Spatially Aware Dictionary Learning and Coding for Fossil Pollen Identification

Shu Kong[1], Surangi Punyasena[2], Charless Fowlkes[1]
[1]UC Irvine {skong2, fowlkes}@ics.uci.edu
[2]UIUC punyasena@life.illinois.edu

Abstract

We propose a robust approach for performing automatic species-level recognition of fossil pollen grains in microscopy images that exploits both global shape and local texture characteristics in a patch-based matching methodology. We introduce a novel criteria for selecting meaningful and discriminative exemplar patches. We optimize this function during training using a greedy submodular function optimization framework that gives a near-optimal solution with bounded approximation error. We use these selected exemplars as a dictionary basis and propose a spatially-aware sparse coding method to match testing images for identification while maintaining global shape correspondence. To accelerate the coding process for fast matching, we introduce a relaxed form that uses spatially-aware soft-thresholding during coding. Finally, we carry out an experimental study that demonstrates the effectiveness and efficiency of our exemplar selection and classification mechanisms, achieving 86.13% *accuracy on a difficult fine-grained species classification task distinguishing three types of fossil spruce pollen.*[1]

1. Introduction

As one of the most ubiquitous of terrestrial fossils, pollen has an extraordinarily rich record and has been used to test hypotheses from a broad cross-section of biological and geological sciences and a diverse array of disciplines. Detecting and classifying pollen grains in a collected sample allows one to estimate the diversity of plant species in a particular area, carry out paleoecological and paleoclimatological investigations across hundreds to millions of years, implement the identification of plant speciation and extinction events, calculate the correlation and biostratigraphic dating of rock sequences, and conduct studies of long-term anthropogenic impacts on plant communities and the study of plant-pollinator relationships [25].

While high-throughput microscopic imaging allows for ready acquisition of large numbers of images of modern or fossilized pollen samples, identifying and counting by eye the number of grains of each species is painstaking work and requires substantial expertise and training. In this paper, we tackle the problem of performing automated species-level classification of individual pollen grains using machine learning techniques based on sparse coding to capture fine-grained distinctions in surface texture and shape.

A number of previous works have proposed to apply machine learning to pollen identification [15, 16, 8, 27, 17, 31, 35, 2, 5, 14, 9]. However, all of these methods have largely avoided the difficult problem of species-level classification, which is significant to the reconstruction of paleoenvironments and discrimination of paleoecologically and paleoclimatically significant taxa [25]. Recently, Punyasena *et al.* proposed two different machine learning-based approaches to identify two pollen species of spruce [25, 30]. Their approach uses three categories of hand-crafted features, including intensity distribution, gross shape, and texture features which are further enriched by varying the parameters for each feature computation. They show effectiveness of the approach in identifying both modern and fossil pollen grains as a three-way classification problem, and suggest that the pollen grain size and texture are important variables in pollen species discrimination. However, they rely on leave-one-out validation to estimate performance and leave open the question of generalization to held-out test data and other species.

In this paper, we propose a robust framework to automatically identify the species of fossil pollen grains in microscopy images. There are several difficulties that arise, including the arbitrary viewpoint of the pollen grains imaged (see Fig. 1) and very limited amounts of expert-labeled training data (relative to many modern computer vision tasks). To address these problems, we introduce an exemplar matching strategy for identification based on local surface patches through several novel technical components. First, we propose a greedy method for selecting discriminative exemplar patches based on optimizing a submodular set function. We show our greedy algorithm is effi-

[1] This work was supported by NSF grants DBI-1262547 and IIS-1253538.

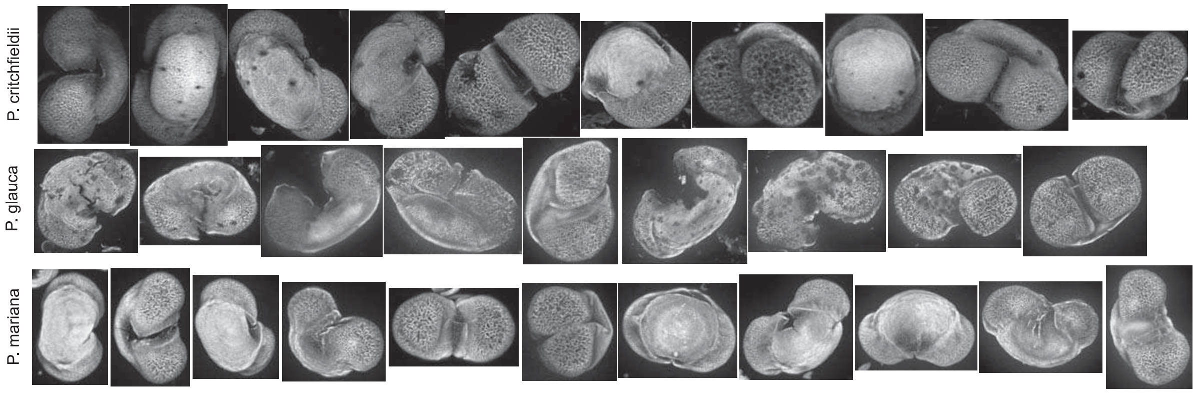

Figure 1. Example fossil pollen grains from three species of spruce, imaged via confocal fluorescence microscopy. The fine-grained identification of pollen species is not a trivial task and depends on subtle differences in the overall pollen grain shape as well as local surface texture. The arbitrary viewpoint, substantial intra-species shape variance and sample degradation of the grains poses further difficulties.

cient and gives a near-optimal solution with a $(1 - 1/e)$-approximation bound. Second, we use the selected exemplar patches as a codebook dictionary and propose a spatially aware sparse coding method to match test image patches for classification. Finally, to accelerate the matching process for classification, we introduce a relaxed form of our weighted sparse coding method for fast matching. Through experimental study on a dataset of spruce pollen grains, we demonstrate the efficiency and effectiveness of our patch selection and classification mechanisms. Our method achieves 86.13% accuracy on a three-way classification task which is quite promising given the visual difficulty of the task and the small training set size.

2. Discriminative Patch Selection

In order to allow robust matching of surface texture and local shape features of pollen grains while maintaining invariance to arbitrary viewpoint (as shown in Figure 1), we use a patch-based representation of appearance. Our first step is to select a small number of exemplar patches from the training dataset. The selected patches or exemplars should not only represent the pollen grains well in the feature space, but also have the capability to distinguish species-level characteristics by preserving the spatial structure of the grains. We use the selected patches as a dictionary basis to match testing images for identification.

To this end, we formulate an objective function that scores a set of candidate patches selected from the training set based on several criteria including representational and discriminative power, and balanced sampling across classes and spatial locations. Selecting a subset of patches that optimizes this objective reduces to a well studied problem of maximizing a *submodular set function*, which we introduce briefly before describing the specific terms in our patch selection objective function.

2.1. Submodular Function Optimization

Given a finite ground set $\mathcal{V}$, a set function $\mathcal{F} : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ assigns a value to each possible subset of $\mathcal{V}$. We say $\mathcal{F}$ is monotonically increasing if $\mathcal{F}(A) \leq \mathcal{F}(B)$ for all $A \subseteq B$. A set function $\mathcal{F}$ is submodular if $\mathcal{F}(A \cup a) - \mathcal{F}(A) \geq \mathcal{F}(A \cup \{a, b\}) - \mathcal{F}(A \cup b)$, for all $A \subseteq \mathcal{V}$ and $a, b \in \mathcal{V}/A$. This is often referred to as *diminishing return property*, as the benefit of adding each additional element to the set decreases as the size of the set grows.

While maximizing submodular set functions is NP-hard in general [4], a simple heuristic of greedy forward selection works well in practice and can be shown to have a $(1-1/e)$-approximation guarantee for monotonic functions [4, 23].

2.2. Patch Selection Objective Function

We generate a large set of candidate patches by sampling randomly and uniformly over spatial locations across the collection of training images. The patches could be represented by pixel values or other features such as SIFT [19]. In our experiments, we use activations from a pretrained CNN as our feature descriptor [13, 29]. We assume the subset of selected patches should be representative of all the patches in the feature space and yield discriminative compact clusters that are balanced across classes. In addition patches should be spatially cover most regions of the pollen grain. We now describe terms that encode each of these criteria.

Representative in feature space: Given a set of M patches which we denote $\mathcal{V}$, we construct a K-nearest neighbor weighted affinity graph specified by the matrix $\mathbf{S} \in \mathbb{R}^{M \times M}$ where $\mathbf{S}_{ij}$ is the similarity (a non-negative value) between patch i and patch j measured by the Euclidean distance. Our aim is to select a subset $A \subseteq \mathcal{V}$ consisting of patches that are representative in the sense that every patch in $\mathcal{V}$ is similar to some patches in the set A. We define the score of a set exemplars A as:

$$\mathcal{F}_R(A) = \sum_{j \in \mathcal{V}} \max_{i \in A} \mathbf{S}_{ij}, \tag{1}$$

This function is a monotonically increasing submodular function and can be seen as a special case of the facility location problem [4] where the costs of all the nodes are the same.

Spatially distributed in input space: Similar to the first term $\mathcal{F}_R$ which assures patches are representative in feature space, we would also like selected patches to be well distributed spatially in the input training images. We construct an affinity graph that stores the proximity of pairs of patches according to their coordinates on the pollen surface to assure that the selected exemplars to spread over the whole pollen grain. We denote this graph similarity matrix by $\mathbf{L} \in \mathbb{R}^{M \times M}$, and formulate it as the following

$$\mathcal{F}_S(A) = \sum_{j \in \mathcal{V}} \max_{i \in A} \mathbf{L}_{ij} \tag{2}$$

Discriminative power: Inspired by [10], we adopt a discriminative term to encourage selection of patches with discriminative power. For a given exemplar patch $i \in A$, we refer to the i^{th} cluster as the set of all patches in $\mathcal{V}$ which are more similar to i than to any other exemplar $C_i = \{j \in \mathcal{V} : S_{ij} > S_{kj} \, \forall k \in A/i\}$, breaking ties arbitrarily. We measure the discriminative power of such a clustering based on how pure the clusters are with respect to the category labels, while favoring a smaller number of clusters, given by:

$$\mathcal{F}_D(A) = \frac{1}{C} \sum_{i \in A} \max_c N_c^i - |A|, \tag{3}$$

where N_c^i is the number of exemplars from the c^{th} class that are assigned to the i^{th} cluster, and $C = \sum_{i \in A} C_i$. Eq. 3 is also a submodular function, and partial proof can be found in [10].

Class balance: We further adopt the balancing term introduced in [11] to balance the number of exemplars belonging to different classes:

$$\mathcal{F}_B(A) = \sum_c \log(|A_c| + 1) \tag{4}$$

where A_c is the subset of exemplars in A belonging to class c. The proof can be found in [11] that the above term is monotonically increasing and a submodular function.

Cluster compactness: In addition to balancing the size of exemplars of different classes, we would also like the clusters to be compact so that the total number of clusters is small and each exemplar represents roughly the same number of patches. We utilize the compactness term introduced in [18] as below:

$$\mathcal{F}_C(A) = -\sum_{i \in A} p(i) \log(p(i)) - |A| \tag{5}$$

where $p(i) = \frac{|C_i|}{|\mathcal{V}|}$ is the prior probability of a patch belonging to the i^{th} exemplar cluster. This is also a submodular function as shown in [18]. The above term will also favor a smaller number of clusters.

By combining these terms, our final objective function for selecting patches is given by:

$$\begin{aligned}\mathcal{F}(A) \equiv & \sum_{j=1}^{M} \max_{i \in A} \mathbf{S}_{ij} + \lambda_S \sum_{j=1}^{M} \max_{i \in A} \mathbf{L}_{ij} \\ & + \lambda_D \left(\frac{1}{C} \sum_{i \in A} \max_c N_c^i - |A| \right) \\ & + \lambda_B \sum_c \log(|A_c| + 1) \\ & + \lambda_C \left(-\sum_{i \in A} p(i) \log(p(i)) - |A| \right)\end{aligned} \tag{6}$$

where $\{\lambda_S, \lambda_D, \lambda_B, \lambda_C\}$ are hyperparameters that weigh the relative contribution of each term. We note that $\mathcal{F}(\varnothing) = 0$. As each term is a submodular function, our objective summing up all the five terms is also a submodular function. Therefore, we can easily use standard greedy approximation algorithms to approximately maximize the objective function.

2.3. Greedy Lazy Forward Selection

We sketch the naive greedy forward selection algorithm in Algorithm 1 to maximize our objective function. It is well known in literature that solving the submodular function by the greedy algorithm can yield near-optimal solution with a $(1-1/e)$-approximation bound [22]. However, while the complexity of this algorithm is linear in the number of exemplars selected and bounded by K, the computation in each iteration can be very time consuming. Each update has to recalculate the gains Δ for all the unselected patches remaining in $\mathcal{V}$ which makes direct application of the greedy method infeasible in practice.

Instead, we utilize the lazy greedy algorithm introduced in [22] using a max heap structure. The lazy greedy algorithm, sketched in Algorithm 2, maintains an expected gain for selecting each patch but only recomputes this gain when a patch becomes a candidate for selection. This avoids updating many of the gains associated with patches in $\mathcal{V}$ which

Algorithm 1 Greedy Selection Algorithm

```
Input: $\mathcal{V}, \mathcal{F}, K$
Output: a subset $A$ with $|A| \le K$
  initialize $A = \varnothing$, $k = 0$
  while $k \le K$ do
    for all $i \in \mathcal{V}/A$ do
      compute $\Delta(i) = \mathcal{F}(A \cup \{i\}) - \mathcal{F}(A)$
    end for
    $i^* = \arg\max_{i \in \mathcal{V}/A} \Delta(i)$
    if $\Delta(i^*) < 0$ then
      return $A$
    else
      $A = A \cup \{i^*\}, \quad k = k + 1$
    end if
  end while
  return $A$
```

Algorithm 2 Lazy Greedy Selection Algorithm

```
Input: $\mathcal{V}, \mathcal{F}, K$
Output: a subset $A$ with $|A| \le K$
  initialize $A = \varnothing$, iteration $k = 0$
  for all $i \in \mathcal{V}$, compute $\Delta(i) = \mathcal{F}(\{i\})$
  while $k \le K$ do
    $i^* = \arg\max_{i \in \mathcal{V}/A} \Delta(i)$
    compute $\Delta(i^*) = \mathcal{F}(A \cup \{i^*\}) - \mathcal{F}(A)$
    if $\Delta(i^*) \ge \max_{i \in \mathcal{V}/A} \Delta(i)$ then
      if $\Delta(i^*) < 0$ then
        return $A$
      else
        $A = A \cup \{i^*\}, \quad k = k + 1$
      end if
    end if
  end while
```

are already “covered” by an exemplar. This greedy algorithm with lazy updates is analyzed in [22] and provides a good approximation to the optimal solution of the NP-hard optimization problem. In our experiments, the lazy greedy Algorithm 2 yields good solutions and is hundreds of times faster than the naive greedy Algorithm 1. Specifically, the run time is less than ten minutes to select $K = 600$ exemplars from a pool of $10,000$ candidates on a single CPU.

3. Spatially Aware Coding for Fast Matching

The framework of sparse coding has been exploited for a number of computer vision tasks [33], *e.g.* image classification [12] and face recognition [34]. In standard coding-based classification, the individual patch appearance is represented by an abstract code vector while the spatial location of the patch in a test image is typically ignored. However, the spatial coordinates of a patch can be useful to encode information about the overall shape of a pollen grain and limit comparisons of local texture between grains to corresponding locations. Therefore, we propose to make use of the coordinates in the sparse coding procedure.

3.1. Spatially-aware Sparse Coding

Given a dictionary $\mathbf{D} \in \mathbb{R}^{p \times m}$, one can compute a sparse representation $\hat{\mathbf{a}} \in \mathbb{R}^m$ of a given input $\mathbf{x} \in \mathbb{R}^p$ over that dictionary by solving a sparse reconstruction problem:

$$\mathbf{a}^* = \operatorname*{argmin}_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_0. \tag{7}$$

The ℓ_0 norm $\|\cdot\|_0$ counts the number of non-zeros in a vector. We follow the standard approach of replacing this by an ℓ_1 norm $\|\cdot\|_1$ which yields a convex relaxation [1].

When learning a sparse coding model, it is common practice to learn a dictionary that is adapted to the dataset by minimizing the reconstruction error or other discriminative performance measures with respect to the dictionary elements [1, 33]. In our setup, we use the selected set of discriminative patch exemplars, as described in the previous section, directly as dictionary elements for sparse coding-based classification. One can thus view the selection process as a discriminative dictionary learning method (see, e.g. [12, 26, 20]).

In order to make use of patch coordinates, we modify the standard sparse coding objective by including a weight w_i associated with each dictionary element which encourages codes that are spatially coherent with respect to the training data. This weighting can be incorporated into the ℓ_1 sparsity term

$$\mathbf{a}^* = \operatorname*{argmin}_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda_1 \|\mathsf{diag}(\mathbf{w})\mathbf{a}\|_1, \tag{8}$$

or alternately by an additional weighted ℓ_2-norm penalizer

$$\mathbf{a}^* = \operatorname*{argmin}_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2 + \lambda_2 \|\mathsf{diag}(\mathbf{w})\mathbf{a}\|_2^2 + \lambda_1 \|\mathbf{a}\|_1. \tag{9}$$

The weight vector $\mathbf{w}$ will depend on the spatial location of the patch $\mathbf{x}$ in the test image. In particular, $\mathbf{w}_i$ depends on the difference in relative spatial location of the patch $\mathbf{x}$ and the location of the dictionary atom (exemplar patch) i in the training image. Dictionary atoms that were selected from a very different part of the pollen grain than the patch being coded are thus more heavily penalized for taking part in the reconstruction.

3.2. Fast Spatially-aware Coding

The spatially-aware sparse coding described above works quite well for performing classification. However, as the number of patches sampled in a test image increases, which will be a case if we desire better classification performance, the sparse coding process becomes computationally intensive. To address this problem, we propose a fast (relaxed) version of spatially aware sparse coding, which we term SACO for short.

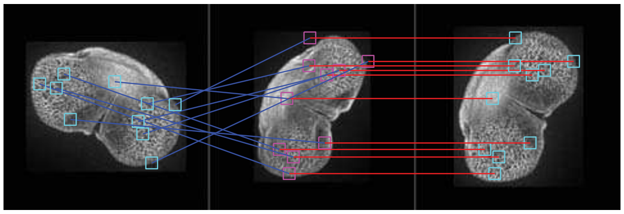

Figure 2. The success of our patch-based matching methodology requires that the two images are reasonably well aligned w.r.t viewpoint. We perform alignment to remove in-plane rotation and use k-medoids clustering to group training examples into canonical viewpoints. After alignment, the comparison of patches at corresponding spatial locations between a training (center) and test image (right) provides much stronger discriminative information than with an unaligned image (left). We exploit this alignment by utilizing a spatially adaptive sparse coding scheme we term SACO.

To motivate our approach, suppose we have an under-complete dictionary[2] $\mathbf{D} \in \mathbb{R}^{p\times m}$, $p \geq m$. Without the sparsity regularization, the reconstruction problem $\operatorname{argmin}_{\mathbf{a}} \|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2^2$ has a simple least-squares solution given by:

$$\mathbf{a}^* = \mathbf{\Omega}\mathbf{x}, \text{where } \mathbf{\Omega} \equiv (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T. \tag{10}$$

We thus consider an alternate cost function that seeks a sparse approximation to the (dense) least-squares code:

$$\mathbf{a}^* = \operatorname*{argmin}_{\mathbf{a}} \|\mathbf{\Omega}\mathbf{x} - \mathbf{a}\|_2^2 + \lambda_1\|\mathbf{a}\|_1 \tag{11}$$

For orthonormal dictionaries, e.g. as used in wavelet denoising [7, 28], $\mathbf{\Omega} = \mathbf{D}^{-1}$ and this problem is equivalent to the sparse reconstruction problem. In general, it provides an upper-bound on sparse reconstruction since

$$\|\mathbf{\Omega}(\mathbf{x} - \mathbf{D}\mathbf{a})\|_2 \geq \sigma(\mathbf{\Omega})\|\mathbf{x} - \mathbf{D}\mathbf{a}\|_2 \tag{12}$$

where $\sigma(\mathbf{\Omega})$ is a constant that depends on the dimension and smallest singular value of $\mathbf{D}$.

The primary appeal of this relaxed formulation is that we can easily obtain the optimal solution by applying a simple soft-thresholding or “shrinkage” function independently to each element of the least squares solution:

$$a_i^* = \operatorname{sgn}(u_i) \cdot \max(0, |u_i| - \lambda_1), \text{where } \mathbf{u} = \mathbf{\Omega}\mathbf{x}. \tag{13}$$

Spatial weighting In our problem, suppose we have an under-complete dictionary $\mathbf{D} \in \mathbb{R}^{p\times m}$ consisting of the selected patches and precompute corresponding pseudo-inverse $\mathbf{\Omega}$. We then solve the spatially-weighted variant corresponding to Eq. 8 by

$$\mathbf{a}^* = \operatorname*{argmin}_{\mathbf{a}} \|\mathbf{\Omega}\mathbf{x} - \mathbf{a}\|_2^2 + \lambda_1\|\mathsf{diag}(\mathbf{w})\mathbf{a}\|_1, \tag{14}$$

The solution is then given by $\mathbf{a}^*$ whose i^{th} element is the following:

$$a_i^* = \operatorname{sgn}(u_i) \cdot \max(0, |u_i| - \lambda_1 w_i), \text{where } \mathbf{u} = \mathbf{\Omega}\mathbf{x} \tag{15}$$

We term this scheme SACO-I.

Alternatively, for the counterpart of the ℓ_2 weighting used in Eq.9 we have

$$\begin{aligned} \mathbf{\Omega} \equiv& (\mathbf{D}^T\mathbf{D} + \lambda_2\mathsf{diag}(\mathbf{w})^2)^{-1}\mathbf{D}^T \\ \mathbf{u} =& \mathbf{\Omega}\mathbf{x} \\ a_i^* =& \operatorname{sgn}(u_i) \cdot \max(0, |u_i| - \lambda_1) \\ \mathbf{a}^* =& [a_1^*, \ldots, a_i^*, \ldots, a_m^*]^T. \end{aligned} \tag{16}$$

We term this scheme SACO-II.

Both versions of spatial structure aware shrink coding (SACO) enable us to do the coding in a feed-forward way without iterative optimization required by sparse reconstruction. This makes the classification process significantly more efficient than full reconstructive sparse coding. We find that in practice, using a non-overcomplete dictionary is not a limitation and that the SACO approximation leads to very good classification performance in our experiments. SACO-I has additional computational advantage over SACO-II as it avoids inverting a different matrix at each patch location. This makes it feasible to perform coding densely over the whole image by performing correlation over the whole image feature map with each element of $\mathbf{\Omega}$ followed by application of a spatially varying shrinkage function. Beyond SACO, we utilize global pooling and linear SVM for classification, as detailed in the next section.

4. Implementation Details

4.1. k-medoids Clustering for Viewpoint Alignment

As demonstrated by Figure 2, the success of our spatially-aware patch-based matching methodology lies in that the images are well aligned w.r.t viewpoint. To align the images, we perform unsupervised pre-processing of both training and test images in order to automatically improve alignment.

We use the all the training images (ignoring the species labels) to build an affinity graph, where the similarity of image $\mathbf{I}_A$ and $\mathbf{I}_B$ is measured by

$$similarity(\mathbf{I}_A, \mathbf{I}_B) = \frac{1}{\min_\theta \|\mathbf{I}_A - R_\theta(\mathbf{I}_B)\|}, \tag{17}$$

where $R_\theta(\mathbf{I}_B)$ is an operator that rotates image $\mathbf{I}_B$ by θ degrees. We resize all images and rotated intermediates to 40×40 pixel resolution, and calculate the distance between two images as the sum of squared pixel-wise differences. We use the resulting similarity graph to perform k-medoids clustering. Empirically, we find that once in-plane rotation is removed, clustering the images into two canonical

[2] In our experiments this is indeed the case since the patch feature dimension is larger than the number of exemplar patches

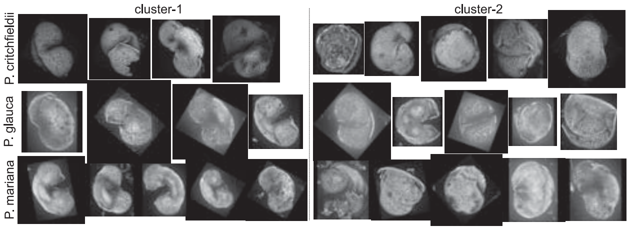

Figure 3. Rotated images according to two canonical viewpoints determined by k-medoids clustering.

viewpoints is enough to achieve good performance. Figure 3 shows two viewpoint clusters with examples from all three species. Equatorial views of pollen grains appear in the first cluster while top-down views are assigned to the second cluster.

4.2. Classification Pipeline

Using the dictionary $\mathbf{D} \in \mathbb{R}^{p\times m}$ constructed by the selected exemplar patches from training images, we perform spatially-aware aware coding (SACO) for patches of the test image, resulting in m-dimensional sparse codes for each patch. For each test image we use 50 patches sampled at random. We pool the m-dimensional code vectors over the entire set of test patches using average pooling to produce a final m-dimensional feature vector which is fed into a linear SVM classifier to predict the species.

Rather than using raw image pixel values, we use a feature vector extracted via the pretrained VGG19 model of [29]. We found that using the features at layer-$conv4\_3$ of VGG19 performed best. We also analyzed the performance of SIFT and features at other layers of VGG19 in our experiments (Section 5). The receptive field at this layer spans a patch of 52×52 pixels in the original image. Figure 2 shows a visualization of these selected patches relative to the scale of the pollen grain and shows qualitatively that patches capture meaningful local textures.

5. Experiments

In this section, we introduce our dataset, show the effectiveness of the proposed exemplar selection method on synthetic data, study different features used for classification and several hyper-parameters in our pipeline, and report the classification performance of our models and comparisons to several strong baselines.

5.1. Dataset

We test our method on samples of fossil pollen from three species of spruce, *Picea critchfieldii*, *Picea glauca*, and *Piciea mariana*. Samples were chemically extracted from lake sediments as detailed in [25, 21] and imaged using a Zeiss Apotome fluorescence microscope (a form of structured illumination) [32] to produce high-resolution, three-dimensional image stacks. Imaging was carried out by multiple operators, with no single person responsible for a single species. The full shape of the grain was captured by multiple z-focal planes at intervals of half the Nyquist frequency [25]. Grains were cropped manually, using a bounding box that reached from the maximum width of the grain in the x axis and the maximum length of the grain in the y-axis. The z-stack is limited to the uppermost and lowermost in-focus planes of the grain. Details of the imaging procedure can be found in [anon]. For each grain, we use maximum intensity projection over the top half of the grain to produce a single in focus 2D image. Some examples are show in Figure 1.

Table 1. Statistics of our fossil pollen grain dataset.

	#train	#test	#total
P. critchfieldii	65	43	108
P. glauca	65	355	420
P. mariana	65	287	352
Summary	195	685	880

Experts provided a nominal species label for each grain along with a confidence score. We note that unlike some other image classification tasks, there is no “ground-truth” for species identification. However, fossil pollen grains were taken from strata containing other macro-fossil evidence (e.g., leaves) of these species and we restricted our analysis to samples with high-confidence labels. We randomly split the dataset into training and testing (validation) sets (statistics are listed in Table 1). The dataset will be released to public in the near future.

5.2. Exemplar Selection on Synthetic Data

We first verified the effectiveness of the proposed exemplar selection method using synthetic toy data for which we can easily visualize the results qualitatively. In this setting we merge term $\mathcal{F}_R$ and $\mathcal{F}_S$ in the objective function (Eq. 6) to simplify our analysis as the 2D data themselves are both features and spatial coordinates. As can be seen in Figure 4, the greedy lazy algorithm selects representative exemplars that cover the data points from each class while maintaining discriminative power by sampling near class boundaries but avoid non-discriminative areas of feature space with high inter-class overlap.

5.3. Choice of Feature Representation

To study the choice of feature representation, we compare the performance of SIFT descriptors [19] and features extracted at different layers of VGG19 model [29]. Using SACO-I with SIFT feature yields 54.40% classification accuracy while CNN features perform much better. Figure 5 shows performance of features extracted at different layers of the VGG19 hierarchy with layer $conv4\_3$ achieving a performance at 77.62% classification accuracy. Receptive fields at this layer span 52×52 pixel patches in the original image and are visualized in in Figure 2. We use these features in our remaining classification experiments.

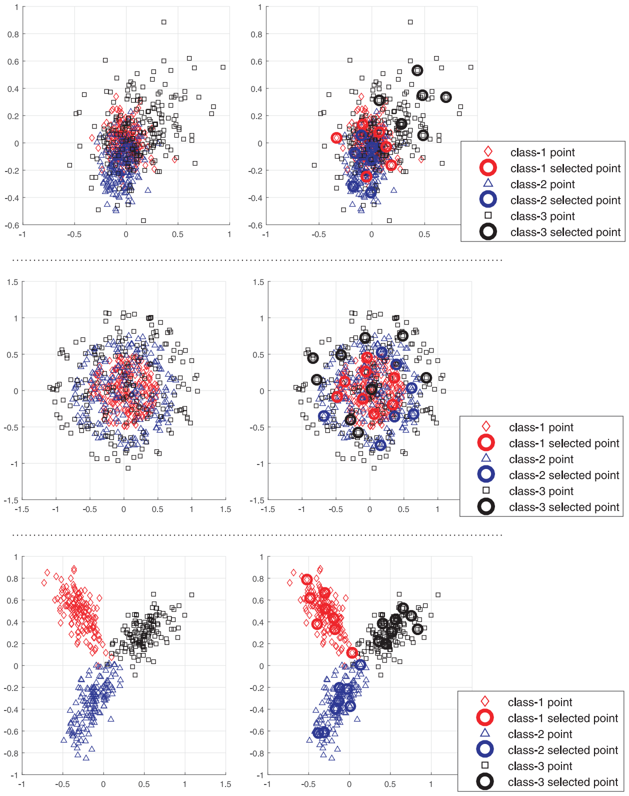

Figure 4. Qualitative demonstration of the effectiveness of the proposed method in exemplar selection on synthetic data (best seen in color and zoom-in)

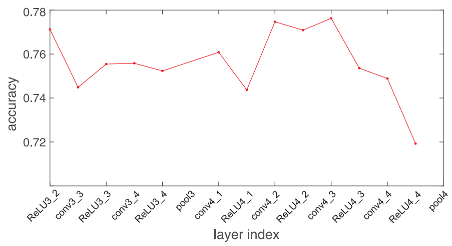

Figure 5. Classification accuracy vs. layer index in VGG19 model. We use features extracted from $conv4\_3$ in the remainder of our experiments.

dictionary size	300	512	600
Random Selection	77.66	76.49	77.23
Discriminative Selection	81.75	81.60	82.34

Table 2. Classification accuracy (%) for different sized dictionaries constructed by our discriminative exemplar selection algorithm. Our method consistently outperforms a baseline that selects patches at random from the training set.

5.4. Evaluation of Dictionary Learning

In addition to the synthetic tests in Section 5.2, we verify the effectiveness of our exemplar selection method in the pollen identification task by comparing the classification performance of dictionaries consisting of randomly sampled patches. We also report the performance as a function of varying dictionary size.

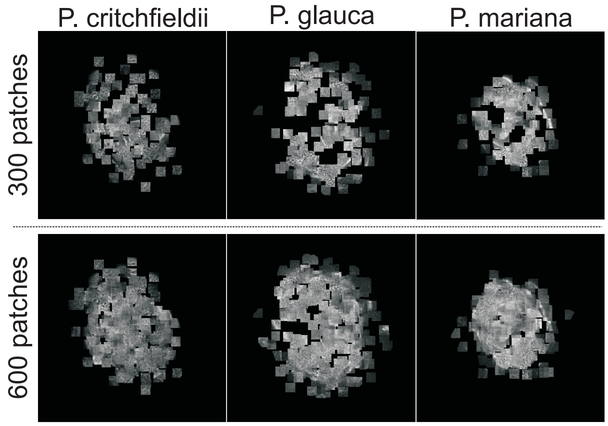

Figure 6. To visualize the selected patches, we paste patches selected from the same species on a black background to the places according to their coordinates. The two panels show the 300 and 600 patches respectively. Our dictionary selection approach favors patches that cover the pollen grain spatially, focusing on more discriminative regions when the number of dictionary elements is limited (300 patches) but eventually covering the whole grain at larger dictionary sizes (600 patches).

SRC	VGG19+SVM	FV+SVM	SACO-I	SACO-II
62.04	65.11	61.46	83.21	86.13

Table 3. Performance of baselines and our SACO methods measured by classification accuracy (%).

We use SACO-I for this experiment, and vary the dictionary size by (randomly) selecting 300, 512 and 600 patches. The results are listed in Table 2. First, it is clear that a dictionary built from our selected exemplars performs much better than the counterpart consisting of randomly sampled patches. Second, a smaller dictionary of 300 atoms is sufficient for our classification task. However, it appears larger dictionaries do not harm performance. We expect that some hyper-parameters will have an important impact on the performance for larger dictionaries, in particular regularization of the SVM. We study the effect of hyper-parameters in Section 5.6. We use a 300-basis dictionary for the rest of our experiments.

To visualize the selected patches in the dictionary, we paste them on a black panel according to their coordinates. Figure 6 shows the patches of the three species. We can see that these patches not only capture local texture information, but also convey a global shape and average size of the three species.

5.5. Comparison of SACO Methods and Baselines

We report the classification performance of our proposed two variants of SACO along with several baselines in Table 3. "SRC" is the sparse-representation based classification method of [34] using the reconstruction error to identify species. "VGG19+SVM" is a standard CNN-based image classification approach that applies the VGG19 model to the whole image and performs classification using a lin-

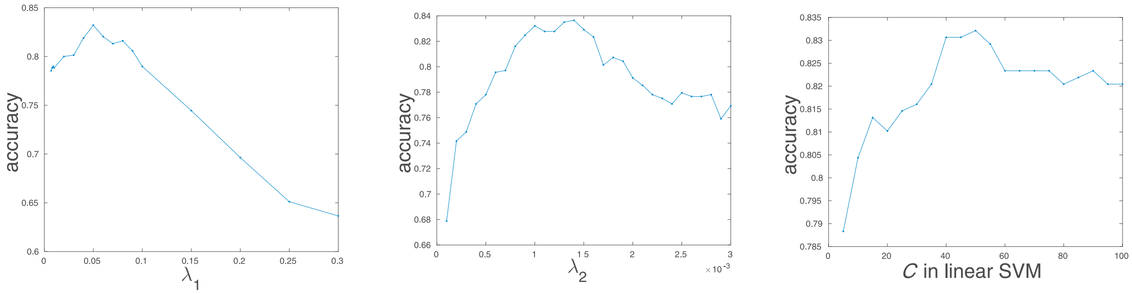

Figure 7. The effect of sparisty (λ_1 in Eq. 16), spatial weighting (λ_2 in Eq. 16), and SVM regularization (C) parameter on performance.

ear SVM applied to features from a high-level layer [6]. "FV+SVM" uses Fisher Vector [24] to pool features at a specific layer of VGG19 to represent the entire image, and applies an SVM classifier [3]. For the VGG19 baselines, we tried features at different layers of VGG19, and report the best result here.

VGG19+SVM provides a strong baseline for shape-based object recognition while FV+SVM has shown strong performance on texture classification [3]. However, neither of these standard methods is competitive with SACO. It is worth noting that, if fine-tuning the VGG model with softmax loss, we only obtain 52.41% accuracy. We posit two reasons. First the original images are of high resolution, so it is easy to overfit the training set. Second, if we down size the images, the valuable textural characteristics will be eliminated. SRC is closer to our approach but also performs significantly worse. When using random patches without the spatial information as a dictionary in SRC, we only achieve an accuracy of 57.12%. Adding spatial information and using selected patches in SRC improves performance to 62.04%. Performing average pooling over the sparse codes, *i.e.* using our SACO-I method, provides a substantial improvement in performance, reaching 83.21% classification accuracy. The alternative method SACO-II yields even better performance, 86.13%. This shows the importance of the spatial information of patches in our task and the need to fuse both shape and texture cues.

5.6. Parameter sensitivity

There are several important hyper-parameters in our pipeline, including the sparsity λ_1, spatial weighting λ_2 (see Eq. 16), and regularization parameter C in the linear SVM. Figure 7 shows accuracy as a function of each of these hyper-parameters. The curve showing accuracy as a function of λ_1 shows that inducing sparsity improves classification performance notably. The second curve showing accuracy vs. λ_2, makes it clear that incorporating spatially-varying weights on the dictionary elements also improves the classification performance remarkably. However, it is necessary to jointly tune both λ_1 and λ_2 for best performance. Last, we note the performance is stable w.r.t the parameter C in linear SVM over a large range.

5.7. Dense convolutional SACO

An intriguing aspect of the SACO-I formulation is that it is amenable to a dense implementation that performs coding at every patch location in the test image. This is implemented by first correlating the input image or feature map with each element of the pseudo-inverse dictionary $\mathbf{\Omega}$ followed by soft-thresholding of each response map with a spatially varying threshold and pooling the result. In theory SACO-II could also be applied densely but demands significantly more computation since $\mathbf{\Omega}$ is spatially varying and would require computing the matrix inverse (Eq. 16) at every location.

Using a fully convolutional implementation of SACO-I achieved 83.86% classification accuracy. Although we do not see significant improvement over the sparse sampling of test patches, we believe better performance may ultimately be achieved in the dense evaluation by incorporating automatic segmentation of the pollen grain from background noise and masking of uninformative damaged areas. We plan to explore these possibilities in future work.

6. Conclusion and Future Work

We propose a robust framework for pollen grain identification by matching testing images with a set of discriminative patches selected beforehand from a training set. To select the discriminative patches, we introduce a novel selection approach based on submodular maximization, which is very efficient and effective in practice. To identify pollen grains using the selected patches as a dictionary, we present two spatially-aware sparse coding methods. We further accelerate these two methods using a relaxed formulation that can be computed in an efficient non-iterative manner.

As our experiments show, this spatially aware exemplar-based coding approach significantly outperforms strong baselines built on state-of-the-art CNN features. We leave open as future work the question of how such a matching mechanism could be fully embedded in a neural network architecture, how to exploit confidence scores provided with expert labels, and extending the approach to perform cross-domain matching of fossil and modern pollen samples.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006. 4

[2] C. Chen, E. A. Hendriks, R. P. Duin, J. H. Reiber, P. S. Hiemstra, L. A. de Weger, and B. C. Stoel. Feasibility study on automated recognition of allergenic pollen: grass, birch and mugwort. *Aerobiologia*, 22(4):275–284, 2006. 1

[3] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3828–3836, 2015. 8

[4] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. Technical report, DTIC Document, 1983. 2, 3

[5] R. DellAnna, P. Lazzeri, M. Frisanco, F. Monti, F. M. Campeggi, E. Gottardini, and M. Bersani. Pollen discrimination and classification by fourier transform infrared (ft-ir) microspectroscopy and machine learning. *Analytical and bioanalytical chemistry*, 394(5):1443–1452, 2009. 1

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013. 8

[7] D. L. Donoho, I. M. Johnstone, et al. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes Rendus de l'Academie des Sciences-Serie I-Mathematique*, 319(12):1317–1322, 1994. 5

[8] I. France, A. Duller, G. Duller, and H. Lamb. A new approach to automated pollen analysis. *Quaternary Science Reviews*, 19(6):537–546, 2000. 1

[9] K. Holt, G. Allen, R. Hodgson, S. Marsland, and J. Flenley. Progress towards an automated trainable pollen location and classifier system for use in the palynology laboratory. *Review of Palaeobotany and Palynology*, 167(3):175–183, 2011. 1

[10] Z. Jiang, G. Zhang, and L. S. Davis. Submodular dictionary learning for sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3418–3425. IEEE, 2012. 3

[11] S. Kong, Z. Jiang, and Q. Yang. Collaborative receptive field learning. *arXiv preprint arXiv:1402.0170*, 2014. 3

[12] S. Kong and D. Wang. A dictionary learning approach for classification: separating the particularity and the commonality. In *Computer Vision–ECCV 2012*, pages 186–199. Springer, 2012. 4

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[14] S. H. Landsmeer, E. A. Hendriks, L. A. De Weger, J. H. Reiber, B. C. Stoel, et al. Detection of pollen grains in multifocal optical microscopy images of air samples. *Microscopy research and technique*, 72(6):424–430, 2009. 1

[15] M. Langford, G. Taylor, and J. Flenley. Computerized identification of pollen grains by texture analysis. *Review of Palaeobotany and Palynology*, 64(1):197–203, 1990. 1

[16] P. Li and J. R. Flenley. Pollen texture identification using neural networks. *Grana*, 38(1):59–64, 1999. 1

[17] P. Li, W. Treloar, J. Flenley, and L. Empson. Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification of optical images of pollen grains. *Journal of quaternary science*, 19(8):755–762, 2004. 1

[18] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011. 3

[19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 2, 6

[20] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009. 4

[21] L. Mander, J. Rodriguez, P. G. Mueller, S. T. Jackson, and S. W. Punyasena. Identifying the pollen of an extinct spruce species in the late quaternary sediments of the tunica hills region, south-eastern united states. *Journal of Quaternary Science*, 29(7):711–721, 2014. 6

[22] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978. 3, 4

[23] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978. 2

[24] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010. 8

[25] S. W. Punyasena, D. K. Tcheng, C. Wesseln, and P. G. Mueller. Classifying black and white spruce pollen using layered machine learning. *New Phytologist*, 196(3):937–944, 2012. 1, 6

[26] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010. 4

[27] O. Ronneberger, E. Schultz, and H. Burkhardt. Automated pollen recognition using 3d volume images from fluorescence microscopy. *Aerobiologia*, 18(2):107–115, 2002. 1

[28] E. P. Simoncelli and E. H. Adelson. Noise removal via bayesian wavelet coring. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 379–382. IEEE, 1996. 5

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 6

[30] D. K. Tcheng, A. K. Nayak, C. C. Fowlkes, and S. W. Punyasena. Visual recognition software for binary classification and its application to spruce pollen identification. *PloS one*, 11(2):e0148879, 2016. 1

[31] W. Treloar, G. Taylor, and J. Flenley. Towards automation of palynology 1: analysis of pollen shape and ornamentation using simple geometric measures, derived from scanning electron microscope images. *Journal of quaternary science*, 19(8):745–754, 2004. 1

[32] A. Weigel, D. Schild, and A. Zeug. Resolution in the apotome and the confocal laser scanning microscope: comparison. *Journal of biomedical optics*, 14(1):014022–014022, 2009. 6

[33] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010. 4

[34] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009. 4, 7

[35] Y. Zhang, D. Fountain, R. Hodgson, J. Flenley, and S. Gunetileke. Towards automation of palynology 3: pollen pattern recognition using gabor transforms and digital moments. *Journal of quaternary science*, 19(8):763–768, 2004. 1