Insect Species Recognition using Discriminative Local Soft Coding

An Lu† Xinwen Hou† Cheng-Lin Liu† Xiaolin Chen‡
†NLPR Institute of Automation, Chinese Academy of Sciences
‡Institute of Zoology, Chinese Academy of Sciences
†{alu, xwhou, liucl}@nlpr.ia.ac.cn ‡xlchen@ioz.ac.cn

Abstract

Insect species recognition is more difficult than generic object recognition because of the similarity between different species. In this paper, we propose a hybrid approach called discriminative local soft coding (DLSoft) which combines local and discriminative coding strategies together. Our method takes use of neighbor codewords to get a local soft coding and class specific codebooks (sets of codewords) for a discriminative representation. On obtaining the vector representation of image via spatial pyramid pooling of patches, a linear SVM classifier is used to classify images into species. Experimental results show that the proposed method performs well on insect species recognition and outperforms the state-of-the-art methods on generic object categorization.

1. Introduction

Insect species recognition is widely applied in agriculture, ecology, and environmental science. In traditional way, it is always manually operated by expert entomologists. However for laymen without any professional knowledge, it is hard to discriminate different insect categories in the level of species. In this case, to deal with the problem, insect species recognition using computer vision methods may be a feasible solution. Comparing to generic object recognition, most insects are composed of several sub-parts (legs, antennae, tails, wings, etc.) and many degrees of freedom so that they are more difficult to be identified. The goal of our research is to develop a computer vision application to partly take place of expert entomologists in some areas and help entomologists lighten heavy labor in their researches.

To solve the difficult insect recognition problem we propose a method motivated by soft coding [1] and salient coding [2] methods. These two methods are state-of-the-art for vision object recognition, but both lose some discriminative information in the codebook generation process and local information in the subsequent coding process. So, we take use of class specific codebooks to improve the discriminability of codewords and get more robust representations of feature vectors with smaller reconstruction error in a local and soft coding strategy. We find that the locality of coding has a manifold explanation which can prove the rationality of our approach. Experiments on insect species dataset and generic image categorization dataset show the validity of our approach.

2. Related Works

Over these years several efficient approaches to generic object categorization have appeared. Among them, the most popular methods are based on bag-of-features framework. These methods work by partitioning an image into small patches, computing a codebook and take some coding strategies to represent an image by a vector.

Let \( X \) be a set of feature vectors such as SIFT appearance descriptors in a d-dimensional feature space. \( X \) can be represent as \( X = [x_1, x_2, \ldots] \in \mathbb{R}^{d \times n} \), where \( d \) is the dimensional number of feature vectors and \( n \) is the number of feature vectors in the set. Codebook can be expressed as \( B = [b_1, b_2, \ldots] \in \mathbb{R}^{d \times m} \) and the response (coefficients) of a feature vector is present as \( V = [v_1, v_2, \ldots] \in \mathbb{R}^{1 \times m} \), where \( m \) is the number of codewords.

To represent a feature vector \( x \), hard coding methods [3] assign 1 to the nearest codeword and 0 to the others:

\[
v_i = \begin{cases} 
1, & \text{if } i = \arg \min_j (\|x - b_j\|_2) \\
0, & \text{otherwise}
\end{cases}
\]  

(1)

So a coefficient is equal to a frequency histogram. There is a reconstruction bias between \( x \) and \( b_j \).
Consequently the coefficient can not exactly express the distribution of feature vectors.

In soft coding strategies [1], a feature vector \( x \) is encoded by a kernel function of distance between the feature vectors and each codeword:

\[
v_i = K_\sigma(\|x-b_i\|_2) \tag{2}
\]

These methods outperform hard coding methods, however, they do not take use of any local and salient information.

Reconstruction methods are popular in recent years. The sparse coding [4] method relaxes the cardinality constraint of hard coding methods as:

\[
V = \arg \min_S \|x-BS\|_2^2 + \lambda |S| \tag{3}
\]

Other reconstruction methods such as LCC [5] and its variant LLC [6] achieve good performance on many image databases such as Caltech101 and PASCAL VOC2009. However, the computation costs of these methods are high because of the iteration process.

Salient coding [2] is a hybrid method that combines hard coding and reconstruction coding:

\[
v_i = \begin{cases} 
\Psi(x, b_i), & \text{if } i = \arg \min_j (\|x-b_j\|) \\
0, & \text{otherwise} 
\end{cases} \tag{4}
\]

\[
\Psi(x, b_i) = \Phi \left( \frac{\|x-b_i\|_2}{\sum_{j=1}^{k} ||x-b_j||_2} \right), \quad \Phi(z) = 1-z
\]

\( k \) is the number of neighbor codewords near the feature vector \( x \). The drawback of this method is that only one dimension of \( V \) is reserved while other dimensions that may have useful information are discarded.

3. Discriminative Local Soft Coding

Considering the drawbacks of the methods above, we propose a new coding strategy that is hybrid of the soft coding and the local reconstruction coding. [7, 8] also refer to some similar local and soft strategies. However in [7], their work focuses on image retrieval. And in [8], the authors treat each neighbor assignment separately in order to use Multiple Kernel Learning (MKL) for classification. The formulation is as following:

\[
v_i = \begin{cases} 
K_\sigma(\|x-b_i\|_2)/Z, & \text{if } b_i \in N_k(x) \\
0, & \text{otherwise} 
\end{cases} \tag{5}
\]

\( K_\sigma \) is a kernel function, \( N_k \) is the \( k \) nearest neighbor codewords of \( x \), and \( Z \) is a normalization factor. The kernel function we use here is Gaussian kernel:

\[
K_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \tag{6}
\]

The comparison of local soft coding method and others are shown in Fig.1 below.

![Figure 1: A demonstration of the five kinds of coding strategies.](image)

After that a reconstruction residual \( \sigma = x-\tilde{x}^p \) is calculated for each coefficient vector \( v_i^p \), where \( \tilde{x}^p \) is the reconstruction of \( x \). We choose the coefficient vector with the smallest residual, set the others to zero and concatenate all the \( C \) coefficient vectors together according to their cardinate:

\[
ind = \arg \min_p \|x-\tilde{x}^p\| \\
V = \left( \sqrt{\sum_{i=1}^{m_1} \hat{v}_{i1}^p \sum_{i=2}^{m_2} \hat{v}_{i2}^p \sum_{i=3}^{m_3} \hat{v}_{i3}^p} \right)
\tag{8}
\]

After calculating the concatenated coefficients for each feature vector, max pooling is used to combine these coefficients belonging to the same image together to obtain the final feature vectors. Then any learning method such as neural networks or SVM is competent for the recognition task.

Our approach also has a manifold explanation. The feature vector of each class lies on respective manifold.
Each manifold can be reconstructed from neighbor codewords with weights computed by heat kernel [9]. The class label of a new feature vector is determined by minimal reconstruction error on C manifolds.

4. Application to insect species recognition

Tephritidae (fruit fly) is a family of insect which contains about 500 genera and about 4200 species. The application requirement of our approach is to recognize different Tephritidae species without any interaction in both training and testing stages. A dataset is divided into two parts: training dataset and testing dataset. For both datasets, a sample is an image of Tephritidae. We transform all the images into gray scale and extract SIFT features of patches by densely sampled from each image. So an image sample can be represented by a set of SIFT features. For local soft coding method, we get the codebook $B$ by k-means clustering and compute the local soft codes by Eq.5 and Eq.6. For discriminative local soft coding method, we get a codebook $B'$ for each species also by k-means clustering and calculate the final coefficient codes by Eq.7 and Eq.8. Subsequently we calculate max pooling over all the vectors of patches in the same image. The spatial pyramid pooled vector [4] is the final representation of an image sample which can be used by any machine learning method for classification.

Our Tephritidae dataset is composed of 3 genus and 20 species. Each specimen is taken one photograph respectively of its whole body, head, thorax, abdomen and wing (as shown in Fig.2). So we divide the whole dataset into 5 sub-dataset according to different part of the specimen. Tab.1 shows the number of species, photographs and codewords in the 5 sub-datasets.

<table>
<thead>
<tr>
<th>Sub-dataset</th>
<th>Insect</th>
<th>Head</th>
<th>Thorax</th>
<th>Abdomen</th>
<th>Wing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td>19</td>
<td>22</td>
<td>20</td>
<td>17</td>
<td>14</td>
</tr>
<tr>
<td>Images</td>
<td>152</td>
<td>143</td>
<td>151</td>
<td>144</td>
<td>103</td>
</tr>
<tr>
<td>#CWs</td>
<td>1520</td>
<td>1520</td>
<td>1600</td>
<td>1360</td>
<td>1120</td>
</tr>
</tbody>
</table>

Table 1: row2-Number of species, row3-Number of images, row4-Number of codewords for each sub-dataset.

5. Experimental results

In the following experiments, we evaluate our local soft coding method and its discriminative variation on the Tephritidae dataset and Caltech101 dataset [10]. We extract SIFT descriptor from $16\times16$ pixel patches which are densely sampled from each image on a grid with a step size of 8 pixels. We use multi-layer max pooling [4] to combine patch features into a coding vector with linear SVM for training and evaluate our result by randomly splitting training and testing datasets 10 times.

Firstly, we show our results on the Tephritidae dataset. We respectively take one and two samples from each species for training and the remaining for testing. Then we set the number of layers of spatial pyramid as 3 and the number of codewords as 80 for each species. The numbers of total codewords (80xnumber of species) for each sub-dataset are shown in the 3rd row in Tab.1. We compare our local soft coding (LS) method and discriminative local soft coding method (DLS) with sparse coding spatial pyramid matching (ScSPM) [4], locality-constrained linear coding (LLC) [6], soft coding (Soft) [1] and salient coding (SaC) [2]. Discriminative soft coding (DSoft) is similar to DLS just without the locality process of Eq.5. We use it to evaluate the efficiency of the locality process in DLS. We set the number of nearest neighbor as 5 or 10 in SaC, LLC and DLS. As shown in Tab.2 and Tab.3, the accuracy of DLS outperforms other coding methods almost in all sub-datasets. The better results comparing to those of DSoft prove the efficiency of locality process. And the high accuracies of the two discriminative methods (DLS and DSoft) indicate that using codebooks calculated from different species is beneficial to catch some discriminative information for categorization.

Secondly, we test our methods on Caltech101 dataset (with 102 categories). We set the total number of codewords as 2040 and the number of nearest neighbor codewords as 5. We randomly take 15 or 30 images as training dataset and the remaining images as testing dataset. The result is shown in Tab.4. According to the result, we can find that our DLS method can also outperform the other ones in generic image datasets. The result shows that the locality and discriminative process are both efficient so that our DLS method hybrid of the two strategies get the best accuracy. As we notice, the cause that the SaC method result is worse than ScSPM may be that the number of codewords is not very large (in [7] the number of codewords is 24000). Unlike SaC, our method only makes use of the k nearest neighbor

<table>
<thead>
<tr>
<th>Sub-dataset Methods</th>
<th>Insect</th>
<th>Head</th>
<th>Thorax</th>
<th>Abdomen</th>
<th>Wing</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLSoft</td>
<td>24.23</td>
<td>29.56</td>
<td>37.87</td>
<td>33.30</td>
<td>35.54</td>
</tr>
<tr>
<td>DSoft</td>
<td>23.41</td>
<td>29.02</td>
<td>35.22</td>
<td>30.96</td>
<td>27.99</td>
</tr>
<tr>
<td>LSoft</td>
<td>18.53</td>
<td>19.86</td>
<td>25.01</td>
<td>26.07</td>
<td>33.94</td>
</tr>
<tr>
<td>Sac</td>
<td>20.00</td>
<td>19.31</td>
<td>26.70</td>
<td>27.85</td>
<td>29.23</td>
</tr>
<tr>
<td>Soft</td>
<td>17.92</td>
<td>21.66</td>
<td>25.62</td>
<td>27.42</td>
<td>23.25</td>
</tr>
<tr>
<td>ScSPM</td>
<td>17.41</td>
<td>20.69</td>
<td>25.33</td>
<td>24.40</td>
<td>31.23</td>
</tr>
<tr>
<td>LLC</td>
<td>17.62</td>
<td>21.85</td>
<td>26.96</td>
<td>24.35</td>
<td>33.27</td>
</tr>
</tbody>
</table>

Table 2: Recognition accuracy of the 7 methods with 1 training sample.
codewords to represent the feature vector. So the number of codewords needs not to be so large and the computational cost can be lowered to adapt to realistic applications.

Table 3: Recognition accuracy of the 7 methods with 2 training samples.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Insect</th>
<th>Head</th>
<th>Thorax</th>
<th>Abdomen</th>
<th>Wing</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLSoft</td>
<td>34.56</td>
<td>37.02</td>
<td>54.69</td>
<td>43.38</td>
<td>45.02</td>
</tr>
<tr>
<td>Dsoft</td>
<td>33.57</td>
<td>35.61</td>
<td>53.08</td>
<td>40.16</td>
<td>41.47</td>
</tr>
<tr>
<td>Lsoft</td>
<td>25.75</td>
<td>21.74</td>
<td>37.97</td>
<td>31.58</td>
<td>40.99</td>
</tr>
<tr>
<td>Sac</td>
<td>32.80</td>
<td>23.46</td>
<td>33.50</td>
<td>31.91</td>
<td>42.02</td>
</tr>
<tr>
<td>Soft</td>
<td>22.97</td>
<td>22.62</td>
<td>35.82</td>
<td>33.60</td>
<td>36.87</td>
</tr>
<tr>
<td>ScSPM</td>
<td>25.12</td>
<td>24.44</td>
<td>33.86</td>
<td>33.02</td>
<td>37.05</td>
</tr>
<tr>
<td>LLC</td>
<td>27.40</td>
<td>27.55</td>
<td>35.41</td>
<td>34.71</td>
<td>37.65</td>
</tr>
</tbody>
</table>

Table 4: Recognition accuracy of different methods on Caltech101 dataset. The number in the first row indicates the number of training data.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Number of training data</th>
<th>15</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLSoft</td>
<td>68.80±1.22</td>
<td>75.26±1.26</td>
<td></td>
</tr>
<tr>
<td>Dsoft</td>
<td>67.21±0.81</td>
<td>74.67±0.76</td>
<td></td>
</tr>
<tr>
<td>Lsoft</td>
<td>65.77±0.72</td>
<td>74.21±0.92</td>
<td></td>
</tr>
<tr>
<td>Sac</td>
<td>61.21±0.66</td>
<td>69.77±1.04</td>
<td></td>
</tr>
<tr>
<td>Soft</td>
<td>63.72±0.98</td>
<td>71.44±0.78</td>
<td></td>
</tr>
<tr>
<td>ScSPM</td>
<td>65.46±0.69</td>
<td>73.71±0.53</td>
<td></td>
</tr>
<tr>
<td>LLC</td>
<td>66.72±0.59</td>
<td>74.13±0.72</td>
<td></td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper we proposed a discriminative local soft coding method for insect species recognition. This method takes use of the local soft coding within k nearest neighbor codewords of a feature vector and keeps discriminative information by generating codebooks for each species respectively. The experimental results on the Tephritidae dataset and Caltech101 dataset demonstrate the efficiency and effectiveness of our method. We believe that the locality process preserves the manifold structure of feature vectors and the class specific codebook generation strategy gains more discriminative information for classification.

In the future, we will study the influence of different kinds of kernel functions and analyze the theoretical basis of DLSoft method by experiments on more image datasets. Some prior attributes of the insect species may also be used to improve the recognition accuracy.

Acknowledgements

This work was supported in part by the 973 Program Grant 2012CB316302, NSFC Grant 60825301, and the Strategic Priority Research Program of the CAS (Grant XDA06030300).

References


Figure 2: Example of our Tephritidae dataset: each column is corresponding to one species and the rows are respectively whole body, head, thorax, abdomen and wing photographs of the corresponding species taken by a microscope camera.