Learning to Detect Basal Tubules of Nematocysts in SEM images

Michael Lam, Janardhan Rao Doppa, Xu Hu, Sinisa Todorovic, and Thomas Dietterich
Oregon State University
Department of EECS
{lamm,doppa,huxu,sinisa,tgd}@eecs.oregonstate.edu

Abigail Reft and Marymegan Daly
Ohio State University
Department of Evolution, Ecology and Organismal Biology
{reft.1,daly.66}@osu.edu

Abstract

This paper presents a learning approach for detecting nematocysts in Scanning Electron Microscope (SEM) images. The image dataset was collected and made available to us by biologists for the purposes of morphological studies of corals, jellyfish, and other species in the phylum Cnidaria. Challenges for computer vision presented by this biological domain are rarely seen in general images of natural scenes. We formulate nematocyst detection as labeling of a regular grid of image patches. This structured prediction problem is specified within two frameworks: CRF and $\mathcal{HC}$-Search. The CRF uses graph cuts for inference. The $\mathcal{HC}$-Search approach is based on search in the space of outputs. It uses a learned heuristic function ($\mathcal{H}$) to uncover high-quality candidate labelings of image patches, and then uses a learned cost function ($\mathcal{C}$) to select the final prediction among the candidates. While locally optimal CRF inference may be sufficient for images of natural scenes, our results demonstrate that CRF with graph cuts performs poorly on the nematocyst images, and that $\mathcal{HC}$-Search outperforms CRF with graph cuts. This suggests biological images of flexible objects present new challenges requiring further advances of, or alternatives to existing methods.

1. Introduction

This paper addresses the problem of object detection in scanning electron microscope (SEM) images for the purposes of morphological characterization of cnidae. This work focuses on nematocysts, one kind of cnida, illustrated in Figure 1.

A cnida (plural cnidae) is an explosive sub-cellular capsule that fires toxins when it discharges. It is produced by a special cell called a cnidocyte. Because cnidae manifest both extreme morphological cell-level simplicity and wide biological diversity, cnidae provide a great opportunity to investigate fundamental questions in biology, including constraints and convergence in morphology [1]. Of particular interest is a morphological characterization of the basal tubules of nematocysts, marked yellow in the images shown in Figure 1. This is because surfaces of the basal tubules are characterized by spines whose shapes, lengths, and density of placement along the surface represent important phonemic characters for evolutionary studies [7].

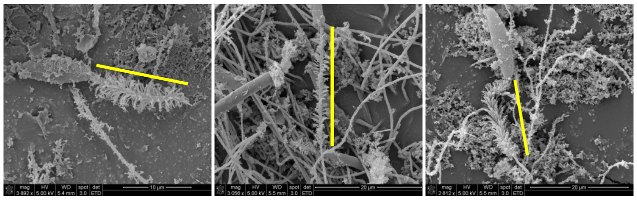


Figure 1: Example images of nematocysts from our dataset. Detecting textured, elongated, highly deformable basal tubules of nematocysts (marked yellow) against background clutter is very challenging.

Biological studies of nematocyst images are currently conducted by visual inspection and manual annotation, taking prohibitive amounts of expert time. This, in turn, typically limits the studies to small image collections of narrow scope. In this paper, we explore an opportunity for computer vision to help biologists in their analysis of nematocyst images by automatically detecting the basal tubules. As the image resolution (i.e., pixel size) is calibrated to the real size of observed specimens, detection of the basal tubules readily gives information about the size and shape of the nematocyst useful for morphological studies.

As can be seen in Figure 1, images of nematocysts present significant challenges to the state of the art in computer vision. The basal tubules are relatively thin, elongated, and highly deformable objects covered with spines. They are typically imaged against significant background clutter, consisting of mucus and cellular debris. The clutter is unavoidable, since it is extremely difficult to isolate individual nematocysts during image acquisition. Thin, elongated particles of debris appear very similar to the basal tubule. The texture of debris appears very similar to the texture of spines along the surface of the basal tubule. In addition, some images may not show the entire basal tubule, because it may be partially occluded by clutter, or extend beyond the image frame. Nematocysts are often damaged naturally and sometimes damaged through preparation, so that large parts of the basal tubules may not be physically present in the image. Rarely do we see the aforementioned challenges in general images of natural scenes.

Related work mostly focuses on image classification for accelerating biological studies [6]. In contrast, this paper focuses on object detection and localization for accelerating biological studies. We formulate detection of the basal tubules as binary labeling of a regular grid of image patches. Patches that fall on the basal tubule are assigned label “1”, and patches that fall on background are assigned label “0”. One solution for this problem is to learn a binary classifier to predict each patch label independently. However, this approach is limited, since it does not account for relationships among neighboring patches. An alternative is to specify object detection as a structured prediction problem. To this end, we employ two state-of-the-art structured prediction frameworks: CRFs (e.g., [5, 4]), and $\mathcal{HC}$-Search [3, 2]. $\mathcal{HC}$-Search has a number of advantages over CRFs in our detection problem. For example, $\mathcal{HC}$-Search allows us to use higher-order features with negligible overhead.

Our evaluation on the nematocyst images demonstrates that locally-optimal CRF inference produces poor detection results. This is in contrast to the literature, which usually reports very good CRF performance on images of outdoor- and indoor-scenes. Our results demonstrate that $\mathcal{HC}$-Search outperforms CRFs. Although both CRFs and $\mathcal{HC}$-Search are considered the most powerful, state-of-the-art frameworks for structured prediction, their relatively modest performance on the nematocyst images suggests that, in general, these kinds of biological images present new challenges for computer vision.

Our key contributions include: (I) Addressing new vision challenges in SEM images; and (II) Evaluating the most powerful structured prediction approaches – namely, CRFs and $\mathcal{HC}$-Search – on these images, and identifying key advantages and weaknesses of each approach.

In the following, we describe approaches that we use for our detection problem: IID Classifier in Sec. 2.1, CRFs in Sec. 2.2, and $\mathcal{HC}$-Search in Sec. 2.3. Sec. 3 presents the dataset of nematocyst images and our results.

2. Technical Approach

In this section, we first state the formal problem setup, and then describe the different approaches used in this work. **Problem Setup.** We are provided with a training set of input-output pairs $\{(x, y^*)\}$, where input $x \in \mathcal{X}$ is the regular grid of patches of a nematocyst image and output $y^* \in \mathcal{Y}$ corresponds to the ground-truth binary labeling of the patches. Let L be a non-negative loss function such that $L(x, y', y^*)$ is the loss associated with labeling a particular input x by output y' when the true output is y^* (e.g., Hamming and F1 loss). Our goal is to learn a predictor from inputs to outputs whose predicted outputs have low loss.

2.1. IID Classifier

A simple baseline approach for our problem is to learn an IID classifier (e.g., SVM, Logistic Regression) on patch features, and make independent predictions for every image patch. This solution is unsatisfactory, as it does not account for relationships among neighboring patches.

Structured approaches such as Conditional Random Fields (CRFs) [5, 4] and $\mathcal{HC}$-Search [3] leverage the structure in the problem by accounting for relationships between inputs and outputs. In what follows, we formulate the basal tubule detection problem within the framework of CRFs and $\mathcal{HC}$-Search.

2.2. Conditional Random Fields (CRFs)

The CRF is one of the most popular models for structured learning and inference in computer vision [5, 4]. A CRF defines a parametric posterior distribution over the outputs (labels), y, given observed image features, x, in a factored form: $P(y|x, w) = \frac{1}{Z(x,w)} e^{w \cdot \phi(x,y)}$, where w are the parameters, $Z(x, w)$ is the partition function, and the features, $\phi(x, y)$, decompose over the cliques in the underlying graphical model.

Inference is typically posed as finding the joint MAP assignment that maximizes the posterior distribution: $\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y|x, w)$, which is generally intractable. Parameter learning is usually formulated as minimizing the negative conditional log-likelihood of the data. It involves repeated calls to the inference procedure, and thus is also generally intractable. Well-known approximate inference algorithms in vision include Loopy Belief Propagation (LBP), Iterated Conditional Modes (ICM), and Graph Cuts.

In our model, the patches are organized in a graph, $G = (V, E)$, where V and E are sets of nodes and edges. The nodes $i = 1, 2, \cdots, |V|$ correspond to patches in the image, and edges $(i, j) \in E$ capture their spatial relations as a regular grid with 4-connected neighbors. Every node i is described by a 128-dimensional SIFT descriptor vector,

$\Psi_{\text{u}}(x_i, y_i)$, referred to as unary feature. Every edge $(i, j) \in E$ is described by a pairwise feature, $\Psi_{\text{pair}}(x_i, x_j, y_i, y_j)$, indicating the compatibility between patches i and j with the corresponding labeling y_i and y_j

$$\Psi_{\text{pair}}(x_i, x_j, y_i, y_j) = \begin{cases} 0 & \text{if } y_i = y_j, \\ \exp(-\beta|x_i - x_j|^2) & \text{if } y_i \neq y_j, \end{cases} \tag{1}$$

where β is a parameter. $\Psi_{\text{pair}}(x_i, x_j, y_i, y_j)$ encourages neighboring patches to take the same label.

Let the set of all patch descriptors be denoted $x = \{x_i : i = 1, \cdots, |V|\}$, and let the set of all patch labels be denoted $y = \{y_i : i = 1, \cdots, |V|\}$, where $y_i \in \{0, 1\}$. We investigate two different CRF formulations, referred to as *pairwise CRFs* and *pyramid CRFs*, as explained below.

Pairwise CRF. The pairwise CRF, given by (2), corresponds to the formulation that contains the unary and pairwise features of image patches, with the standard 4-connected neighborhood of every patch on the image lattice:

$$w \cdot \phi(x, y) = \sum_{i \in V} w_{\text{u}} \cdot \Psi_{\text{u}}(x_i, y_i) + \sum_{\substack{i \in V \\ j \in N_i}} w_{\text{pair}} \cdot \Psi_{\text{pair}}(x_i, x_j, y_i, y_j), \tag{2}$$

Pyramid CRFs. The pyramid CRF, given by (3), contains additional pyramid features, $\Psi_{\text{pyr}}(x_i, x_j, y_i, y_j)$. The graphical model now contains a grid of patches from a downsampled image by a factor of 2, in order to approximate higher-order features. Each node i from the downsampled layer is connected to its four corresponding child nodes $k \in C_i$ in the original image.

$$\begin{aligned} w \cdot \phi(x, y) = & \sum_{i \in V} w_{\text{u}} \cdot \Psi_{\text{u}}(x_i, y_i) + \sum_{\substack{i \in V \\ j \in N_i}} w_{\text{pair}} \cdot \Psi_{\text{pair}}(x_i, x_j, y_i, y_j) \\ & + \sum_{i \in V, k \in C_i} w_{\text{pyr}} \cdot \Psi_{\text{pyr}}(x_i, x_k, y_i, y_k). \end{aligned} \tag{3}$$

We investigate these two CRF models combined with the well-known inference algorithms: ICM, LBP, and Graph-Cuts.

2.3. $\mathcal{HC}$-Search

The key elements of $\mathcal{HC}$-Search [3] include the Search space over complete outputs $\mathcal{S}_o$; Search strategy $\mathcal{A}$; Heuristic function $\mathcal{H} : \mathcal{X} \times \mathcal{Y} \mapsto \Re$ to guide the search towards high-quality outputs; and Cost function $\mathcal{C} : \mathcal{X} \times \mathcal{Y} \mapsto \Re$ to score the candidate outputs generated by the search procedure. A high level overview of $\mathcal{HC}$-Search framework is shown in Figure 2. Below we explain all these elements and then describe how to learn the heuristic and cost functions.

Search Space. Every state in $\mathcal{S}_o$ consists of an input-output pair, (x, y), representing the possibility of predicting y as the output for input image x (see Figure 2). Such a search space is defined in terms of two functions: 1) *Initial state function*, I, such that $I(x)$ returns an initial state for input x; and 2) *Successor function*, S, such that for any state (x, y), $S((x, y))$ returns a set of next states $\{(x, y_1), \cdots, (x, y_k)\}$ that share the same input x.

The specific search space that we investigate leverages the IID classifier. Our $I(x)$ corresponds to the predictions made by a logistic regression classifier. S generates a set of next states by computing a set of image patches where the classifier has low confidence and generating one successor for each patch with the corresponding y value flipped. We use the conditional probability of the logistic regression IID classifier as the confidence measure. This search space is similar to the *Flipbit space* defined in [2].

The effectiveness of $\mathcal{HC}$-Search depends critically on the quality of the search space being used. The quality of a search space can be understood in terms of the expected number of search steps needed to uncover the target output y^*. For most search procedures, the time required to find y^* will grow as the depth of the target in the search space increases. Thus, one way to quantify the expected amount of search, independently of the specific search strategy, is by considering the expected depth of target outputs y^*. In particular, for a given input-output pair (x, y^*), the target depth d is defined as the minimum depth at which we can find a state corresponding to the target output y^*. By this definition, the expected target depth of our search space is equal to the expected number of errors in the output corresponding to the initial state.

Search Strategy. The role of the search procedure is to uncover high-quality outputs, guided by the heuristic function $\mathcal{H}$. Prior work [2, 3] has shown that greedy search works quite well when used with an effective search space. We investigate $\mathcal{HC}$-Search with greedy search. Given an input x, greedy search traverses a path of length τ through the search space, selecting as the next state, the best successor of the current state according to the heuristic. Specifically, if s_i is the state at search step i, greedy search selects $s_{i+1} = \arg\min_{s \in S(s_i)} \mathcal{H}(s)$, where $s_0 = I(x)$.

Making Predictions. Given an input image x, and a prediction time bound τ, $\mathcal{HC}$-Search traverses the search space starting at $I(x)$, using the search procedure $\mathcal{A}$, guided by the heuristic function $\mathcal{H}$, until the time bound is exceeded. It then scores each visited state s according to $\mathcal{C}(s)$ and returns the $\hat{y}$ of the lowest-cost state as the predicted output.

Let $y^*_{\mathcal{H}}$ denote the best output that $\mathcal{HC}$-Search could possibly return when using $\mathcal{H}$, and let $\hat{y}$ denote the output that it actually returns. Also, let $\mathcal{Y}_{\mathcal{H}}(x)$ be the set of candidate outputs generated using heuristic $\mathcal{H}$ for a given input x. Then, we define

$$y^*_{\mathcal{H}} = \arg\min_{y \in \mathcal{Y}_{\mathcal{H}}(x)} L(x, y, y^*), \quad \hat{y} = \arg\min_{y \in \mathcal{Y}_{\mathcal{H}}(x)} \mathcal{C}(x, y). \tag{4}$$

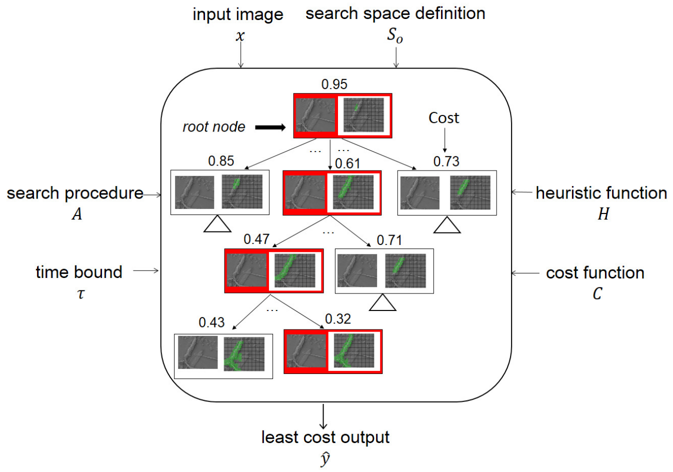


Figure 2: A high level overview of $\mathcal{HC}$-Search. Given input x and a search space, S_o, we first instantiate a search space over complete outputs. Each search node in this space consists of a input-output pair (i.e., input image and basal tubule detection). Next, we run a search procedure $\mathcal{A}$ guided by the heuristic function $\mathcal{H}$ for a time bound τ (no. of search steps). The highlighted nodes correspond to the search trajectory traversed by the search procedure, in this case greedy search. We return the least cost output $\hat{y}$ (basal tubule detection) that is uncovered during the search as the prediction for input x.

Heuristic and Cost Function Learning. The error of $\mathcal{HC}$-Search, $\epsilon_{\mathcal{HC}}$, for a given $\mathcal{H}$ and $\mathcal{C}$ can be decomposed into two parts: 1) *Generation error*, $\epsilon_{\mathcal{H}}$, due to $\mathcal{H}$ not generating high-quality outputs; and 2) *Selection error*, $\epsilon_{\mathcal{C}|\mathcal{H}}$, the additional error (conditional on $\mathcal{H}$) due to $\mathcal{C}$ not selecting the best loss output generated by $\mathcal{H}$. Guided by the error decomposition in (5), the learning approach optimizes the overall error, $\epsilon_{\mathcal{HC}}$, in a greedy stage-wise manner by first training $\mathcal{H}$ to minimize $\epsilon_{\mathcal{H}}$, and then, training $\mathcal{C}$ to minimize $\epsilon_{\mathcal{C}|\mathcal{H}}$ conditioned on $\mathcal{H}$.

$$\epsilon_{\mathcal{HC}} = \underbrace{L\left(x, y^*_{\mathcal{H}}, y^*\right)}_{\epsilon_{\mathcal{H}}} + \underbrace{L\left(x, \hat{y}, y^*\right) - L\left(x, y^*_{\mathcal{H}}, y^*\right)}_{\epsilon_{\mathcal{C}|\mathcal{H}}} \quad (5)$$

$\mathcal{H}$ is trained by imitating the search decisions made by the true loss function (available only for training data). We run the search procedure $\mathcal{A}$ for a time bound of τ for input x using a heuristic equal to the true loss function, i.e. $\mathcal{H}(x, y) = L(x, y, y^*)$, and record a set of ranking constraints that are sufficient to reproduce the search behavior. For greedy search, at every search step i, we include one ranking constraint for every node $(x, y) \in C_i \setminus (x, y_{best})$, such that $\mathcal{H}(x, y_{best}) < \mathcal{H}(x, y)$, where (x, y_{best}) is the best node in the candidate set C_i (ties are broken by a random tie breaker). The aggregate set of ranking examples is given to a rank learner (e.g., SVM-Rank) to learn $\mathcal{H}$.

$\mathcal{C}$ is trained to score the outputs $\mathcal{Y}_{\mathcal{H}}(x)$ generated by $\mathcal{H}$ according to their true losses. Specifically, this training is formulated as a bi-partite ranking problem to rank all the best loss outputs $\mathcal{Y}_{best}$ higher than all the non-best loss outputs $\mathcal{Y}_{\mathcal{H}}(x) \setminus \mathcal{Y}_{best}$.

Advantages of $\mathcal{HC}$-Search relative to other structured prediction approaches, including CRFs, are as follows. First, it scales gracefully with the complexity of the dependency structure of features. In particular, we are free to increase the complexity of $\mathcal{H}$ and $\mathcal{C}$ (e.g., by including higher-order features) without considering its impact on the inference complexity. [2, 3] show that the use of higher-order features results in significant improvements. Second, the terms of the error decomposition in (5) can be easily measured for a learned $(\mathcal{H}, \mathcal{C})$ pair, which allows for an assessment of which function is more responsible for the overall error. Third, $\mathcal{HC}$-Search makes minimal assumptions about the loss function, requiring only that we have a "blackbox" evaluation of any candidate output. Theoretically, it can even work with non-decomposable loss functions, such as F1 loss.

3. Experiments and Results

We evaluate IID classifiers (Sec. 2.1), CRFs (Sec. 2.2), and $\mathcal{HC}$-Search (Sec. 2.3) on a dataset of SEM images containing nematocysts. The image dataset was prepared by an expert biologist. Fresh specimens of cnidarian tissue were: (a) Exposed to 1M sodium citrate for 10 minutes; (b) Rinsed in water; (c) Preserved in 70% ethanol; (d) Dehydrated in a graded series; (e) Sputter-coated with gold palladium in a Cressington sputter coater; and, finally, (f) Imaged using a FEI NOVA nanoSEM microscope. The dataset consists of 130 images, each with resolution of 1024×864 pixels. The images often show multiple instances of nematocysts within cluttered background, as illustrated in Figures 1, 4, 5. The dataset is very challenging. First, the background clutter consists of mucus and debris. These appear quite similar to the target basal tubules. Mucus and debris often latch onto parts of nematocysts, which may partially occlude the basal tubules or create foreground-background confusion even to the human eye. Parts of nematocysts may also be physically missing, or may simply be out of the field of view. SEM images suffer from low contrast. The ground truth for each image is manually annotated by dividing the image into a regular grid of 32x32 pixel patches, and labeling each patch as belonging to the basal tubule of a nematocyst or background.

Evaluation Setup and Metrics. We use 80 images for training, 20 for hold-out validation, and 30 for testing. Given a test image, our structured prediction assigns one of the classes to each image patch on a regular grid. Performance is evaluated by precision, recall, and F1 measure, where true positives are patches that fall on the ground truth basal tubules. For $\mathcal{HC}$-Search, we evaluate our sensitivity to the time bound (τ), the number of greedy search steps that are allowed before making the final prediction.

Methods. An image is divided into a regular grid of patches. Each patch is described by a 128-dimensional SIFT descriptor. Assigning labels to the patches is performed using the following methods. *IID Classifier* applies either SVM or Logistic Regression independently on each image patch. *Pairwise CRF* is the standard CRF that models the image using the unary and pairwise potentials of the image patches. *Pyramid CRF* augments the pairwise potential with hierarchical relationships between (larger) parent patches and their (embedded smaller) children patches. The notations *w/ ICM*, *w/ LBP*, and *w/ GraphCuts* indicate that inference of CRF is conducted using ICM, LBP, or Graph-Cuts algorithms, respectively. $\mathcal{HC}$-Search uses the following variants: *No Global*, *Max Global*, and *Sum Global*, which differ in the feature representation for the heuristic and cost functions. *No Global* uses only the unary and pairwise features of image patches, given by (1). *Max Global* additionally uses a higher-order feature describing the largest connected component of positive detections. *Sum Global* additionally uses a higher-order term describing all connected components of positive detections. The higher-order feature is defined as the standard Bag-of-Words (BoW) of 300 codewords, found by K-means over SIFTs of all image patches from the entire dataset.

Table 1 presents the detection results of IID Classifiers, CRF, and $\mathcal{HC}$-Search. The results of Logistic Regression are reported for the detection threshold set at the maximum F1 score. The $\mathcal{HC}$-Search results are obtained for time bound $\tau = 100$ (greedy search steps). Table 1 shows that $\mathcal{HC}$-Search outperforms the two types of IID Classifiers, improving upon the initial prediction of logistic regression. Also, $\mathcal{HC}$-Search yields higher recall and F1 than all variants of CRFs. Interestingly, the CRFs with ICM inference gave better recall and F1 than the CRFs with LBP and Graph-Cuts inference. From Table 1, the inclusion of standard higher-order features (BoW) in $\mathcal{HC}$-Search does not lead to significant performance improvements. This contrasts with common reports in the literature and requires further investigation.

We also test sensitivity to (i) Image patch size, (ii) Choice of the descriptor used for patches, and (iii) Training time bound τ for $\mathcal{HC}$-Search.

First, for patch sizes of 16x16 and 64x64 pixels, and appropriately adjusted ground truth, all the approaches underperform relative to the results presented in Table 1. For all the approaches, for 16x16 pixels, F1 decreases by 8%–11%, and, for 64x64 pixels, F1 decreases by 8%–9%. Thus, our default patch size of 32x32 pixels empirically works best.

Second, when replacing SIFTs with 496-dimensional HOG descriptors, the F1 of all the approaches decreases by 2%–4%.

(a) IID Classifier Results

	Precision	Recall	F1
SVM	.675	.147	.241
Logistic Regression	.605	.129	.213

(b) CRF Results

	Precision	Recall	F1
Pairwise w/ ICM	.432	.360	.393
Pairwise w/ LBP	.545	.091	.156
Pairwise w/ GraphCuts	.537	.070	.124
Pyramid w/ ICM	.565	.258	.354
Pyramid w/ LBP	.500	.013	.025
Pyramid w/ Graph Cuts	.732	.013	.026

(c) $\mathcal{HC}$-Search Results

	Precision	Recall	F1
No Global	.472	.545	.506
Max Global	.445	.508	.475
Sum Global	.457	.533	.492

Table 1: Performance on the nematocyst images.

Finally, Figure 3 shows the plots of precision, recall, and F1 of $\mathcal{HC}$-Search *No Global* for increasing time bounds τ. The plots show four types of curves: $\mathcal{LL}$-Search, $\mathcal{HL}$-Search, $\mathcal{LC}$-Search and $\mathcal{HC}$-Search. $\mathcal{LL}$-Search uses the loss function as both the heuristic and the cost function, and thus serves as an upper bound on the performance of the selected search architecture. $\mathcal{HL}$-Search uses the learned heuristic function, and the loss function as cost function, and thus serves to illustrate how well the learned heuristic performs in terms of the quality of generated outputs. $\mathcal{LC}$-Search uses the loss function as an oracle heuristic, and learns a cost function to score the outputs generated by the oracle heuristic. From Figure 3, for $\mathcal{HC}$-Search, we see that as τ increases, precision drops, but recall and F1 improve up to a certain point before decreasing. This is understandable, because as τ increases, the generation error ($\epsilon_{\mathcal{H}}$) will monotonically decrease, since strictly more outputs will be encountered. Simultaneously, difficulty of cost function learning can increase as τ grows, since it must learn to distinguish among a larger set of candidate outputs. In addition, we can see that the $\mathcal{LC}$-Search curve is very close to the $\mathcal{LL}$-Search curve, while the $\mathcal{HL}$-Search curve is far below the LL-Search curve. This suggests that the overall error of $\mathcal{HC}$-Search, $\epsilon_{\mathcal{HC}}$, is dominated by the heuristic error $\epsilon_{\mathcal{H}}$. A better heuristic is thus likely to lead to better performance overall.

We also report the error decomposition results of $\mathcal{HC}$-Search in Table 2. Recall that from Equation 5, we can compute the decomposition of overall error $\epsilon_{\mathcal{HC}}$ in terms of the heuristic error $\epsilon_{\mathcal{H}}$ and cost function error $\epsilon_{\mathcal{C}|\mathcal{H}}$ ($\epsilon_{\mathcal{HC}} = \epsilon_{\mathcal{H}} + \epsilon_{\mathcal{C}|\mathcal{H}}$) for a given pair of heuristic and cost functions $(\mathcal{H}, \mathcal{C})$. As noted above, we can use the true loss function L as a heuristic function and/or cost function. Results in Table 2 more precisely indicate that heuristic error $\epsilon_{\mathcal{H}}$ dominates the overall error $\epsilon_{\mathcal{HC}}$ for the $\mathcal{HC}$-Search approach and cost function error $\epsilon_{\mathcal{C}|\mathcal{H}}$ for $\mathcal{LC}$-Search is very small indicating that the cost function learner is able to leverage the better outputs produced by the oracle heuristic.

Error	$\epsilon_{\mathcal{HC}}$	$\epsilon_{\mathcal{H}}$	$\epsilon_{\mathcal{C}\|\mathcal{H}}$
LL-Search	.027	.027	0
$\mathcal{HC}$-Search	.116	.075	.041
$\mathcal{LC}$-Search	.034	.027	.007

Table 2: Error decomposition of $\mathcal{HC}$-Search *No Global* for time bound $\tau = 100$.

Figures 5 and 4 illustrate detection results of *CRF Pairwise w/ ICM* and $\mathcal{HC}$-Search *No Global* for the time bounds of 5 and 50 steps. As can be seen, the CRF tends to produce islands of false positives, whereas recall of $\mathcal{HC}$-Search improves by "growing" the initial connected component of positives as the time bound increases. Figure 5 also shows some false detections of $\mathcal{HC}$-Search, because the background clutter appears very similar to the textured surface of the basal tubule.

4. Conclusions and Future Work

We have evaluated the state-of-the-art structured prediction methods, CRF and $\mathcal{HC}$-Search, on the problem of detecting basal tubules of nematocysts appearing in SEM images. $\mathcal{HC}$-Search gives better recall than CRF. While predicting locally optimal solutions may not be critical for images of general scenes, CRF and $\mathcal{HC}$-Search yield recall and precision that are less than 50% on the nematocyst images. Our experimental results indicate that the overall error of the $\mathcal{HC}$-Search approach is dominated by the heuristic error and we can significantly improve our current results with $\mathcal{HC}$-Search by improving the heuristic function. We are currently investigating this direction by employing more effective search spaces, advanced imitation learning algorithms and non-linear representations (e.g., Boosted Regression Trees) for the heuristic and cost functions.

Acknowledgements

This research has been sponsored in part by NSF DEB 1208272 to Oregon State University, and NSF DEB 1208523 to Ohio State University.

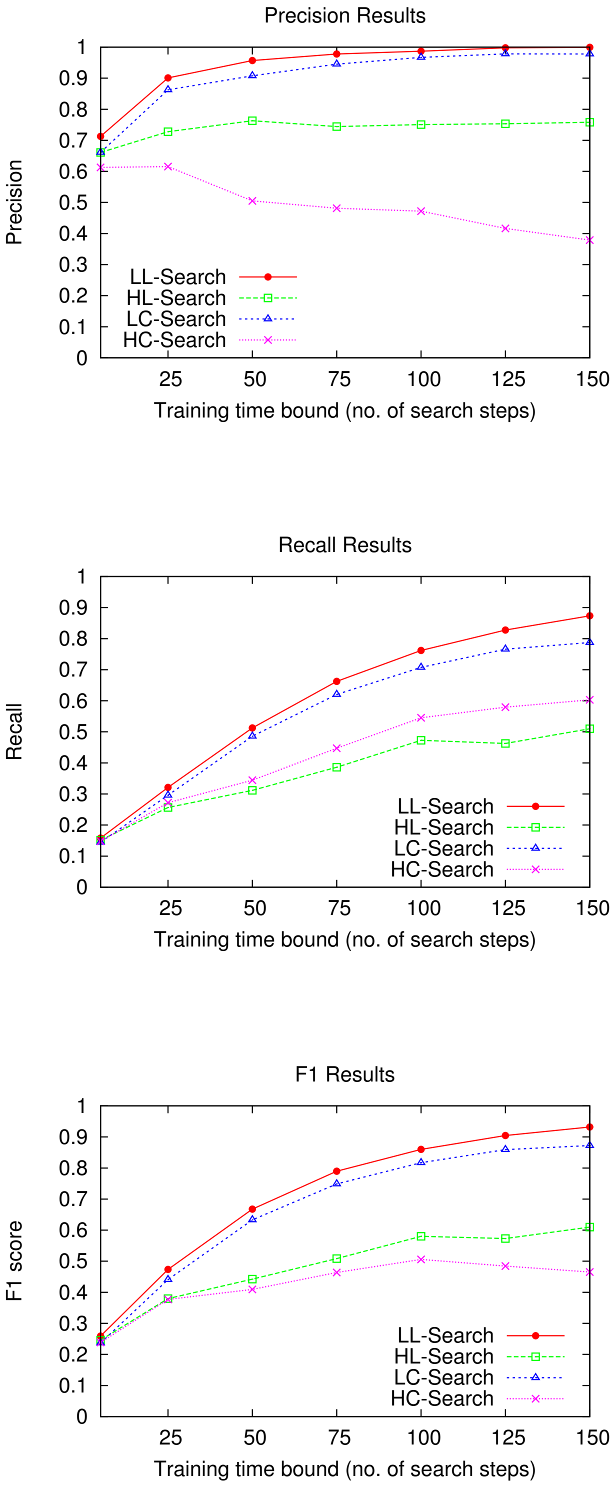
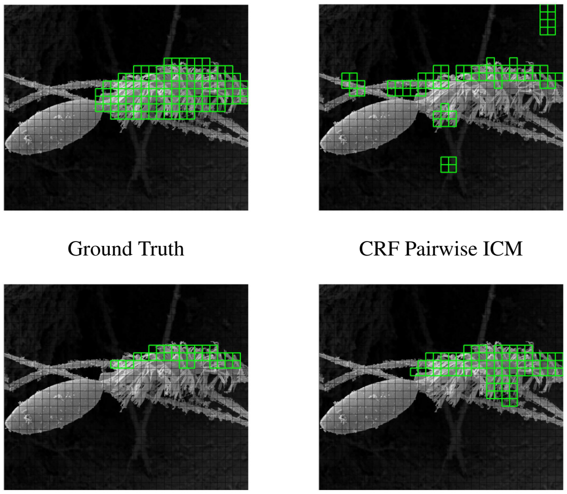


Figure 3: The plots of precision, recall and F1 of $\mathcal{HC}$-Search versus the training time bound.



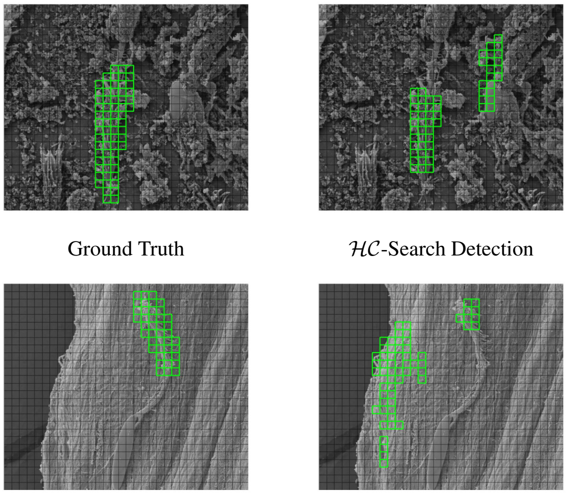
Ground Truth

CRF Pairwise ICM

$\mathcal{HC}$-Search τ=5

$\mathcal{HC}$-Search τ=50

Figure 4: An example nematocyst with the basal tubule (green). $\mathcal{HC}$-Search gives better precision and recall than CRF, and performance of $\mathcal{HC}$-Search improves as the time bound (τ) increases.



Ground Truth

$\mathcal{HC}$-Search Detection

Ground Truth

$\mathcal{HC}$-Search Detection

Figure 5: The left column shows the ground truth labels of the basal tubule (green) for two example images of nematocysts with significant background clutter. The right column shows the corresponding $\mathcal{HC}$-Search detection results (green) with the time bound of 50. $\mathcal{HC}$-Search falsely detects other instances of basal tubules, which appear similar to the target basal tubules.

References

[1] M. Daly and et al. The phylum Cnidaria: A review of phylogenetic patterns and diversity three hundred years after Linnaeus. *Zootaxa*, 1668:127–186, 2007. 1

[2] J. R. Doppa, A. Fern, and P. Tadepalli. Output space search for structured prediction. In *ICML*, 2012. 2, 3, 4

[3] J. R. Doppa, A. Fern, and P. Tadepalli. HC-Search: Learning heuristics and cost functions for structured prediction. In *AAAI*, 2013. 2, 3, 4

[4] P. Kohli, L. Ladický, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, May 2009. 2

[5] M. P. Kumar and D. Koller. Efficiently selecting regions for scene understanding. In *CVPR*, 2010. 2

[6] G. Martinez, W. Zhang, N. Payet, S. Todorovic, N. Larios, A. Yamamuro, D. Lytle, A. Moldenke, E. Mortensen, R. Paasch, L. Shapiro, and T. Dietterich. Dictionary-free categorization of very similar objects via stacked evidence trees. In *CVPR*, 2009. 2

[7] A. Reft and M. Daly. Morphology, distribution, and evolution of apical structure of nematocysts in Hexacorallia. *Journal of Morphology*, 273(2):121–136, 2012. 1