UC Berkeley
UC Berkeley Electronic Theses and Dissertations

Title
Visual Representations for Fine-grained Categorization

Permalink
https://escholarship.org/uc/item/3g69z0fw

Author
Zhang, Ning

Publication Date
2015

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

Visual Representations for Fine-grained Categorization

by

Ning Zhang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Trevor Darrell, Chair
Professor Jitendra Malik
Professor Bruno Olshausen

Fall 2015

Visual Representations for Fine-grained Categorization

Copyright 2015
by
Ning Zhang

Abstract

Visual Representations for Fine-grained Categorization

by

Ning Zhang

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Trevor Darrell, Chair

In contrast to basic-level object recognition, fine-grained categorization aims to distinguish between subordinate categories, such as different animal breeds or species, plant species or man-made product models. The problem can be extremely challenging due to the subtle differences in the appearance of certain parts across related categories and often requires distinctions that must be conditioned on the object pose for reliable identification. Discriminative markings are often highly localized, leading traditional object recognition approaches to struggle with the large pose variations often present in these domains. Face recognition is the classic case of fine-grained recognition, and it is noteworthy that the best face recognition methods jointly discover facial landmarks and extract features from those locations. We propose pose-normalized representations, which align training exemplars, either piecewise by part or globally for the whole object, effectively factoring out differences in pose and in camera viewing angle.

I first present the methods of using the idea of pose-normalization for two related applications: human attribute classification and person recognition beyond frontal face. Following the recent success of deep learning, we use deep convolutional features as feature representations. Next, I will introduce the part-based RCNN method as an extension of state-of-art object detection method RCNN for fine-grained categorization. The model learns both whole-object and part detectors, and enforces learned geometric constraints between them. I will also show the results of using the recent compact bilinear features to generate the pose-normalized representations. However, bottom-up region proposals is limited by hand-engineered features and in the final work, I will present a fully convolution deep network, trained end-to-end for part localization and fine-grained classification.

To Shuo, and my parents.

Contents

Contents **ii**

List of Figures **iv**

List of Tables **vii**

1 Introduction **1**

2 Human Attribute Classification **4**
2.1 Background 6
2.2 Pose Aligned Networks for Deep Attribute modeling (PANDA) 7
2.3 Datasets 9
2.4 Experiments 10
2.5 Summary 16

3 Person recognition beyond frontal faces **17**
3.1 Background 18
3.2 People In Photo Albums Dataset 21
3.3 Pose Invariant Person Recognition (PIPER) 23
3.4 Experiments 26
3.5 Summary 30

4 Part-based RCNN model **34**
4.1 Background 36
4.2 Method 37
4.3 Experiments 40
4.4 Bilinear CNN model and compact bilinear pooling 45
4.5 Dense window sampling for keypoint localization 47
4.6 Summary 50

5 Fully Convolution Part Model **52**
5.1 Background 53
5.2 Fully convolutional part model for fine-grained categorization 54

5.3 Experiments 57
5.4 Summary 60

6 Conclusion **62**

Bibliography **63**

List of Figures

2.1 **Overview of Pose Aligned Networks for Deep Attribute modeling (PANDA)**. One convolutional neural net is trained on semantic part patches for each poselet and then the activations of all nets are concatenated to obtain a pose-normalized deep representation. The final attributes are predicted by linear SVM classifier using the pose-normalized representations. 5

2.2 **Part-based Convolutional Neural Nets.** For each poselet, one convolutional neural net is trained on patches resized 64x64. The network consists of 4 stages of convolution/pooling/normalization and followed by a fully connected layer. Then, it branches out one fully connected layer with 128 hidden units for each attribute. We concatenate the activation from fc_attr from each poselet to obtain the pose-normalized representation. The details of filter size, number of filters and stride we used are depicted above. 7

2.3 **Poselet Input Patches.** For each poselet, we use the detected patches to train a convolution neural net. Here are some examples of input poselet patches and we are showing poselet patches with high scores for poselet 1,16 and 79. 8

2.4 Statisitcs of the number of groundtruth labels on Attribute 25k Dataset. For each attribute, green is the number of positive labels, red is the number of negative labels and yellow is the number of uncertain labels. 10

2.5 **Example of PANDA queries.** Top results returned by our proposed method PANDA on Berkeley Attributes of People Dataset for query about multiple attributes. The prediction scores of multiple attributes are computed as a linear combination of attribute prediction scores. 11

2.6 Example of failure cases on Berkeley Attributes of People Dataset. On the top we show highest scoring failure cases for "wears t-shirt" and on the bottom – for "wears long sleeves". 13

3.1 Recognizing people beyond frontal face. We are able to easily recognize people we know in unusual poses, and even if their face is not visible. In this chapter we explore the subtle cues necessary for such robust viewpoint-independent recognition. 18

3.2 **Example photos from our dataset.** These are taken from a single album and show the associated identities. Each person is annotated with a ground truth bounding box around the head, with each color representing one identity. If the head is occluded, the expected position is annotated. 19
3.3 **Challenges of our dataset.** 20
3.4 Example of sparsity filling. On the left we show the predictions of the global model for every identity and every instance. The poselet classifier in the middle does not activate for two instances (the white rows) and is not trained to recognize two identities (the white columns). On the right we show how in the normalized probability we fill-in missing rows from the global model as in the top of Equation 3.2. In addition we account for the missing columns by linearly interpolating each row with the global model based on the likelihood that the identity is not coming from one of the missing columns (the triangles) 22
3.5 Examples that the combination of the Global model and DeepFace misclassify and are recovered by using all of PIPER. **(a)** In a closeup shot the full body falls outside the image and the extracted full-body patch, shown on the right, is severely misaligned. A profile-face poselet should handle this case without misalignment. **(b)** In unusual pose the full body patch may fall on the background or **(d)** on another person which will further confuse the classifier. In **(c)** people have the same clothes and similar pose which will confuse the global model. . . 28
3.6 Recognition accuracy as a function of number of training examples per identity, with $\sigma = 1$ error bar. As we increase the number of training examples our system's accuracy grows faster than the full-body baseline. Chance performance is 0.0017. 30
3.7 Performance of our method on identity retrieval. 31
3.8 Example of PIPER results on unsupervised identity retrieval. For each row we show the query image followed by the top 5 ranked retrieval images. Those are cropped bounding box images on test split and are stretched to make visualization aligned. 32
3.9 **Interfaces for our annotation system.** 33

4.1 **Overview of our part localization.** Starting from bottom-up region proposals (top-left), we train both object and part detectors based on deep convolutional features. During test time, all the windows are scored by all detectors (middle), and we apply non-parametric geometric constraints (bottom) to rescore the windows and choose the best object and part detections (top-right). The final step is to extract features on the localized semantic parts for fine-grained recognition for a pose-normalized representation and then train a classifier for the final categorization. 35

4.2 Illustration of geometric constant δ^{NP}. In each row, the first column is the test image with an R-CNN bounding box detection, and the rest are the top-five nearest neighbors in the training set, indexed using `pool5` features and cosine distance metric. 39
4.3 Cross-validation results on fine-grained accuracy for different values of α (left) and K (right). We split the training data into 5 folds and use cross-validate each hyperparameter setting. 44
4.4 Examples of bird detection and part localization from strong DPM [4] (left); our method using Δ_{box} part predictions (middle); and our method using δ^{NP}(right). All detection and localization results without any assumption of bounding box. 46
4.5 Failure cases of our part localization using δ^{NP}. 47
4.6 Examples of keypoint prediction on five classes of the PASCAL dataset: aeroplane, cat, cow, potted plant, and horse. Each keypoint is associated with one color. The first column is the ground truth annotation, the second column is the prediction result of SIFT+prior and the third column is conv5+prior. (Best viewed in color). 50

5.1 **Overview of our model.** The network consists of two main modules: 1) a localization network for learning where to look and 2) a classification network which uses a coordinate transfer layer to semantically pool part features; these are jointly used to learn the fine-grained classifier. 53
5.2 The coordinate transfer layer takes two inputs: the keypoint heatmap and the feature representation activations. For each part, the layer pools the feature based on the small surrounding neighborhood around the argmax point of the heatmap, as shown in white rectangle. For P different parts, the layer pools P semantic features and stacks them together as the output. 55
5.3 Visualizations of keypoint predictions. Ground truth annotations are shown in the left and our prediction results are shown in the right. Each color map to one keypoint. No bounding box information is used during training or test time. The images shown are warped to align the visualizations. 60

List of Tables

2.1 Attribute classification results on Berkeley Attributes of People Dataset as compared to the methods of Bourdev *et al.* [11] and Zhang *et al.* [119] 12
2.2 Average Precision on the Attributes25K-test dataset. 12
2.3 Relative performance of baselines and components of our system on the Berkeley Attributes of People test set. 14
2.4 Performance of PANDA on front-facing, profile-facing and back-facing examples of the Berkeley Attributes of People test set. 15
2.5 Average precision of PANDA on the gender recognition of the LFW dataset. . . 16

3.1 Statistics of our dataset. 21
3.2 Person recognition results on PIPA test set using 6442 training examples over 581 identities 27
3.3 Performance on the test set when split into the subset where frontal face is visible (52%) and when it is not (48%). 27
3.4 Person recognition performance on the PIPA test set using 6442 training examples over 581 identities as we disable some of the components of our method. PIPER gets more than 3% gain over the very strong baseline of using the fine-tuned CNN combined with the DeepFace model. DeepFace' s score is low because it only fires on 52% of the test images and we use chance performance for the rest. 29

4.1 Fine-grained categorization results on CUB200-2011 bird dataset. -ft means extracting deep features from finetuned CNN models using each semantic part. Oracle method uses the ground truth bounding box and part annotations for both training and test time. 42
4.2 Fine-grained categorization results on CUB200-2011 bird dataset with *no parts*. We trained a linear SVM using deep features on all the methods. Therefore only the bounding box prediction is the factor of difference. -ft is the result of extracting deep features from fine-tuned CNN model on bounding box patches. 43
4.3 Recall of region proposals produced by selective search methods on CUB200-2011 bird dataset. We use ground truth part annotations to compute the recall, as defined by the proportion of ground truth boxes for which there exists a region proposal with overlap at least 0.5, 0.6 and 0.7 respectively. 43

4.4 Part localization accuracy in terms of PCP (Percentage of Correctly Localized Parts) on the CUB200-2011 bird dataset. There are two different settings: with given bounding box and without bounding box. 44
4.5 Fine-grained categorization results using our model and compact bilinear representations. 48
4.6 Convnet receptive field sizes and strides of AlexNet [55], for an input of size 227×227. 48
4.7 Keypoint prediction results on PASCAL VOC 2011. The numbers give average accuracy of keypoint prediction using $\alpha = 0.1$. 49

5.1 Comparison with other methods on fine-grained Classification results. 58
5.2 **Keypoint Localization results.** We use PCK as our evaluation metric. The prediction is correct if lies within $\alpha \times max(h, w)$ of the annotated keypoint with the corresponding object's dimension being (h, w). We show results on different α. No bounding box information is used. 59
5.3 Part localization accuracy in terms of PCP on the CUB200-2011 bird dataset. . 59

Acknowledgments

I am very fortunate to work with my wonderful PhD advisor Trevor Darrell. First and foremost, I would like to thank him, for guiding me to the field of computer vision and then later deep learning. He gave me countless encouragement and support. He has not only taught me how to conduct research, also given me advices on life and careers. I would also like to thank Professor Jitendra Malik, Alyosha Efros, Bruno Olshausen for high-level ideas and for providing such an open environment between research groups.

The most amazing thing during my PhD is the opportunity to work with so many talented postdocs and graduate students in Berkeley. I would like to thank Professor Ryan Farrell, who helped me write my first paper and start the research about fine-grained categorization. I am very fortunate to know Yangqing Jia, Sergey Karayev, Jon Barron, Hyun Oh Song, Judy Huffman, Jon Long, Jeff Donahue, Evan Shelhamer, Georgia Gkioxari, Saurabh Gupta, Bharath Hariharan, Lisa Hendricks, Eric Tzeng, Brian Kulis, Kate Saenko, Oriol Vinyals, Ross Girshick, Erik Rodner, Sergio Guadarrama, and so many others.

I have done two amazing summers internships at Facebook AI Research and I would like to thank my internship mentor Dr. Lubomir Bourdev. Also many thanks to Manohar Paluri, Marc'Aurelio Ranzato, Yaniv Tagiman and Rob Fergus who offer me great learning experience at Facebook.

Last but not least, I want to dedicate my thesis to Shuo and my parents for their love and support along this long journey.

Chapter 1

Introduction

Fine-grained classification is a challenging task due to striking variations in pose, subtle differences in appearance between classes, and strong intra-class variety. Fine-grained classification has made progress in improving accuracy and now covers several domains including aircraft [73], cars [97], plants [1, 2, 7, 77, 78, 92], and animal breeds [70]. In contrast to basic-level recognition, fine-grained categorization aims to distinguish between different breeds, species or product models, and often requires distinctions that must be conditioned on the object pose for reliable identification. Facial recognition is the classic case of fine-grained recognition, and it is noteworthy that the best facial recognition methods jointly discover facial landmarks and extract features from those locations.

The motivation of my work is to use pose-normalized representation that relies on correspondence to align part representations so that the subtle distinguishing features of different classes can be learned. Localizing the parts in an object is therefore central to establishing correspondence between object instances and discounting object pose variations and camera view position. Several part localization methods using strong supervision are investigated in this dissertation. The Poselet [12] and DPM [33] methods have been utilized to obtain part localizations with a modest degree of success and they achieve adequate localization performance when given a known bounding box at test time. RCNN [41], the state-of-art object detection approach which uses bottom-up region proposals and deep features, has been extended to jointly predict bounding box and part locations without the assumption of bounding box at test time. All these methods are seperated into two modules: localization and feature learning. To overcome that, we propose an end-to-end trainable deep network for simultaneously learning localization and category prediction.

In the past few years, deep convolutional networks originally pioneered by LeCun et al. [62] have been a tremendous success by achieving the state-of-art performance in image classification [55], object detection [41, 90], face recognition [99], keypoint prediction [102], semantic segmentation [71] and other computer vision tasks. The strength of the deep nets is its ability to learn discriminative features from raw image input unlike hand-engineered features. DeCAF [26] showed the deep features extracted from the network pretrained on large datasets can be generalized to other recognition problems.

The fine-grained recognition task requires not only recognition but localization. Whether or not a deep network can effectively learn all necessary invariances to pose is an open question. We believe it is still critical to model parts and pool features into a pose-independent representation to best distinguish between similar subordinate classes. This makes good use of limited training data by reserving model capacity for appearance given the discounting of pose. While recent work [49, 68, 54] has made impressive progress without strong supervision, we show that the correspondence gained from end-to-end keypoint and classification training improves fine-grained recognition accuracy.

The main contributions of my thesis are:

- Present approaches of using pose-normalized representations for two related tasks, including human attribute classification and person recognition beyond frontal face (Chapter 2 and Chapter 3).
- Introduce a novel part-based RCNN method which leverages deep convolutional features for both part localizations and category prediction (Chapter 4).
- Propose an end-to-end trainable fully convolutional deep network that directly predicts part locations from pixel and generate pose-normalized descriptors using the estimated keypoints. The approach achieves state-of-art performance on the CUB200-2011 benchmark dataset (Chapter 5).

Some of the work in this dissertation have been presented over the course of several research papers [119, 121, 72, 118, 116, 120]. The outline of the dissertation is as follows:

Chapter 2 presents using the idea of pose-normalization for a related application to fine-grained categorization, i.e. human attribute classification. The signal associated with some attributes, e.g. wear glasses, long hair, is very subtle and the image is dominated by the effects of pose and viewpoint, just like the fine-grained classification task. Specifically, we propose the PANDA model, which augments deep convolutional networks to have input layers based on semantically aligned part patches by poselets. While this chapter only focuses on using poselet as the part model, our model can be generalized to other part-based models as well.

Following Chapter 2, **Chapter 3** discusses another important application of recognizing people beyond frontal face. We rely a variety of subtle cues from other parts to recognize people we know from unusual poses. Non-frontal views are very common in real-world photo albums but no large-scale public dataset is available. We introduce a new large-scale dataset and propose a part-level person recognizers to account for pose variations. Though we use the same part detectors in Chapter 2, we find that combining parts by concatenating their feature in the manner of Chapter 2 is not effective for this task and we introduce a novel way to combine the representations from different parts.

Having discusses the effectiveness of pose-normalized representations in two related applications, **Chapter 4** moves on to the fine-grained categorization task. The part model Poselets relies on hand-engineered features and report adequate part localization only when

given a known bounding box at test time. In this chapter, we propose a part-based RCNN model for fine-grained categorization without the assumption of bounding box at test time. The motivation is still to use strong supervision to localize the parts and establish correspondence between object instances in order to discount object pose variations and camera view points. Specifically, we extend the state-of-art detection model, RCNN, to predict the bounding box and part locations and then generate pose-normalized representations to predict the fine-grained category. We will show the results of using compact bilinear pooling features. We will also show a dense window sampling method for keypoint localization to overcome the limitation of bottom-up region proposals.

Chapter 5 proposes an end-to-end trainable network supervised by keypoint locations and class labels that localizes parts by a fully convolutional network for the fine-grained classification task. All the methods discussed in the previous chapters have separated fine-grained recognition into stages which first localize parts using hand-engineered or coarsely-localized proposal features, and then separately learns deep descriptors centered on inferred part positions. Our model simultaneously learn the part locations and fine-grained category prediction in a unified network. We design a semantic pooling layer—which is called the coordinate transfer layer—to pool feature maps using predicted keypoints and the classification errors can be back-propagated.

Chapter 2

Human Attribute Classification

In this chapter, we show an important application of human attribute classification by using the idea of pose-normalized representations. Recognizing human attributes, such as gender, age, hair style, and clothing style, has many applications, such as facial verification, visual search and tagging suggestions. This is, however, a challenging task when dealing with non-frontal facing images with low image quality, occlusion, and pose variations.

The signal associated with some attributes is very subtle and the image is dominated by the effects of pose and viewpoint, just like the fine-grained classification task. For example, consider the problem of detecting whether a person wears glasses. The signal (glasses frame) is weak at the scale of the full person and the appearance varies significantly with the head pose, frame design and occlusion by the hair. Therefore, localizing object parts and establishing their correspondences with model parts can be key to accurately predicting the underlying attributes.

Deep learning methods, and in particular convolutional nets [62], have achieved very good performance on several tasks, from generic object recognition [55] to pedestrian detection [91] and image denoising [17]. Moreover, Donahue *et al.* [26] show that features extracted from the deep convolutional network trained on large datasets can benefit related tasks because they provide good generic visual features. However, as we report below, they may underperform compared to conventional methods which exploit explicit pose or part-based normalization. We conjecture that available training data, even ImageNet-scale, is presently insufficient for learning pose normalization in a CNN, and propose a new class of deep architectures which explicitly incorporate such representations. We combine a part-based representation with convolutional nets in order to obtain the benefit of both approaches. By decomposing the input image into parts that are pose-specific we make the subsequent training of convolutional nets drastically easier, and therefore, we can learn very powerful pose-normalized features from relatively small datasets.

Part-based methods have gained significant recent attention as a method to deal with pose variation and are the state-of-the-art method for attribute prediction today. For example, spatial pyramid matching [60] incorporates geometric correspondence and spatial correlation for object recognition and scene classification. The DPM model [33] uses a mixture of

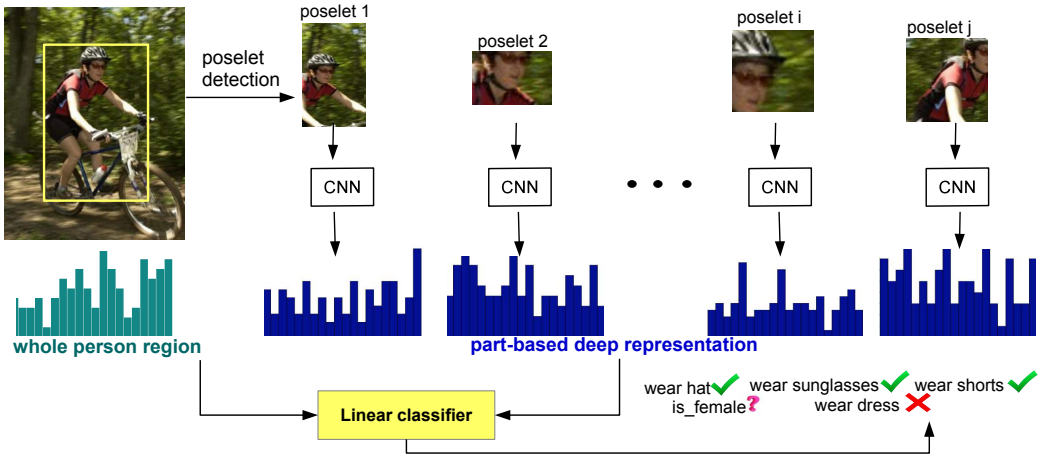


Figure 2.1: **Overview of Pose Aligned Networks for Deep Attribute modeling (PANDA)**. One convolutional neural net is trained on semantic part patches for each poselet and then the activations of all nets are concatenated to obtain a pose-normalized deep representation. The final attributes are predicted by linear SVM classifier using the pose-normalized representations.

components with root filter and part filters capturing viewpoint and pose variations. Zhang *et al.* proposed deformable part descriptors [119], using DPM part boxes as the building block for pose-normalized representations for fine-grained categorization task. Poselets [12, 13] are part detectors trained on positive examples clustered using keypoint annotations; they capture a salient pattern at a specific viewpoint and pose. Several approaches [32, 117] have used poselets as a part localization scheme for fine-grained categorization tasks which are related to attribute prediction. Although part-based methods have been successful on several tasks, they have been limited by the choice of the low-level features applied to the image patches.

In this chapter, we propose the PANDA model, Pose Alignment Networks for Deep Attribute modeling, which augments deep convolutional networks to have input layers based on semantically aligned part patches. Our model learns features that are specific to a certain part under a certain pose. We then combine the features produced by many such networks and construct a pose-normalized deep representation. In this work, we use poselets, but the method can also be generalized to other part-based models, such as DPM [33]. We demonstrate the effectiveness of PANDA on attribute classification problems and present state-of-the-art experimental results on three datasets, a large-scale attribute dataset from the web, the Berkeley Attributes of People Dataset [11] and the Labeled Faces in the Wild dataset [56].

2.1 Background

2.1.1 Attribute classification

Attributes are used as an intermediate representation for knowledge transfer in [58, 31] on object recognition tasks. By representing the image as a list of human selected attributes enables it to recognize unseen objects with few or zero examples. Other related work on attributes includes that by Parikh *et al.* [81] exploring the relative strength of attributes by learning a rank function for each attribute, which can be applied to zero-shot learning as well as generating richer textual descriptions. There are also some related work in automatic attribute discovery. Berg *et al.* [8] proposed automatic attribute vocabularies discovery by mining unlabeled text and image data sampled from the web. Duan *et al.* [27] proposed an interactive crowd-sourcing method to discover both localized and discriminative attributes to differentiate bird species.

In [56], facial attributes such as gender, mouth shape, facial expression, are learned for face verification and image search tasks. Some of the attributes used by them are similar to what we evaluate in this work. However, the difference is that all of their attributes are about human faces and most of images in their dataset are just frontal face subjects while our dataset is much more challenging in terms of image quality and pose variations.

A very closely related work on attribute prediction is Bourdev *et al.* [11], which is a three-layer feed forward classification system and the first layer predicts each attribute value for each poselet type. All the predicted scores of first layer are combined as a second layer attribute classifier and the correlation between attributes are leveraged in the third layer. Our method is also built on poselets, from which the part correspondence is obtained to generate a pose-normalized representation.

2.1.2 Deep learning

The most popular deep learning method for vision, namely the convolutional network, has been pioneered by LeCun and collaborators [62] who initially applied it to OCR [63] and later to generic object recognition tasks [51]. As more labeled data and computational power has become recently available, convolutional nets have become the most accurate method for generic object category classification [55] and pedestrian detection [91].

Although very successful when provided very large labeled datasets, convolutional nets usually generalize poorly on smaller dataset because they require the estimation of millions of parameters. This issue has been addressed by using unsupervised learning methods leveraging large amounts of unlabeled data [87, 51, 61]. In this work, we take instead a different perspective: we make the learning task easier by providing the network with pose-normalized inputs and we also train on a related task using a larger dataset.

While there has already been some work on using deep learning methods for attribute prediction [21], in this work we explore many more ways to predict attributes, we incorporate the use of poselets in the deep learning framework and we perform a more extensive empirical

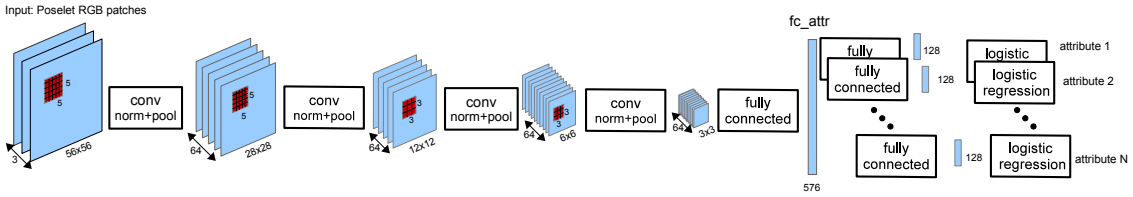


Figure 2.2: **Part-based Convolutional Neural Nets.** For each poselet, one convolutional neural net is trained on patches resized 64x64. The network consists of 4 stages of convolution/pooling/normalization and followed by a fully connected layer. Then, it branches out one fully connected layer with 128 hidden units for each attribute. We concatenate the activation from fc_attr from each poselet to obtain the pose-normalized representation. The details of filter size, number of filters and stride we used are depicted above.

validation which compares against conventional baselines and deep CNNs evaluated on the whole person region.

2.2 Pose Aligned Networks for Deep Attribute modeling (PANDA)

We explore part-based models, specifically poselets, and deep learning to obtain pose-normalized representation for attribute classification tasks. Our goal is to use poselets for part localization and incorporate these normalized parts into deep convolutional nets in order to extract pose-normalized representations. Towards this goal, we leverage both the power of convolutional nets for learning discriminative features from data and the ability of poselets to simplify the learning task by decomposing the objects into their canonical poses. We develop Pose Aligned Networks for Deep Attribute modeling (PANDA), which incorporates part-based and whole-person deep representations.

While convolutional nets have been successfully applied to large scale object recognition tasks, they do not generalize well when trained on small datasets. In this work, we propose a part-based deep convolutional neural nets, using poselets based part model (but DPM could also be used). Starting from well-aligned poselet patches, a deep net is trained on patches from each poselet yielding highly localized and discriminative feature representations.

Specifically, we start from aligned poselet patches and resize each patch to 64x64 pixels and some example poselet patches are shown in Figure 2.3. The overall convolutional net architecture is shown in Figure 2.2. First, we randomly jitter the image and flip it horizontally with probably 0.5 in order to improve generalization. The network consists of four convolutional, max pooling, local response normalization stages, then followed by a fully connected layer with 576 hidden units. After that, the network branches out one fully connected network with 128 hidden units for each attribute and each of the branch outputs a binary classifier of the attribute. The last two layers are split to let the network develop customized features for each attribute (e.g., detecting whether a person wears a “dress” or

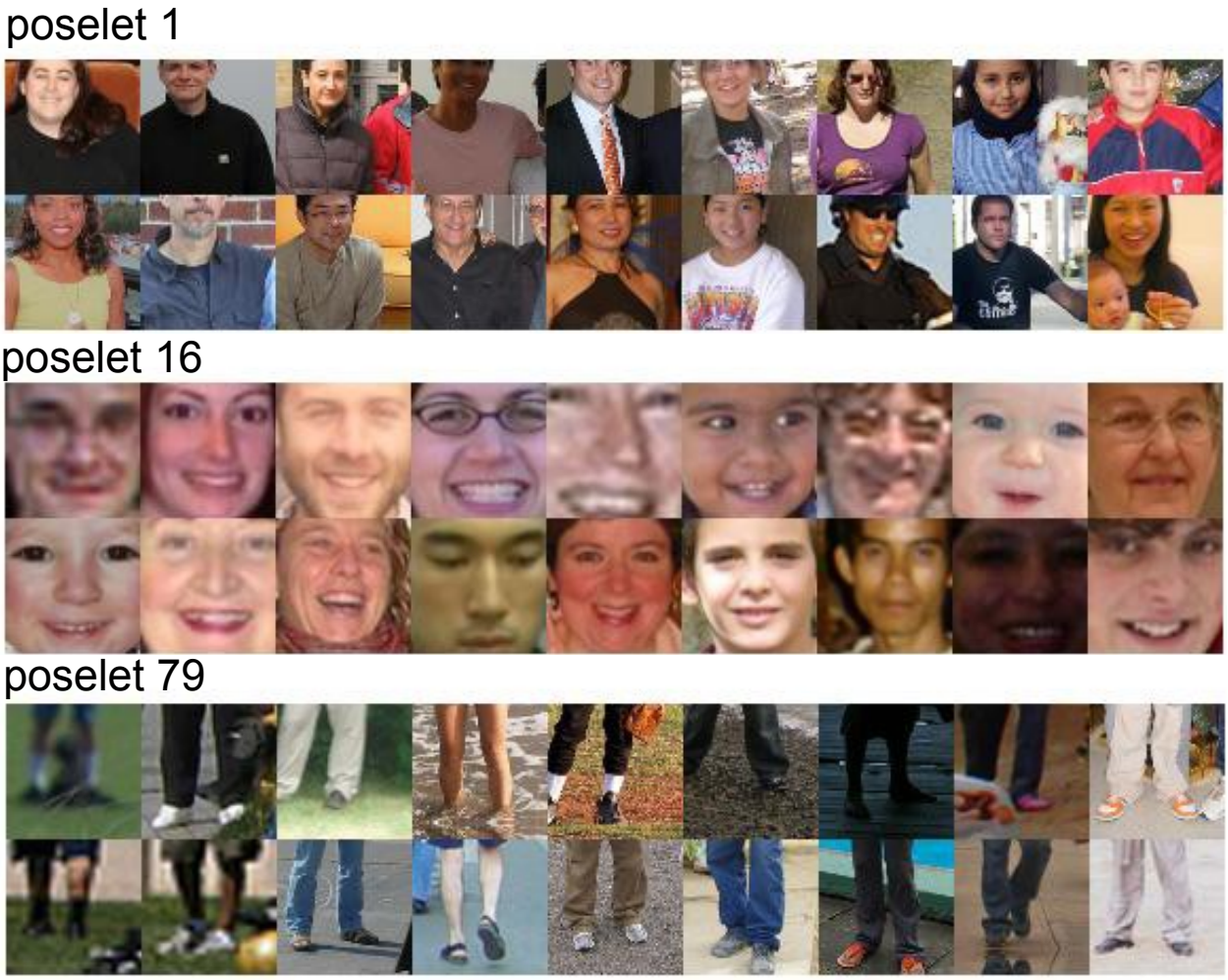


Figure 2.3: **Poselet Input Patches.** For each poselet, we use the detected patches to train a convolution neural net. Here are some examples of input poselet patches and we are showing poselet patches with high scores for poselet 1,16 and 79.

"sunglasses" presumably reuqires different features) while the bottom layers are shared to a) reduce the number of parameters and b) to leverage common low-level structure.

The whole network is trained jointly by standard back-propagation of the error [88] and stochastic gradient descent [10] using as a loss function the sum of the log-losses of each attribute for each training sample. The details of the layers are given in Figure 2.2, further implementation details can be found in [55]. To deal with noise and inaccurate poselet detections, we train on patches with high poselet detection scores and then we gradually add more low confidence patches.

Different parts of the body may have different signals for each of the attributes and sometimes signals coming from one part cannot infer certain attributes accurately. For example, deep net trained on person leg patches contains little information about whether the

person wears a hat. Therefore, we first use deep convolutional nets to generate discriminative image representations for each part separately and then we combine these representations for the final classification. Specifically, we extract the activations from fc_attr layer in Figure 2.2, which is 576 dimensional, for the CNN at each poselet, and concatenate the activations of all poselets together into 576*150 dimensional feature. If a poselet does not activate for the image, we simply leave the feature representation to zero.

The part-based deep representation mentioned above leverages both the discriminative deep convonvolutional features and part correpondence. However, poselet detected parts may not always cover the whole image region and in some degenerate cases, images may have few poselets detected. To deal with that, we also incorporate a deep network covering the whole-person bounding box region as input to our final pose-normalized representation.

Based on our experiments, we find a more complex net is needed for the whole-person region than for the part regions. We extract deep convolutional features from the model trained on Imagenet [55] using the open source package provided by [26] as our deep representation on whole image patch.

As shown in Figure 2.1, we incorporate both part-based deep representation and deep representation on whole image patch in our model to obtain the deep pose-normalized representation. A linear SVM classifier is trained using the pose-normalized representation for each of the attribute to get the final prediction.

2.3 Datasets

2.3.1 The Berkeley Human Attributes Dataset

We tested our method on the Berkeley Human Attributes Dataset [11]. This dataset consists of 4013 training, and 4022 test images collected from PASCAL and H3D datasets. The dataset is challenging as it includes people with wide variation in pose, viewpoint and occlusion. About 60% of the photos have both eyes visible, so many existing attributes methods that work on frontal faces will not do well on this dataset.

2.3.2 Attributes 25K Dataset

Unfortunately the training portion of the Berkeley dataset is not large enough for training our deep-net models (they severely overfit when trained just on these images). We collected an additional dataset from the web of 24963 people split into 8737 training, 8737 validation and 7489 test examples. We made sure the images do not intersect those in the Berkeley dataset. The statistics of the images are similar, with large variation in viewpoint, pose and occlusions.

We train on our large training set and report results on both the corresponding test set and the Berkeley Attributes test set. We chose to use a subset of the categories from

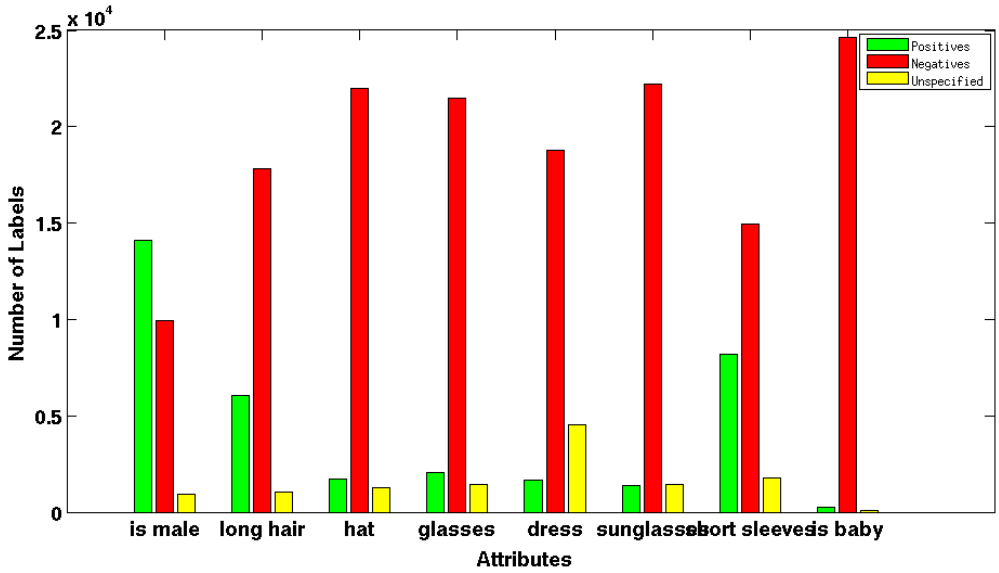


Figure 2.4: Statisitcs of the number of groundtruth labels on Attribute 25k Dataset. For each attribute, green is the number of positive labels, red is the number of negative labels and yellow is the number of uncertain labels.

the Berkeley dataset and add a few additional categories. This will allow us to explore the transfer-learning ability of our system.

Not every attribute can be inferred from every image. For example, if the head of the person is not visible, we cannot enter a label for the "wears hat" category. The statistics of ground truth labels are shown on Figure 2.4.

2.4 Experiments

In this section, we will present a comparative performance evaluation of our proposed method.

2.4.1 Results on the Berkeley Attributes of People Dataset

On Table 2.1 we show the results on applying our system on the publicly available Berkeley Attributes of People dataset. **Poselets** and **DPD** rows show the results on that dataset as reported by [11] and [119]. For our method, **PANDA**, we use Attributes25K dataset to train the poselet-level CNNs of our system, and we used the Berkeley dataset validation examples to train the SVM.

As the table shows, our system outperforms all the prior methods across most attributes. In the case of t-shirt, [11] performs slightly better, perhaps due to the fact that they use skin-tone channel and part masks. It should be noted that our method is conceptually

(a) Query: Women with long hair who wear glasses.

(b) Query: people who wear hats and glasses.

(c) Query: men with short pants and glasses.

Figure 2.5: **Example of PANDA queries.** Top results returned by our proposed method PANDA on Berkeley Attributes of People Dataset for query about multiple attributes. The prediction scores of multiple attributes are computed as a linear combination of attribute prediction scores.

very simple: We feed the raw RGB pixels to each poselet CNN, then we concatenate the features extracted from each CNN and train a linear SVM classifier. In contrast, [11] uses a combination of HOG features, color histogram features, skin channels, and part-dependent soft masks and combines the results using context by training a polynomial SVM. [119] use gradient, LBP, RGB and normalized RGB combined in Spatial Pyramid Match Kernel.

Note also that the attributes t-shirt, shorts, jeans and long pants are not present in the Attributes25K dataset. In Figure 2.5, we show the attribute prediction results returned by PANDA by generating queries of several attributes. To search for person images wearing both hat and glasses, we return the images with the largest cumulative score for those attributes.

In Figure 4.5, we show the top failure cases for wear tshirts and wear long sleeves on the test dataset. In the case of wearing tshirt, the top failure cases are picking the sleeveless,

Attribute	male	long hair	glasses	hat	tshirt
Poselets[11]	82.4	72.5	55.6	60.1	**51.2**
DPD[119]	83.7	70.0	38.1	73.4	49.8
PANDA	**91.7**	**82.7**	**70.0**	**74.2**	49.8
Attribute	longsleeves	shorts	jeans	long pants	Mean AP
Poselets[11]	74.2	45.5	54.7	90.3	65.18
DPD[119]	78.1	64.1	78.1	93.5	69.88
PANDA	**86.0**	**79.1**	**81.0**	**96.4**	**78.98**

Table 2.1: Attribute classification results on Berkeley Attributes of People Dataset as compared to the methods of Bourdev *et al.* [11] and Zhang *et al.* [119] .

Attribute	male	long hair	hat	glasses
Poselets150[11]	86.00	75.31	29.03	36.72
DPD[119]	85.84	72.40	27.55	23.94
DeCAF [26]	82.47	65.03	19.15	14.91
PANDA	**94.10**	**83.17**	**39.52**	**72.25**
dress	sunglasses	short sleeves	baby	mean AP
34.73	50.16	55.25	41.26	51.06
48.55	34.36	54.75	41.38	48.60
44.68	26.91	56.40	50.19	44.97
59.41	**66.62**	**72.09**	**78.76**	**70.74**

Table 2.2: Average Precision on the Attributes25K-test dataset.

which look very similar to tshirts. And for the case of wearing long sleeves, some of failures are due to the occlusion of the arms and presence of jacket.

2.4.2 Results on the Attributes25K Dataset

Table 2.2 shows results on the Attributes25K-test Dataset.

Poselets150. shows the performance of our implementation of the three-layer feed-forward network proposed by [11]. Instead of the 1200 poselets in that paper we used the 150 publicly released poselets, and instead of multiple aspect ratios we use 64x64 patches. Our system underperforms [11] and on the Berkeley Attributes of People dataset yields mean AP of 60.6 vs 65.2, but it is faster and simpler and we have adopted the same setup for our CNN-based poselets. This allows us to make more meaningful comparisons between the two methods.

DPD and **DeCAF** We used the publicly available implementations of [119] based on deformable part models and [26] based on CNN trained on ImageNet. The results in this table confirm that our method performs very well.

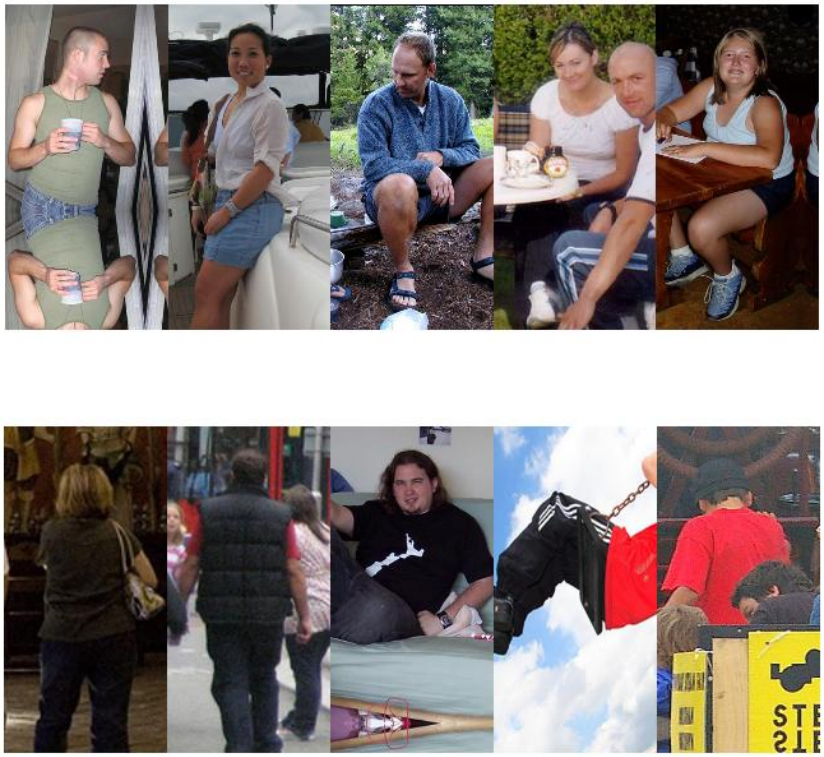


Figure 2.6: Example of failure cases on Berkeley Attributes of People Dataset. On the top we show highest scoring failure cases for "wears t-shirt" and on the bottom – for "wears long sleeves".

2.4.3 Component Evaluation

We now explore the performance of individual components of our system as shown on Table 2.3 using the Berkeley dataset. Our goal is to get insights into the importance of using deep learning and the importance of using parts.

How well does a conventional deep learning classifier perform? We first explore a simple model of feeding the raw RGB image of the person into a deep network. To help with rough alignment and get signal from two resolutions we split the images into four 64x64 patches – one from the top, center, and bottom part of the person's bounds, and one from the full bounding box at half the resolution.

We resize each image to 64x128 pixels and crop it into three overlapping 64x64 squares (top, middle and bottom). We also resize the input image to 64x64 pixels to provide the network with information about the whole image at a coarser scale. In total we have 4 concatenated 64x64 square color images as input (12 channels). We train a CNN on this 12x64x64 input on the full Attributes-25K dataset. The structure we used is similar to the

Attribute	male	long hair	glasses	hat	tshirt
DL-Pure	80.65	63.23	30.74	57.21	37.99
DeCAF	79.64	62.29	31.29	55.17	41.84
Poselets150 L2	81.70	67.07	44.24	54.01	42.16
DLPoselets	**92.10**	82.26	**76.25**	65.55	44.83
PANDA	91.66	**82.70**	69.95	**74.22**	**49.84**
Attribute	longsleeves	short	jeans	long pants	Mean AP
DL-Pure	71.76	35.05	60.18	86.17	58.11
DeCAF	78.77	**80.66**	**81.46**	96.32	67.49
Poselets150 L2	71.70	36.71	42.56	87.41	58.62
DLPoselets	77.31	43.71	52.52	87.82	69.15
PANDA	**86.01**	79.08	80.99	**96.37**	**78.98**

Table 2.3: Relative performance of baselines and components of our system on the Berkeley Attributes of People test set.

CNN in Figure 2.2 and it consists of two convolution/normalization/pooling stages, followed by a fully connected layer with 512 hidden units followed by nine columns, each composed of one hidden layer with 128 hidden units. Each of the 9 branches outputs a single value which is a binary classifier of the attribute. We then use the CNN as a feature extractor on the validation set by using the features produced by the final fully connected layer. We train a logistic regression using these features and report its performance on the ICCV test set as **DL-Pure** on Table 2.3.

We also show the results of our second baseline – DeCAF, which is the global component of our system. Even though it is a convolutional neural net originally trained on a completely different problem (ImageNet classification), it has been exposed to millions of images and it outperforms **DL-Pure**.

How important is deep learning at the part level?. By comparing the results of **Poselets150L2** and **DLPoselets** we can see the effect of deep learning at the part level. Both methods use the same poselets, train poselet-level attribute classifiers and combine them at the person level with a linear SVM. The only difference is that Poselets150L2 uses the features as described in [11] (HOG features, color histogram, skin tone and part masks) whereas DLPoselets uses features trained with a convolutional neural net applied to the poselet image patch. As our table shows, deep-net poselets result in increased performance.

PANDA shows the results of our proposed system which combines DeCAF and DLPoselets. As Table shows, our part and holistic classifiers use complementary features and combining them together further boosts the performance.

Partition	male	long hair	glasses	hat	tshirt
Frontal	92.55	88.40	77.09	74.40	51.69
Profile	91.42	59.38	37.06	69.47	49.02
Back-facing	88.65	63.77	72.61	72.19	55.20
All	91.66	82.70	69.95	74.22	49.84
Partition	longsleeves	shorts	jeans	long pants	Mean AP
Frontal	86.84	78.00	79.63	95.70	80.47
Profile	84.61	85.57	82.71	98.10	73.04
Back-facing	84.32	74.01	86.12	96.68	77.06
All	86.01	79.08	80.99	96.37	78.98

Table 2.4: Performance of PANDA on front-facing, profile-facing and back-facing examples of the Berkeley Attributes of People test set.

2.4.4 Robustness to viewpoint variations

In Table 2.4, we show the performance of our method as a function of the viewpoint of the person. We considered as *frontal* any image in which both eyes of the person are visible, which includes approximately 60% of the dataset. *Profile* views are views in which one eye is visible and *Back-facing* are views where both eyes are not visible. As expected, our method performs best for front-facing people because they are most frequent in our training set. However, the figure shows that PANDA can work well across a wide range of viewpoints.

2.4.5 Results on the LFW Dataset

We also report results on the Labeled Faces in the Wild dataset [56]. The dataset consists of 13233 images of cropped, centered frontal faces. Such constrained environment does not leverage the strengths of our system in its ability to deal with viewpoint, pose and partial occlusions. Nevertheless it provides us another datapoint to compare against other methods. This dataset contains many attributes, but unfortunately the ground truth labels are not released. We used crowd-sourcing to collect ground-truth labels for the gender attribute only. We split the examples randomly into 3042 training and 10101 test examples with the only constraint that the same identity may not appear in both training and test sets. We used our system whose features were trained on Attribute-25K to extract features on the 3042 training examples. Then we trained a linear SVM and applied the classifier on the 10101 test examples. We also used the publicly available gender scores of [56] to compute the average precision of their system on the test subset. The results are shown on Table 2.5.

PANDA's AP on LFW is 99.54% using our parts model, a marked improvement over the previous state of the art. Our manual examination of the results shows that roughly 1 in 200 test examples either had the wrong ground truth or we failed to match the detection results with the correct person. Thus PANDA shows nearly perfect gender recognition performance

Method	Gender AP
Simile [56]	95.52
FrontalFace poselet	96.43
PANDA	**99.54**

Table 2.5: Average precision of PANDA on the gender recognition of the LFW dataset.

in LFW. This experiment also shows the difficulty of the Berkeley Attributes of People and the Attributes25K datasets.

One interesting observation is that, even though the dataset consists of only frontal-face people, the performance of our frontal-face poselet is significantly lower than the performance of the full system. This suggests that our system benefits from combining the signal from multiple redundant classifiers, each of which is trained on slightly different set of images.

2.5 Summary

In this chapter, we propose a part-based method for human attribute classification. The motivation comes from the same idea of using parts to discount pose and viewpoint variations to distinguish the subtle differences. The method is conceptually simple and leverages the strength of convolutional neural nets without requiring datasets of millions of images. It uses poselets to factor out the pose and viewpoint variation which allows the convolutional network to focus on the pose-normalized appearance differences. We concatenate the deep features at each poselet and add a deep representation of the whole input image. This might not be ideal and we will show an alternative way of combining features in the next chapter.

Chapter 3

Person recognition beyond frontal faces

Recognizing people we know from unusual poses is easy for us, as illustrated on Figure 3.1. In the absence of a clear, high-resolution frontal face, we rely on a variety of subtle cues from other body parts, such as hair style, clothes, glasses, pose and other context. We can easily picture Charlie Chaplin's mustache, hat and cane or Oprah Winfrey's curly volume hair. Yet, examples like these are beyond the capabilities of even the most advanced face recognizers. While a lot of progress has been made recently in recognition from a frontal face, non-frontal views are a lot more common in photo albums than people might suspect. For example, in our dataset which exhibits personal photo album bias, we see that only 52% of the people have high resolution frontal faces suitable for recognition. Thus the problem of recognizing people from any viewpoint and without the presence of a frontal face or canonical pedestrian pose is important, and yet it has received much less attention than it deserves. We believe this is due to two reasons: first, there is no high quality large-scale dataset for unconstrained recognition, and second, it is not clear how to go beyond a frontal face and leverage these subtle cues. We address both of these problems in this chapter and show pose-normalized representations can also improve the application of person recognition.

To address the first problem, we introduce the *People In Photo Albums (PIPA)* dataset, a large-scale recognition dataset collected from Flickr photos with creative commons licenses. It consists of 37,107 photos containing 63,188 instances of 2,356 identities and examples are shown in Figure 3.2. We tried carefully to preserve the bias of people in real photo albums by instructing annotators to mark every instance of the same identity regardless of pose and resolution. Our dataset is challenging due to occlusion with other people, viewpoint, pose and variations in clothes. While clothes are a good cue, they are not always reliable, especially when the same person appears in multiple albums, or for albums where many people wear similar clothes (sports, military events), as shown in Figure 3.3. As an indication of the difficulty of our dataset, the DeepFace system [99], which is one of the state-of-the-art recognizers on LFW [47], was able to register only 52% of the instances in our test set and, because of that, its overall accuracy on our test set is 46.66%. The dataset is publicly

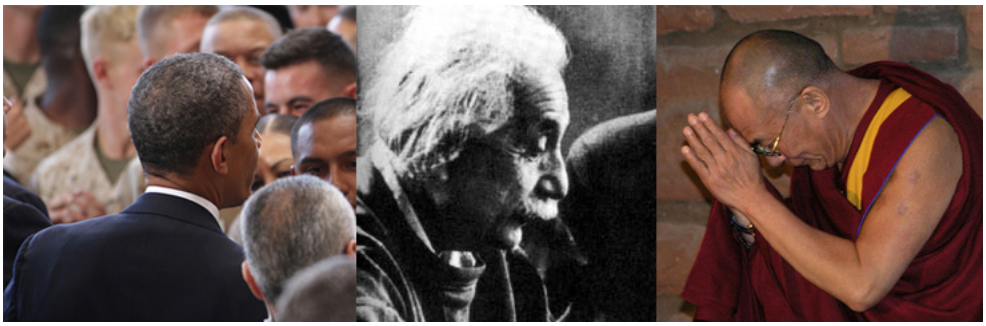


Figure 3.1: Recognizing people beyond frontal face. We are able to easily recognize people we know in unusual poses, and even if their face is not visible. In this chapter we explore the subtle cues necessary for such robust viewpoint-independent recognition.

available[1].

To address the second problem, we propose a Pose Invariant PErson Recognition (PIPER) method, which uses part-level person recognizers to account for pose variations. Inspired by Chapter 2, we also use poselets [12] as our part models and train identity classifiers for each poselet. However this task is significantly harder than attribute classification since we have many more classes with significantly fewer training examples per class. Also, we found that combining parts by concatenating their feature in the manner of Chapter 2 is not effective for this task. It results in feature vectors that are very large and overfit easily when the number of classes is large and training examples are few. Instead, we found training each part to do identity recognition and combining their predictions achieves better performance. Unlike [11], we propose a new way to handle the sparsity from poselet detections which boosts the performance by a large margin.

We demonstrate the effectiveness of PIPER by using three different experimental settings on our dataset. Our method can achieve 83.05% accuracy over 581 identities on the test set. Moreover when a frontal face is available, it improves the accuracy over DeepFace from 89.3% to 93.4%, which is close to 40% decrease in relative error.

3.1 Background

3.1.1 Face recognition

There has been dramatic progress made in face recognition in the past few decades from EigenFace [104] to the state-of-art face recognition system [99] by using deep convolutional nets. Most of the existing face recognition systems require constrained setting of frontal faces and explicit 3D face alignment or facial keypoint localizations. Some other works [109, 22] have addressed robust face recognition systems to deal with varying illumination, occlusion

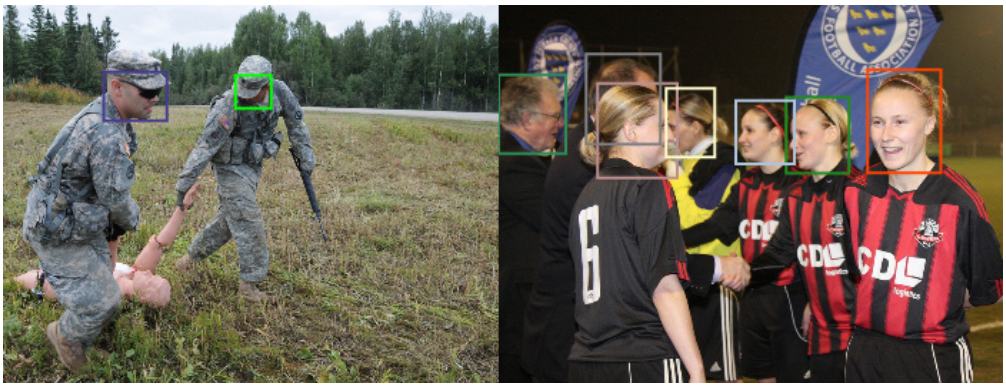
[1] `http://www.eecs.berkeley.edu/~nzhang/piper.html`

Figure 3.2: **Example photos from our dataset.** These are taken from a single album and show the associated identities. Each person is annotated with a ground truth bounding box around the head, with each color representing one identity. If the head is occluded, the expected position is annotated.

and disguise. Due to our unconstrained setting, most conventional face recognition systems have limited performance on our dataset.

3.1.2 Person identification in photo albums

Tagging of personal photo albums is an active research topic. To address the limitation of face recognition systems, various approaches have been proposed to incorporate context. For example, the authors in [3, 76] proposed methods to incorporate contextual cues including clothing appearance and meta data from photos for person identification in photo collections. Sivic et al. [96] proposed a simple pictorial structure model to retrieve all the occurrences



(a) While clothing can be discriminative it does not help for military or business activities, for example, where people dress similarly.

(b) The same individual may appear in multiple albums wearing different clothes.

Figure 3.3: **Challenges of our dataset.**

of the same individual in a sequence of photographs. Lin et al. [66] presented a generative probabilistic approach to model cross-domain relationships to jointly tag people, events and locations. In [39], the authors try to find all images of each person in the scene from a crowed public event.

There is also some related work to discover the social connection between people in the photo collections. Wang et al. [107] proposed a model to represent the relationship between the position and pose of people and their identities. In [37], the authors investigated the different factors that are related to the positions of people in a group image. Another interesting direction is to name characters in TV series. In [29, 95, 28], the authors proposed approach to automatically label the characters by using aligned subtitle and script text. Tapaswi et al. [100] proposed a Markov Random Field (MRF) method to combine face recognition and clothing features and they tried to name all the appearance of characters in TV series including non frontal face appearance. Later they presented another semi-supervised learning method [16] for the same task.

Split	All	Train	Val	Test	Leftover
Photos	37,107	17,000	5,684	7,868	6,555
Albums	1,438	579	342	357	160
Instances	63,188	29,223	9,642	12,886	11,437
Identities	2,356	1,409	366	581	-
Avg/identity	26.82	20.74	26.34	22.18	-
Min/identity	5	10	5	10	-
Max/identity	2928	99	99	99	-

Table 3.1: Statistics of our dataset.

3.1.3 Person re-identification in videos

The task of person re-identification is to match pedestrian images from different cameras and it has important applications in video. Existing work is mainly focused on metric learning [85, 45, 64] and mid-level feature learning [44, 123, 122, 30, 59, 80]. Li et al. [65] propose a deep network using pairs of people to encode photometric transformation. Yi et al. [115] used a siamese deep network to learn the similarity metric between pairs of images.

3.2 People In Photo Albums Dataset

To our knowledge, there is no existing large scale dataset for the task of person recognition. The existing datasets for person re-identification, such as VIPeR [43] and CUHK01 [64], come mostly from videos and they are low resolution images taken from different cameras from different viewpoints. The Gallagher Collection Person Dataset [36] is the closest to what we need, however the released subset has only 931 instances of 32 identities which is approximately 1.5% of the size of our dataset. Furthermore, [36] have only labeled the frontal faces. The Buffy dataset used in [95, 29] is a video dataset and it only has less than 20 different characters.

Our problem setting is to identify a person in the "wild" without any assumption of frontal face appearance or straight pedestrian pose. We don't assume that the person is detected by a detector; our instruction to annotators is to mark the head (even if occluded) for any people they can co-identify, regardless of their pose, and the image resolution.

3.2.1 General statistics

We collected our dataset, People In Photo Albums (PIPA) Dataset, from public photo albums uploaded to Flickr [2]. Those albums were uploaded from 111 users. Photos of the same person have been labeled with the same identity but no other identifying information is preserved. Table 3.1 shows statistics of our dataset.

[2] `https://www.flickr.com/`

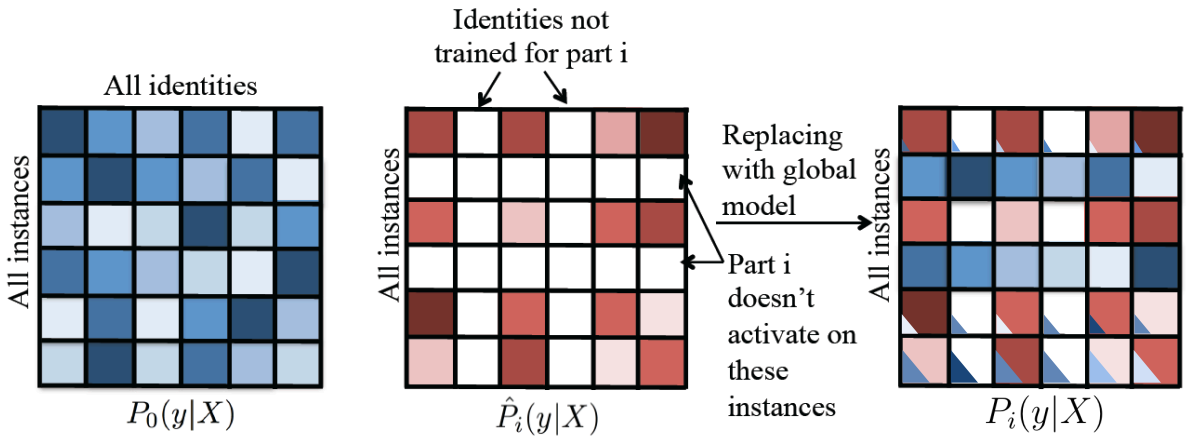


Figure 3.4: Example of sparsity filling. On the left we show the predictions of the global model for every identity and every instance. The poselet classifier in the middle does not activate for two instances (the white rows) and is not trained to recognize two identities (the white columns). On the right we show how in the normalized probability we fill-in missing rows from the global model as in the top of Equation 3.2. In addition we account for the missing columns by linearly interpolating each row with the global model based on the likelihood that the identity is not coming from one of the missing columns (the triangles)

3.2.2 Collection Method

Our data collection consists of the following steps:

1. **Album Filtering**. After downloading thousands of albums from Flickr, we first show the annotators a set of photos from the album and ask them to filter out albums which are not of people albums, such as landscape, flowers, or photos where person co-occurrence is very low.

2. **Person Tagging**. Then given each album, we ask the annotators to select all the individuals that appear at least twice in that album and draw a bounding box around their heads with different color indicating different identity. If the head is partially/fully occluded, we mark the region of where the head should be. The head bounds may also be partially/fully outside the image. Not every person is tagged. In a crowd scene we ask the annotators to tag no more than 10 people. The interface of person tagging is shown in Fig 3.9a. The annotator can scroll over the faces on the left. If clicking on one individual, it will show all the instances of that person. The original images and head annotations are shown on the right where the annotators draw all the heads of different individuals. If the person's head is occluded, the annotator will draw a bounding box around the expected position of the head.

3. **Cross-Album Merging**. Often the same identities appear in multiple albums, in which case their identities should be merged. While it is not feasible to do so across

all albums, we consider the set of albums uploaded by the same uploader and we try to find the same identities appearing in multiple albums and merge them. Showing all identities from all albums is a challenging UI task for uploaders that have dozens of large albums. We show our annotation interface in Figure 3.9b. The top row shows a set of merged individuals. Each column in the bottom section corresponds to an album from the same uploader. Each face is an example face of a different individual. Merging is done by selecting a set of faces across albums, optionally selecting an individual from the top row to merge into, and clicking the merge button.

4. **Instance Frequency Normalization**. After merging, we count the number of instances for each individual and discard individuals that have less than 10 instances. In addition, a few individuals have a very large number of instances which could bias our dataset. To prevent such bias, we restrict the maximum number of instances per individual to be 99. We randomly sample 99 instances and move the remaining ones into a "leftover" set. Our leftover set consists of 11,437 instances of 54 individuals.

5. **Dataset Split**. We split the annotations randomly into three sets – training, validation and test. To ensure complete separation between the sets, all the photos of the same uploader fall in the same set. That ensures that the set of photos, identities and instances across the three sets is disjoint. We do a random permutation of the uploaders and we pick the first K of them so that the number of person instances reaches about 50% and we assign those to the training set. We assign the next 25% to validation and the remaining 25% to test. While we target 50-25-25 split we cannot assure that the instances will be partitioned precisely due to the constraints we impose. See Table 3.1 for more details about the splits.

3.3 Pose Invariant Person Recognition (PIPER)

We introduce a novel view invariant approach to combine information of different classifiers for the task of person recognition. It consists of three components:

- The global classifier, a CNN trained on the full body of the person.
- A set of 107 poselet classifiers, each is a CNN trained on the specific poselet pattern using [12]. [3]
- An SVM trained on the 256 dimensional features from DeepFace [99].

In total we have 109 part classifiers. The identity prediction of PIPER is a linear combination of the predicted probabilities of all classifiers:

[3]The original set of poselets is 150 but some of them did not train well.

$$s(X, y) = \sum_i w_i P_i(y|X) \tag{3.1}$$

Here $P_i(y|X)$ is the normalized probability of identity label y given by part i for feature X and w_i is the associated weight of the part. The final identity prediction is $y^*(X) = \text{argmax}_y\, s(X, y)$.
Here is an overview of the training steps. The next sections provide a more detailed description.

1. We run poselets [12] over our dataset and match the person predictions coming from poselets to our ground truths.

2. Using the poselet patches of step 1, we train a CNN for each poselet to recognize the identities on our training set. In addition, we train a CNN for the global classifier using the patches corresponding to the full body images. In all cases we use the Convolutional Neural Net architecture by Krizhevsky *et al.* . [55]. We fine-tune the network pre-trained on ImageNet on the task of identity recognition. While recent architectures have improved the state-of-the art [94, 98] and might further improve our performance, we decided to use the Krizhevsky architecture because its performance is well studied on a number of visual tasks [26]. We then discard the final FC8 layer and treat the activations from the FC7 layer as a generic feature on which we train SVMs in the next steps.

3. We partition the validation set into two halves. We train an SVM for each part using the FC7 layer feature from Step 2 on the first half of validation and use it to compute the identity predictions $P_i(y|X)$ on the second half, and vice versa.

4. We use the identity predictions of all parts on the validation set to estimate the mixing components w_i .

5. We split the test set in half and train SVMs on top of the FC7 features on the first half of the test set and use them to compute the identity predictions $P_i(y|X)$ on the second half, and vice versa.

6. We use the identity predictions on the test set for each part $P_i(y|X)$ as well as the mixing components w_i to compute the combined identity prediction $S(X, y)$ using equation 3.1.

In the next sections we will describe the training steps, and way we compute $P_i(y|X)$ and w_i.

3.3.1 Computing Part Activations

Our groundtruth annotations consist of bounding boxes of heads. From the head boxes, we estimate the bounding box locations by setting approximate offset and scaling factor. We run poselets [12] on the images, which returns bounding boxes of detected people in the images, each of which comes with a score and locations of associated poselet activations. We use a bipartite graph matching algorithm to match the ground truth bounds to the ones predicted by the poselets. This algorithm performs globally optimal matching by preferring detections with higher score and higher overlap to truths. The output of the algorithm is a set of poselet activations associated with each ground truth person instance. We extract the image patches at each poselet and use them to train part-based classifiers.

3.3.2 Training the Part Classifiers $P_i(y|X)$

Global classifier $P_0(y|X)$

Using the FC7 layer of the CNN trained for the full body area of each instance, we train a multi-class SVM to predict each identity y. We refer to its prediction as $P_0(y|X)$.

Part-level SVM classifier $\hat{P}_i(y|X)$

Given the FC7 layer features X extracted from detected part i patch and identity labels y, we train a multi-class SVM on X to predict y and we denote the softmax of the output score as $\hat{P}_i(y|X)$.

Notice that $\hat{P}_i$ is sparse in two ways:

- Each poselet activates only on instances that exhibit the specific pose of that poselet. Some poselets may activate on 50% while others on as few as 5% of the data.
- Not all identities have examples for all poselets and thus each poselet level SVM classifier for part i is only trained on a subset F_i of all identities. Thus $\hat{P}_i(y|X)$ is inflated when $y \in F_i$ and is zero otherwise.

The sparsity pattern is correlated with the pose of the person and has almost no correlation to the identity that we are trying to estimate. Thus properly accounting for the sparsity is important in order to get high accuracy identity recognition.

Sparsity filling

We address both of these sparsity issues by using the probability distribution of our global model P_0 which is defined for all identities and activates on all instances:

$$P_i(y|X) = \begin{cases} P_0(y|X), \text{if part } i \text{ doesn't activate, or} \\ P(y \in F_i)\hat{P}_i(y|X) + P(y \notin F_i)P_0(y|X) \end{cases} \tag{3.2}$$

$$P(y \in F_i) = \sum_{y' \in F_i} P_0(y'|X) \tag{3.3}$$

In Figure 3.4 we give a visual intuition behind this formula.

3.3.3 Computing the part weights w_i

We use the validation set to compute w. We split the validation set into two equal subsets. We train the part-based SVMs on one subset and use them to compute $P_i(y|X)$ on all instances of the second subset, and vice versa. Let $P_i^j(y|X)$ denote the probability that the classifier for part i assigns to instance j being of class y given feature X. We formulate a binary classification problem which has one training example per pair of instance j and label y. If we have K parts its feature vector is $K+1$ dimensional: $[P_0^j(y|X); P_1^j(y|X); \ ... \ P_k^j(y|X)]$. Its label is 1 if instance j's label is y and -1 otherwise. We solve this by training a linear SVM. The weights w are the weights of the trained SVM.

We use the first split of validation to do a grid search for the C parameter of the SVM and test on the second split. Once we find the optimal C we retrain on the entire validation set to obtain the final vector w.

3.4 Experiments

We report results of our proposed method on our PIPA dataset and compare it with baselines. Specifically, we conduct experiments in three different settings: 1) Person recognition, 2) One-shot person identification, and 3) Unsupervised identity retrieval.

In all experiments we use the training split of our dataset to train the deep networks for our global model and each poselet and we use the validation set to compute the mixing weights w and tune the hyper-parameters. All of our results are evaluated on the test split.

3.4.1 Person Recognition

We first present experimental results on the person recognition task with our *PIPA* dataset. It is a standard supervised classification task as we train and test on same set of identities. Since the set of identities between training and test sets is disjoint, we split our test set in two equal subsets. We train an SVM on the first, use it to compute $P_i(y|X)$ on the second and vice versa. We then use the weights w trained on the validation set to get our final prediction

Method	Classification accuracy
Chance Performance	0.17%
DeepFace [99]	46.66%
FC7 of Krizhevsky *et al.* [55]	56.07%
Global Model	67.60%
PIPER w/out sparsity filling	75.35%
PIPER	83.05%

Table 3.2: Person recognition results on PIPA test set using 6442 training examples over 581 identities

Method	Non-faces subset	Faces subset
Global Model	64.3%	70.6%
DeepFace[99]	0.17%	89.3%
PIPER	71.8%	93.4%

Table 3.3: Performance on the test set when split into the subset where frontal face is visible (52%) and when it is not (48%).

as the identity that maximizes the score in equation 3.1 and we average the accuracy from both halves of the test set. Qualitative examples are shown on Figure 3.5.

Overall Performance

Table 3.2 shows the accuracy in this setting compared to some standard baselines. We compared it against DeepFace [99], which is one of the state-of-the-art face recognizers. Although it is very accurate, it is a frontal face recognizer, so it doesn't trigger on 48% of our test set and we use chance performance for those instances. As a second baseline we trained an SVM using the FC7 features of the CNN proposed by Krizhevsky *et al.* . and pretrained on ImageNet[25]. We use its activations after showing it the full body image patch for each instance. The Global Model baseline is the same CNN, except it was fine-tuned for the task of identity recognition on our training set. We also tested the performance of our model by combining the sparse part predictions, i.e. using $\hat{P}_i(y|X)$ instead of $P_i(y|X)$ in equation 3.1. The performance gap of more than 7% shows that sparsity filling is essential to achieve high recognition accuracy.

Ablation Study

Our method consists of three components – the fine-tuned Krizhevsky CNN (the Global Model), the DeepFace recognizer, and a collection of 107 Poselet-based recognizers. In this

(a) Body falls outside image

(b) Unusual pose

(c) Same clothing similar pose

(d) Confusion with other person

Figure 3.5: Examples that the combination of the Global model and DeepFace misclassify and are recovered by using all of PIPER. **(a)** In a closeup shot the full body falls outside the image and the extracted full-body patch, shown on the right, is severely misaligned. A profile-face poselet should handle this case without misalignment. **(b)** In unusual pose the full body patch may fall on the background or **(d)** on another person which will further confuse the classifier. In **(c)** people have the same clothes and similar pose which will confuse the global model.

section we explore using all combinations of these three components[4]. For each combination we retrain the mixture weights w and re-tune the hyper parameters. Table 3.4 shows the performance of each combination of these components. As the table shows, the three parts of PIPER are complementary and combining them is necessary to achieve the best performance.

[4]Since our method relies on sparsity filling from a global model $P_0(y|X)$, to remove the effect of the global model we simply set $P_0(y|X)$ to be uniform distribution.

Global Model	DeepFace[99]	Poselets	Accuracy
✓	–	–	67.60%
–	✓	–	46.66%
–	–	✓	72.18%
✓	✓	–	79.95%
✓	–	✓	78.79%
–	✓	✓	78.08%
✓	✓	✓	83.05%

Table 3.4: Person recognition performance on the PIPA test set using 6442 training examples over 581 identities as we disable some of the components of our method. PIPER gets more than 3% gain over the very strong baseline of using the fine-tuned CNN combined with the DeepFace model. DeepFace' s score is low because it only fires on 52% of the test images and we use chance performance for the rest.

Performance on face and non-face instances

Since the presence of a high resolution frontal face provides a strong cue for identity recognition and allows us to use the face recognizer, it is important to consider the performance when a frontal face is present and when it is not. Table 3.3 shows the performance on the face and non-face part of our test set. We considered the instances for which DeepFace generated a signature as the face subset. As the figure shows, when the face is not present we can significantly outperform a fine-tuned CNN on the full image. More importantly, the contextual cues and combinations of many classifiers allow us to significantly boost the recognition performance even when a frontal face is present.

3.4.2 One-Shot Learning

Figure 3.6 shows the performance of our system when the training set is tiny. We randomly pick one, two or three instances of each identity in our test set, train on those and report results on the rest of the test set. Our system performs very well even with one training example per identity, achieving 28.1% accuracy for 581 identities. This result illustrates the powerful generalization capability of our method. The generalization capabilities of deep features are well studied, but we believe the mixture of multiple part-based classifiers also helps here, since our system improves faster than the global fine-tuned Krizhevsky's CNN method.

3.4.3 Unsupervised identity retrieval

We evaluate our method on the task of retrieval: Given an instance, we measure the likelihood that one of the K nearest neighbors will have the same identity.

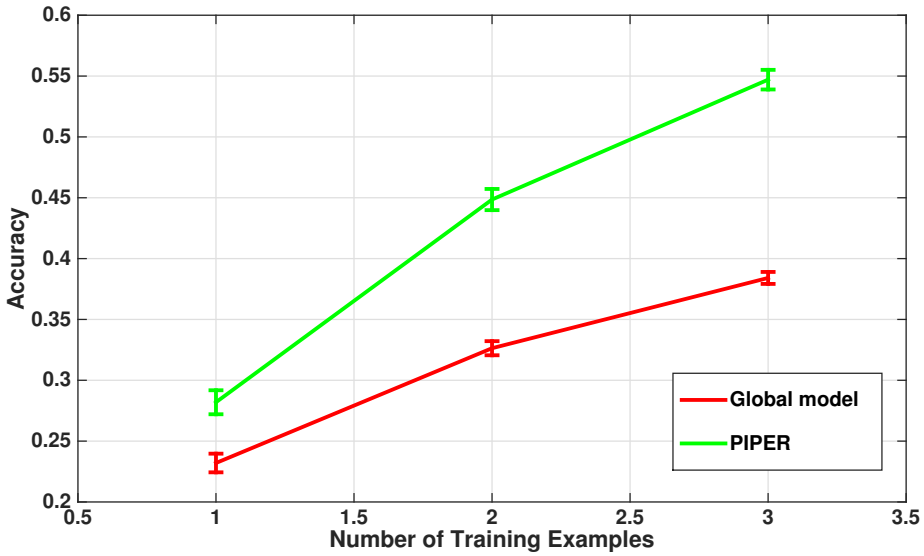


Figure 3.6: Recognition accuracy as a function of number of training examples per identity, with $\sigma = 1$ error bar. As we increase the number of training examples our system's accuracy grows faster than the full-body baseline. Chance performance is 0.0017.

To do this we used the SVMs trained on split 0 of the validation set to predict the 366 identities in the validation set. We applied them to the instances in the test set to obtain a 366-dimensional feature vector for each part and we combine the part predictions using equation 3.1 with w trained on the validation set to obtain a single 366-dimensional feature for each instance in the test set. We then, for each instance of the test set, compute the K nearest neighbors using Euclidean distance and we consider retrieval as successful if at least one of them is of the same identity. This has the effect of using the identities in the validation set as exemplars and projecting new instances based on their similarities to those identities. As Figure 3.7 shows our method is quite effective on this task – despite the low dimensional feature and without any metric learning, the nearest neighbor of 64% of the examples is of the same identity as the query. If we use the predictions of the Krizhevsky's CNN trained on ImageNet and fine-tuned on our training set, which is known to be a very powerful baseline, the nearest neighbor is of the same class in only 50% of the examples. This suggests that our model is capable of building a powerful and compact identity vector independent of pose and viewpoint. Examples of our method are shown in Figure 3.8.

3.5 Summary

This chapter complements Chapter 2 by using pose-normalized CNN model for another application: person recognition beyond frontal face. Specifically, the method combines state-of-art frontal face recognizer, CNN models trained on full body images and cropped-out poselet parts. PIPER can learn effectively even with a single training example and performs

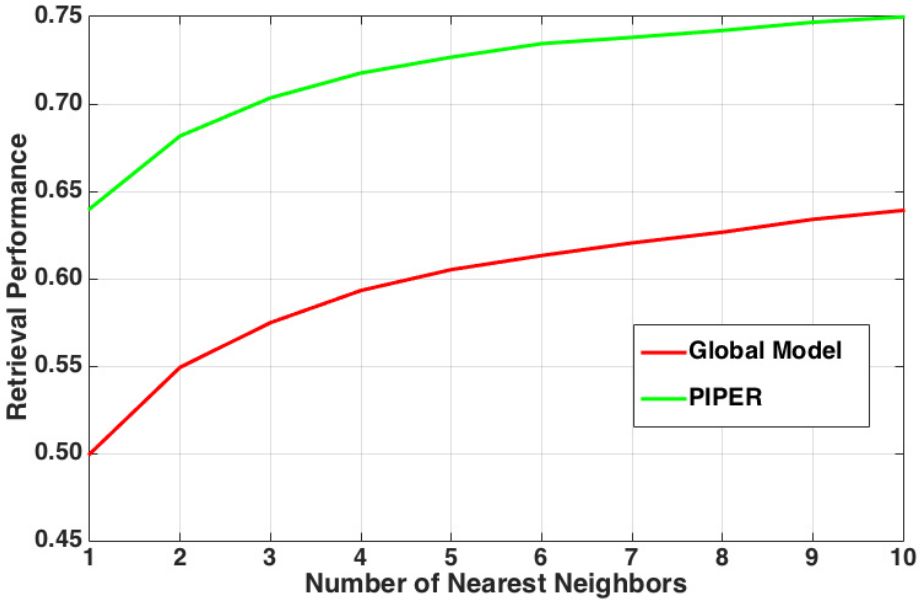


Figure 3.7: Performance of our method on identity retrieval.

surprisingly well at the task of image retrieval. While we have used PIPER for person recognition, the algorithm readily applies to generic instance co-identification, such as finding instances of the same car or the same dog. The other contribution of this chapter is the *People In Photo Albums* dataset, which is the first of its kind large scale data set for person coidentification in photo albums.

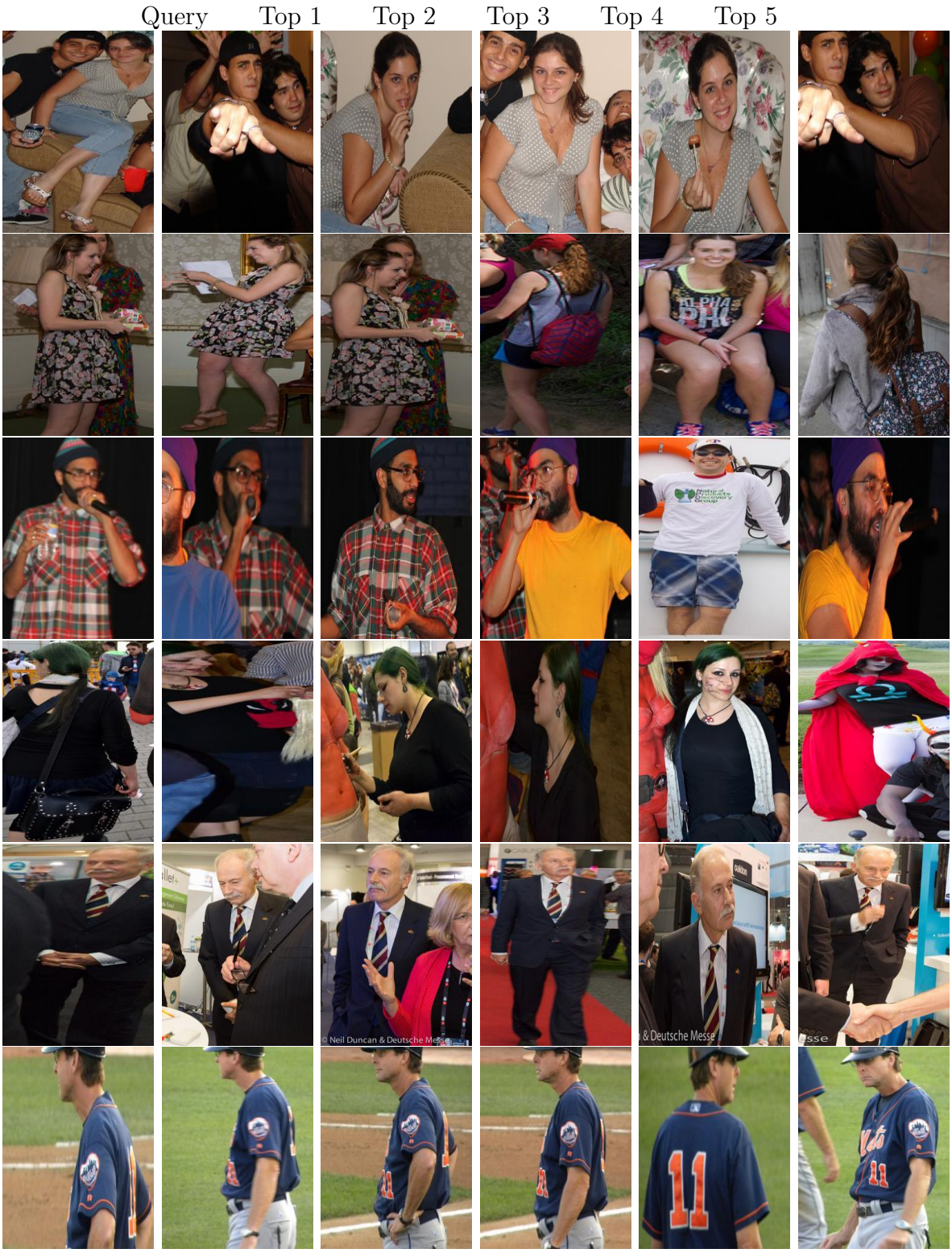
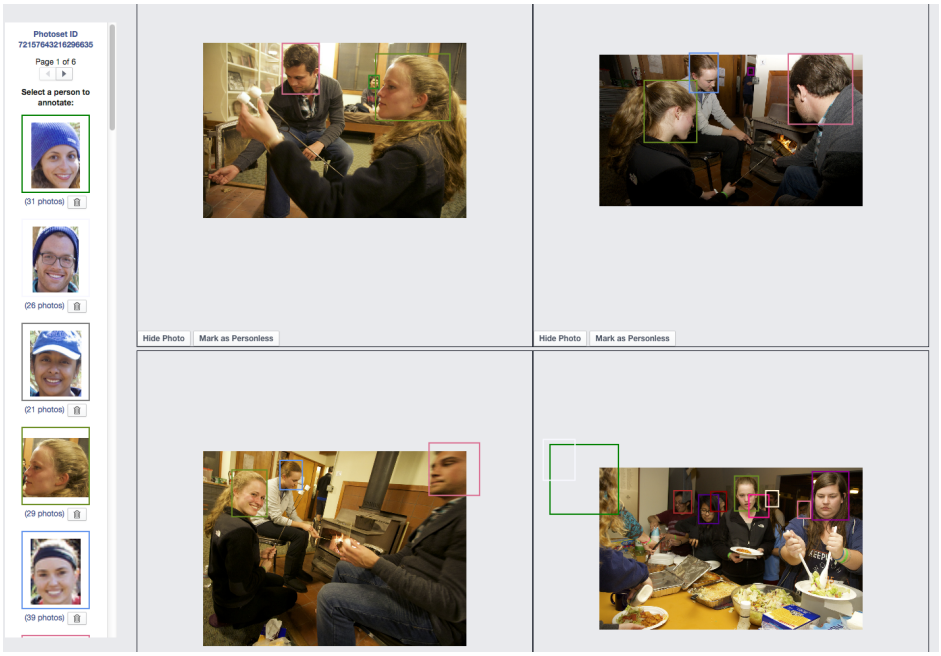
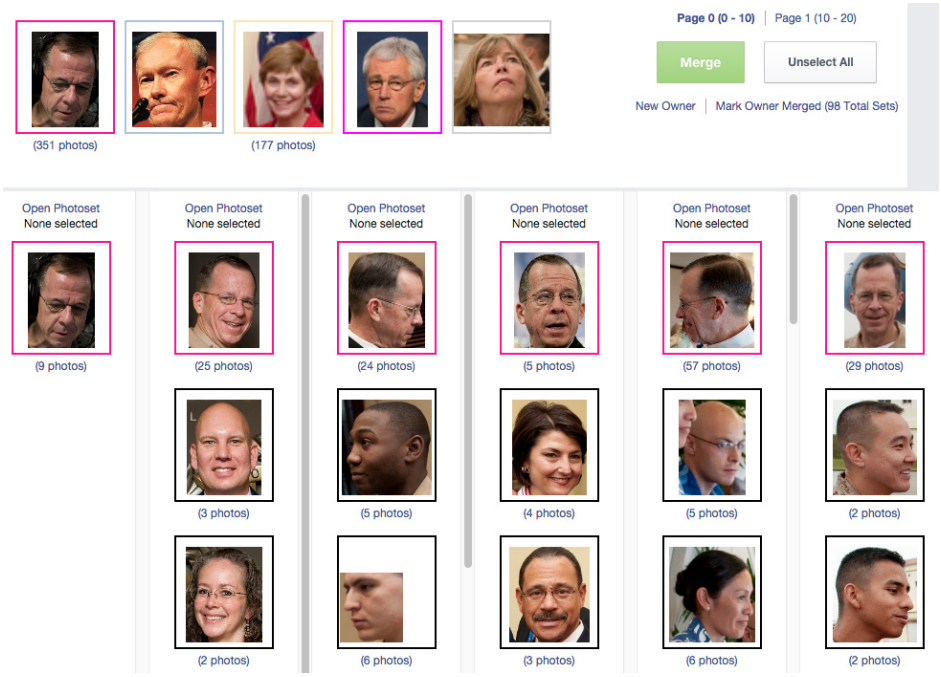


Figure 3.8: Example of PIPER results on unsupervised identity retrieval. For each row we show the query image followed by the top 5 ranked retrieval images. Those are cropped bounding box images on test split and are stretched to make visualization aligned.



(a) Interface for annotating identities in one album.



(b) Interface for merging identities across albums.

Figure 3.9: **Interfaces for our annotation system.**

Chapter 4

Part-based RCNN model

Previous chapters address using strong supervised parts to normalize the visual representations for the application of attribute classification and person recognition. In this chapter and next chapter, we will focus on the main fine-grained categorization task. As mentioned before, our motivation is to localize the parts and establish correspondence between object instances in order to discount object pose variations and camera view points.

Previous work has investigated part-based approaches to this problem [9, 32, 69, 112, 117, 42]. The bottleneck for many pose-normalized representations is indeed accurate part localization. The Poselet [12] used in the previous two chapters and DPM [33] methods have previously been utilized to obtain part localizations with a modest degree of success; methods generally report adequate part localization only when given a known bounding box at test time [20, 40, 83, 82, 111]. By developing a novel deep part detection scheme, we propose a fine grained categorization system which requires no knowledge of object bounding box at test time, and can achieve performance rivaling previously reported methods requiring the ground truth bounding box at test time to filter false positive detections.

The recent success of convolutional networks, like [55], on the ImageNet Challenge [48] has inspired further work on applying deep convolutional features to related image classification [26] and detection tasks [41]. In [41], Girshick et al. achieved breakthrough performance on object detection by applying the CNN of [55] to a set of bottom-up candidate region proposals [105], boosting PASCAL detection performance by over 30% compared to the previous best methods. Independently, OverFeat [90] proposed localization using a CNN to regress to object locations. However, the progress of leveraging deep convolutional features is not limited to basic-level object detection. In many applications such as fine-grained recognition, attribute recognition, pose estimation, and others, reasonable predictions demand accurate part localization.

Feature learning has been used for fine-grained recognition and attribute estimation, but was limited to engineered features for localization. DPD-DeCAF [119] used DeCAF [26] as a feature descriptor, but relied on HOG-based DPM [33] for part localization. PANDA [121] learned part-specific deep convolutional networks whose location was conditioned on HOG-based poselet models. These models lack the strength and detection robustness of R-

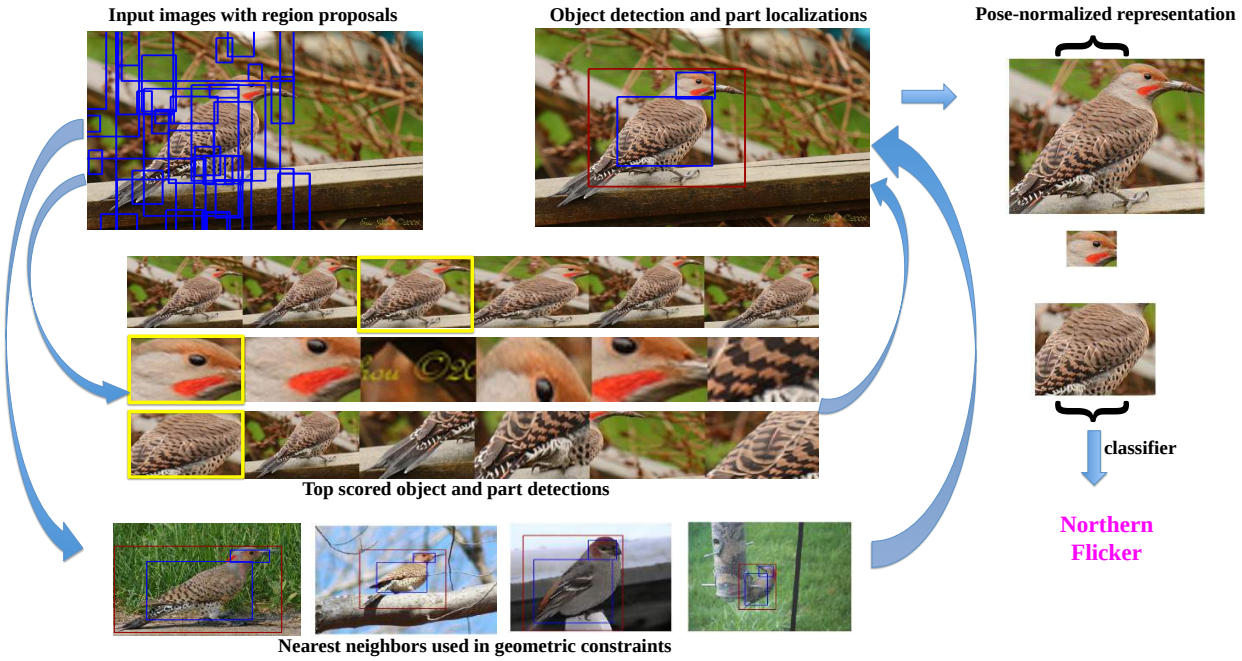


Figure 4.1: **Overview of our part localization.** Starting from bottom-up region proposals (top-left), we train both object and part detectors based on deep convolutional features. During test time, all the windows are scored by all detectors (middle), and we apply non-parametric geometric constraints (bottom) to rescore the windows and choose the best object and part detections (top-right). The final step is to extract features on the localized semantic parts for fine-grained recognition for a pose-normalized representation and then train a classifier for the final categorization.

CNN [41]. In this work we explore a unified method that uses the same deep convolutional representation for detection as well as part description.

We conjecture that progress made on bottom-up region proposal methods, like selective search [105], could benefit localization of smaller parts in addition to whole objects. As we show later, average recall of parts using selective search proposals is 95% on the Caltech-UCSD bird dataset.

In this chapter, we propose a part localization model which overcomes the limitations of previous fine-grained recognition systems by leveraging deep convolutional features computed on bottom-up region proposals. Our method learns part appearance models and enforces geometric constraints between parts. An overview of our method is shown in Figure 4.1. We have investigated different geometric constraints, including a non-parametric model of joint part locations conditioned on nearest neighbors in semantic appearance space. We present state-of-the-art results evaluating our approach on the widely used fine-grained benchmark Caltech-UCSD bird dataset [108].

4.1 Background

4.1.1 Fine-grained categorization

Recently, a large body of computer vision research has focused on the fine-grained classification problem in a number of domains, such as animal breeds or species [32, 53, 70, 74, 82, 114], plant species [1, 2, 57, 77, 78, 92], and man-made objects [73, 97].

Several approaches are based on detecting and extracting features from certain parts of objects. Farrell et al. [32] proposed a pose-normalized representation using poselets [12]. Deformable part models [33] were used in [82, 119] for part localization. Based on the work of localizing fiducial landmarks on faces [6], Liu et al. [70] proposed an exemplar-based geometric method to detect dog faces and extract highly localized image features from keypoints to differentiate dog breeds. Furthermore, Berg et al. [9] learned a set of highly discriminative intermediate features by learning a descriptor for each pair of keypoints. Moreover, in [69], the authors extend the non-parametric exemplar-based method of [6] by enforcing pose and subcategory consistency. Yao et al. [113] and Yang et al. [112] have investigated template matching methods to reduce the cost of sliding window approaches. Recent work by Göring et al. [42] transfers part annotations from objects with similar global shape as non-parametric part detections. All these part-based methods, however, require the ground truth bounding box at test time for part localization or keypoint prediction.

Human-in-the-loop methods [15, 24, 27] ask a human to name attributes of the object, click on certain parts or mark the most discriminative regions to improve classification accuracy. Segmentation-based approaches are also very effective for fine-grained recognition. Approaches such as [20, 40, 83, 82, 111] used region-level cues to infer the foreground segmentation mask and to discard the noisy visual information in the background. Chai et al. [19] showed that jointly learning part localization and foreground segmentation together can be beneficial for fine-grained categorization. Similar to most previous part-based approaches, these efforts require the ground truth bounding box to initialize the segmentation seed. In contrast, the aim of our work is to perform end-to-end fine-grained categorization with no knowledge at test time of the ground truth bounding box.

Our part detectors use convolutional features on bottom-up region proposals, together with learned non-parametric geometric constraints to more accurately localize object parts, thus enabling strong fine-grained categorization.

Recent work by [14] analyze the space of staged detection, alignment, and classification pipelines to reveal the role of feature learning in fine-grained recognition. The choice of learned features makes the most difference but pose has its own contribution. [67] formulate a joint localization, alignment, and classification model for fine-grained classification. [54] achieve fine-grained recognition without part annotations.

4.1.2 Part-based models for detection and pose localization

Previous work has proposed explicit modeling of object part appearances and locations for more accurate recognition and localization. Starting with pictorial structures [34, 35], and continuing through poselets [12] and related work, many methods have jointly localized a set of geometrically related parts. The deformable parts model (DPM) [33], until recently the state-of-the-art PASCAL object detection method, models parts with additional learned filters in positions anchored with respect to the whole object bounding box, allowing parts to be displaced from this anchor with learned deformation costs. The "strong" DPM [4] adapted this method for the strongly supervised setting in which part locations are annotated at training time. A limitation of these methods is their use of weak features (usually HOG [23]).

Most recently, generic object detection methods have begun to leverage deep CNNs and outperformed any competing approaches based on traditional features. OverFeat [90] uses a CNN to regress to object locations in a coarse sliding-window detection framework. Of particular inspiration to our work is the R-CNN method [41] which leverages features from a deep CNN in a region proposal framework to achieve unprecedented object detection results on the PASCAL VOC dataset. Our method generalizes R-CNN by applying it to model object parts in addition to whole objects, which our empirical results will demonstrate is essential for accurate fine-grained recognition.

4.2 Method

While [41] demonstrated the effectiveness of the R-CNN method on a generic object detection task (PASCAL VOC), it did not explore the application of this method to simultaneous localization and fine-grained recognition. Because our work operates in this regime, we extend R-CNN to detect objects and localize their parts under a geometric prior. With hypotheses for the locations of individual semantic parts of the object of interest (e.g., the location of the head for an animal class), it becomes reasonable to model subtle appearance differences which tend to appear in locations that are roughly fixed with respect to these parts.

In the R-CNN method, for a particular object category, a candidate detection x with CNN feature descriptor $\phi(x)$ is assigned a score of $w_0^\intercal \phi(x)$, where w_0 is the learned vector of SVM weights for the object category. In our method, we assume a strongly supervised setting (e.g., [4]) in which at training time we have ground truth bounding box annotations not only for full objects, but for a fixed set of semantic parts $\{p_1, p_2, ..., p_n\}$ as well.

Given these part annotations, at training time all objects and each of their parts are initially treated as independent object categories: we train a one-versus-all linear SVM on feature descriptors extracted over region proposals, where regions with ≥ 0.7 overlap with a ground truth object or part bounding box are labeled as positives for that object or part, and regions with ≤ 0.3 overlap with any ground truth region are labeled as negatives. Hence for a single object category we learn whole-object ("root") SVM weights w_0 and part

SVM weights $\{w_1, w_2, ..., w_n\}$ for parts $\{p_1, p_2, ..., p_n\}$ respectively. At test time, for each region proposal window we compute scores from all root and part SVMs. Of course, these scores do not incorporate any knowledge of how objects and their parts are constrained geometrically; for example, without any additional constraints the *bird head* detector may fire outside of a region where the *bird* detector fires. Hence our final joint object and part hypotheses are computed using the geometric scoring function detailed in the following section, which enforces the intuitively desirable property that pose predictions are consistent with the statistics of poses observed at training time.

4.2.1 Geometric constraints

Let $X = \{x_0, x_1, \ldots, x_n\}$ denote the locations (bounding boxes) of object p_0 and n parts $\{p_i\}_{i=1}^n$, which are annotated in the training data, but unknown at test time. Our goal is to infer both the object location and part locations in a previously unseen test image. Given the R-CNN weights $\{w_0, w_1, \ldots, w_n\}$ for object and parts, we will have the corresponding detectors $\{d_0, d_1, \ldots, d_n\}$ where each detector score is $d_i(x) = \sigma(w_i^\intercal \phi(x))$, where $\sigma(\cdot)$ is the sigmoid function and $\phi(x)$ is the CNN feature descriptor extracted at location x. We infer the joint configuration of the object and parts by solving the following optimization problem:

$$X^* = \arg\max_X \Delta(X) \prod_{i=0}^{n} d_i(x_i) \tag{4.1}$$

where $\Delta(X)$ defines a scoring function over the joint configuration of the object and root bounding box. We consider and report quantitative results on several configuration scoring functions Δ, detailed in the following paragraphs.

Box constraints. One intuitive idea to localize both the object and parts is to consider each possible object window and all the windows inside the object and pick the windows with the highest part scores. In this case, we define the scoring function

$$\Delta_{\text{box}}(X) = \prod_{i=1}^{n} c_{x_0}(x_i) \tag{4.2}$$

where

$$c_x(y) = \begin{cases} 1 & \text{if region } y \text{ falls outside region } x \text{ by at most } \epsilon \text{ pixels} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

In our experiments, we let $\epsilon = 10$.

Geometric constraints. Because the individual part detectors are less than perfect, the window with highest individual part detector scores is not always correct, especially when there are occlusions. We therefore consider several scoring functions to enforce constraints

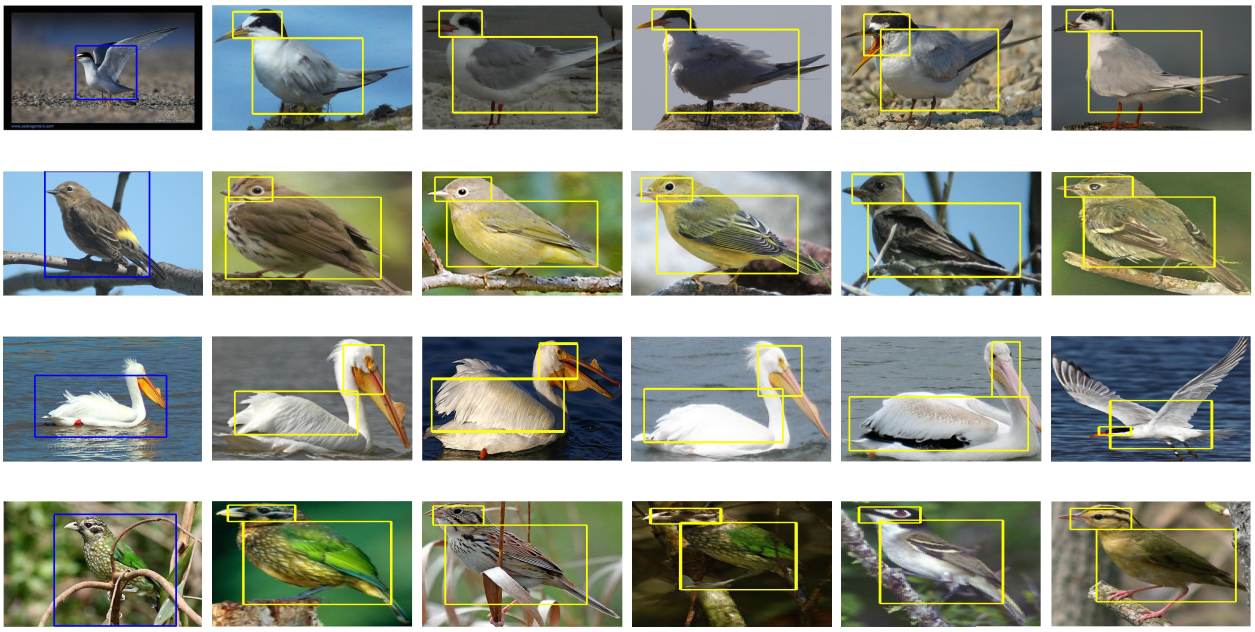


Figure 4.2: Illustration of geometric constant δ^{NP}. In each row, the first column is the test image with an R-CNN bounding box detection, and the rest are the top-five nearest neighbors in the training set, indexed using `pool5` features and cosine distance metric.

over the layout of the parts relative to the object location to filter out incorrect detections. We define

$$\Delta_{\text{geometric}}(X) = \Delta_{\text{box}}(X) \left(\prod_{i=1}^{n} \delta_i(x_i) \right)^{\alpha} \tag{4.4}$$

where δ_i is a scoring function for the position of the part p_i given the training data. Following previous work on part localization from, e.g. [6, 33, 35], we experiment with three definitions of δ:

- $\delta_i^{MG}(x_i)$ fits a mixture of Gaussians model with N_g components to the training data for part p_i. In our experiments, we set $N_g = 4$.
- $\delta_i^{NP}(x_i)$ finds the K nearest neighbors in appearance space to $\tilde{x}_0$, where $\tilde{x}_0 = \arg\max d_0(x_0)$ is the top-scoring window from the root detector. We then fit a Gaussian model to these K neighbors. In our experiments, we set $K = 20$. Figure 4.2 illustrates some examples of nearest neighbors.

The DPM [33] models deformation costs with a per-component Gaussian prior. R-CNN [41] is a single-component model, motivating the δ^{MG} or δ^{NP} definitions. Our δ^{NP} definition is inspired by Belhumeur et al. [6], but differs in that we index nearest neighbors on appearance rather than geometry.

4.2.2 Fine-grained categorization

We extract semantic features from localized parts as well as the whole object. The final feature representation is $[\phi(x_0) \dots \phi(x_n)]$ where x_0 and $x_{1\dots n}$ are whole-object and part location predictions inferred using one of the models from the previous section and $\phi(x_i)$ is the feature representation of part x_i.

In one set of experiments, we extract deep convolutional features $\phi(x_i)$ from an ImageNet pre-trained CNN, similar to DeCAF [26]. In order to make the deep CNN-derived features more discriminative for the target task of fine-grained bird classification, we also fine-tune the ImageNet pre-trained CNN for the 200-way bird classification task from ground truth bounding box crops of the original CUB images. In particular, we replace the original 1000-way `fc8` classification layer with a new 200-way `fc8` layer with randomly initialized weights drawn from a Gaussian with $\mu = 0$ and $\sigma = 0.01$. We set fine-tuning learning rates as proposed by R-CNN [41], initializing the global rate to a tenth of the initial ImageNet learning rate and dropping it by a factor of 10 throughout training, but with a learning rate in the new `fc8` layer of 10 times the global learning rate. For the whole object bounding box and each of the part bounding boxes, we independently finetune the ImageNet pre-trained CNN for classification on ground truth crops of each region warped to the 227×227 network input size, always with 16 pixels on each edge of the input serving as context as in R-CNN [41]. At test time, we extract features for the predicted whole object or part region using the network fine-tuned for that particular whole object or part.

For training the classifier, we employ a one-versus-all linear SVM using the final feature representation. For a new test image, we apply the whole and part detectors with the geometric scoring function to get detected part locations and use the features for prediction. If a particular part i was not detected anywhere in the test image (due to all proposals falling below the part detector's threshold, set to achieve high recall), we set its features $\phi(x_i) = \mathbf{0}$ (zero vector).

4.3 Experiments

In this section, we present a comparative performance evaluation of our proposed method. Specifically, we conduct experiments on the widely-used fine-grained benchmark Caltech-UCSD birds dataset [108] (CUB200-2011). The classification task is to discriminate among 200 species of birds, and is challenging for computer vision systems due to the high degree of similarity between categories. It contains 11,788 images of 200 bird species. Each image is annotated with its bounding box and the image coordinates of fifteen keypoints: the beak, back, breast, belly, forehead, crown, left eye, left leg, left wing, right eye, right leg, right wing, tail, nape and throat. We train and test on the splits included with the dataset, which contain around 30 training samples for each species. Following the protocol of [119], we use two semantic parts for the bird dataset: head and body.

We use the open-source package Caffe [52] to extract deep features and fine-tune our

CNNs. For object and part detections, we use the Caffe reference model, which is almost identical to the model used by Krizhevsky et al. in [55]. We refer deep features from each layer as `conv`n, `pool`n, or `fc`n for the nth layer of the CNN, which is the output of a convolutional, pooling, or fully connected layer respectively. We use `fc6` to train R-CNN object and part detectors as well as image representation for classification. For δ^{NP}, nearest neighbors are computed using `pool5` and cosine distance metric.

4.3.1 Fine-grained categorization

We first present results on the standard fine-grained categorization task associated with the Caltech-UCSD birds dataset. The first set of results in Table 4.1 are achieved in the setting where the ground truth bounding box for the entire bird is known at test time, as most state-of-art methods assume, making the categorization task somewhat easier. In this setting, our part-based method with the local non-parametric geometric constraint δ^{NP} works the best without fine-tuning, achieving 68.1% classification accuracy without fine-tuning. Fine-tuning improves this result by a large margin, to over 76%. We compare our results against three state-of-the-art baseline approaches with results assuming the ground truth bounding box at test time. We use deep convolutional features as the authors of [26], but they use a HOG-based DPM as their part localization method. The increase in performance is likely due to better part localization (see Table 4.4). Oracle method uses the ground truth bounding box and part annotations for both training and test time.

The second set of results is in the less artificial setting where the bird bounding box is *unknown* at test time. Most of the literature on this dataset doesn't report performance in this more difficult, but more realistic setting. As Table 4.1 shows, in this setting our part-based method works much better than the baseline DPD model. We achieve 66.0% classification accuracy without finetuning , almost as good as the accuracy we can achieve when the ground truth bounding box is given. This means there is no need to annotate any box during test time to classify the bird species. With finetuned CNN models, our method achieves 73.89% classification accuracy. We are unaware of any other published results in this more difficult setting, but we note that our method outperforms previous state-of-the-art even without knowledge of the ground truth bounding box.

Another interesting experiment we did is to remove the part descriptors by only looking at the image descriptors inside the predicted bounding box. By having geometric constraints over part locations relative to object location, our method is able to help localize the object. As Table 4.2 shows, our method outperforms a single object detector using R-CNN, which means the geometric constraints helps our method better localize the object window. The detection of strong DPM is not as accurate as our method, which explains the performance drop. The "oracle" method uses the ground truth bounding box and achieves 57.94% accuracy, which is still much lower than the method in Table 4.1 of using both image descriptors inside object and parts.

Table 4.1: Fine-grained categorization results on CUB200-2011 bird dataset. -ft means extracting deep features from finetuned CNN models using each semantic part. Oracle method uses the ground truth bounding box and part annotations for both training and test time.

Bounding Box Given	
DPD [119]	50.98%
DPD+DeCAF feature [26]	64.96%
POOF [9]	56.78%
Symbiotic Segmentation [19]	59.40%
Alignment [40]	62.70%
Oracle	72.83%
Oracle-ft	82.02%
Ours (Δ_{box})	67.55%
Ours ($\Delta_{\text{geometric}}$ with δ^{MG})	67.98%
Ours ($\Delta_{\text{geometric}}$ with δ^{NP})	68.07%
Ours-ft (Δ_{box})	75.34%
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{MG})	**76.37%**
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{NP})	76.34%
Bounding Box Unknown	
DPD+DeCAF [26] with no bounding box	44.94%
Ours (Δ_{null})	64.57%
Ours (Δ_{box})	65.22%
Ours ($\Delta_{\text{geometric}}$ with δ^{MG})	65.98%
Ours ($\Delta_{\text{geometric}}$ with δ^{NP})	65.96%
Ours-ft (Δ_{box})	72.73%
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{MG})	72.95%
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{NP})	**73.89%**

4.3.2 Part localization

We now present results evaluating in isolation the ability of our system to accurately localize parts. Our results in Table 4.4 are given in terms of the Percentage of Correctly Localized Parts (PCP) metric. For the first set of results, the whole object bounding box is given and the task is simply to correctly localize the parts inside of this bounding box, with parts having ≥ 0.5 overlap with ground truth counted as correct.

For the second set of results, the PCP metric is computed on top-ranked parts predictions using the objective function described in Sec. 3.2. Note that in this more realistic setting we do not assume knowledge of the ground truth bounding box at test time – despite this limitation, our system produces accurate part localizations.

As shown in Table 4.4, for both settings of given bounding box and unknown bounding box, our methods outperform the strong DPM [4] method. Adding a geometric constraint

Table 4.2: Fine-grained categorization results on CUB200-2011 bird dataset with *no parts*. We trained a linear SVM using deep features on all the methods. Therefore only the bounding box prediction is the factor of difference. -ft is the result of extracting deep features from fine-tuned CNN model on bounding box patches.

Oracle (ground truth bounding box)	57.94%
Oracle-ft	68.29%
Strong DPM [4]	38.02%
R-CNN [41]	51.05%
Ours (Δ_{box})	50.17%
Ours ($\Delta_{\text{geometric}}$ with δ^{MG})	51.83%
Ours ($\Delta_{\text{geometric}}$ with δ^{NP})	52.38%
Ours-ft (Δ_{box})	62.13%
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{MG})	62.06%
Ours-ft ($\Delta_{\text{geometric}}$ with δ^{NP})	**62.75%**

Table 4.3: Recall of region proposals produced by selective search methods on CUB200-2011 bird dataset. We use ground truth part annotations to compute the recall, as defined by the proportion of ground truth boxes for which there exists a region proposal with overlap at least 0.5, 0.6 and 0.7 respectively.

Overlap	0.50	0.60	0.70
Bounding box	96.70%	97.68%	89.50%
Head	93.34%	73.87%	37.57%
Body	96.70%	85.97%	54.68%

δ^{NP} improves our results (79.82% for body localization compared to 65.42%). In the fully automatic setting, the top ranked detection and part localization performance on head is 65% better than the baseline method. $\Delta_{\text{null}} = 1$ is the appearance-only case with no geometric constraints applied. Although the fine-grained classification results don't show a big gap between $\Delta_{\text{geometric}}$ and Δ_{box}, we can see the performance gap for part localization. The reason for the small performance gap might be that deep convolutional features are invariant to small translations and rotations, limiting the impact of small localization errors on our end-to-end accuracy.

We also evaluate the recall performance of selective search region proposals [105] for bounding box and semantic parts. The results of recall given different overlapping thresholds are shown in Table 4.3. Recall for the bird head and body parts is high when the overlap requirement is 0.5, which provides the foundation for localizing these parts given the region proposals. However, we also observe that the recall for head is below 40% when the overlap threshold is 0.7, indicating the bottom-up region proposals could be a bottleneck for precise part localization.

Other visualizations are shown in Figure 4.4. We show three detection and part localiza-

Table 4.4: Part localization accuracy in terms of PCP (Percentage of Correctly Localized Parts) on the CUB200-2011 bird dataset. There are two different settings: with given bounding box and without bounding box.

Bounding Box Given		
	Head	Body
Strong DPM [4]	43.49%	75.15%
Ours (Δ_{box})	61.40%	65.42%
Ours ($\Delta_{\text{geometric}}$ with δ^{MG})	66.03%	76.62%
Ours ($\Delta_{\text{geometric}}$ with δ^{NP})	**68.19%**	**79.82%**
Bounding Box Unknown		
	Head	Body
Strong DPM [4]	37.44%	47.08%
Ours (Δ_{null})	60.50%	64.43%
Ours (Δ_{box})	60.56%	65.31%
Ours ($\Delta_{\text{geometric}}$ with δ^{MG})	**61.94%**	70.16%
Ours ($\Delta_{\text{geometric}}$ with δ^{NP})	61.42%	**70.68%**

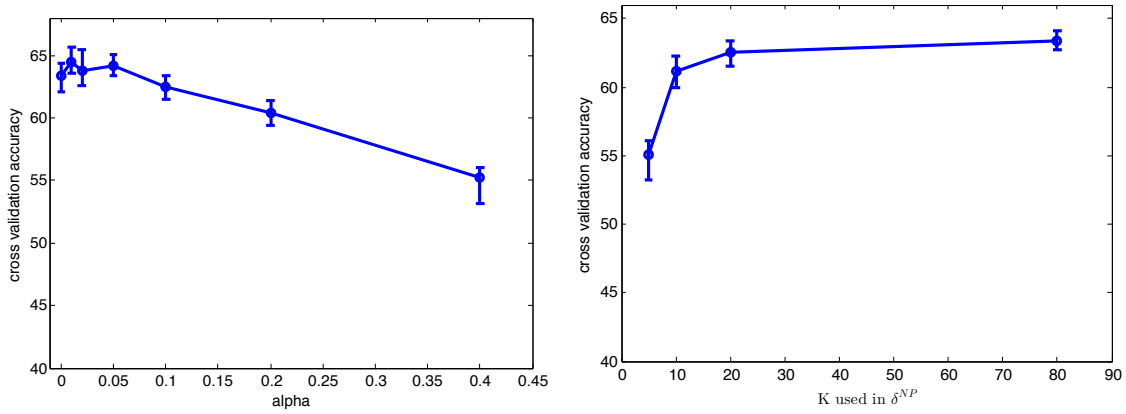


Figure 4.3: Cross-validation results on fine-grained accuracy for different values of α (left) and K (right). We split the training data into 5 folds and use cross-validate each hyperparameter setting.

tion for each image, the first column is the output from strong DPM, the second column is our methods with individual part predictions and the last column is our method with local prior. We used the model pretrained from [4] to get the results. We also show some failure cases of our method in Figure 4.5.

4.3.3 Component Analysis

To examine the effect of different values of α and K used in $\Delta_{\text{geometric}}$, we conduct cross-validation experiments. Results are shown in Figure 4.3. We fix $K = 20$ in Figure 4.3, left and fix $\alpha = 0.1$ in Figure 4.3, right. All the experiments on conducted on training data in a cross-validation fashion and we split the training data into 5 folds. As the results show, the end-to-end fine-grained classification results are sensitive to the choice of α and $\alpha = 0$ is the case of Δ_{box} predictions without any geometric constraints. The reason why we have to pick a small α is the pdf of the Gaussian is large compared to the logistic score function output from our part detectors. On the other hand, the choice of K cannot be too small and it is not very sensitive when K is larger than 10.

4.4 Bilinear CNN model and compact bilinear pooling

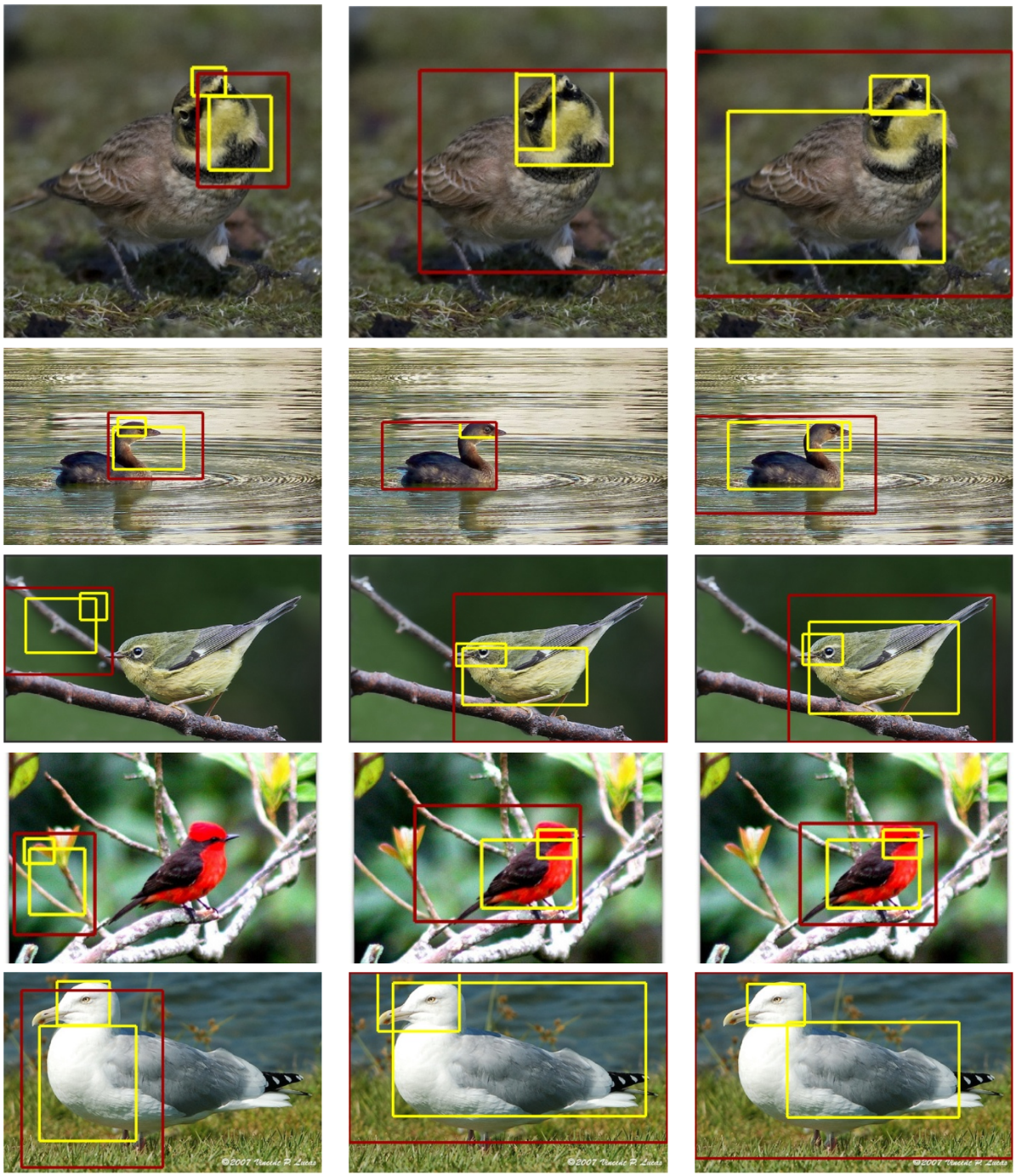
Bilinear models were first introduced in Tenenbaum and Freeman [101] to separate style and content. Second order pooling have since been considered for semantic segmentation and fine grained recognition respectively, both using hand-tuned features [18], and CNN features [68]. The recent work of Lin et al. [68] used bilinear pooling to encode the activations of a deep convolutional network for fine-grained visual recognition and achieved state-of-art results on CUB200-2011 dataset. The details of bilinear CNN model can be found in [68].

4.4.1 Compact Bilinear Pooling

Inspired by the bilinear CNN models, we want to deploy bilinear pooling features into our part-based RCNN model as the feature representations for each part. However, due to the high dimensionality of the bilinear pooling features (more than $250,000$ dimensions in [68]), it is impractical to replace deep feature with the bilinear pooling features in our model. Gao *et al.* propose compact bilinear representations in [38]. The goal is to project the large bilinear pooling features to a lower dimensional space ($\sim$ 5k dimension) with the same discriminative power. It also allows back-propagation of classification errors, enabling an end-to-end optimized model. The method relies on the low dimensional feature maps for kernel functions, specifically the Random Maclaurin Projection [86] and Tensor Sketch Projection [84].

4.4.2 Results using Compact Bilinear Pooling

We now show the results of part-based RCNN model using compact bilinear pooling features. We use the same part detections as in Section 4.2. However, for the feature representations, we use compact bilinear models to finetune the CNN models for each part and use the last



Strong DPM Ours (Δ_{box}) Ours (δ^{NP})

Figure 4.4: Examples of bird detection and part localization from strong DPM [4] (left); our method using Δ_{box} part predictions (middle); and our method using δ^{NP}(right). All detection and localization results without any assumption of bounding box.

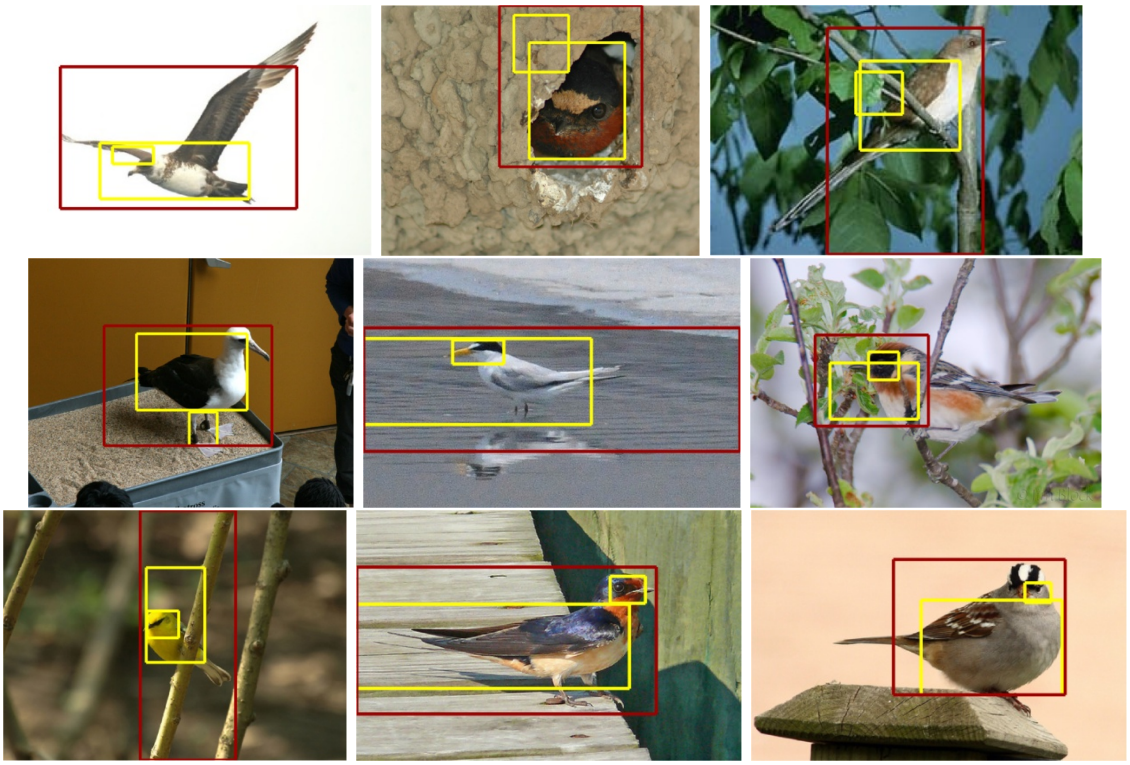


Figure 4.5: Failure cases of our part localization using δ^{NP}.

layer features to replace the original deep features. The classification results on CUB200-2011 dataset are shown in Table 4.5. Ours-fc6 is the result using the setting of bounding box unknown and parts predicted by non-parametric geometric prior, the $\Delta_{\text{geometric}}$ with δ^{NP} entry in Table 4.1. By using the same predicted parts but the compact bilinear representations, we improve the classification accuracy to 85.31% from 73.89%. We also show the oracle result by using the groundtruth part detections at test time. The gap between 85.31% to the oracle 86.34% comes from the errors made in part detections. We also show the results of using bilinear CNN model without any parts. It demonstrates that even with the power bilinear CNN features, it is still beneficial to model the object's pose by means of parts for this challenge task.

4.5 Dense window sampling for keypoint localization

During the experiments in Section 4.3, we observe that the recall of using selective search on small parts is very low, which limits us using RCNN to localize small parts. In this section, we propose a dense window sampling method for keypoint localization. We use the pretrained AlexNet [55] on ImageNet ILSVRC 2012 and report keypoint localization results

Method	Classification accuracy
Ours-fc6	73.89%
Bilinear CNN Model [68]	84.1%
Ours-compact bilinear	**85.31%**
Oracle-compact bilinear	86.33%

Table 4.5: Fine-grained categorization results using our model and compact bilinear representations.

layer	rf size	stride
`conv1`	11×11	4×4
`conv2`	51×51	8×8
`conv3`	99×99	16×16
`conv4`	131×131	16×16
`conv5`	163×163	16×16
`pool5`	195×195	32×32

Table 4.6: Convnet receptive field sizes and strides of AlexNet [55], for an input of size 227×227.

on PASCAL VOC 2011 dataset and assume groundtruth bounding box at test time.

Inspired by our part-based RCNN method and [90, 50], we propose dense sample sliding window part detectors to predict keypoint locations independently. To do so, we first compute all features at a particular layer (`conv5` in our experiment), resulting in an 2d grid of feature vectors. We associate each feature vector with a patch in the original image at the center of the corresponding receptive field and with size equal to the receptive field stride. (Note that the strides of the receptive fields are much smaller than the receptive fields themselves, which overlap. Refer to Table 4.6 above for specific numbers.)

We rescale each bounding box to 500×500 and compute `conv5` (with a stride of 16 pixels). Each cell of `conv5` contains one 256-dimensional descriptor. We concatenate `conv5` descriptors from a local region of 3×3 cells, giving an overall receptive field size of 195×195 and feature dimension of 2304. For each keypoint, we train a linear SVM with hard negative mining. We consider the ten closest features to each ground truth keypoint as positive examples, and all the features whose rfs do not contain the keypoint as negative examples. We also train using dense SIFT descriptors for comparison. We compute SIFT on a grid of stride eight and bin size of eight using VLFeat [106]. For SIFT, we consider features within twice the bin size from the ground truth keypoint to be positives, while samples that are at least four times the bin size away are negatives.

We augment our SVM detectors with a diagonal Gaussian prior over candidate locations constructed by nearest neighbor matching. The mean of each Gaussian is taken to be the location of the keypoint in the nearest neighbor in the training set found using cosine similarity on `pool5` features, and we use a fixed standard deviation of 22 pixels. Let $s(X_i)$

	aero	bike	bird	boat	bttl	bus	car	cat	chair	cow
SIFT	12.3	13.6	12.8	7.8	20.6	19.1	16.8	12.9	13.6	6.0
SIFT+prior	21.7	29.2	18.5	12.0	33.6	36.4	29.6	23.7	17.4	17.8
conv5	31.3	31.9	25.0	14.2	42.6	45.6	23.6	38.4	9.5	27.0
conv5+prior	**37.8**	**40.1**	**29.6**	**17.7**	**51.0**	**55.0**	**34.9**	**39.3**	**19.2**	**33.5**
table	dog	horse	mbike	prsn	plant	sheep	sofa	train	tv	mean
7.1	12.3	15.9	11.9	11.9	17.2	13.6	13.4	6.9	27.1	13.6
11.6	19.8	26.2	25.0	20.9	23.3	21.3	21.2	17.4	47.6	23.7
22.5	28.8	33.4	22.7	30.2	32.5	27.3	22.3	28.4	54.1	29.5
25.5	**31.3**	**40.5**	**34.3**	**35.7**	**34.9**	**32.5**	**32.7**	**37.6**	**59.5**	**36.2**

Table 4.7: Keypoint prediction results on PASCAL VOC 2011. The numbers give average accuracy of keypoint prediction using $\alpha = 0.1$.

be the output score of our local detector for keypoint X_i, and let $p(X_i)$ be the prior score. We combine these to yield a final score $f(X_i) = s(X_i)^{1-\eta} p(X_i)^{\eta}$, where $\eta \in [0, 1]$ is a tradeoff parameter. In our experiments, we set $\eta = 0.1$ by cross validation. At test time, we predict the keypoint location as the highest scoring candidate over all feature locations.

Recent work by Jain et al. [50] proposes a convolutional network architecture for human pose estimation. They train multiple small convnets on 64×64 patches from scratch for independent binary body-part detection and apply a chain model on top to enforce pose consistency while we use the pretrained ImageNet reference model and share mid-level feature representations among all parts. We also use a simpler nonparametric prior to filter out outliers.

We consider a ground truth keypoint to be correctly predicted if the prediction lies within a Euclidean distance of α times the bounding box width. We compute the overall accuracy across keypoints on each instance, and average over all instances for a final score. We do not penalize predicted keypoints that are invisible in the target image. In our evaluation, we take $\alpha = 0.1$. The results of 20 PASCAL classes using conv5 and SIFT with and without the prior are shown in Table 4.7. From the table, we can see that local part detectors trained on the conv5 feature outperform SIFT by a large margin and that the prior information is helpful in both cases. To our knowledge, these are the first keypoint prediction results reported on this dataset. We show example results from five different categories in Figure 4.6. Each set consists of rescaled bounding box images with ground truth keypoint annotations and predicted keypoints using SIFT and conv5 features, where each color corresponds to one keypoint. As the figure shows, conv5 outperforms SIFT, often managing satisfactory outputs despite the challenge of this task. A small offset can be noticed for some keypoints like eyes and noses. We believe this is caused by the limited stride of our scanning windows. A final regression or finer stride could mitigate this issue.

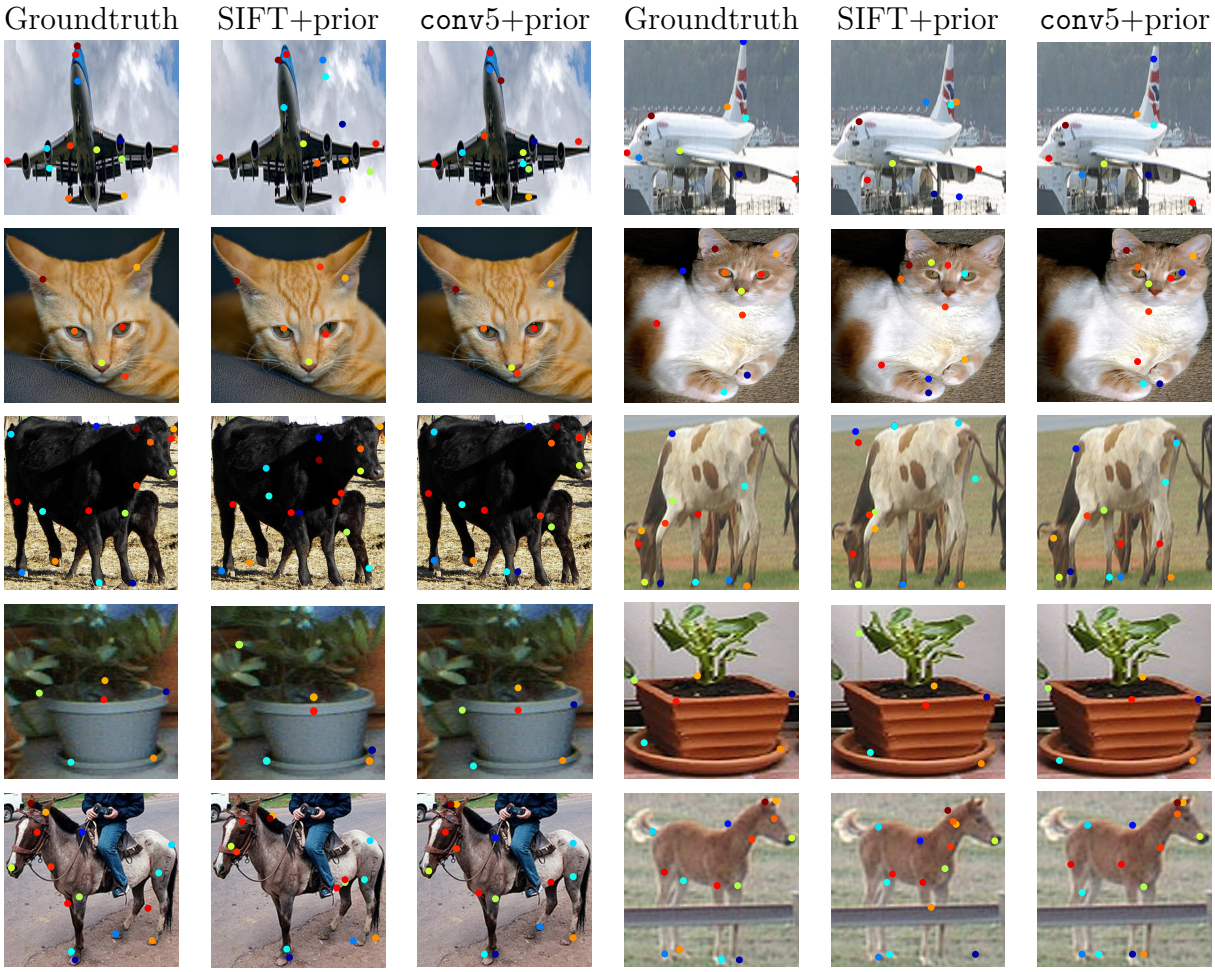


Figure 4.6: Examples of keypoint prediction on five classes of the PASCAL dataset: aeroplane, cat, cow, potted plant, and horse. Each keypoint is associated with one color. The first column is the ground truth annotation, the second column is the prediction result of SIFT+prior and the third column is conv5+prior. (Best viewed in color).

4.6 Summary

In this chapter, we have proposed a system for joint object detection and part localization capable of state-of-the-art fine-grained object recognition. Our method learns detectors and part models and enforces learned geometric constraints between parts and with the object frame. We also show the results of using the recent state-of-art compact bilinear representations. It shows even with the power feature representations, our idea of using pose-normalization still can improve upon the model without any knowledge of pose. Furthermore, we show how to relax the use of selective search for smaller parts and employing dense window sampling for keypoint localization. We are first to report keypoint localization results on

PASCAL dataset. We still see the accuracy gap between predicted parts and groundtruth parts. As an extension of this work, we will show a fully convolutional part modelthat learns more accurate part locations and object category in the next chapter.

Chapter 5

Fully Convolution Part Model

Part-based RCNN model shown in the last chapter relies on bottom-up region proposals from hand-engineered features for part detection and the localization model is not an end-to-end trainable network. We also show that identifying parts to find correspondence discounts pose variation can improve the fine-grained classification performance even with powerful deep features. In this chapter, we propose an end-to-end trainable network supervised by keypoint locations and class labels that localizes parts by a fully convolutional network to focus on the learning of feature representations for the fine-grained classification task.

Pose variation and subtle differences in appearance are key challenges to fine-grained classification. While deep networks have markedly improved general recognition, many approaches to fine-grained recognition rely on anchoring networks to parts for better accuracy. Identifying parts to find correspondence discounts pose variation so that features can be tuned to appearance. To identify these correspondences, previous work [116, 14] has focused on using strong supervisions for part localizations and feature transformations that can discount the nuisance variables of pose and viewpoint.While recent work [49, 68, 54] has made impressive progress without strong supervision, we show that the correspondence gained from end-to-end keypoint and classification training improves fine-grained recognition accuracy.

Deep learning has made dramatic progress on not only image classification, but detection [90, 41], keypoint prediction [102], and semantic segmentation [71]. These tasks require not only recognition but localization. Whether or not a deep network can effectively learn all necessary invariances to pose is an open question. We believe it is still critical to model parts and pool features into a pose-independent representation to best distinguish between similar subordinate classes. This makes good use of limited training data by reserving model capacity for appearance given the discounting of pose.

In this chapter, we propose an end-to-end trainable deep network to simultaneously localize parts using a spatially fine-grained detection model, form descriptors over inferred part locations, and classify a categorically fine-grained label space. The network goes directly from pixels to parts via fully convolutional keypoint localization network. We design a semantic pooling layer—which we call the coordinate transfer layer—to pool feature maps

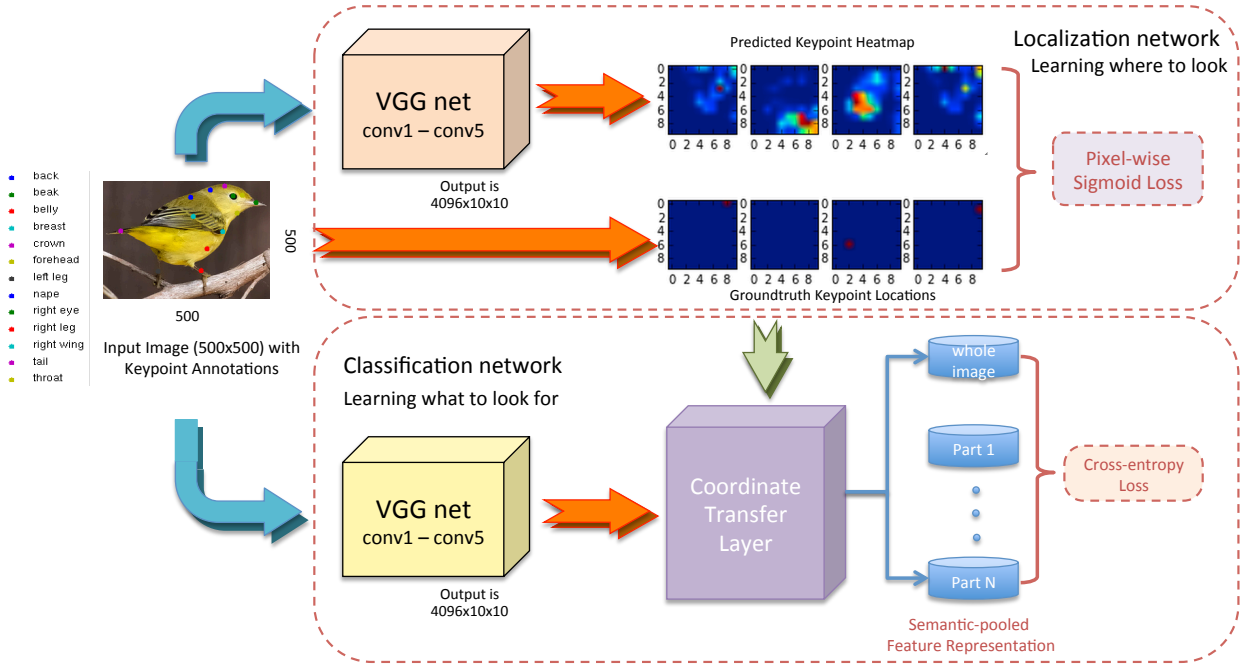


Figure 5.1: **Overview of our model.** The network consists of two main modules: 1) a localization network for learning where to look and 2) a classification network which uses a coordinate transfer layer to semantically pool part features; these are jointly used to learn the fine-grained classifier.

using predicted keypoints. The fine-grained classification is then decided from the combined representations. Classification errors are back-propagated to the selected parts to drive the learning of discriminative, pose-normalized features. To our knowledge, ours is the first approach to incorporate pose prediction and normalization in a unified deep fine-grained model. Our end-to-end model achieves state-of-the-art results on the standard CUB200-2011 bird benchmark [108] for fine-grained recognition.

5.1 Background

The literature on fine-grained classification methods and on recent successes of deep neural networks have been covered in the previous chapters. While most of the approaches use strong supervisions, weakening supervision is an exciting direction for fine-grained recognition. The spatial transformer network of [49] strives for spatial invariance by end-to-end learning transformations and is shown to be effective for fine-grained categorization.

Our method is also related to attention models. Olshausen et al. proposed a neural model of visual attention which is based on dynamic gating of information flow at stage of representation within a deep network [79]. Recent attention models [75, 5] use sequential

deep learning models, such as recurrent neural network or LSTM to learn glimpse location based on last glimpse. In [89], the authors use attention models to learn where to look for as glimpses for fine-grained classification. Compared with attention models, our method is not restricted to sequential and can directly exploit available strong supervision. While appealing in generality, these attention-based approaches do not exceed performance of recent direct vision approaches.

5.1.1 Fully convolutional networks

The fully convolutional networks (FCNs) of Long et al. [71] are designed for pixelwise prediction. Every layer in an FCN computes a local operation on relative spatial coordinates. There are no fully connected layers to restrict dimensions. In this way, an FCN can take an input of any size and produce an output of corresponding dimensions.

Inference and learning are done on whole images at a time. The dense forward pass and backpropagation are highly efficient since intermediate features and gradients are effectively cached. This makes it possible to take advantage of the complete ground truth in training pixel-to-pixel tasks such as semantic segmentation or in our case keypoint prediction.

Tompson et al. [102] learn an end-to-end fully convolutional network for human pose. However, this model predicts keypoints alone and does not explore their role in recognition. In this work, the keypoints guide the fine-grained classification model by pooling deep features into a pose-normalized form that is learned end-to-end.

Shubham et al. [103] combine convolutional score maps with inferred viewpoint priors to regress to keypoint locations. Their viewpoint and keypoint prediction depend on detections from R-CNN. Joint learning is prevented by the independent proposal and bounding box recognition stages. The fully convolutional keypoints to fine-grained classification network proposed in this work does not rely on separate detections or bounding box knowledge.

5.2 Fully convolutional part model for fine-grained categorization

The overview architecture of our model is shown in Figure 5.1. The goal is to define a unified deep network to simultaneously learn to localize parts and semantically pool the feature representations accordingly. The network consists of two main modules: 1) localization net learning where to look and 2) classification net which has semantic pooling to learn the fine-grained classifier.

We initialize our network from the VGG-16 network [94] as pretraining. With its further parameters and layers of nonlinearity, the VGG net was a winner of the ILSVRC 2014 challenge. It has since proven useful through transfer learning to other vision tasks [46, 71] like semantic segmentation that require both recognition and localization. We use the publicly available pretained VGG-16 weights and implement our model with the Caffe [52] framework.

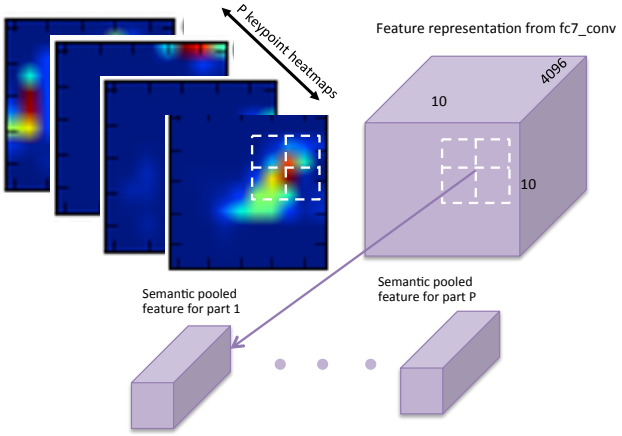


Figure 5.2: The coordinate transfer layer takes two inputs: the keypoint heatmap and the feature representation activations. For each part, the layer pools the feature based on the small surrounding neighborhood around the argmax point of the heatmap, as shown in white rectangle. For P different parts, the layer pools P semantic features and stacks them together as the output.

5.2.1 Keypoint prediction

First, we cast the VGG network into fully convolutional form following [71]. This equips the network for dense feature extraction on any input image size. Although the original input size of VGG net is 224×224 pixels, the last layer receptive field size is actually 404×404. This is the size of the spatial context for each keypoint prediction. For finer part resolution, we upsample the input image (which we note can be composed with the filters of the first layer) to 500×500 for an ultimate feature map of dimensions $10 \times 10 \times 4096$. We score keypoints by a 1×1 convolution layer.

The keypoint location scoring is learned through end-to-end training from a pixelwise sigmoid cross-entropy loss function. The ground truth keypoint locations are coded as a binary map over output locations. The output centered at closest point to the true keypoint is coded as a positive while all other outputs are negatives. A keypoint may not be visible in all inputs; these missing keypoints are coded as negatives at every location.

The pixelwise sigmoid cross entropy loss function is

$$l = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} [p_{ij}^n log \hat{p}_{ij}^n + (1 - p_{ij}^n) log(1 - \hat{p}_{ij}^n)]$$

where p_{ij}^n is the ground truth keypoint map of nth example at location (i, j) and $\hat{p}_{ij}^n$ is the sigmoid output at location (i, j) of nth example.

5.2.2 Coordinate transfer layer

It is not enough to have keypoint locations: rich, part-tuned features are needed to capture fine-grained differences. In order to direct feature learning to this purpose, we pool features semantically by keypoint identity. This is carried out by our novel coordinate transfer layer. It takes two inputs: the keypoint location map and the feature activations. In the forward pass, the layer pools the features maps in a focused, surrounding neighborhood of the part, as shown by the white rectangle in Figure 5.2 and then stack the pooled features together as the pose-normalized part representation. During back-propagation, the layer propagates the classification error to the corresponding part region in the lower feature representation. During training the ground truth part locations are pooled whereas the predicted keypoint locations are pooled at test time.

5.2.3 Joint representation

For fine-grained class recognition, we concatenate the pose-normalized part feature and the holistic representation of the whole image together and add a fully connected layer on top of the fused representation. The classification training is done by the softmax loss. At testing neither the bounding box or part annotations are given; everything is inferred from the image. For the holistic representation, we also optionally employ an additional layer implementing the recent bilinear feature space of [68] and its compact extension [38].

5.2.4 Training

The whole network can be trained end-to-end for joint learning of all model parameters. The architecture is a directed acyclic graph and the coordinate transfer layer gradient back-propagates fine-grained recognition error while respecting localization.

Given the size of the network, we adopt a staged approach to learning before end-to-end fine-tuning. This ensures reasonable accuracy of the keypoint and classification tasks before complete feature learning to avoid divergence from the pre-training.

Our training proceeds as follows:

1. Keypoint localization is fine-tuned from the VGG net for fixed features and random initialization of the last score layer.

2. Once converged, we fine-tune the localization network at a smaller learning rate.

3. Once localization network has converged, the classifiers are trained for fixed semantic pooling on ground truth part annotations.

4. Once converged, we fine-tune all layers at a smaller learning rate.

5.2.5 Compact bilinear representation

[68] have shown impressive results on fine-grained classification by using bilinear pooling to encode the deep representation. However, those features are high dimensional ($\sim$ 262k) and impractical for spatial tasks like keypoints or segmentation. [38] propose compact representations with same discriminative power which allow back-propagation and end-to-end training. We use the "Random Maclaurin Projection" of [38] to yield a deep representation that of only 5k dimensions.

5.2.6 Finetuned part nets

The coordinate transfer layer provides a way to use the feature representation around each keypoint and unify the whole pose-normalized representation into one network. However, one disadvantage is that it lacks context due to the smaller corresponding receptive windows. An alternative way of utilizing keypoints for pose-normalized representation is to crop part images as [116]. Following [116], we crop two parts: head and body using the predicted keypoints and finetuned part networks using the crop part images.

5.3 Experiments

In this section, we present a comparative performance evaluation of our proposed method. Specifically, we conduct experiments on the widely-used fine-grained benchmark Caltech-UCSD bird dataset [108] (CUB200-2011). The classification task is to discriminate among 200 species of birds, and is challenging for computer vision systems due to the high degree of similarity between categories. It contains 11,788 images of 200 bird species. Each image is annotated with its bounding box and the image coordinates of fifteen keypoints: the beak, back, breast, belly, forehead, crown, left eye, left leg, left wing, right eye, right leg, right wing, tail, nape and throat. We train and test on the splits included with the dataset, which contain around 30 training samples for each species.

For both the localization net and classification net, we fine-tuned from the 16-layer VGG network used in [94] pretrained on ImageNet data. Like [71], we change the last two fully connected layer to convolutional layer and make the net fully convolutional. All of our experiments are implemented in the open source Caffe framework [52] and we will make our architecture and weights publicly available.

5.3.1 Fine-grained classification

We first present classification results on the standard CUB200-2011 dataset. The classification accuracies of our method and other methods are reported in Table 5.1, along with the different annotations required during the training and test phases. Our method with compact bilinear pooling achieves 83.00% classification accuracy, and fine-tuning part networks

Method	Train phase		Test phase		Accuracy
	BBox	Parts	BBox	Parts	
DPD+DeCAF feature [26]	✓	✓	✓		64.96%
POOF [9]	✓	✓	✓		56.78%
Symbiotic Segmentation [19]	✓		✓		59.40%
Alignment [40]	✓		✓		62.70%
Part-based RCNN with BBox[116]	✓	✓	✓		76.37%
Part-based RCNN[116]	✓	✓			73.89%
Pose normalized CNNs [14]	✓	✓			75.70%
Co-segmentation [54]	✓				82.0%
Two-level attention [110]					69.7%
Bilinear model [68]					84.0%
Spatial transformer networks [49]					84.1%
fc7 feature of VGG on whole image					58.34%
fc7 feature of VGG-ft on whole image					71.73%
Keypoint features baseline		✓			65.10%
Our model with fc7 feature		✓			75.04%
Our model with compact bilinear feature		✓			83.00%
Our model with compact bilinear feature on ft-part		✓			**85.92%**

Table 5.1: Comparison with other methods on fine-grained Classification results.

improves accuracy to 85.92%. This is state-of-the-art and improves over other methods supervised by part annotations.

VGG net pretrained on Imagenet alone can achieve 58.34% classification accuracy and finetuning on our dataset can improve the performance to 71.73%. By adding pose-normalized representation, our method can achieve 75.04% accuracy. We also tested the keypoint feature baseline results, i.e. finetune the keypoint localization net using classification loss and the accuracy dropped to 65.10%. It means it is necessary to split out the keypoint localization network and the fine-grained classification network.

5.3.2 Part localization

We now present results of part localization using the localization net in our method. We use PCK (Percentage of Correctly Localized Keypoints) as our evaluation metric and the results are shown in Table 5.2. For each annotated instance, the keypoint is said to have been predicted correctly if the corresponding prediction lies within $\alpha \times max(h, w)$ of the annotated keypoint with the corresponding object's dimension being (h, w). For each keypoint, the PCK is measured as the fraction of objects where it was found correctly.

As shown in the Table 5.2, although no bounding box information is used, our method

Parts	$\alpha = 0.02$	$\alpha = 0.05$	$\alpha = 0.08$	$\alpha = 0.10$
Back	9.4%	46.8%	74.8%	85.6%
Beak	12.7%	62.5%	89.1%	94.9%
Belly	8.2%	40.7%	70.3%	81.9%
Breast	9.8%	45.1%	74.2%	84.5%
Crown	12.2%	59.8%	87.7%	94.8%
Forehead	13.2%	63.7%	91.0%	96.0%
Left eye	11.3%	66.3%	91.0%	95.7%
Left leg	7.8%	33.7%	56.6%	64.6%
Left wing	6.7%	31.7%	56.7%	67.8%
Nape	11.5%	54.3%	82.9%	90.7%
Right eye	12.5%	63.8%	88.4%	93.8%
Right leg	7.3%	36.2%	56.4%	64.9%
Right wing	6.2%	33.3%	58.6%	69.3%
Tail	8.2%	39.6%	65.0%	74.7%
Throat	11.8%	56.9%	87.2%	94.5%

Table 5.2: **Keypoint Localization results.** We use PCK as our evaluation metric. The prediction is correct if lies within $\alpha \times max(h, w)$ of the annotated keypoint with the corresponding object's dimension being (h, w). We show results on different α. No bounding box information is used.

	Head	Body
RCNN [116]	61.42%	70.68%
Ours	**63.82** %	**89.06**%

Table 5.3: Part localization accuracy in terms of PCP on the CUB200-2011 bird dataset.

can still have strong localization performances even when the requirement α is small. To our knowledge, these are the first reported PCK numbers on this dataset. We also show part localization visualizations in Figure 5.3. We show the ground truth annotation and our prediction side-by-side. As you can see, almost all of the part predictions lie on the bird body. Some errors are caused by the confusion of left and right, for example, localizing left leg on the position of right leg. The small localization errors could be corrected by fusing multiple feature layers [71, 46] or multi-scale modeling [102].

We also show the localization results on part box prediction in Table 5.3 in terms of PCP (Percentage of correctly localized parts). The prediction is correct if the overlap between it and grountruth part box is over 0.5. We compared with Chapter 4 and we outperform the RCNN detection.

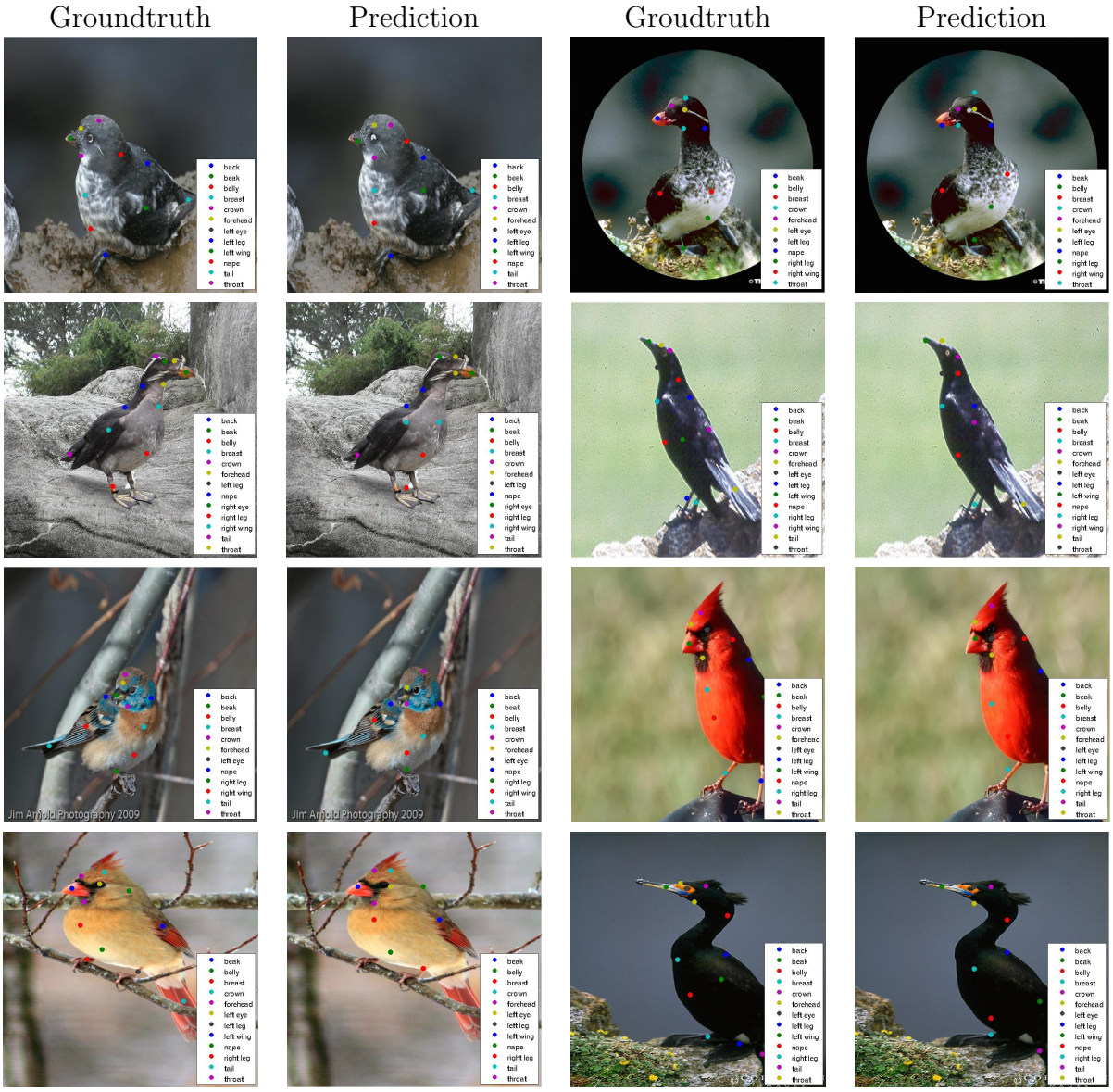


Figure 5.3: Visualizations of keypoint predictions. Ground truth annotations are shown in the left and our prediction results are shown in the right. Each color map to one keypoint. No bounding box information is used during training or test time. The images shown are warped to align the visualizations.

5.4 Summary

We have presented an approach to classification that embodies fine-grained approaches to part/keypoint localization, pose-normalized descriptor learning, and category learning in this chapter. Our model is fine-grained in its spatial localization ability, and in its ability

to distinguish closely related classes in a fine-grained recognition task. A fully convolutional network can localize instance keypoints using a pixel-level map, and deep pose-normalized descriptors can make distinctions across fine-grained category labels learned in a coordinate space that is defined using the estimated keypoints. We unify these steps in an end-to-end trainable network that estimates part locations, pose-normalized descriptor representations, and fine-grained classifier weights using a joint loss on keypoint predictions and the classification task.

Chapter 6

Conclusion

In this thesis, I have proposed using pose-normalized representations for fine-grained classification task. To this end, I have demonstrated that the idea of pose-normalization can be applied to two similar tasks: human attribute classification and person recognition beyond frontal face. I then move to the deep learning based localization method: part-based RCNN for more accurate part predictions without bounding box. I also show the results by combining part-based RCNN and recent compact bilinear pooling features and show the dense window sampling method to overcome the limitation by region proposals. Lastly, I present the end-to-end trainable fully convolutional network that simultaneously learn part predictions and category prediction.

A careful comparison of strong keypoint supervision and weak class label supervision is valuable future work to guide further improvements to fine-grained recognition. While strong supervision aids with correspondence, weak supervision may learn from more data. This comparison could take the form of a strongly supervised spatial transformer network or an unsupervised variant of our network incorporating part discovery as in [93]. With end-to-end multi-task learning it should be possible to unify these approaches in semi-supervised models for fine-grained recognition.

Bibliography

[1] Anelia Angelova and Shenghuo Zhu. "Efficient object detection and segmentation for fine-grained recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[2] Anelia Angelova, Shenghuo Zhu, and Yuanqing Lin. "Image segmentation for large-scale subcategory flower recognition". In: *WACV*. 2013.

[3] Dragomir Anguelov et al. "Contextual Identity Recognition in Personal Photo Albums." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[4] Hossein Azizpour and Ivan Laptev. "Object Detection Using Strongly-Supervised Deformable Part Models". In: *European Conference on Computer Vision (ECCV)*. 2012.

[5] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. "Multiple Object Recognition with Visual Attention". In: *ICLR* (2015).

[6] Peter N. Belhumeur et al. "Localizing Parts of Faces Using a Consensus of Exemplars". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[7] Peter Belhumeur et al. "Searching the World′s Herbaria: Visual Identification of Plant Species". In: *European Conference on Computer Vision (ECCV)*. Oct. 2008,

[8] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. "Automatic Attribute Discovery and Characterization from Noisy Web Data". In: *European Conference on Computer Vision (ECCV)*. 2010.

[9] Thomas Berg and Peter N. Belhumeur. "POOF: Part-Based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[10] Léon Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks: Tricks of the Trade*. Vol. 7700. 2012.

[11] Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. "Describing People: Poselet-Based Approach to Attribute Classification". In: *International Conference on Computer Vision (ICCV)*. 2011.

[12] Lubomir Bourdev and Jitendra Malik. "Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations". In: *International Conference on Computer Vision (ICCV)*. 2009.

[13] Lubomir Bourdev et al. "Detecting People Using Mutually Consistent Poselet Activations". In: *European Conference on Computer Vision (ECCV)*. 2010.

[14] Steve Branson et al. "Bird Species Categorization Using Pose Normalized Deep Convolutional Nets". In: *British Machine Vision Conference (BMVC)*. 2014.

[15] Steve Branson et al. "Visual Recognition with Humans in the Loop". In: *European Conference on Computer Vision (ECCV)*. 2010.

[16] Martin Bfffdfffduml, Makarand Tapaswi, and Rainer Stiefelhagen. "Semi-supervised learning with constraints for person identification in multimedia data." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[17] H.C. Burger, C.J. Schuler, and S. Harmeling. "Image denoising: Can plain Neural Networks compete with BM3D?" In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[18] Joao Carreira et al. "Semantic segmentation with second-order pooling". In: *European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 430–443.

[19] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. "Symbiotic Segmentation and Part Localization for Fine-Grained Categorization". In: *International Conference on Computer Vision (ICCV)*. 2013.

[20] Yuning Chai et al. "TriCoS: A Tri-level Class-Discriminative Co-segmentation Method for Image Classification". In: *European Conference on Computer Vision (ECCV)*. 2012.

[21] Junyoung Chung et al. "Deep Attribute Networks". In: *Deep Learning and Unsupervised Feature Learning NIPS Workshop*. 2012.

[22] Zhen Cui et al. "Fusing Robust Face Region Descriptors via Multiple Metric Learning for Face Recognition in the Wild." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[23] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005.

[24] Jia Deng, Jonathan Krause, and Li Fei-Fei. "Fine-Grained Crowdsourcing for Fine-Grained Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[25] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[26] Jeff Donahue et al. "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition". In: *International Conference on Machine Learning (ICML)*. 2014.

[27] Kun Duan et al. "Discovering Localized Attributes for Fine-grained Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[28] M. Everingham, J. Sivic, and A. Zisserman. ""Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video". In: *Proceedings of the British Machine Vision Conference*. 2006.

[29] Mark Everingham, Josef Sivic, and Andrew Zisserman. "Taking the Bite out of Automated Naming of Characters in TV Video". In: *Image and Vision Computing*. 2009.

[30] Michela Farenzena et al. "Person re-identification by symmetry-driven accumulation of local features." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010.

[31] Ali Farhadi et al. "Describing Objects by their Attributes". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[32] Ryan Farrell et al. "Birdlets: Subordinate Categorization using Volumetric Primitives and Pose-normalized Appearance". In: *International Conference on Computer Vision (ICCV)*. 2011. DOI: 10.1109/ICCV.2011.6126238.

[33] P. F. Felzenszwalb et al. "Object Detection with Discriminatively Trained Part Based Models". In: vol. 32. 9. 2010, pp. 1627–1645.

[34] Pedro Felzenszwalb and Daniel Huttenlocher. "Efficient Matching of Pictorial Structure". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2000.

[35] Martin A. Fischler and Robert A. Elschlager. "The Representation and Matching of Pictorial Structures". In: *IEEE Transactions on Computers*. Jan. 1973.

[36] A. Gallagher and T. Chen. "Clothing Cosegmentation for Recognizing People". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008.

[37] Andrew C. Gallagher and Tsuhan Chen. "Understanding Images of Groups of People". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[38] Yang Gao et al. "Compact Bilinear Pooling". In: *CoRR* abs/1511.06062 (2015).

[39] Rahul Garg et al. "Where's Waldo: Matching People in Images of Crowds". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[40] E. Gavves et al. "Fine-Grained Categorization by Alignments". In: *International Conference on Computer Vision (ICCV)*. 2013.

[41] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[42] Christoph Göring et al. "Nonparametric Part Transfer for Fine-grained Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[43] Doug Gray, Shane Brennan, and Hai Tao. "Evaluating appearance models for recognition, reacquisition, and tracking". In: *In IEEE International Workshop on Performance Evaluation for Tracking and Surveillance, Rio de Janeiro*. 2007.

[44] Douglas Gray and Hai Tao. "Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features". In: *Proceedings of the 10th European Conference on Computer Vision: Part I*. European Conference on Computer Vision (ECCV). 2008.

[45] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. "Is that you? Metric learning approaches for face identification". In: *International Conference on Computer Vision*. Kyoto, Japan, Sept. 2009.

[46] Bharath Hariharan et al. "Hypercolumns for Object Segmentation and Fine-grained Localization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[47] Gary B. Huang et al. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. Tech. rep. 07-49. University of Massachusetts, Amherst, Oct. 2007.

[48] ILSVRC. "ImageNet Large-scale Visual Recognition Challenge". In: 2010-2012. URL: `http://www.image-net.org/challenges/LSVRC/2011/`.

[49] Max Jaderberg et al. "Spatial Transformer Networks". In: (2015).

[50] Arjun Jain et al. "Learning Human Pose Estimation Features with Convolutional Networks". In: *ICLR*. 2014.

[51] Kevin Jarrett et al. "What is the Best Multi-Stage Architecture for Object Recognition?" In: *International Conference on Computer Vision (ICCV)*. 2009.

[52] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *arXiv preprint arXiv:1408.5093* (2014).

[53] Aditya Khosla et al. "Novel Dataset for Fine-Grained Image Categorization". In: *FGVC Workshop, Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[54] Jonathan Krause et al. "Fine-Grained Recognition without Part Annotations". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[55] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. 2012.

[56] Neeraj Kumar et al. "Attribute and Simile Classifiers for Face Verification". In: *International Conference on Computer Vision (ICCV)*. 2009.

[57] Neeraj Kumar et al. "Leafsnap: A Computer Vision System for Automatic Plant Species Identification". In: *European Conference on Computer Vision (ECCV)*. Oct. 2012.

[58] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. "Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[59] Ryan Layne, Timothy Hospedales, and Shaogang Gong. "Person Re-identification by Attributes". In: *British Machine Vision Conference (BMVC)*. 2012.

[60] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[61] Quoc V. Le et al. "Building High-level Features Using Large Scale Unsupervised Learning". In: *ICML*. 2012.

[62] Yang LeCun et al. "Backpropagation applied to hand-written zip code recognition". In: *Neural Computation*. 1989.

[63] Yann Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.

[64] Wei Li, Rui Zhao, and Xiaogang Wang. "Human Reidentification with Transferred Metric Learning." In: *ACCV*. Vol. 7724. Lecture Notes in Computer Science. Springer, 2012, pp. 31–44.

[65] Wei Li et al. "DeepReID: Deep Filter Pairing Neural Network for Person Re-Identification." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[66] Dahua Lin et al. "Joint People, Event, and Location Recognition in Personal Photo Collections Using Cross-domain Context". In: *Proceedings of the 11th European Conference on Computer Vision: Part I*. European Conference on Computer Vision (ECCV). Heraklion, Crete, Greece, 2010, pp. 243–256.

[67] Di Lin et al. "Deep LAC: Deep Localization, Alignment and Classification for Fine-grained Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[68] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. "Bilinear CNN Models for Fine-grained Visual Recognition". In: (2015).

[69] Jiongxin Liu and Peter N. Belhumeur. "Bird Part Localization Using Exemplar-Based Models with Enforced Pose and Subcategory Consistenty". In: *International Conference on Computer Vision (ICCV)*. 2013.

[70] Jiongxin Liu et al. "Dog Breed Classification Using Part Localization". In: *European Conference on Computer Vision (ECCV)*. 2012.

[71] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[72] Jonathan Long, Ning Zhang, and Trevor Darrell. "Do Convnets Learn Correspondence?" In: *NIPS*. 2014.

[73] S. Maji et al. *Fine-Grained Visual Classification of Aircraft*. Tech. rep. 2013. arXiv: 1306.5151 [cs-cv].

[74] G. Martinez-Munoz et al. "Dictionary-free categorization of very similar objects via stacked evidence trees". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009.

[75] Volodymyr Mnih et al. "Recurrent Models of Visual Attention". In: *NIPS*. 2014.

[76] Mor Naaman et al. "Leveraging Context to Resolve Identity in Photo Albums". In: *JCDL'05*. 2005.

[77] Maria-Elena Nilsback and Andrew Zisserman. "A Visual Vocabulary for Flower Classification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006.

[78] Maria-Elena Nilsback and Andrew Zisserman. "Automated Flower Classification over a Large Number of Classes". In: *ICVGIP*. 2008.

[79] Bruno A. Olshausen, Charles H. Anderson, and David C. Van Essen. "A Neurobiological Model of Visual Attention and Invariant Pattern Recognition Based on Dynamic Routing of Information". In: *Journal of Neuroscience* 13 (1993), pp. 4700–4719.

[80] Omar Oreifej, Ramin Mehran, and Mubarak Shah. "Human identity recognition in aerial images". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 709–716.

[81] Devi Parikh and Kristen Grauman. "Relative Attributes". In: *International Conference on Computer Vision (ICCV)*. 2011.

[82] O. M. Parkhi et al. "Cats and Dogs". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[83] O. M. Parkhi et al. "The Truth About Cats and Dogs". In: *International Conference on Computer Vision (ICCV)*. 2011.

[84] Ninh Pham and Rasmus Pagh. "Fast and scalable polynomial kernels via explicit feature maps". In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2013, pp. 239–247.

[85] Bryan Prosser et al. "Person Re-identification by Support Vector Ranking." In: *British Machine Vision Conference (BMVC)*. 2010.

[86] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems*. 2007, pp. 1177–1184.

[87] Marc'Aurelio Ranzato et al. "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2007.

[88] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature*. 1986.

[89] Pierre Sermanet, Andrea Frome, and Esteban Real. "Attention for Fine-Grained Categorization". In: *CoRR* abs/1412.7054 (2014).

[90] Pierre Sermanet et al. "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks". In: vol. abs/1312.6229. 2013.

[91] Pierre Sermanet et al. "Pedestrian Detection with Unsupervised Multi-Stage Feature Learning". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[92] Asma Rejeb Sfar, Nozha Boujemaa, and Donald Geman. "Vantage Feature Frames For Fine-Grained Categorization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013.

[93] Marcel Simon and Erik Rodner. "Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks". In: *International Conference on Computer Vision (ICCV)*. 2015.

[94] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. eprint: `arXiv:1409.1556`.

[95] J. Sivic, M. Everingham, and A. Zisserman. ""Who are you?" – Learning Person Specific Classifiers from Video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2009.

[96] Josef Sivic, C. Lawrence Zitnick, and Richard Szeliski. "Finding people in repeated shots of the same scene". In: *British Machine Vision Conference (BMVC)*. 2006.

[97] Michael Stark et al. "Fine-Grained Categorization for 3D Scene Understanding". In: *British Machine Vision Conference (BMVC)*. 2012.

[98] Christian Szegedy et al. *Going Deeper with Convolutions*. 2015.

[99] Yaniv Taigman et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[100] Makarand Tapaswi, Martin Bfffdfffduml, and Rainer Stiefelhagen. "Knock! Knock! Who is it? probabilistic person identification in TV-series". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[101] Joshua B Tenenbaum and William T Freeman. "Separating style and content with bilinear models". In: *Neural computation* 12.6 (2000), pp. 1247–1283.

[102] Jonathan Tompson et al. "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation". In: *NIPS*. 2014.

[103] Shubham Tulsiani and Jitendra Malik. "Viewpoints and Keypoints". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[104] Matthew Turk and Alex Pentland. "Eigenfaces for Recognition". In: *J. Cognitive Neuroscience*. Vol. 3. 1. Jan. 1991, pp. 71–86.

[105] J. Uijlings et al. "Selective Search for Object Recognition". In: *IJCV*. 2013.

[106] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. http://www.vlfeat.org/. 2008.

[107] Gang Wang et al. "Seeing People in Social Context: Recognizing People and Social Relationships". In: *Proceedings of the 11th European Conference on Computer Vision: Part V*. European Conference on Computer Vision (ECCV). Heraklion, Crete, Greece, 2010, pp. 169–182.

[108] P. Welinder et al. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.

[109] John Wright et al. "Robust Face Recognition via Sparse Representation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 31. 2. Feb. 2009, pp. 210–227.

[110] Tianjun Xiao et al. "The Application of Two-level Attention Models in Deep Convolutional Neural Network for Fine-grained Image Classification". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[111] Lingxi Xie et al. "Hierarchical Part Matching for Fine-Grained Visual Categorization". In: *International Conference on Computer Vision (ICCV)*. 2013.

[112] Shulin Yang et al. "Unsupervised Template Learning for Fine-Grained Object Recognition". In: *NIPS*. 2012.

[113] Bangpeng Yao, Gary Bradski, and Li Fei-Fei. "A Codebook-Free and Annotation-Free Approach for Fine-grained Image Categorization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[114] Bangpeng Yao, A. Khosla, and Li Fei-Fei. "Combining Randomization and Discrimination for Fine-grained Image Categorization". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011.

[115] Dong Yi, Zhen Lei, and Stan Z. Li. *Deep Metric Learning for Practical Person Re-Identification*. 2014. eprint: `arXiv:1407.4979`.

[116] Ning Zhang, Jeff Donahue, and Trevor Darrell. "Part-based R-CNNs for Fine-grained Category Detection". In: *European Conference on Computer Vision (ECCV)*. 2014.

[117] Ning Zhang, Ryan Farrell, and Trevor Darrell. "Pose Pooling Kernels for Sub-Category Recognition". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[118] Ning Zhang et al. "Beyond Frontal Faces: Improving Person Recognition Using Multiple Cues". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[119] Ning Zhang et al. "Deformable Part Descriptors for Fine-grained Recognition and Attribute Prediction". In: *International Conference on Computer Vision (ICCV)*. 2013.

[120] Ning Zhang et al. "Fine-grained pose prediction, normalization, and recognition". In: *CoRR*. Vol. abs/1511.07063. 2015.

[121] Ning Zhang et al. "PANDA: Pose Aligned Networks for Deep Attribute Modeling". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[122] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. "Learning Mid-level Filters for Person Re-identification." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[123] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. "Semi-supervised learning with constraints for person identification in multimedia data." In: *International Conference on Computer Vision*. 2013.