AN ABSTRACT OF THE THESIS OF

Yaofei Feng for the degree of Master of Science in Computer Science presented on May 31, 2013.

Title: Fine-grained Detection and Localization of Objects in Images

Abstract approved: ____

Sinisa Todorovic

Object recognition is a fundamental problem in computer vision. Recognition is required by many applications. This thesis presents a distance based approach to recognize objects. We are interested in objects that belong to very similar classes, where each class has large variations. This problem is called fine-grained object recognition. Given a set of training images our approach identifies a sparse number of image patches in the training set which cover the most parts of the target object in the test image. We use Hungarian algorithm to match the image patches, based on a linear combination of appearance and geometric image features. We also specify a voting scheme for each possible location of the target object in the test image. The location which is close to the training image center is more likely to be the object center in the test image. Our results on a set of the challenging benchmark datasets are promising. This suggests that our approach is suitable to effectively address fine-grained object recognition.

Fine-grained Detection and Localization of Objects in Images

by

Yaofei Feng

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented May 31, 2013
Commencement June 2013

Master of Science thesis of Yaofei Feng presented on May 31, 2013.

APPROVED:

Major Professor, representing Computer Science

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Yaofei Feng, Author

ACKNOWLEDGEMENTS

I would like to thank everyone who has helped me on my thesis, specially my advisor Sinisa Todorovic. He has been extremely supportive and helpful in my graduate studies and guided me through my master degree.

I also would like to express my appreciation to all my committee members for serving on my thesis committee and giving me assistance and suggestions. Thanks to Prof. Metoyer, I really enjoy the experience of working with you. Thanks to Prof. Fern for giving me precious guidance in machine learning research area.

Last, thanks to my parents and my friends. With your love and understanding, I found the value of myself. Your encouragement made me more stronger and your smile made more optimistic.

TABLE OF CONTENTS

LIST OF FIGURES

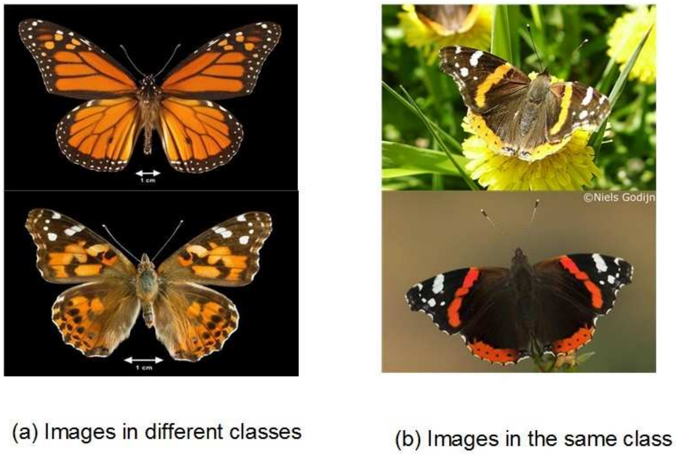
LIST OF TABLES

Chapter 1: Introduction

Object recognition is a fundamental problem in computer vision. Recognition is required by many applications. In particular, in biological research it is required to automatically detect the species of specimens in images (eg. bird, butterfly). However, the target objects in images generated by biological research are very similar to each other. Similarity is in terms of color and texture patterns of the objects. At the same time, objects within the same classes can be very different, due to variations in age, gender, posture etc. of objects. We call this problem fine-grained object recognition. This problem is different from standard object recognition because it requires higher discriminative power for classifying objects belonging to very similar classes.

The major challenges include the difficulty of handling occlusions, articulations, illumination changes and different object locations in images. As shown in figure 1.1, images in different classes may look very similar, and also images in the same class may look very different due to different posture and appearance. Even people have a hard time recognizing or discriminating these objects. In this thesis, I present a novel approach to address this problem.

Our underlying assumption is that, given a test image, there is a limited number of image patches needed to identify the object. If those patches are similar enough to patches of training images that belong to a particular class, in terms of



(a) Images in different classes

(b) Images in the same class

Figure 1.1: Examples of challenges in fine-grained object recognition. (a):The images from two different classes, Danaus plexippus and Vanessa cardui. (b): The images from the same class, Vanessa atalanta. Images in different classes may look very similar, and also images in the same class may look very different due to different posture and appearance. Even people have a hard time to recognize these objects.

both appearance and geometric properties, then we can classify the test image as belonging to that class. In particular, for every training image, we first hypothesize that the test image belongs to the same class of the training image. Then, we try to match patches of the test and training images, so that their total cost of matching is minimized. The target object is identified as the class with smallest matching cost across all training images.

To combine the appearance and geometric information of image patches simultaneously, we weight these two terms by a coefficient. To find this unknown coefficient, we exhaustively search for its optimal value. Then, we specify the cost of matching two patches—one from the training image and the other from the test image—as the Euclidean distance between their appearance and geometric properties. Next, we apply the Hungarian algorithm to find the lowest cost match of test patches with training patches. Since the geometric features are based on the guess of potential center of object in the test image, we evaluate different candidate locations of the center. Finally, we perform a voting process over all train images, and candidate center locations to identify the class of the test image.

For evaluation, we use the Leeds butterfly dataset. This is a challenging benchmark dataset, since all the butterfly images are captured in the wild. The butterflies are articulated and show various postures. Additionally, the images usually contain various and complex backgrounds. Our method outperforms all the existing methods.

We also evaluate our approach on a significantly more challenging Caltech-UCSD Birds 200 dataset. This dataset is more challenging than the Leeds butterfly dataset because the birds are assumed various poses, e.g., wings spread-out, wings folded, zoomed-in bird heads,etc.

The rest of this thesis is organized as follows: Chapter 2 discusses related work in computer vision and machine learning. Chapter 3 briefly introduces the workflow of our approach. Chapter 4 describes how to compute the appearance term, geometric term and weight term for matching image patches. Chapter 5 introduces two possible optimizations to find the best matching, and make our final prediction for the recognition task. The experiments are presented in Chapter 6, while the conclusion is given in Chapter 7.

Chapter 2: Related Work

There has been much work in recent years using semilocal patch-based features such as SIFT [9] and geometric blur [1] for object classification. When the Caltech101 data set [23] was introduced in 2004, the initial result was approximately only 16% mean recognition across categories. Since then, there has been great improvements in recognition performance on the 2004 benchmark, with most algorithms making use of some variant of geometric blur or SIFT [2, 18, 3, 29, 20, 21, 26, 30]. Of this work, [21],[29] and [20] focused specifically on defining good image-to-image kernel functions over sets of patch-based features for use with support vector machines(SVMs). In the first two of the three, the distance function is designed in advance and does not make use of the training data. In the third, the training data is used to structure a hierarchy over the feature space, but the class labels are not used. In [18] the geometric blur descriptor was used with DAG SVMs and a nearest-neighbor pruning of the training set at test time to yield strong result. That work also linearly combined different types of feature information, though their combination was parameterized by a single variable tuned by cross-validation.

Fine-grained object recognition demands an approach to discriminate among highly similar object classes that are often differentiated by only subtle differences [11]. Traditional image classification approaches, however, often fail to perform this task satisfactorily [12]. We hypothesize that a key weakness might reside in the way features are encoded in most of todays state-of-the-art image classification systems [38, 29, 14, 19]. Specifically, image patches are often encoded by a universal dictionary of visual codewords built by clustering a large number of image patches. Such a procedure, while computationally efficient and effective for generic object categorization, results in a large loss of finer details that are important for differentiating fine-grained object classes.

An increasing number of papers have focused on fine-grained object recognition in recent years [12, 7, 22, 32, 34]. In [7], multiple kernel learning is used to combine different types of features and serves as a baseline fine-grained recognition algorithm, and human help is used to discover useful attributes. In [38], a random forest is proposed for fine-grained object recognition that uses different depths of the tree to capture dense spatial information. In [22], a multi-cue combination is used to build discriminative compound words from primitive cues learned independently from training images. In [37], bagging is used to select discriminative ones from the randomly generated templates. In [10], image regions are considered as discriminative attributes and CRF is used to learn the attributes on training set with human in the loop. Pose pooling [39] adapted Poselets [6] to fine-grained recognition problems and learned different poses from fully annotated data. Though deformable parts model [13] is powerful for object detection, it might be insufficient to capture the flexibility and variability in fine-grained tasks considered here [28].

Our approach deviates from the previous approaches in that they focus on exploiting the sparsity in the feature space [31, 25, 17, 8] or in the over complete dictionary [16, 4, 24, 36]. Instead, we employ a different parsimony here: the combination of appearance and geometric.

Chapter 3: Overview of Our Approach

Our approach is motivated by findings on human vision. Our algorithm simulates the gaze behaviour of people. Given an image, a person will not immediately see all the details. On the contrary, a person will first see only familiar parts of the image that are consistent geometrically with previously seen objects. The recognition is performed based on these selected regions.

The basic flow of our object recognition algorithm is shown in Figure 3.1. The figure depicts the process for one identity-hypothesis. In practice, this flow will repeat for M times for M training images.

As can been see in Figure 3.1, each training image is uniformly partitioned into training patches $b_1, b_2, \ldots, b_Q$. Each patch is associated with a descriptor, shown on the upper left in the Figure 3.1. In the test image, we first extract the same number of patches $p_1, p_2, \ldots, p_T$. Each patch of test image is compared with training patches using the appearance properties. The appearance term compares the similarities of two patches' by linear regression. The appearance similarities are then combined with the geometric properties of the image. The latter is defined by the distance between the object center and coordinate of patches in both our test and training images. We combine the appearance and geometric terms as a weighted sum, with α as a weighting parameter. We match the patches using the Hungarian algorithm. To this end, we generate a cost matrix. Since we have

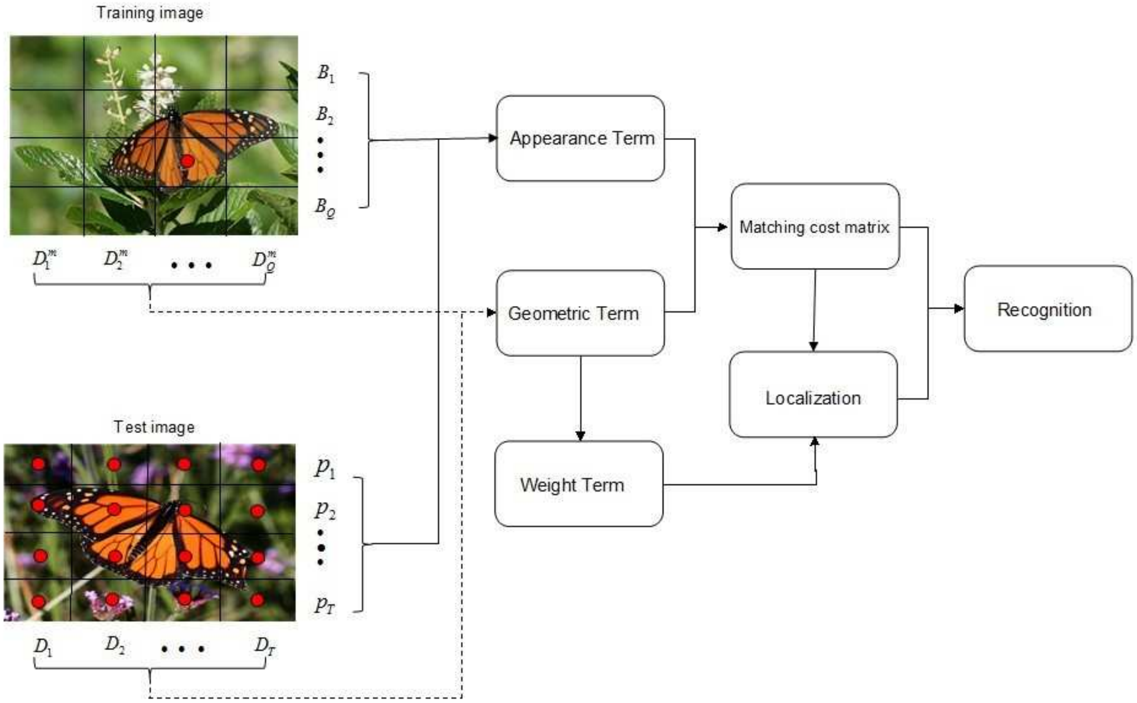


Figure 3.1: The pipeline of my proposed algorithm.

two different formulation of the cost matrix, we present two approaches to find the closest training image. In the following chapter we describe each step of our approach in further detail.

Chapter 4: Feature Extraction

This chapter describe the appearance and geometric terms that we use to compute the cost of matching image patches.

4.1 Appearance Term

In our approach, all the training images and test images are normalized to the same size. Then, we use the HOG descriptor to describe the training and test patches. The HOG descriptor is specified in section 4.1.1. For the mth training image, let b_q^m denote the HOG of qth vectorized training patch. Assume that each training image has a total of I neighbor patches. Then, we form a matrix $B_q^m = [b_{m,q}^1, b_{m,q}^2, \ldots, b_{m,q}^I]$. B denotes the block collection of patches for training images that are distributed around q. In this thesis, B_q^m plays a role of linear basis. The test patches are denoted as $p_1, p_2, \ldots, p_T$, where t denotes the tth patch of test image.

A series of linear regressions are performed to capture the similarities of each test patch to the training patches, as

$$\underset{\beta}{min} \left\| p_t - B_q^m \beta_q^m \right\| . \tag{4.1}$$

Equation 4.1 is a least squares problem and has a closed-form solution

$$\beta_q^m = (B_q^{mT} B_q^m)^{-1} B_q^{mT} p_t. \tag{4.2}$$

The regression residuals can be calculated as:

$$r_{t,q}^m = \left\| p_t - B_q^m \beta_q^m \right\|_2 . \tag{4.3}$$

The smaller the residual, the larger confidence that the test patch t is similar to the training patch q in the particular training image m. The whole process is shown in the Figure 4.1

Now, we can denote this residual value as our appearance term. We normalize the residual value as

$$R_{t,q}^m = 1 - \exp(\frac{-r_m^{t,q}}{\sigma}) \in [0, 1] . \tag{4.4}$$

The residual is based on linear regression. Therefore its computation cost is low. Also the accuracy is acceptable, as previous work shows that linear regression based methods can achieve highly discriminative ability [27]. Also, some work in the literature proved that if the target could be viewed as a convex Lambertian surface, linear regression is very robust to illumination changes [15, 5].

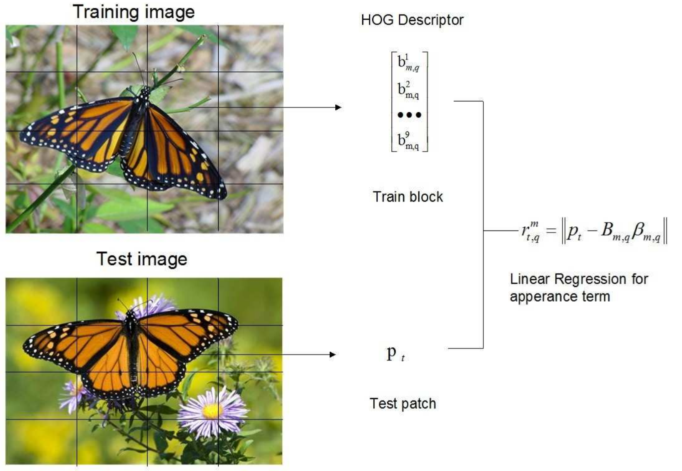

Training image
HOG Descriptor
$\begin{bmatrix} b^1_{m,q} \\ b^2_{m,q} \\ \bullet\bullet\bullet \\ b^9_{m,q} \end{bmatrix}$
Train block
$r^m_{t,q} = \|p_t - B_{m,q}\beta_{m,q}\|$
Linear Regression for apperance term
Test image
p_t
Test patch

Figure 4.1: Using linear regression to measure the residual as our appearance term

4.1.1 HOG Descriptor

Histogram of Oriented Gradients(HOG) is a very popular feature descriptor in computer vision and image processing for the purpose of object detection. It counts the occurrences of specific gradient orientation in particular portions of an image or a video called cells. This method is very similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. Each pixel within the cell gives a weighted vote for an orientation-based histogram. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing. The HOG descriptor has a few key advantages over other descriptor methods. Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions.

4.2 Geometric Term

4.2.1 Object Center

We observe that the appearance term is not enough for object recognition. We compute the geometric term as the additional cue for our recognition task. In conventional detection methods, the geometric term is specified in terms of the object scale or rotation angle of the object. For some recognition tasks, test images may have arbitrary rotation angles. Then a 4-dimension matrix is required for estimating the location,scale and angle variations. However, extraction of this matrix is very difficult for a single 2D image. Therefore, we employ a relatively concise retrieval strategy, based on the object center, which is depicted in Figure 4.2.

As shown in Figure 4.2, given a training image we can easily obtain its object

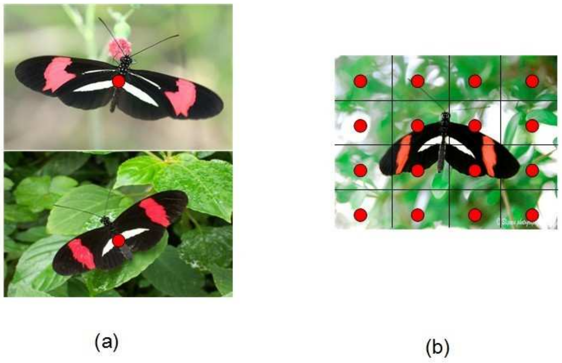


Figure 4.2: The demonstration of object center. (a)For the training images we retrieve the object center by annotation. (b)For the test images we take every possible location into consideration

center by annotation. The object center is depicted as the center of the bounding box. For the test image, we do not know the location of the center of the object. Therefore, we take every possible location in the test image into consideration. Not all locations are equally likely to be the object center of test image. We introduce the weight term in section 4.2.3 to measure the probability of a particular location to be the object center.

With the help of the object center, we can focus on the important patches, and reduce the computation complexity of our algorithm. Also we can check the consistency of our test patches to the training patches. This will improve the accuracy of recognition.

4.2.2 Specification of Geometric Term

For each training image, we can obtain its object center by annotation, as the center of the bounding box around the object. We denote the object center of training image m as c_m. Since each training image has the same number of patches, they share the same locations. Then, we can denote the center of each patch as c_q. We compute the vector distance between the patch q and the object center as

$$\mathbf{d}_q^m = c_m - c_q. \tag{4.5}$$

Also for the test image we consider candidate object centers. We denote the candidate object center of the test image as c_l. Then we compute the vector distance between the center of test patches c_t, and our guess object center as is given by

$$\mathbf{d}_t^l = c_t - c_l. \tag{4.6}$$

From the Equation 4.5 and Equation 4.6, we can achieve the geometric consistency of the test patch with the training patch when the following expression is close to zero

$$w_{t,q}^{m,l} = \left\| \mathbf{d}_q^m - \mathbf{d}_t^l \right\|. \tag{4.7}$$

$w_{t,q}^{m,l}$ is specified as our geometric term. We normalize the geometric consistency as

$$W_{t,q}^{m,l} = 1 - \exp(\frac{-w_{t,q}^{m,l}}{\sigma}) \in [0,1]. \tag{4.8}$$

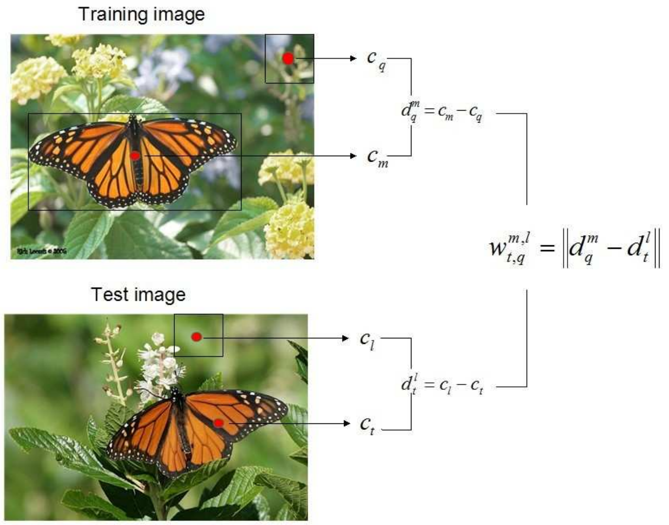


Figure 4.3: Using geometric distance to measure the consistency as the geometrical term

4.2.3 Weight Term

In Equation 4.7, the geometric term has parameter l which is a candidate object center in the test image. Not all the locations have the same likelihood to be the object center. We evaluate the likelihood as

$$D^{m,l} = \exp(-\frac{\|\mathbf{c}_m - \mathbf{c}_l\|}{\sigma}). \tag{4.9}$$

From Equation 4.9 we can see that the object center of test image is more likely to be the location that is close to the training image object center than the far away ones. With this weight term we will have a good constraint to select a good object center from the candidate. The weight term will apply to our two different optimize process in Chapter 4.

4.3 Combining the Appearance and Geometric Terms

We combine appearance and geometric terms as

$$\Psi_{t,q}^{m,l} = R_{t,q}^{m} + \alpha W_{t,q}^{m,l}. \tag{4.10}$$

In Equation 4.10, T is the combination of our appearance term and geometric term, and α is a constant. Because we do not know whether appearance term is more important than geometric term for recognition. We use different α values in our experiments to achieve the best results.

Chapter 5: Recognition

This section describes our approach to estimating: 1) The cost of matching between the patches in the training and test images, 2) The object center in the test image. The object center is referred to our estimate of the center of the object. Image patches that are far from the object center are then less likely to belong to the object. These far away patches are appropriately downweighted in recognition.

5.1 Matching Image Patches

For each training image $m \in \{1, 2, ...M\}$ and each candidate location of the object center $l \in \{1, 2, ...L\}$, we compute a $Q \times T$ cost matrix $\Psi^{m,l}$. An element of $\Psi^{m,l}$ is denoted as $\psi^{m,l}_{t,q}$. Given $\Psi^{m,l}$, our goal is to find the legal mapping

$$f := \{(q,t) : q = 1, \ldots, Q, t = 1, \ldots, T\} . \tag{5.1}$$

Where q denotes patches in the training image m, and t denotes patches in the test image.

We want to minimize the total cost of matching, defined as:

$$\sum_{(q,t) \in f} \psi^{m,l}_{t,q} \cdot x(q,t), x(q,t) \in \{0, 1\} . \tag{5.2}$$

Where $x(q,t)$ is the indicator of matching with following meaning: $x(q,t) = 1$ means that patch q in training image matches patch t in the test image and $x(q,t) = 0$ means they do not match. Equation 5.2 is equivalent to

$$\min_{X} tr\left((\Psi^{l}_{m})^{T} X\right), x(q,t) \in \{0,1\}. \tag{5.3}$$

To avoid trivial solutions, we constrain X as $\sum_q x(q,t) = 1$, $\sum_t x(q,t) = 1$.

The constraint in Equation 5.3 ensures one-to-one matching, where every patch q finds a single matching patch t and every patch t finds a single matching patch q.

To solve this fundamental matching problem, we use the Hungarian algorithm. The Hungarian algorithm is presented in section 5.1.1. After applying the Hungarian algorithm to find the best matching between training patches q and test patches t, we also get a total cost value of our matches. The smaller the cost value, the better our matching results.

5.1.1 Hungarian Algorithm

The Hungarian algorithm is a combinatorial optimization algorithm which solves the assignment problem in polynomial time. The assignment problem consists of finding a minimun cost matching in a weighted bipartite graph. Here, our goal is to find an assignment of training block to test patches so that no training block is assigned more than one test patch and no test patch is assigned to more than one training block in such a manner so as to minimize the total cost of completing the assignment task. The following are the key steps of the Hungarian algorithm:

1. Given a cost value matrix Ψ, from each row of Ψ, find the row minimum, and subtract it from all elements in that row.

2. From each column of Ψ, find the column minimum, and subtract it from all elements in that column.

3. Cross out the minimum number of rows and columns in ψ to cover all zero elements.

4. If all rows of Ψ are crossed out, we are done.

5. Otherwise, find the minimum entry of Ψ that is not crossed out. Subtract it from all entries of Ψ that are not crossed out. Also, add it to all elements that are crossed out. Return to step 2 with the new matrix.

6. Solutions are zero elements of Ψ. Go first for the zero element which is unique in its row and column. Then, delete that row and column from Ψ. Repeat until you delete all rows or columns from Ψ.

5.2 Finding the Object Center

After solving the Hungarian algorithm for all training images m and candidate location l, we form a new matrix of the total cost of matching C. Each element of C is equal $C^{m,l} = \sum_{q,t} \psi_{q,t}^{m,l} \cdot x\,(q,t)$, where $x(q,t)$ is solved by the Hungarian algorithm. As we mentioned earlier, we need to estimate object center in the test image and make a final prediction based on this object center. This object center will be used to downweight the patches in the test image that are far away, since they are less likely to belong to the target object. In section 4.2.3 we explained that not all the candidate locations have the same likelihood to be the object center. We use $D^{m,l}$ to denote the likelihood that a location l is the object center given training image m. This was defined in Equation 4.9. our goal is to find the most possible location of object center. The optimization can be written as

$$\min_{l,m} C^{m,l} + \beta(-\log D^{m,l}). \tag{5.4}$$

The objective function in Equation 5.4 evaluates the match between test image and particular training image m when the object center of the test image is l. The smaller the objective, the more similar our test image to the training image.

The location with the lowest average value in the objective of Equation 5.4 is our target object center. Once the test image object center is found, we take the majority vote for the classes those training images belong to.

This optimization approach has high dependency on the correctness of localizing the object center. As shown in Figure 5.1, it is possible that we fail to localize the location of object center in the test image. The recognition can be increased if we improve the estimation of object center in the test image, as explained in the next section.

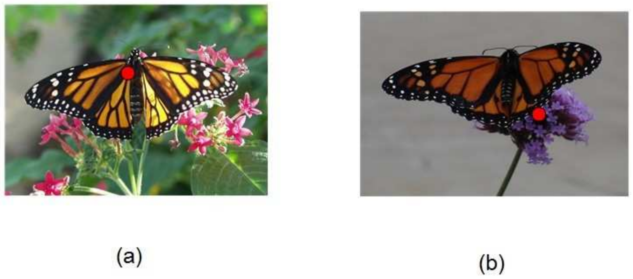


Figure 5.1: (a)A successful localization to object center of test image. (b)A unsuccessful example that we localize the object center as the center of butterfly and flower.

5.3 Finding the Object Center by Weighting

As explained in the previous section, our approach is sensitive to correctness of localizing the object center. We propose another optimization approach which relaxes the location of the object center by estimating its expected values. In other words, each location can be the object center but with different probabilities. The probability is exactly our previously defined weight term. The optimization approach based on weighting is described in Figure 5.2.

Given a cost value matrix, we check every candidate location to find the minimal value in this particular location denoted as c^l_{min}. Then localize the training image m corresponds to the minimal value. After obtaining the training image we perform voting for a particular class which the training image belongs to. The voting process is associated with weighted score $W^{m,l}$. We can retrieve the weighted score from the weight term. Finally, we make the prediction based on the score in

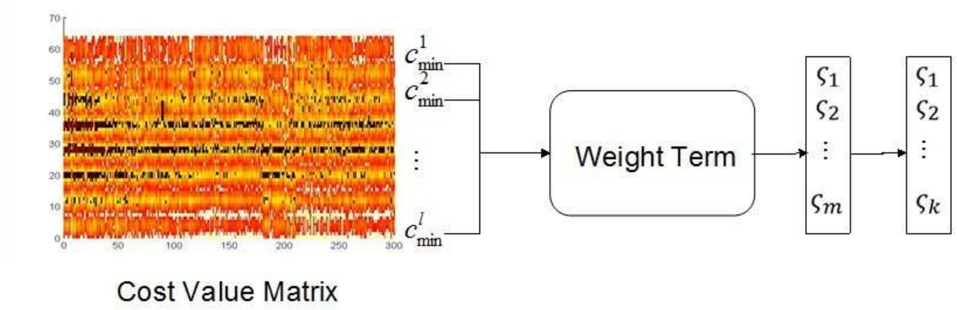


Figure 5.2: The pipeline of finding the object center by weighting

the vote list. The majority vote class will be our choice.

$$\zeta^m = \sum_l \min_m D^{m,l}. \tag{5.5}$$

As shown in Equation 5.5, the total voting score of particular image m is defined as the sum of each possible candidate's vote associates with the weight term. Then the particular image voting score projects to the class voting score which this training image belongs to,

$$\zeta^k = \sum_{m^k} \zeta_{m^k}, m^k \in k. \tag{5.6}$$

After the project process, we obtain a score list over classes. The final prediction is achieved by choosing the class with highest score ζ^k.

Chapter 6: Experiments and Result

We test our approach on the Leeds butterfly dataset [33] and the Caltech-UCSD Birds-200-2011 dataset [35]. The Leeds butterfly dataset contains 832 butterfly images belong to 10 classes, we take 30 images in each class as our training images and 30 images in each class as our testing images. For the bird dataset, we choose 10 classes from 200 categories and each class has 30 images as our training images and the other 30 images as our test images.

The object center for a training butterfly is the middle point of the two joints of the forewings, as shown in Figure 6.1. For the bird dataset shown in Figure 6.2, the object center for a training bird image is the center of bird's body. We partition each training image and test image to $8 * 8$ patches. Each patch is described by HOG descriptor.

All the experiments are conducted in Matlab-R2012b, on a laptop with a 2.6GHz quad-core CPU and 8GB RAM.

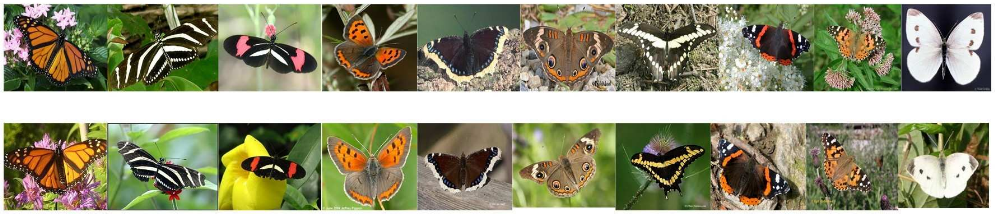


Figure 6.1: The training(top) and test(below) butterflies in Leeds butterfly dataset. All the images are captured in the wild, which makes the task very challenging.

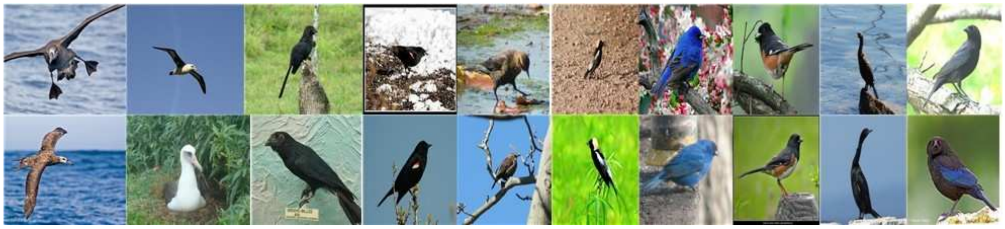


Figure 6.2: The training(top) and test(below) birds in Caltech-UCSD Birds-200-2011 dataset. Since the birds are assumed various poses, e.g., wings spread-out, wings folded, zoomed-in bird heads,etc, it is even hard than the butterfly dataset.

Below, we review our main steps and provide implementation details. First, to compute the appearance term in Equation 4.4, we merge each training patch its 8 nearest patches, and form a block. The appearance term is the residual value of linear regression between test patches and training blocks. Second, we consider to the runtime of our approach. Rather than using every possible pixel as the object center of our test image, we sample every 16 pixels as a trade-off strategy. Third, to compute Equation 4.10 we directly delete the test patches that are far away from our guess location of object center. This strategy updates the Equation 4.10 as

$$\Psi^{m,l}_{t',q} = R^{m}_{t',q} + \alpha W^{m,l}_{t',q}. \tag{6.1}$$

These kind of strategies ensure that our algorithm is efficient and has a good performance.

We organize our results as follows. First, we show how α affects accuracy. Second, we show voting confidence for every class. Third, we present the confusion matrix. Finally, we show some qualitative examples.

6.1 Butterfly Dataset Result

In Equation 6.1, the importance weight of appearance term and geometric term, α is unknown. We explore different α values which maximize our accuracy. The α value is varied as $0, 0.5, 1, 1.2, 2.0$. Table 6.1 shows the influence of different α values on the accuracy.

	α=0	α=0.5	α=1	α=1.2	α=2.0
Find the object center	27.12	35.93	40.00	44.08	42.71
By weighting	51.19	56.27	60.00	**61.36**	49.83

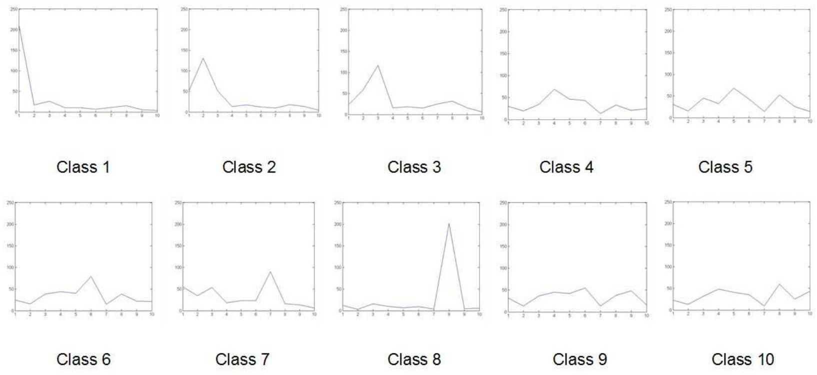
Table 6.1: The influence of different α value to the accuracy of our two optimization process in butterfly dataset. We set α as $0, 0.5, 1, 1.2, 2$. We notice that the overall accuracy of optimization by weighting is better than the optimization by finding the object center.

According to the Table 6.1, both optimization process reach highest accuracy when $\alpha = 1.2$. The accuracy when $\alpha = 0$ means that we got the results without considering the geometric term. As can be seem, with the help of geometric information, we improve our result by 14%.

In order to show the confidence during our recognition process, we illustrate the vote score for each class in Figure 6.3 and Figure 6.4.

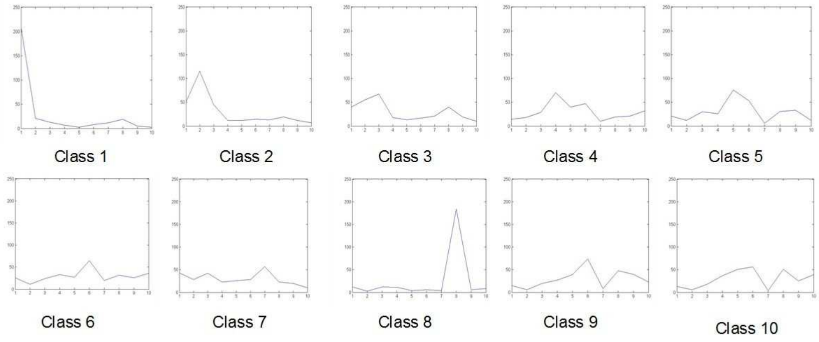
From these confidence figures one may notice that we have enough confidence to recognize the objects belong to class 1, class 2, class3 and class 8. However, for class 9 and class 10 the voting score for the true class is even less than other classes which means we almost fail to recognize the objects in the test images. This trend can also be seem in the confusion matrix Table 6.2.

In the following, we present some qualitative to illustrate how our optimization


Class 1 Class 2 Class 3 Class 4 Class 5

Class 6 Class 7 Class 8 Class 9 Class 10

Figure 6.3: Vote score for each class by finding the object center optimization process in butterfly dataset


Class 1 Class 2 Class 3 Class 4 Class 5

Class 6 Class 7 Class 8 Class 9 Class 10

Figure 6.4: Vote score for each class by finding the object center by weighting in butterfly dataset

works, we present a successful recognition process in Figure 6.5. The dark color in the cost value matrix of Figure 6.5 means low value. First, we localize a object center in the test image by Equation 5.4. As the color map of cost value matrix shows, most of lowest cost values are located in our guess location $No.36$. The location $No.36$ is exactly the closest guess location to the the middle point of the two joints of the forewings. Now we focus on the fixed object center and check

30	0	0	0	0	0	0	0	0	0
2	25	2	0	0	0	0	1	0	0
1	3	26	0	0	0	0	0	0	0
2	0	4	16	3	2	0	3	0	0
1	0	5	0	13	2	0	6	3	0
2	0	2	2	3	17	0	3	0	1
8	0	4	1	0	0	17	0	0	0
1	0	0	0	0	1	0	23	0	0
7	0	1	2	3	6	0	4	7	0
1	1	1	6	4	1	0	8	0	7

Table 6.2: Confusion matrix when set $\alpha = 1.2$ and finding the object center by weighting in butterfly dataset. This confusion matrix return the total accuracy of our recognition task as **61.36%**

the value in this particular location. We can see that, in the color map of cost value matrix in location $No.36$, the low values are more likely to appear in the training image region belongs to class-1. This observation leads us to make the final classification that our test image is from class-1.

In Table 6.3 we show the recognition accuracies of human recognition and some state of the art method. According to the table, our approach achieves the highest recognition rates, among all the computer-based algorithms. The accuracy of our approach is merely 10% lower than the best result of human recognition.

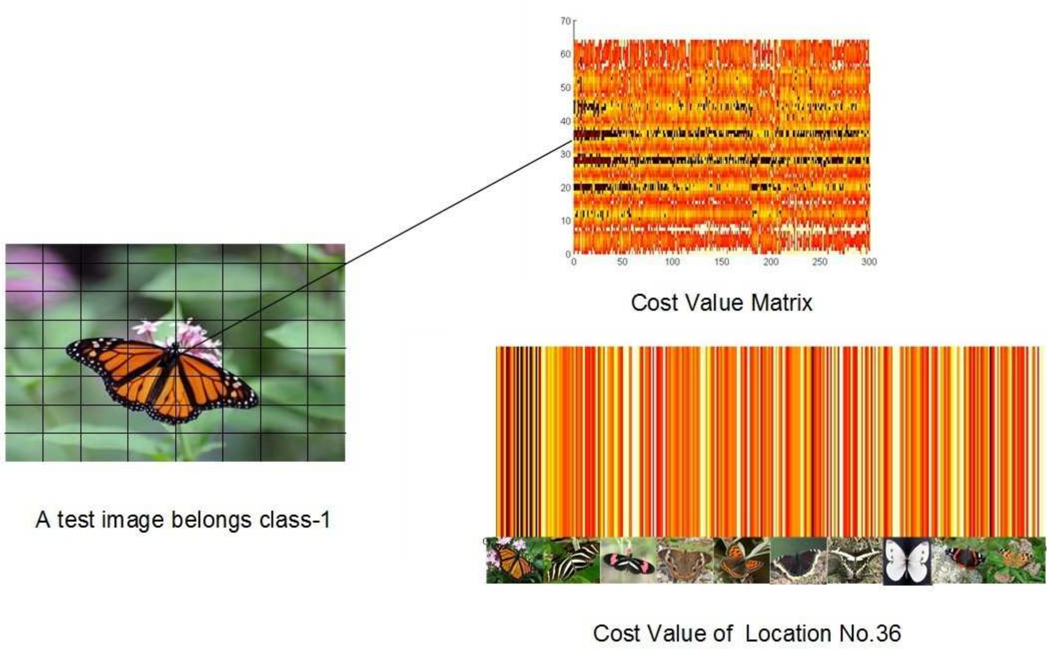


Figure 6.5: The demonstration of our optimization process by finding the object center. Dark color means low value. Localize the object center by the cost value matrix, make classification based on the cost value of a fixed location.

	Accuracy
Human (native English Speakers)	72.0
Human (nonnative English Speakers)	51.0
Leeds(Learned templates)	54.4
Bag-of-Features(SVM + l_2-norm)	29.6
Bag-of-Features(SVM + l_1-norm)	44.7
Our approach	61.4

Table 6.3: The accuracies of butterfly recognition.

6.2 Bird Dataset Result

For the bird dataset we reach a highest accuracy with $\alpha = 1.2$. The confusion matrix is shown in Table 6.4.

25	0	1	1	0	2	0	0	1	0
9	4	1	6	0	3	4	0	1	2
2	1	9	5	1	1	1	2	2	6
3	0	2	12	1	2	7	0	2	1
3	0	4	9	4	2	1	1	3	3
1	1	6	2	1	11	3	0	4	1
2	2	4	4	2	6	6	1	1	2
2	0	2	5	2	4	3	2	3	2
14	0	5	6	2	2	0	1	0	0
1	0	3	10	3	2	3	0	4	4

Table 6.4: Confusion matrix when set $\alpha = 1.2$ and finding the object center by weighting. This confusion matrix return the total accuracy of our recognition task as **26.10%**

In the following, we illustrate one of our qualitative result in Figure 6.6.

For the bird dataset, there is still no published results for comparison.

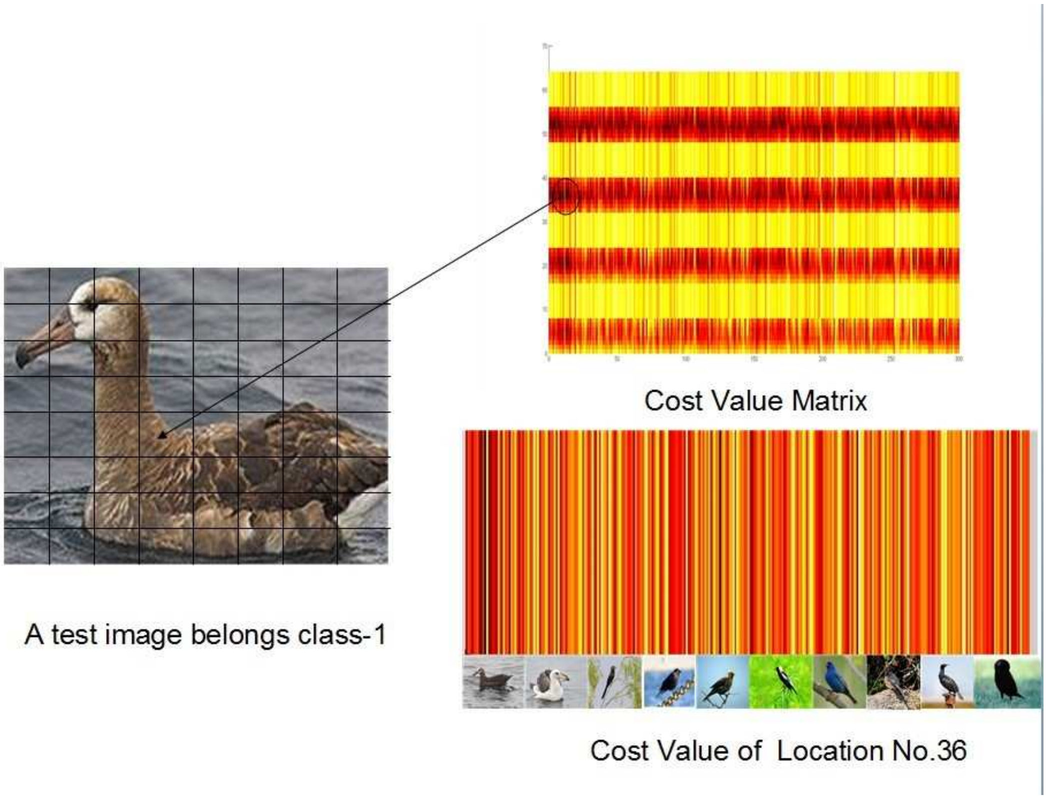


Figure 6.6: The demonstration of our optimization process by finding the object center. Dark color means low value. Localize the object center by the cost value matrix, make classification based on the cost value of a fixed location.

Chapter 7: Conclusion

In this thesis, we present a novel approach to object recognition. In particular, we addressed the fine-grained recognition problem, where each object class has large variations, and there are only small differences between the classes. Our approach is based on the estimation of distance between patches in the test image and a given labeled set of training images. The distance is specified in terms of appearance and geometric properties of the object. As a critical step, we estimate the center of the object called the object center. The concept of object center makes our approach similar to the gaze behaviour of people. We use weighting to improve the localization accuracy of object center. At the same time, we use object center sampling, and elimination of irrelevant patches. This makes our approach scalable and easy to perform without losing information. The experiments on the butterfly dataset and bird dataset demonstrate the superiority of our proposed approach.

From our experiment results, our approach can not handle the large variation in the scale problem very well. This is because our distance function does not account for changes in size of objects. In our future work, we will take the scale of objects into consideration. This is likely to improve our current results.

Bibliography

[1] A.Berg and J.Malik. Geometric blur for template matching. In *CVPR*, pages 607–614, 2001.

[2] T.Berg A.Berg and J.Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, 2005.

[3] M.Welling A.D.Holub and P.Perna. Combining generative models and fisher kernels for object recognition. In *ICCV*, 2005.

[4] Roth D Agarwal, S. Learning a sparse representation for object detection. In *ECCV*, 2006.

[5] Jacobs D Basri, R. Lambertian reflectance and linear subspaces. In *IEEE Trans. Pattern Anal.Mach.Intell*, 2003.

[6] Malik J. Bourdev, L. Poselets: body partddetectors trained using 3d human pose annotations. In *ICCV*, 2009.

[7] Wah C. Babenko B. Schroff F. Welinder P. Perona P. Belongie S. Branson, S. Visual recognition with humans in the loop. In *ECCV*, 2010.

[8] Bach F. Ghaoui L d'Aspremont, A. Optimal solutions for sparse principal component analysis. In *J.Mach.Learn*, 2008.

[9] D.Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[10] Parikh D. Crandall D. Grauman K. Duan, K. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[11] W. Gray D. Johnson E. Rosch, C. Mervis and P. BoyesBraem. Basic objects in natural categories. In *Cognitive Sci.,8(3):382439*, 1976.

[12] Oza O. Zhang N. Morariu V. Darrell T. Davis L. Farrell, R. Birdlets: subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[13] Girshick R. McAllester D. Ramanan D. Felzenszwalb, P. Object detection with discriminatively trained part based models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 2010.

[14] L. Fan J. Willamowski G. Csurka, C. Dance and C. Bray. Visual categorization with bags of keypoints. In *Workshop on SLCV*, 2004.

[15] Belhumeur P. Kriegman D Georghiades, A. From few to many: Illumination cone models for face recognition under variable lighting and pose. In *IEEE Trans. Pattern Anal.Mach.Intell*, 2001.

[16] P Hoyer. Non-negative matrix factorization with sparseness constraints. In *J.Mach.Learn*, 2004.

[17] Aviyente S Huang, K. Sparse representation for signal classcification. In *Proc.Adv.Neural Inf.Process.Syst*, 2007.

[18] M.Maire H.Zhang, A.Berg and J.Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[19] K. Yu F. Lv T. Huang J. Wang, J. Yang and Y. Gong. Learning locality-constrained linear coding for image classication. In *CVPR*, 2010.

[20] K.Grauman and T.Darrell. Approximate correspondences in high dimensions. In *NIPS*, 2006.

[21] K.Grauman and T.Darrell. Pyramic match kernels: Discriminative classificiation with sets of image features. In *Technical Report USB/CSD-0401366, MIT*, March 2006.

[22] van de Weijer J. Bagdanov A. Vanrell M. Khan, F. Portmanteau vocabularies for multi-cue image representations. In *NIPS*, 2011.

[23] R.Fergus L.Fei-Fei and P.Perona. Learning generative visual models from few training examples: an incremental bayesian approach testing on 101 object categories. In *Workshop on Generative-Model Based Vision, CVPR*, 2004.

[24] Ling H Mei, X. Robust visual tracking using 11 minimization. In *ICCV*, 2009.

[25] Weiss Y. Avidan S Moghaddam, B. Generalized spectral bounds for sparse ida. In *Proc.Int.Conf.Mach.Learn.*, 2006.

[26] M.Schutlz and T.joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003.

[27] Togmero R. Bennamoun M Naseem, I. Linear regression for face recognition. In *IEEE Trans. Pattern Anal.Mach.Intell*, 2010.

[28] Vedaldi A. Zisserman A. Jawahar C. Parkhi, O. Cats and dogs. In *CVPR*, 2010.

[29] C.Schmid S.Lazebnik and J.Ponce. Beyond bags of feature: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[30] L.Wolf T.serre and T.Poggio. Object recognition with feature inspired by visual cortex. In *CVPR*, 2005.

[31] V Vapnik. The nature of statistical learning theory. In *Springer Verlag*, 2000.

[32] Branson S. Perona P. Belongie S. Wah, C. Interactive localization and recognition of fine-grained visual categories. In *ICCV*, 2011.

[33] Markert K. Everingham M Wang, J. Learning models for object recognition from natural language descriptor. In *Bri.Mach.Vis.Conf*, 2009.

[34] Branson S. Mita T. Wah C. Schroff F. Belongie S. Perona P. Welinder, P. Caltech-ucsd birds 200. In *Technical Report CNS-TR-201, Caltech*, 2010.

[35] Mita T. Wah C. Schroff F. Belongie S. Perona P. Welinder P., Branson S. Caltech-ucsd birds 200. In *California Institute of Technology. CNS-TR-2010-001.*, 2010.

[36] Yang A.Y. Canesh A. Sastry S.S. Ma Y Wright, J. Robust face recognition via sparse representation. In *Pattern Anal.Mach.Intell*, 2009.

[37] Bradski G. Fei-Fei L. Yao, B. A codebook-free and annotation-free approach for negrained image categorization. In *CVPR*, 2012.

[38] Khosla A. Fei-Fei L. Yao, B. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011.

[39] Farrell R. Darrell-T. Zhang, N. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012.