Aligned to the Object, not to the Image: A Unified Pose-aligned Representation for Fine-grained Recognition

Pei Guo, Ryan Farrell
Computer Science Department
Brigham Young University
{peiguo,farrell}@cs.byu.edu

Abstract

Dramatic appearance variation due to pose constitutes a great challenge in fine-grained recognition, one which recent methods using attention mechanisms or second-order statistics fail to adequately address. Modern CNNs typically lack an explicit understanding of object pose and are instead confused by entangled pose and appearance. In this paper, we propose a unified object representation built from pose-aligned regions of varied spatial sizes. Rather than representing an object by regions aligned to image axes, the proposed representation characterizes appearance relative to the object's pose using pose-aligned patches whose features are robust to variations in pose, scale and viewing angle. We propose an algorithm that performs pose estimation and forms the unified object representation as the concatenation of pose-aligned region features, which is then fed into a classification network. The proposed algorithm attains state-of-the-art results on two fine-grained datasets, notably 89.2% on the widely-used CUB-200 [46] dataset and 87.9% on the much larger NABirds [45] dataset. Our success relative to competing methods shows the critical importance of disentangling pose and appearance for continued progress in fine-grained recognition.

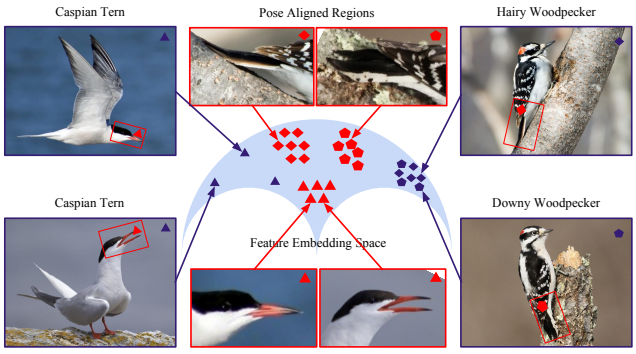

Figure 1: **Motivation for Pose-Aligned Regions.** Two terns of the same species (images on left), but in different poses, have dramatically different appearances while two different species of woodpecker (images on right) appear nearly identical except for the barring pattern on the outer tail (and the shape of the beak). The large intra-class variance and small inter-class variance of the full images make the feature space distance inaccurately reflect the true class relationships. Such observations motivate the use of *pose-aligned regions* that disentangle intrinsic part appearance from variations in object pose, leading to a feature space that facilitates correctly classifying the species or fine-grained category.

1. Introduction

What makes fine-grained visual categorization (FGVC), commonly referred to as fine-grained recognition, different from general visual categorization? One important distinction lies in the difficulty of the datasets. General-purpose visual categorization often involves the classification of everyday objects, such as chairs, bicycles and dogs, which are easy for humans to identify. Fine-grained recognition, on the other hand, consists of more detailed classifications such as identifying the species of a bird. This is extremely difficult for non-expert humans as it requires familiarity with domain knowledge and hundreds of hours of training. Computer algorithms for fine-grained recognition have the potential to be far more accurate than most humans and can thus benefit millions of people by providing services like species recognition through mobile applications [7, 2, 28].

An intrinsic challenge of fine-grained recognition is small inter-category variance coupled with large intra-category variance. Discriminative features for two visually similar categories often lie in a few key locations; while the appearances of two objects from the same category may dramatically different due simply to pose variation. The entangling of appearance and pose presents a great challenge and motivates the need for stable appearance features that

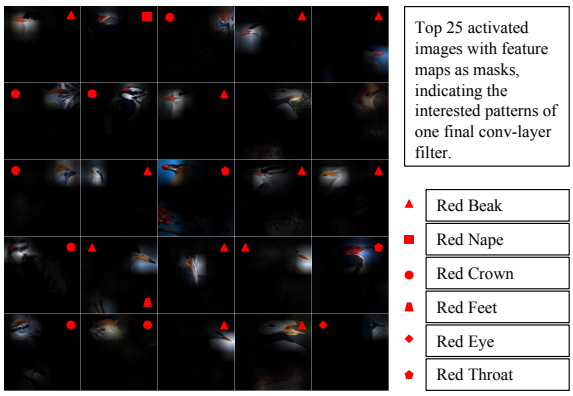

Figure 2: **Filter Visualization.** We visualize the 25 images that maximally activate an example filter. Different semantic parts (*e.g*. red beak, red eye, *et al*.) with similar appearance can all activate this filter, causing confusion for the classifier when such semantic parts would be discriminative. This problem can be solved by disentangling pose and appearance.

are invariant to variations in pose, scale and rotation.

It's almost instinct for humans to identify and visually compare key locations across objects in different poses, establishing correspondences. Convolutional neural networks, however, struggle on this task because the convolutional mechanisms are purely appearance-based and lack an understanding of geometry or pose. Their built-in pooling mechanisms can tolerate a certain amount of scale and rotation variation [4, 5, 6, 36, 47], but exactly how much is still largely an open question [39]. We show this in Figure 2 via the visualization of some final convolutional layer responses. We show the top-activated images together with the feature map as a masked region. It's evident that this convolutional filter is attuned to red beaks. However, due to its lack of pose-awareness, this filter also fires strongly at visually similar parts such as red crowns, red throats, red eyes, *etc*. This causes confusion for the classifier because of the noisy entangled pose-appearance representation.

In the feature embedding space, dramatic pose variation can make images of the same category farther separated and images of different but visually-similar categories appear closer together as shown in Figure 1. It is therefore vital that pose-aligned regions, which explicitly factor out pose variation, should be the building block of the disentangled image representation.

Recent efforts in fine-grained recognition have largely focused on two directions. One is algorithms related to second-order statistics[9, 16, 32, 24]. Representative works include Bilinear Pooling[32], its memory efficient variants [16, 24], and those that extend to higher-order statistics [9]. The idea is to project the features into a higher-order space where they can be linearly separated. Such methods have both sound theoretical support and work well in practice. However, they look at the image globally, and thus having little hope of finding subtle highly-localized differences. Also, they lack interpretability and insights for further improvement.

The second direction is attention-based methods [15, 29, 35, 41, 48, 57] that use subnetworks to propose possible discriminative regions to attend to. However, the regions proposed by these networks are often weakly-supervised by some heuristic loss function, lacking proof that they really attend to the right location. Both of these directions suffer from a lack of pose awareness and moreover the entanglement of pose and appearance features limits their performance. Furthermore, training data is often scarce in the long-tailed distributions seen in many fine-grained domains; in such cases, both techniques suffer as the limited training imagery does not adequately span the space of pose and viewing angle for each category, hindering their ability to recognize any species in any pose.

Based on the above observations, we propose to disentagle pose and appearance via a unified object representation built upon pose-aligned regions. Those regions are characterized by rectangular patches defined relative to two keypoints anchors. The final object representation is an aggregation of the features from all pose-aligned regions. This representation comprises a pose-invariant and overcomplete basis of features at multiple scales. We contrast the pose-aligned regions with weakly-supervised regions that are generated in a purely data-driven fashion and with "axis-aligned" rectangular bounding boxes centered around a keypoint or landmark. The features from these types of regions are subject to natural variations in pose, scale and viewing angles. We experimentally demonstrate that axis-aligned regions are less-capable of classifying fine-grained datasets compared to pose-aligned regions (see Figure 6).

To automate the process of applying the unified object representation to fine-grained recognition, we propose an algorithm that first performs pose-estimation via keypoint detection, enabling the generation of pose-aligned region features. The local features from these aligned regions, regions of varying size/scale relative to the object, are concatenated to comprise the unified representation for the object and are then fed into a classification network to produce a final classification prediction. We call the proposed algorithm PAIRS: Pose and Appearance Integration for Recognizing Subcategories. It achieves state-of-the-art results on two key fine-grained datasets: CUB-200-2011 [46] and NABirds [45]. Keypoint annotations are used only during training. In consideration of the annotation cost, keypoint annotations may actually be less expensive and time-consuming than collecting additional training samples because keypoints can be annotated by human non-experts

whereas fine-grained image category annotations require the consensus of multiple domain experts.

2. Background and Related Work

Fine-grained visual categorization (FGVC) lies between generic category-level object recognition like VOC [12], ImageNet [40], COCO [31], *etc*. and instance-level classification like face recognition or other visual biometrics. The challenges inherent to FGVC are many. Differences between similar species are often subtle and highly-localized and thus difficult even for (non-expert) humans to identify. Dramatic pose changes introduce great intra-class variance. Generalization also becomes an issue as the network struggles to find truly useful and discriminative features.

FGVC has drawn broad interest within the computer vision community. Early work includes [10, 13, 34, 50, 51, 53, 54], two of which explicitly tackle the challenging interplay of pose and appearance. Birdlets [13], a volumetric poselet representation, was proposed to account for the pose and appearance variation. Zhang, *et al*. [54] further proposed pose-normalized descriptors based on computationally-efficient deformable part models. While they seek to address pose and appearance, their hand-engineered features result in limited success.

Our work is related to part-based CNN models [3, 21, 25, 30, 52, 55] which seek to decompose the object into semantic parts. Zhang, *et al*. [52] employed the R-CNN [19] object detection framework for object and part detection. Part-Stacked CNN [22] proposes a fully-convolutional network for keypoint detection and a two-stream convolutional network for object- and part-level feature extraction. Deep LAC [30] proposes a valve linkage function for back-propagation chaining, forming a deep localization, alignment and classification system. Zhang *et al*. [55] introduce an end-to-end learning framework for joint learning of pose estimation, normalization and recognition. These models all use a handful of image-aligned patches, which can appear very different depending on object pose and viewpoint.

Our work is perhaps most closely related to POOF [1] which also uses keypoint pair patches. Unlike POOF, we employ a fully-convolutional network for keypoint detection. And where POOF computes 5000 patch features per image, whereas we're only computing 35-70.

Other methods focus on object alignment. Unlike previous methods which relied on detectors for part localization, Gavves *et al*. [17, 18] propose to localize distinctive details by roughly aligning the objects using just the overall shape. Spatial Transformer Networks [23] introduced a differentiable affine transformation learning layer to transform and align the object or part of interest.

Another direction in fine-grained recognition is feature correlation and kernel mapping. Bilinear Pooling [32] computes a second order-polynomial kernel mapping on CNN features. Several extensions [9, 16, 24] followed this simple paradigm. Compact Bilinear Pooling [16] proposes a compact representation to approximate the polynomial kernel, reducing memory usage. Low-rank Bilinear Pooling [24] represents the covariance features as a matrix and applies a low-rank bilinear classifier. Kernel Pooling [9] proposes a general pooling framework that captures the higher-order interactions of features in the form of kernels. This line of work achieves relatively good results with only weak supervision. These approaches, however, attend to the image globally, lacking a mechanism for part-level discovery. This constrains their potential for further improvement.

Inspired by human attention mechanisms, many attempts have been made to guide the attention of CNN models toward informative object parts. Works along this direction include [15, 29, 35, 41, 48, 57]. Zheng *et al*. [58] proposes a multi-attention convolutional neural network (MA-CNN), where part generation and feature learning can reinforce each other. Lam *et al*. [29] leverages long short-term memory (LSTM) networks to unify new patch candidate generation and informative part evaluation. This work establishes the current state-of-the-art performance on CUB-200-2011 dataset, achieving an accuracy of 87.5% with part annotations (excluding works like [8, 27] that utilize outside training data). The key difference in our PAIRS representation is that it integrates pose and appearance information and explicitly achieves multi-scale attention over semantic object parts at the same time.

3. PAIRS - Pose and Appearance Integration

We illustrate our algorithm pipeline in Figure 3. We first apply a simple yet effective fully-convolutional neural network for keypoint-based pose estimation. We follow the prevailing modular design paradigm by stacking convolutional blocks that have similar topology. We show that our pose estimation network achieves superior results on the CUB-200 dataset, both qualitatively and quantitatively. Second, given detected keypoint locations, a rectangular region is aligned to each keypoint pair and cropped from the original image. The region is then similarity-transformed into a uniform-sized patch (see Figure 4), such that both keypoints are at fixed positions across different images. As the representation is normalized to the keypoint locations, the patches are well-aligned, independent of the pose or the camera's angle. Third, we train a separate CNN model as the feature extractor for each pose-aligned patch representation. Last, we explore different classification architectures for the unified representation based on the assumption that parts differ in their respective contributions for different images and classes. We find surprisingly that the Multi-Layer Perception (MLP), while perhaps the most simple method, achieves the best final classification accuracy.

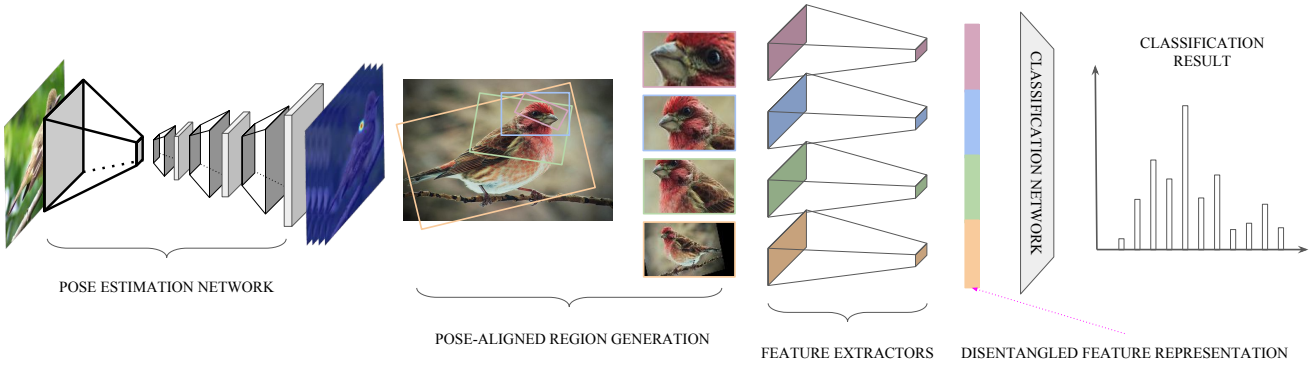

Figure 3: **Overview of the Proposed Framework for Fine-grained Recognition.** We first apply a pose-estimation network to the image for keypoint detection. Pose-aligned regions are then extracted from the image using the predicted keypoint locations. We then extract features from the individual regions using region-specific networks. The concatenated features collectively form a unified multi-scale representation that is invariant to pose, scale and rotation change. This representation is then fed into a classification network for the final fine-grained classification.

3.1. Pose Estimation Network

Pose estimation networks usually follow one of two paradigms for prediction. The first is to directly regress discrete keypoint coordinates, e.g. (x_i, y_i). Representative approaches include [44]. The alternate approach [43] instead uses a two-dimensional probability distribution heat map to represent the keypoint location. We call this resulting multi-channel probability distribution matrix a *pose tensor*.

In this paper, we adopt the second strategy, proposing a fully convolutional network to produce the desired pose tensor. Specifically, we take a pretrained classification network and remove the final classifier layer(s), retaining what can be seen as an encoder network that encodes strong visual semantics. We follow the prevailing modular design to stack repeated building blocks to the end of the network. This building block consists of one upsampling layer, one convolutional layer, one batch normalization layer and one ReLU layer. The parameter-free bilinear interpolation layer is used for upsampling. The convolutional layer uses 1x1 kernel and reduces the input channel size by half. Last, a final convolutional layer and upsampling layer are added to produce the pose tensor. There are many modifications one can make to enhance this basic model, including using larger 3x3 kernels, adding more convolutional layers to the building block, adding residue connection to each block, stacking more building blocks, and using a learnable transposed convolutional layer for upsampling. We find these structures provide only limited improvement but introduce more parameters, and we therefore adopt this simpler architecture.

3.2. Patch Generation

Historically, part-based representations would model parts either as rectangular regions [14, 52] or keypoints. Keypoints are convenient for pose-estimation. However, the square or rectangular patches, each centered on a given keypoint and extracted to characterize the part's appearance, are far from optimal in the presence of rotation or more general pose variation. We instead, propose to use keypoint pairs as anchor points in extracting pose-aligned patches.

Given two keypoints $\vec{p}_i = (x_i, y_i)$ and $\vec{p}_j = (x_j, y_j)$, we define the vectors $\vec{r}_{ij} = \vec{p}_j - \vec{p}_i$, and $\hat{\vec{r}}_{ij} = \vec{r}_{ij}/||\vec{r}_{ij}||_2$. We also define the vector $\hat{\vec{t}}_{ij} = \hat{\vec{z}} \times \hat{\vec{r}}_{ij}$, a unit vector perpendicular to $\vec{r}_{ij}$, and the distances $d = ||\vec{r}_{ij}||_2$ and $h = d/2$ for convenience. We seek to extract a region around $\vec{p}_i$ and $\vec{p}_j$ that is aligned with $\vec{r}_{ij}$ and has dimensions $2d \times d$. The four corners of this rectangular region are then given by:

$$\begin{bmatrix} (\vec{p}_i - h\hat{\vec{r}}_{ij}) + h\hat{\vec{t}}_{ij} & (\vec{p}_j + h\hat{\vec{r}}_{ij}) + h\hat{\vec{t}}_{ij} \\ (\vec{p}_i - h\hat{\vec{r}}_{ij}) - h\hat{\vec{t}}_{ij} & (\vec{p}_j + h\hat{\vec{r}}_{ij}) - h\hat{\vec{t}}_{ij} \end{bmatrix} \quad (1)$$

A similarity-transform is computed from these corners to extract the pose-normalized patch. Patches generated in this way contain stable pose-aligned features – features near these keypoints appear at roughly the same location in a given patch across different images, independent of the object's pose or the camera viewing angle. Details are shown in Figure 4.

3.3. Patch Feature Extraction

A separate patch classification network is trained for each posed-aligned `A | B` patch as a feature extractor (A and B are keypoints). The softmax outputs from those networks are concatenated as the representation for the input image. Alternately, the final convolutional layer output after pooling can also be used and the result is comparable. We find that leveraging symmetry can help reduce the overall number of classifiers by nearly 50%, described in Section 4.2.

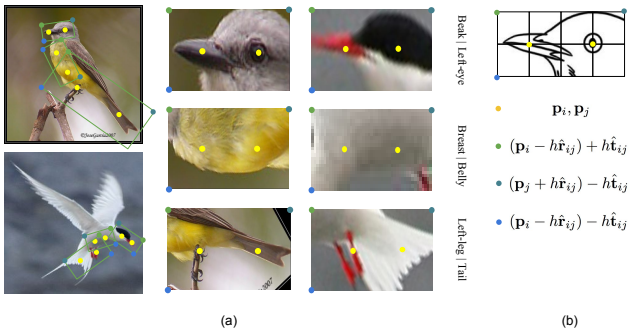

Figure 4: **Pose-aligned Patch Generation.** For each pair of keypoints, we fit a rectangular region whose corners are calculated as in (b). Objects in different poses and/or from different viewpoints can be compared directly by proposed keypoint pair patches shown in (a). Details are described in Section 3.2.

The proposed patch representation can be seen as a spatial pyramid that explicitly captures information from different parts at multiple spatial scales on the object.

3.4. Classification Network

To fully utilize the abundant patch representations, we explore different ways to form a strong combining network. Based on the assumption that only a small fraction of the patches contains discriminative information and patch contribution should therefore be weighted, we explore the following strategies:

Fixed patch selection: take the average score for a fixed number of top ranking patches. This strategy can also predict the performance ceiling of our PAIRS representation.

Dynamic patch selection: employ the sparsely gated network [42] to dynamically learn a selection function to select a fixed number of patches for each given input.

Sequential patch weighting: apply a Long Short Term Memory Network (LSTM) to reweight different patch features in a sequential way.

Static patch weighting: learn a Multi-Layer Perceptron network, which essentially applies a non-linear weighting function to combine the information from different patches.

We find surprisingly that the MLP network, though the simplest network architecture, achieves the best accuracy of all the above approaches. Details are included below in Section 4.3.

4. Experimental Evaluation

We test our algorithm on two datasets, CUB-200-2011 and NABirds. The CUB-200-2011 [46] dataset contains 200 species of birds with 5994 training images and 5794 testing images. The NABirds [45] dataset has 555 common species of birds in North America with a total number of 48,562 images. Class labels and keypoint locations for each image are provided in both datasets.

Figure 5: **Keypoint Detection Results.** Red dots represent the predicted location and black dots are the ground truth locations. Three failure patterns are shown in the third row, which are caused by the visual similarity between symmetric parts and dramatic and rare pose.

4.1. Keypoint Prediction Performance

We use PCK (Percentage of Correct Keypoints) score to measure the accuracy of our keypoint prediction approach. A predicted keypoint p is "correct" if its within a small neighborhood of the ground truth location g, *i.e.* if

$$||p - g||_2 \leq c * \max(h, w)$$

where $c = 0.1$ is the constant factor used previously [21, 55] and $\max(h, w)$ is the longer side of the bounding box.

We evaluate our pose-estimation network on CUB-200-2011 and compare our PCK scores with those of other methods in Table 1. We achieve the highest score on all 15 keypoints with a considerable margin. We do especially well on legs and wings where other models struggle to make precise predictions. Visualization results are shown in Figure 5.

Although we localize the wings and legs better than components, they are still worst predicted parts for our model. This is caused by significant pose changes as well as the inherent appearance similarity between symmetric parts. We also note that using keypoints to denote the wings isn't always appropriate – wings are two-dimensional planar parts that cover a relatively large area. Designating a keypoint for the wing can be ill-posed, it's challenging to decide which point represents the wing location best. In fact, the ground truth keypoint location of the CUB dataset is the average of five annotators' results and it's quite hard for them to reach a consensus.

	back	beak	belly	breast	crown	forehead	left-eye	left-leg
Huang *et al*. [21]	80.7	89.4	79.4	79.9	89.4	88.5	85.0	75.0
Zhang *et al*. [55]	85.6	94.9	81.9	84.5	94.8	96.0	95.7	64.6
Ours	**91.3**	**96.8**	**89.0**	**91.5**	**96.9**	**97.6**	**96.9**	**80.2**
	left-wing	nape	right-eye	right-leg	right-wing	tail	throat	**Overall**
Huang *et al*. [21]	67.0	85.7	86.1	77.5	67.8	76.0	90.8	86.6
Zhang *et al*. [55]	67.8	90.7	93.8	64.9	69.3	74.7	94.5	N/A
Ours	**76.8**	**94.6**	**97.4**	**80.3**	**75.3**	**83.6**	**97.4**	**90.5**

Table 1: **Pose Estimation (Keypoint Prediction).** Accuracy measured with PCK (Percentage of Correct Keypoints).

4.2. Patch Classification Network

We adopt the ResNet-50 [20] architecture for the patch classification network due to its high performance and compact GPU footprint, though alternate architectures like VGG and Inception can easily be adapted. We now discuss two considerations which facilitate training.

Symmetry. For a given object with n keypoints, the total number of patches that can be extracted is

$$\binom{n}{2} = \frac{n(n-1)}{2} = \mathcal{O}(n^2)$$

which increases quadratically with n. Most real world objects show some kind of symmetry. Due to the visual similarity inherent in symmetric pairs of keypoints (for example, right and left eyes, wings and feet), we treat each pair as a hybrid keypoint in the patch generation process. Many real-world objects, like birds, cats, cars, *etc*. are symmetric in appearance. Based on this observation, we propose to merge the patches for a symmetric pair of keypoints into a hybrid patch category, *e.g*. `left-eye | tail` and `right-eye | tail` can be merged into the hybrid `eye | tail` pair, with an appropriate flip of one patch.

As a result, the total number of patch classification networks is reduced from 105 to 69 for the CUB dataset; on the NABirds dataset, the number is reduced from 55 to 37.

Visibility. Due to self-occlusion or foreground-occlusion, not all keypoints are visible in the image. Previous works [21] would eliminate patches with invisible keypoints to purify the input data. Contrarily, we find that this would hurt the performance of the patch classifiers. Details for comparison can be found in Figure 7a. We believe this degradation is caused by the reduction in training set size. This is a similar finding to [27] that noisy but abundant data consistently outperforms clean but limited-size data. Additionally, the pose-estimation network will make a reasonable guess even if the keypoint is invisible. So during patch classifier training, all keypoints are considered visible by taking the maximally-activated location.

4.3. Classification Network

Based on the assumption that image patches should contribute differently to classification, four different strategies are explored and we describe the details in this section.

Fixed patch selection. We assume that only a few patches contain useful information and others are redundant or even act as noise. We propose a fixed patch selection strategy to keep the best k patches. A greedy search algorithm would evaluate all n choose k combinations for $k \in [1, n]$. The number of patches grows as $n!$ and quickly becomes intractable. We thus employ the *beam search* [38] algorithm. Instead of greedily searching the whole parameter space, we iteratively consider larger and larger subsets (values of k), while only keeping a limited number, w, of the best combinations at each iteration. Thus for a given k, we use the $w = 100$ best patch sets from iteration $k-1$, and consider, in turn, the effect of adding each patch among those not already in a given patch set. After all such expanded sets are considered, the w best sets are retained toward iteration $k+1$. To explore the potentially optimal performance of fixed patch selection on our pose-aligned patch representation, we also try this beam search on the test set with results shown in Figure 7d. Our observation is that without overfitting, the potential of fixed patch selection should be well above 89%, compared to 87.5% for the current state-of-the-art [29]. Simply averaging the predictions of all patches achieves 87.6% accuracy.

Dynamic patch selection. One alternative we experiment with is the sparsely gated network [42] for dynamic patch selection. Different from the beam search algorithm which identifies a static set of patches for all input images, the gated network selects different combination of patches depending on the input images. A tiny network is trained to predict weights for each patch and an explicit sparsity constraint is imposed on the weights to only allow k non-zero elements. A Sigmoid layer is added to normalize the weight. The network architecture can be described as,

$$G(x) = \text{softmax}(\text{top-k}(H(x)))$$

where $H(x)$ represents the mapping function from the input image to the patch weights. $G(x)$ is the patch selection

Approach	Annotations	Accuracy
Huang *et al*. [21]	GT+BB+KP	76.2
Zhang *et al*. [52]	GT + BB	76.4
Krause *et al*. [26]	GT+BB	82.8
Jaderberg *et al*. [23]	GT	84.1
Shu *et al*. [24]	GT	84.2
Zhang *et al*. [56]	GT	84.5
Xu *et al*. [49]	GT+BB+KP+WEB	84.6
Lin *et al*. [32]	GT+BB	85.1
Cui *et al*. [9]	GT	86.2
Lam *et al*. [29]	GT+KP	87.5
PAIRS Only	GT + KP	88.7
PAIRS+Single	GT + KP	**89.2**

Table 2: **Classification score on CUB.** Annotation key as follows: GT = class labels; BB = bounding box annotation; KP = keypoint annotations; WEB = images downloaded from the Internet.

function. Different architectures for the tiny network are tried and we find that a simple linear layer works well most of the time. Best accuracy is achieved when k = 105. Interestingly when k=1, Our selected patch performs worse than the best performing patch found fixed patch selection, implying the gated network's inability to learn useful information for decision making.

Sequential patch weighting. Recurrent neural networks (RNN) specialize in processing sequential data like text and speech. RNNs have been widely adopted as an attention mechanism to focus on different image regions sequentially. We instead employ an RNN for sequential patch weighting, aiming to discover different patches for decision making. We employ a one-layer Long Short Term Memory (LSTM) network with 512 nodes. Each node has a hidden layer of size 1024. The last output of the sequence is selected as the final output. We obtain 82.7% in this experiment, confirming the effectiveness of the LSTM network.

Static patch weighting The final, and as it turns out the most effective method that we tried is the MLP network. Our MLP network contains one hidden layer with 1024 parameters, followed by a batch normalization layer, a ReLU layer, and then the output layer. On CUB our final accuracy is 88.7%, 1.2% higher than the current state-of-the-art result. We combine the keypoint pair patches with single keypoint patches and achieve a new state-of-the-art 89.2% accuracy. We compare our result with several other strong baselines in Table 2.

We also test our algorithm on the NAbirds dataset with results shown in Table 3. Our algorithm attains an accuracy of 87.9%, more than 5% better than the best known result.

Approach	Accuracy
ResNet-50 [20] Baseline	79.2%
Bilinear CNN (PAMI 2017) [33]	79.4%
Pairwise Confusion [11]	82.8%
PAIRS	**87.9%**

Table 3: **Performance on the NABirds dataset.**

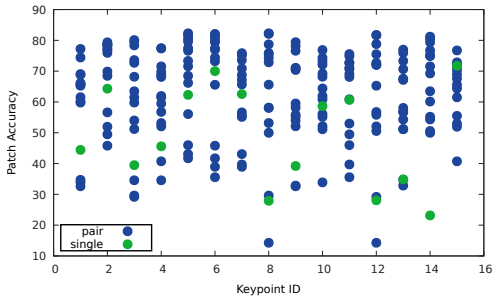

Figure 6: **Single Keypoint vs. PAIRS Patches.** We compare the accuracy of patches around keypoint k (green dots) with PAIRS patches involving keypoint k (blue dots). The x-axis is the keypoint id and the y-axis is the patch accuracy. Most single keypoint patch are inferior to PAIRS patches in terms of isolated patch accuracy.

4.4. Additional Study

Axis-Aligned vs. Pose-Aligned. In Figure 6, we show a comparison of classification accuracy between pose-aligned patches (keypoint pairs) and axis-aligned patches (single keypoints). Axis-aligned patches consistently perform poorly relative to the pose-aligned patches, confirming the effectiveness of our disentangled feature representation.

Patch Size Study. One hyper-parameter in our algorithm is the pose-aligned patch size. We tried several size options on the best performing patch and saw that accuracy is generally higher for larger-size patches. We adopted a 256×512 patch size because our base model is pretrained for this size.

Choice of Pose Estimation Network. To test the influence of the pose-estimation network on the proposed algorithm, we train a separate Stacked Hourglass Network (SHN) [37] model for comparison. While the SHN model is about 2% better than the Fully-convolutional Network (FCN) in it's PCK score, the final classification accuracy numbers are comparable.

4.5. Results Visualization

We show the classification accuracy for each patch when considered independently on the CUB dataset in Figure 7c. The best performing patch corresponds to `belly|crown`, achieving 79.6% accuracy. The worst performing patch is

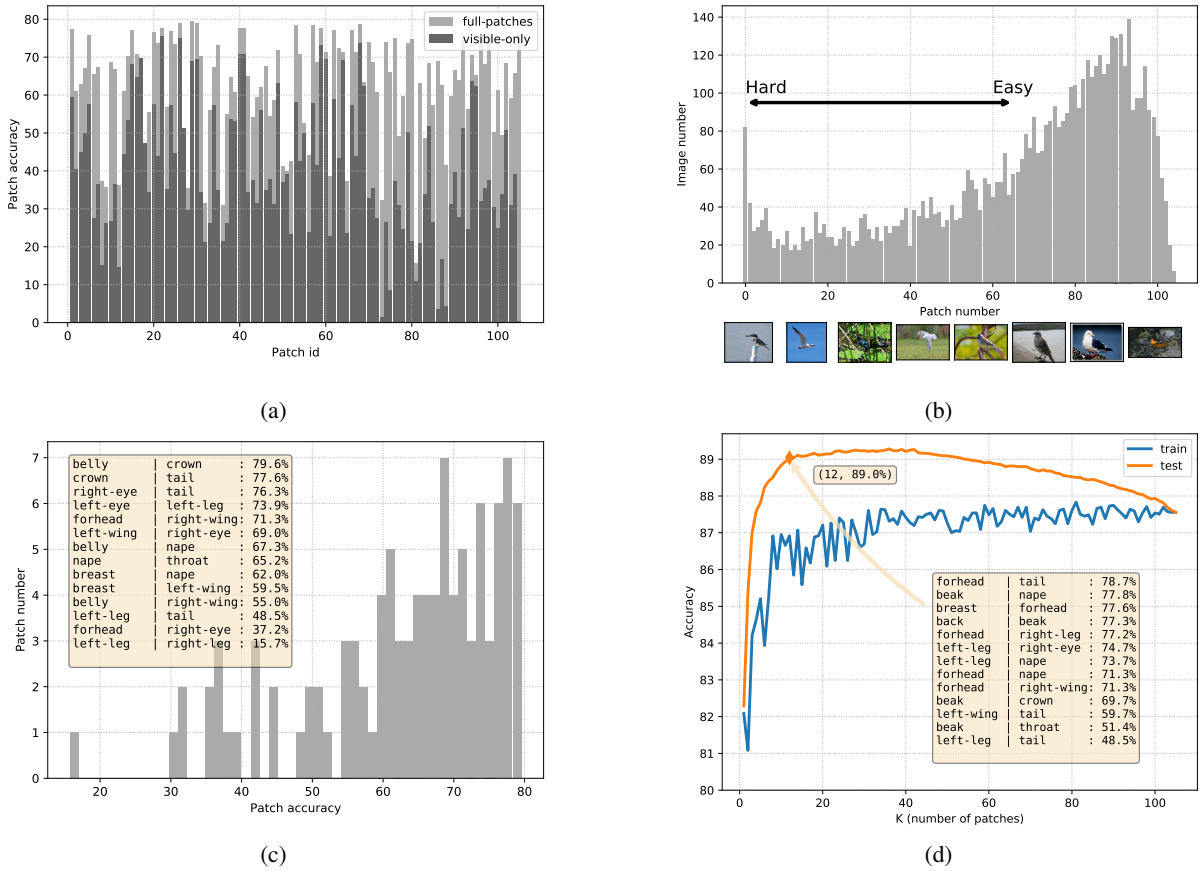

Figure 7: **Visualization of Results.** (a) We show patch classification network accuracies using two strategies, visible-only and all keypoint patches. This affirms that treating all keypoints as visible improves patch classifier accuracy. (b) Hard case mining by correct prediction patch number with sample images ranging from hard to easy. (c) Distributions of patch classifier performance. Some examples are shown in the text box. (d) Beam search results using two strategies, patch finding on the training (blue) and testing (orange) sets. The latter is purely for the estimating of the potential of the PAIRS representation.

the `left-leg | right-leg` pair which achieves only 15.7% accuracy. Empirically, global patches perform better in isolation than local patches, however local patches are also very important for localizing discriminative object parts. The best set of patches found by beam search (see Figure 7d) provides insight – a combination of global and local patches are selected to achieve an optimal result.

As hard cases often can only be classified by a few highly-localized discriminative parts, the number of patches with correct predictions reflects the difficulty of the image. We propose to use the fraction of patches correctly predicting the class of an image as an indicator of image difficulty. In Figure 7b, a histogram is shown, plotting the number of many images (y-axis) for which only the given number of patches (x-axis) correctly predicted the class. Example images are shown below, ranging from hard on the left, to easy on the right; hard cases can be due to very easily-confused classes or to pose-estimation failure.

5. Conclusion

Pose variation constitutes a major challenge in fine-grained recognition, one which recent methods fail to effectively address. This paper introduces a unified object representation built upon pose-aligned patches instead of image-aligned regions – this representation disentangles the intrinsic appearance of an object from confounding influences such as pose variation. Our proposed algorithm attains state-of-the-art performance on two key fine-grained datasets, suggesting the critical importance of disentangling pose and appearance in fine-grained recognition.

Acknowledgements This work was supported by the National Science Foundation under Grant No. IIS1651832. We gratefully acknowledge the support of NVIDIA Corporation for their donation of multiple GPUs that were used in this research.

References

[1] T. Berg and P. N. Belhumeur. POOF: Part-Based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation. In *CVPR*, 2013. 3

[2] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014. 1

[3] S. Branson, G. Van Horn, P. Perona, and S. Belongie. Bird Species Recognition Using Pose Normalized Deep Convolutional Nets. In *BMVC*, 2014. 3

[4] T. Cohen and M. Welling. Group equivariant convolutional networks. In *ICML*, 2016. 2

[5] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 2

[6] T. S. Cohen and M. Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016. 2

[7] Cornell Lab of Ornithology. Merlin Bird ID App. http://merlin.allaboutbirds.org/. Accessed: 2018-07-03. 1

[8] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, 2018. 3

[9] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie. Kernel Pooling for Convolutional Neural Networks. In *CVPR*, 2017. 2, 3, 7

[10] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 3

[11] A. Dubey, O. Gupta, P. Guo, R. Raskar, R. Farrell, and N. Naik. Pairwise Confusion for Fine-Grained Visual Classification. In *ECCV*, 2018. 7

[12] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan. 2015. 3

[13] R. Farrell, O. Oza, V. I. Morariu, N. Zhang, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 3

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 9 2010. 4

[15] J. Fu, H. Zheng, and T. Mei. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition. In *CVPR*, 2017. 2, 3

[16] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact Bilinear Pooling. In *CVPR*, 2016. 2, 3

[17] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars. Fine-Grained Categorization by Alignments. In *ICCV*, 2013. 3

[18] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Local Alignments for Fine-Grained Categorization. *IJCV*, 111(2):191–212, 2015. 3

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014. 3

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 6, 7

[21] S. Huang, Z. Xu, D. Tao, and Y. Zhang. Part-Stacked CNN for Fine-Grained Visual Categorization. In *CVPR*, 2016. 3, 5, 6, 7

[22] S. Huang, Z. Xu, D. Tao, and Y. Zhang. Part-stacked cnn for fine-grained visual categorization. In *CVPR*, 2016. 3

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *NIPS*, 2015. 3, 7

[24] S. Kong and C. C. Fowlkes. Low-rank Bilinear Pooling for Fine-Grained Classification. In *CVPR*, 2017. 2, 3, 7

[25] J. Krause, T. Gebru, J. Deng, L. J. Li, and L. Fei-Fei. Learning Features and Parts for Fine-Grained Recognition. In *ICPR*, 2014. 3

[26] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-Grained Recognition without Part Annotations. In *CVPR*, 2015. 7

[27] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The Unreasonable Effectiveness of Noisy Data for Fine-Grained Recognition. In *ECCV*, 2016. 3, 6

[28] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012. 1

[29] M. Lam, B. Mahasseni, and S. Todorovic. Fine-Grained Recognition as HSnet Search for Informative Image Parts. In *CVPR*, 2017. 2, 3, 6, 7

[30] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, 2015. 3

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 3

[32] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN Models for Fine-Grained Visual Recognition. In *ICCV*, 2015. 2, 3, 7

[33] T. Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear Convolutional Neural Networks for Fine-grained Visual Recognition. *PAMI*, 2018. 7

[34] J. Liu, A. Kanazawa, D. W. Jacobs, and P. N. Belhumeur. Dog Breed Classification Using Part Localization. In *ECCV*, 2012. 3

[35] X. Liu, J. Wang, S. Wen, E. Ding, and Y. Lin. Localizing by Describing: Attribute-Guided Attention Localization for Fine-Grained Recognition. In *AAAI*, 2017. 2, 3

[36] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia. Rotation equivariant vector field networks. *ArXiv e-prints*, 2016. 2

[37] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 7

[38] D. R. Reddy. *Speech Understanding Systems. Summary of Results of the Five-Year Research Effort*. Carnegie-Mellon University, 1977. 6

[39] A. Ruderman, N. Rabinowitz, A. S. Morcos, and D. Zoran. Learned deformation stability in convolutional neural networks. *arXiv preprint arXiv:1804.04438*, 2018. 2

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 12 2015. 3

[41] P. Sermanet, A. Frome, and E. Real. Attention for Fine-Grained Categorization. In *ICLR Workshops*, 2015. 2, 3

[42] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. *CoRR*, abs/1701.0, 2017. 5, 6

[43] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 4

[44] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 4

[45] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a Bird Recognition App and Large Scale Dataset With Citizen Scientists: The Fine Print in Fine-Grained Dataset Collection. In *CVPR*, 2015. 1, 2, 5

[46] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011. 1, 2, 5

[47] M. Weiler, F. A. Hamprecht, and M. Storath. Learning steerable filters for rotation equivariant CNNs. In *CVPR*, 2018. 2

[48] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-Grained Image Classification. In *CVPR*, 2015. 2, 3

[49] Z. Xu, S. Huang, Y. Zhang, and D. Tao. Augmenting Strong Supervision Using Web Data for Fine-Grained Categorization. In *ICCV*, 2015. 7

[50] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, 2012. 3

[51] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011. 3

[52] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-Based R-CNNs for Fine-Grained Category Detection. In *ECCV*, 2014. 3, 4, 7

[53] N. Zhang, R. Farrell, and T. Darrell. Pose pooling kernels for sub-category recognition. In *CVPR*, 2012. 3

[54] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable Part Descriptors for Fine-Grained Recognition and Attribute Prediction. In *ICCV*, 2013. 3

[55] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell. Fine-grained pose prediction, normalization, and recognition. *ICLR Workshops*, 2016. 3, 5, 6

[56] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking Deep Filter Responses for Fine-Grained Image Recognition. In *CVPR*, 2016. 7

[57] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan. Diversified Visual Attention Networks for Fine-Grained Object Classification. *IEEE Transactions on Multimedia*, 19(6):1245–1256, 6 2017. 2, 3

[58] H. Zheng, J. Fu, T. Mei, and J. Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *ICCV*, 2017. 3