy was calculated: see table 2. The same was done for the tobacco data with relation to the dice coefficient, since these were the metrics used in the original papers. The results imply that maximizing the entropy of the pixel value distribution leads to the best results, which can be done without knowing the ground truth. The accuracy for different levels of contrast is shown in table 4.

Metric	Cracks	Tobacco
Entropy	0.598400	0.576224

Table 2: Correlation coefficient between the entropy of the overall pixel distribution and the accuracy of the results for the Cracks data set, and between the entropy and the Dice coefficient for the Tobacco data set.

	0	10	20	30	40	50	60	70	80	90	100
Dice	0.1103	0.1103	0.1103	0.1102	0.1103	0.1100	0.1103	0.1100	0.4872	0.4431	0.7153
Accuracy	0.2318	0.1892	0.1891	0.1877	0.2044	0.1858	0.1892	0.2378	0.9572	0.9466	0.9828

Table 3: Accuracy and Dice coefficient for different percentages of overlap between the coarse mask and the ground truth.

Contrast	Cracks	Tobacco
0.05	0.95243	0.65705
0.1	0.96551	0.76014
0.3	0.97026	0.71628
0.5	0.97755	0.78187
0.8	0.96948	0.82760
1	0.98862	0.76623
1.2	0.98603	0.88113
1.5	0.98139	0.75146
2	0.98476	0.79005
3	0.98073	0.63158
5	0.88450	0.59211

Table 4: Accuracy of the results compared to the ground truth for different contrast levels for the Cracks data set, and the Dice coefficient of the results for the Tobacco data set.

4.2 Empirical validation

In the original paper, the authors obtain an accuracy of 99.4% for the Cracks data by starting using a physics inspired model of the cracks. The tobacco data was obtained from the CVPPP supervised segmentation challenge, where they report a high score Dice coefficient of 0.81. We also ran the regular U-net supervised learning process that uses the ground truth labels to learn from. To compare the performance as an unsupervised segmentation method, different unsupervised methods have been applied. See table 5 for a comparison to our unsupervised results.

4.3 Unlabelled data

4.3.1 Plots

See figure 8.

		Cracks (acc.)	Tobacco (Dice)
Supervised	Original papers [17] [16]	99.4	0.81
	U-net [20]	99.3	0.82
Unsupervised	SLIC [1]	83.4	0.55
	Felzenszwalb [6]	52.0	0.27
	GraphCut [3]	52.9	0.42
	It.Seg.	98.9	0.72

Table 5: Comparison of performances in literature and of other methods.

Acknowledgments

I would like to acknowledge support for this project from Alwin Peppels, whose technical insights into the method led to this article in the first place, and who made substantial contributions in our discussion about validation. I would also like to thank Renske Vroom, who posed the original question if I could figure out a way to easily separate plants from the background without training on any ground truth examples.

References

[1] R. Achanta, A. Shaj, K. Smith, A. Lucchi, P. Fue, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2012.

[2] M. Berman, A.R. Triki, and Matthew Blaschko. The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. *arXiv:1705.08790v2*, 2018.

[3] J. Borovec, J. Svihlik, J. Kybic, and D Habart. Supervised and unsupervised segmentation using superpixels, model estimation, and graph cut. *Journal of Electronic Imaging*, 2017.

[4] L.C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. *arXiv preprint arXiv:1511.03328*, 2015.

[5] M. Chen, T. Artieres, and L. Denoyer. Unsupervised object segmentation by redrawing. *arXiv preprint arXiv:1905.13539v1*, 2019.

[6] P. Felzenzwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[7] D. Houle, D.R. Govindaraju, and S. Omholt. Phenomics: the next challenge. *Nature Review Genetics*, 32(4):126–131, 2010.

[8] Y. Huo, Z. Xu, H. Moon, S. Bao, A. Assad, T. K. Moyo, M. R. Savona, R. G. Abramson, and B. A. Landman. Synseg-net: Synthetic segmentation without target modality ground truth. *EEE Transactions on Medical Imaging*, 2018.

[9] T. Joyce, A. Chartsias, and S.A. Tsaftaris. Deep multi-class segmentation without ground-truth labels. *Medical Imaging with Deep Learning*, 2018.

[10] A. Kanezaki. Unsupervised image segmentation by backpropagation. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[11] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, arXiv:1412.6980, 2015.

[12] E. Maggiori, G. Charpiat, Y. Tarabalka, and P. Alliez. Learning iterative processes with recurrent neural networks to correct satellite image classification maps. *HAL id: hal-01388551*, 2016.

[13] A. Maurer. Sub-modular functions and convexity. *Mathematical Programming The State of the Art*, page 235257, 1983.

[14] A. Maurer. A note on the PAC-Bayesian theorem. *www.arxiv.org*, 2004.

[15] M. Minervini, H. Scharr, and S.A. Tsaftaris. Image analysis: The new bottleneck in plant phenotyping. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, 32(4):126 – 131, 2015.

[16] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*, pages 1–10, 2015.

[17] H. Oliveira and Paulo Lobato Correia. Automatic road crack detection and characterization. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 14(1):155–168, 2013.

[18] A. R. C. Paiva and T. Tasdizen. Fast semi-supervised image segmentation by novelty selection. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010.

[19] D. Pathak, P. Krahlenbuhl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1796–1804, 2015.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*, pages 234–241, 2015.

[21] Y. Seldin and N. Tishby. A PAC-Bayesian Approach to Unsupervised Learning with Application to Co-clustering Analysis. *Journal of Machine Learning Research*, 2018.

[22] Zhiyuan Shi, Yongxin Yang, Timothy M. Hospedales, and Tao Xiang. Weakly-supervised image annotation and segmentation with objects and attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2525 – 2538, 2017.

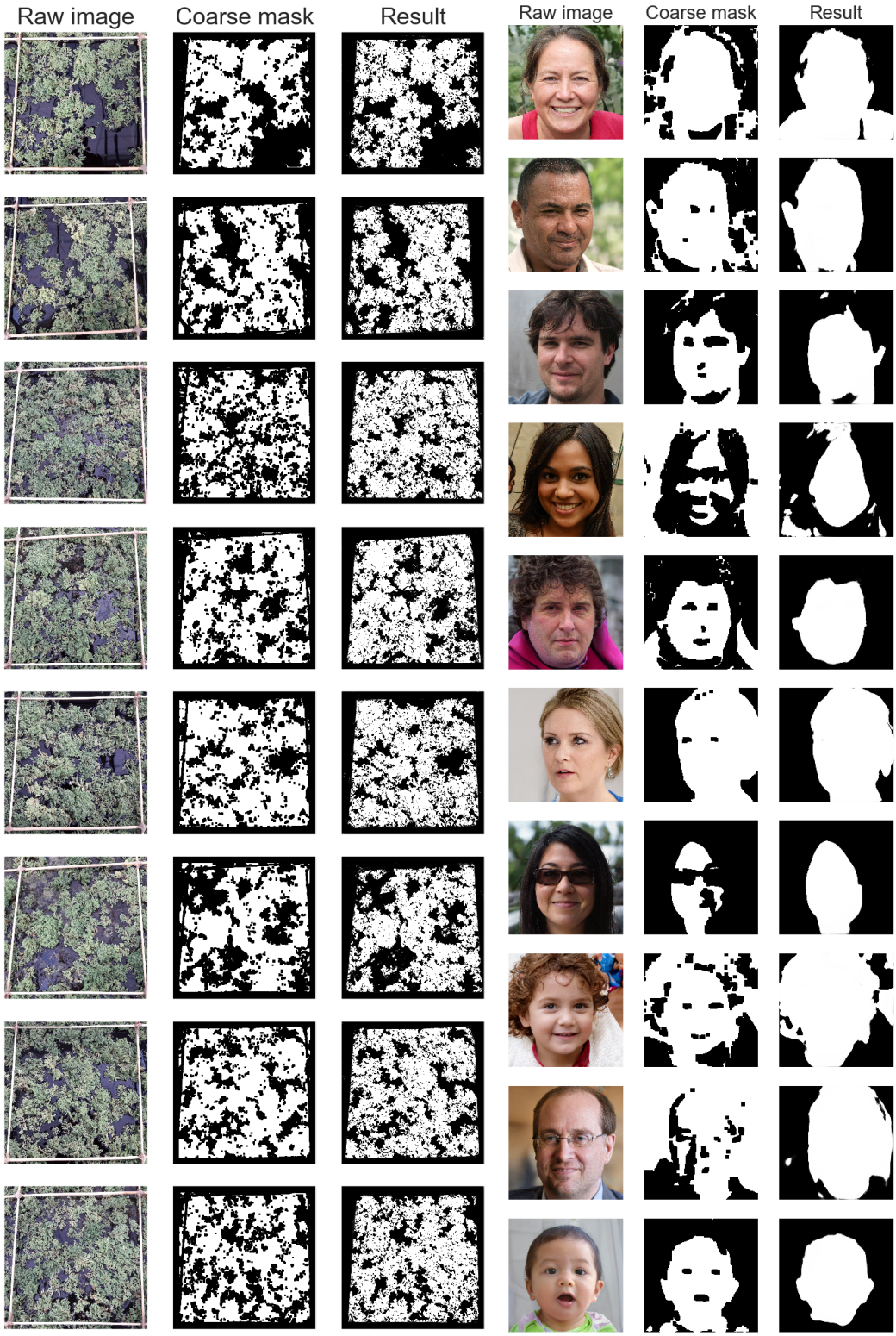
[23] N. Souly, C. Spampinato, and M. Shah. Semi and weakly supervised semantic segmentation using generative adversarial network. *arXiv:1703.09695*, 2018.

[24] N. Tishby, F. Peireira, and W. Bialek. The information bottleneck method. *Allerton Conference on Communication, Control and Computation*, 1999.

[25] R.J.E. Vroom, F. Xie, J.M. Geurts, A. Chojnowska, A.J.P. Smolders, L.P.M. Lamers, and C. Fritz. Typha latifolia paludiculture effectively improves water quality and reduces greenhouse gas emissions in rewetted peatlands. *Ecological Engineering*, 124:88–98, 2018.

[26] Wei Xia, Csaba Domokos, Jian Dong, Loong Fah Cheong, and Shuicheng Yan. Semantic segmentation without annotating segments. *2013 IEEE International Conference on Computer Vision*, pages 2176–2183, 2013.

[27] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Sum, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. *ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1529–1537, 2015.



(a) Plant results.

(b) Face results.

Figure 8