Part Localization using Multi-Proposal Consensus for Fine-Grained Categorization

Kevin J. Shih
kjshih2@illinois.edu
Arun Mallya
amallya2@illinois.edu
Saurabh Singh
ss1@illinois.edu
Derek Hoiem
dhoiem@illinois.edu

University of Illinois
Urbana-Champaign
IL, US

Abstract

We present a simple deep learning framework to simultaneously predict keypoint locations and their respective visibilities and use those to achieve state-of-the-art performance for fine-grained classification. We show that by conditioning the predictions on object proposals with sufficient image support, our method can do well without complicated spatial reasoning. Instead, inference methods with robustness to outliers, yield state-of-the-art for keypoint localization. We demonstrate the effectiveness of our accurate keypoint localization and visibility prediction on the fine-grained bird recognition task with and without ground truth bird bounding boxes, and outperform existing state-of-the-art methods by over 2%.

1 Introduction

Fine-grained image categorization is the task of accurately separating categories where the distinguishing features may be as minute as a different fur pattern, shorter horns, or a smaller beak. The widely accepted and popular approach of dealing with such a task is intuitive: align analogous regions and hone in on where you expect the differences to be. The analogous regions are usually defined by keypoints. Therefore, to perform well one would require not only accurate object-level localization, but also part and/or keypoint localization.

For keypoint localization, the most common approach is to learn a set of keypoints detectors to model appearance and an associated spatial model [5, 22, 23, 31] to capture their spatial relations. Keypoint detectors generate a set of candidates and a spatial model is used to infer the most likely configuration. Keypoint detectors typically model local appearance and thus an approach has to rely on expressive spatial models to capture long range dependencies. Alternatively, the keypoint detectors could condition their predictions on larger spatial support and jointly predict several keypoints [3], then the need for expressive spatial models could be eliminated leading to simpler models.

For effective fine-grained category detection, the keypoint localization method must have high accuracy, low false positive rates, and low false negative rates. Missed or poorly localized predictions make it impossible to extract the relevant features for the task at hand. If a keypoint is falsely determined to be present within a region, it is hard to guarantee that it will appear at a reasonable location. In the case of localizing keypoint-defined regions of an image, such as head or torso of a bird, a single outlier in the keypoint predictions can significantly distort the predicted area. This specific case is noteworthy as several of the current best-performing methods on the CUB 200-2011 birds dataset [29] rely on deep-network based features extracted from localized part regions [2, 5, 22, 31].

In this work, we tackle the problem of learning a keypoint localization model that relies on larger spatial support to jointly localize several keypoints and predict their respective visibilities. Leveraging recent developments in Convolutional Neural Networks (CNNs), we introduce a framework that outperforms the state-of-the-art on the CUB dataset. Further, while CNN-based methods suffer from a loss of image resolution due to the fixed-sized inputs of the networks, we introduce a simple sampling scheme that allows us to work around the issue without the need to train cascades of coarse-to-fine localization networks [26, 27]. Finally, we test our predicted keypoints on the fine-grained recognition task. Our keypoint predictions are able to significantly boost the performance of current top-performing methods on the CUB dataset. Our major contributions include:

1. State-of-the-art keypoint and region (head, torso, body) localization with visibility prediction using a single neural network based on the AlexNet [20] architecture.
2. A sampling scheme to significantly improve keypoint prediction performance without the use of cascades of coarse-to-fine localization networks [26, 27].
3. Improvement of the state-of-the-art performance of [31] on the CUB classification task by using our predicted keypoints with significant gains when the groundtruth bird bounding box is not provided during test time.

2 Related Work

Fine Grained Recognition: Prior work focuses on localizing informative parts of objects and then extracting features from them for classification. Using pairs of localized keypoints, Berg *et al*. [2] learn a set of highly discriminative features for fine-grained classification. Farrell *et al*. [13] and Branson *et al*. [5] use pose normalized representations of birds and their regions (head, torso, entire bird) followed by feature extraction for classification. Liu *et al*. [22] extend the exemplar based model of [1] with pose information for keypoint localization and subsequent classification of birds. Based on the very successful framework of the RCNN [16], Zhang *et al*. [31] perform bird classification using three localized bird regions: head, torso, and full body.

The above mentioned methods are highly dependent on accurate keypoint and bird region localization. In fact, [2, 13] rely on the groundtruth bird bounding box at test time to localize keypoints and to perform classification. Our method overcomes this bottleneck of localization and we demonstrate state-of-the-art classification performance using the framework of [31] along with our localized regions.

Object Region Proposals: Region proposals combined with deep network systems are an efficient solution for finding objects in an image. Recent works use region proposals as initial object candidates to either reduce their search space [17, 25] or to refine their localization

[14]. Instead of exhaustively sliding a window feature extractor on an image at all locations, scales, and aspect ratios, region proposal methods are used to quickly identify a smaller and manageable set of image regions which have high recall for objects present in the image. The time saved enables the use of more expensive feature extraction and processing. Popular region proposal methods include [6, 8, 11, 28, 32]. In our work, we use Edge Boxes [32] for its fast computational speed and its effective scoring method that allows us to further reduce the number of candidates needed test time.

Pose Detection & Regression with Deep Networks: Our method for keypoint localization mainly draws inspiration from the use of regression in networks in the MultiBox approach by Erhan *et al*. [12]. The authors train a deep network which regresses a small number of bounding boxes (~ 100) as object bounding box proposals, along with a confidence value for each bounding box.

Regression for localization of keypoints has previously been explored by Toshev *et al*. [27]. They use a cascade of deep network based regressors for human pose estimation to refine the keypoint predictions. At each stage, the network uses a region around the previous prediction to acquire higher resolution inputs and solve the fixed-resolution network input issue. In contrast, our work relies on multiple regions sampled with Edge Boxes from the image and simultaneously predicts all keypoints. Varying sized regions provide varying resolution and context, and we achieve more robust predictions from multiple regions with statistical outlier removal.

One of the closest works to ours on the CUB dataset is that of Liu *et al*. [22, 23]. They achieve remarkable performance on both keypoint localization and visibility prediction using ensembles of pose exemplars and part-pair detectors. We compare our performance with theirs using metrics defined in their work.

In contemporary works, the Deep LAC model [21] bridges a localization regression network and classification network to train simultaneously to perform on similar tasks to our own. While their setup is very similar to our own, they directly target the localization of entire head and torso boxes whereas we target the keypoints that define said boxes. We include their accuracies for comparison in the localization and recognition experiments.

3 Method

We design our model to simultaneously predict keypoint locations and their visibilities for a given image patch. To share the information across categories, our model is trained in a category agnostic manner. At test time, we efficiently sample each image with Edge Boxes, make predictions from each Edge Box, and reach a consensus by thresholding for visibility and reporting the medoid.

3.1 Training Convolutional Neural Networks for Keypoint Regression

Our network is based on AlexNet [20], but modified to simultaneously predict all keypoint locations and their visibilities for any given image patch. AlexNet is an architecture with 5 convolutional layers and 3 fully connected layers. Henceforth, we refer to the 3 fully connected layers as fc6, fc7, and fc8. We replace the final fc8 layer with two separate output layers for keypoint localization and visibility respectively. Our network is trained on Edge Box [32] crops extracted from each image and is initialized with a pre-trained AlexNet [20] trained on the ImageNet [9] dataset. Each Edge Box is warped to 227×227 pixels before it

can be fed through the network. We apply padding to each Edge Box such that the warped 227×227 pixel crop has 16 pixels of buffer in each direction.

Given N keypoints of interest, we train a network to output an N dimensional vector $\hat{v}$ and a $2N$ dimensional vector $\hat{l}$ corresponding to the visibility and location estimates of each of the keypoints k_i, $i \in \{1,N\}$, respectively. The corresponding groundtruth targets during training are v and l. We define v to consist of indicator variables $v_i \in \{0,1\}$ such that $v_i = 1$ if keypoint k_i is visible in the given Edge Box image before padding is performed, and 0 otherwise. The groundtruth location vector l is of length $2N$ and consists of pairs (l_{x_i}, l_{y_i}) which are the normalized $(\tilde{x}, \tilde{y})$ coordinates of keypoint k_i with respect to the un-padded Edge Box image. Output predicted from the network, $\hat{v}_i \in [0,1]$, acts as a measure of confidence of keypoint visibility, and 2D locations predicted by the network are denoted by $\hat{l}_i$.

We use the *Caffe* framework [19] for training our deep networks. To train a network optimized for both tasks simultaneously, we define our losses as follows:

$$\mathcal{L}_{vis} = ||v - \hat{v}||_2^2 \quad \text{and} \quad \mathcal{L}_{loc} = \sum_{i=1}^{N} v_i \cdot \left[(l_{x_i} - \hat{l}_{x_i})^2 + (l_{y_i} - \hat{l}_{y_i})^2 \right] \tag{1}$$

$$\mathcal{L}_{net} = \mathcal{L}_{vis} + \mathcal{L}_{loc} \tag{2}$$

The visibility loss $\mathcal{L}_{vis}$ is the squared Euclidean distance between the ground truth visibility label vector v, and the predicted visibility vector $\hat{v}$. The values in our $\hat{v}$'s always lie between 0 and 1 as they are obtained after squashing network outputs with a sigmoid function. The keypoint localization loss $\mathcal{L}_{loc}$ is a modified Euclidean loss, in which we set the loss between the prediction and the target to be 0 if $v_i = 0$ i.e. if the keypoint k_i is absent in the given image. The final training loss ($\mathcal{L}_{net}$) is given by the sum of the two losses.

To construct our training set for predicting keypoint visibility and locations, we extract up to 3000 Edge Boxes per image. To train a robust predictor, we need a collection of training images with high variability in which different subsets of keypoints are visible. We generate examples that satisfy this criteria by retaining a subset of Edge Boxes which have at least 50% of their area contained inside the groundtruth bounding box and have at least 20% intersection over union overlap (IOU) with the groundtruth bounding box. We also included up to 50 random boxes per image from outside the bounding box as negative background examples. We augment our dataset with left/right flips. After flipping, appropriate changes were applied to the label vectors. This consisted of swapping orientation-sensitive keypoints such as "left eye" and "left wing" with "right eye" and "right wing", and updating their respective coordinates and visibility indicators. We first train our model on 25 images per class and tune algorithmic and learning rate parameters on a held-out validation set comprising the remaining 4-5 images per class. Finally, we re-train using the entire training set before running our model on the test set.

3.2 Combining Multiple Keypoint Predictions

Our algorithm for dealing with predictions from multiple Edge Boxes at test time is illustrated in Fig. 1. Due to the variance from making predictions from multiple unique subcrops of the image, we need to form a consensus from the multiple predictions. In our experiments, we found that after removing predictions with low visibility confidences, the remaining predictions had a peaky distribution around the ground truth. We use medoid as a robust estimator for this peak and found it to be effective in most cases (Fig. 5). For the task of localizing part regions around keypoints (described in section 3.3), we found on our train/val split that

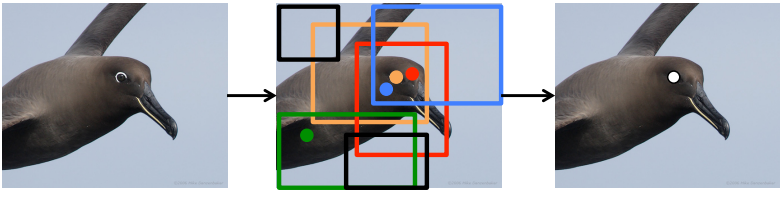


Figure 1: The pipeline of our keypoint localization process: Given an input image, we extract multiple edge boxes. Using each edge box, we make predictions for the location of each of the 15 keypoints, along with their visibility confidences. We then find the best predicted location by performing confidence thresholding and finding the medoid. The process is illustrated for the right eye keypoint (Black edge boxes without associated dots make predictions with confidences below the set threshold, and green is an outlier with a high confidence score).

we achieved better localization performance if we kept a set of good predictions (referred to as *inliers*) instead of using only the medoid. We now describe our procedure for obtaining a tight set of inliers and our choice of parameters. For the keypoint prediction task, we only use the visibility thresholds and report the medoid.

Case 1: Ground Truth Object Box Given: We first describe our method in the case that the ground truth object boxes are given. Using the ground truth object box, we retain the generated Edge Boxes that are mostly contained within and have an IOU of at least 0.2. This results in roughly 50-200 remaining Edge Box subcrops per image. Each subcrop is then independently fed through our keypoint prediction network, returning a set of normalized keypoint predictions and visibilities.

Because each subcrop is expected to cover less than the whole object and contain only a subset of the keypoint predictions, we drop any prediction if its corresponding visibility is below 0.6. Because we make use of multiple overlapping subcrops, it is very likely that at least one of them will lead to a prediction with a sufficiently high visibility score, thereby allowing us to be much more aggressive with the false positive filtering.

Given multiple remaining keypoint predictions per keypoint with sufficiently high visibility scores, we then proceed to remove outliers. To do so, we threshold on a modified Z-score based on a description given by Iglewicz and Hoaglin [13]. The modified Z-score is one that is re-defined using medoids and medians in place of means, as the former estimates are more robust to outliers.

Let p_i where $i = 1, \cdots, M$ be the set of M surviving un-normalized keypoint predictions (for a given keypoint) in (x, y) image coordinates. We first define $\bar{p}$ to be the medoid prediction such that:

$$\bar{p} = \underset{p_j}{\operatorname{argmin}} \sum_{i=1}^{M} ||p_j - p_i||_2, \; j \in \{1, ..., M\} \tag{3}$$

In other words, $\bar{p}$ is the prediction such that its Euclidean distance from all other predictions for that keypoint is minimal. While this optimization is costly at a large scale, we typically deal with only 10-20 predictions at a time after thresholding for visibility scores. To compute the modified Z-score we use:

$$Z_i = \frac{\lambda \; ||p_i - \bar{p}||_2}{\text{median } (||p_i - \bar{p}||_2)}, \; i \in \{1, ..., M\} \tag{4}$$

Here, the denominator is the median absolute deviation, or simply the median distance

from the medoid $\bar{p}$. We use the recommended $\lambda = 0.6745$. The above procedure is separately computed for all 15 sets of keypoint prediction candidates. Finally, we drop any keypoint prediction with $Z_i > 0.35$, a threshold that was experimentally determined on the held-out set.

Case 2: Ground Truth Object Box Not Given: Our ground truth object box not given scenario requires little change from the above case. Using the Edge Box ranking, we found that most of our "good" Edge Boxes fell within the top 600 Edge Boxes per image, saving us a lot of computation. Tuning parameters on our train/val split, we found that an even more aggressive visibility threshold of 0.94 and a Z-score threshold of 0.3 gave the best results.

Medoid-Shift: While the simple Z-score thresholding combined with the medoid achieves excellent results, as we will demonstrate in the results section, we were able to further improve our results by using medoid-shifts [24]. We use the medoid of the largest output cluster from the algorithm instead of the medoid computed over all the visibility-filtered predictions.

3.3 Bird Classification

We verify the effectiveness of our localized parts by implementing the simple classification framework as described in [31]. Using the keypoints, three regions are identified from each bird: head, torso, and whole body. The head is defined as the tightest box surround the beak, crown, forehead, eyes, nape, and throat. Similarly, the torso is the box around the back, breast, wings, tail, throat, belly, and legs. The whole body bounding box is the object bounding box provided in the annotations. To perform classification, fc6 features are extracted from these localized regions, concatenated into a feature vector of length 4096×3, and used for 200-way linear 1-vs-all SVM classification.

To handle the case when ground truth bounding box is not given at test time, we use an overlap heuristic based on the predicted head and torso boxes. We first start by finding the tightest box around the predicted head and torso boxes. While this initial box will do well for birds in their canonical poses, it will result in an undersized box in many cases because the keypoints do not always capture the full extent of the bird. We then assume that there exists an Edge Box with a high edge score that better captures the whole bird. To let the box expand to capture more of the object, we first identify the Edge Boxes such that the tightest box is at least 90% contained within and has at least 50% IOU overlap. The final whole body bounding box is the Edge Box that passes both criteria that also has the highest Edge Box object score. If no Edge Box passes the overlap test, we fall back to the starting tightest box.

4 Experiments and Results

We evaluate our prediction model on the challenging Caltech-UCSD Birds dataset [29]. This dataset contains 200 bird categories with 15 keypoint location and visibility labels for each of the total of 11788 images. We first evaluate our keypoint localization and visibility predictions against other top-performing methods. Next, we demonstrate their effectiveness in the fine-grained categorization task by significantly improving state-of-the-art through better localization.

PCP	Ba	Bk	Be	Br	Cr	Fh	Ey	Le	Wi	Na	Ta	Th	Total
[23]	62.1	49.0	69.0	67.0	72.9	58.5	55.7	40.7	71.6	70.8	40.2	70.8	59.7
[23]	64.5	**61.2**	71.7	70.5	76.8	**72.0**	**70.0**	45.0	74.4	**79.3**	46.2	**80.0**	66.7
Ours	**74.9**	51.8	**81.8**	**77.8**	**77.7**	67.5	61.3	**52.9**	**81.3**	76.1	**59.2**	78.7	**69.1**

Table 2: Comparison of per-part PCP with Liu *et al*. [22, 23]. The abbreviated part names from left to right stand for back, beak, belly, breast, crown, forehead, eye, leg, wing, nape, tail, and throat.

4.1 Keypoint Localization and Visibility Prediction

Table 1 reports our keypoint and visibility performance without using any ground truth bounding box information. Our medoid method reports the medoid of predictions above a visibility threshold, as seen in the red star in Fig. 5. Our "mdshift" method reports the new medoid computed using medoid-shift, which is the blue circle in Fig. 5. We used the evaluation code provided by the authors of [22] to measure our performance using the metrics defined in their work. In short, PCP (Percent Correct Parts) is the percentage of keypoints localized within 1.5 times the annotator standard deviation. We received the pre-computed standard deviatons and evaluation code from the authors of [22] to avoid any discrepancies during evaluation. AE (Average Error) is the mean euclidean prediction error, capped at 5 pixels, computed across examples where a prediction was made and a ground truth location exists. FVR and FIR refer to False Visibility Rate and False Invisibility Rate respectively. The additional methods for comparison are the same as listed in their paper.

Compared to the top-performing methods that also predict visibility, our method achieves the best numbers in three out of four metrics. Our PCP and AE metrics outperform other methods in the table, with our medoid-shift variant performing slightly better. Our FIR is higher because we are using the visibility threshold tuned on the part-localization task. A slightly lowered threshold would lower the FIR and raise the FVR without significantly affecting the PCP.

Method	PCP	AE	FVR	FIR
Poselets [4]	24.47	2.89	47.9	17.15
Consensus [1]	48.70	2.13	43.9	6.72
Exemplar [23]	59.74	1.80	28.48	**4.52**
Ours (medoid)	68.7	1.4	**17.1**	5.2
Ours (mdshift)	**69.1**	**1.39**	**17.1**	5.2
Human [23]	84.72	1.00	20.72	6.03

Table 1: Localization and Visibility Prediction Performance of various methods without using the ground truth Bounding Box

The highest reported PCP is 66.7% due to Liu *et al*. [23], which also predicts visibilities but did not report them. We compare against their PCP in Table 2. Because our method differs significantly from theirs, we outperform them in only 7 of the listed part categories despite having a better overall PCP, suggesting further improvements by targeting the differences in our models' behaviors.

4.2 Head and Torso Localization

We evaluate our ability to localize keypoint-defined part-regions on the test set. Example predictions can be seen in Fig. 3. In Table 3, we compare our part-localization accuracies with other methods. We also compare with the simple case where we make predictions by feeding just the ground truth boxes through the network (single GT Bbox). (We also tried re-training a network on just GT bounding boxes for this case, but it didn't perform as well.) Unlike the keypoint prediction task, we retain a set of inliers after Z-score thresholding to determine the extent of each part box. This was determined to perform best on our validation set. The reported metrics are the percentage of heads, torsos, and whole body boxes that

	Method	Head	Torso	Whole Body
GT Bbox	Part-Based RCNN [31]	68.2	79.8	N/A
	Deep LAC [21]	74.0	**96.0**	N/A
	Ours (single GT bbox)	75.6	90.2	N/A
	Ours (multiple)	88.8	93.9	N/A
	Ours (multiple, mdshift)	**88.9**	94.3	N/A
No GT Bbox	Part-Based RCNN [31]	61.4	70.7	**88.3**
	Exemplar [22]	79.9	78.3	N/A
	Ours (multiple)	87.8	**89.0**	84.5
	Ours (multiple, mdshift)	**88.0**	88.7	84.6

Table 3: Comparison of Part Localization Performance: Our method based on keypoint prediction from Edge Boxes shows significant improvement over previous work.

were correctly localized with a >50% IOU. The ability to perform competitively on this task should correlate with a high PCP and low FVR and FIR.

The results in Table 3 demonstrate that our keypoint predictions are useful in generating accurate part boxes. Our lower performing single GT Bbox method suggests that our use of multiple predictions from Edge Boxes allows for more accurate predictions. Further, we also computed head and torso boxes using the keypoint predictions from Liu *et al*. [22] as shown in the "Exemplar" row. Based on their accuracy, their boxes should also be able to improve the results of [31].

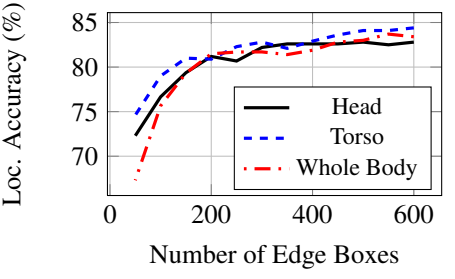


Figure 2: Localization performance on our validation set while varying the number of top Edge Boxes used. We only rely on the Edge Boxes as a means of efficient sampling of the image, so our performance is barely affected by the loss of well-localized boxes.

In Fig. 2, we also look at how our localization ability is affected by the number of top Edge Boxes sampled from the image. As we previously noted, the Edge Box edge scoring is effective enough that most of the sufficiently well localized boxes we used in the ground-truth bounding box given case fell within the top 600. However, as our model predicts individual keypoints and visibilities, it does not need a well localized box at test time at all. It merely needs a set of Edge Boxes that, combined, provide enough coverage over the actual bird for it to predict keypoint locations and visibilities. As such, our model is able to continue to localize over 70% of the head and torso boxes with at least 50% IOU as the number of Edge Boxes drops to 50. While the 50% IOU recall of Edge Boxes for head and torsos on the validation set were 66.36% and 95.12% at 600 boxes and 17.28% and 62.20% respectively at 50 boxes, we demonstrate that we were able to localize these parts with higher accuracy than would have been achievable had we used an RCNN-based approach and tried to map Edge Boxes to the parts.

4.3 Fine-Grained Classification

We now test our part-predictions in a fine-grained classification setting. These results are shown in the right half of Fig. 4. To do this, we train three networks to re-implement the three-part framework as Zhang *et al*. [31] as described in section 3.3. The oracle performance refers to the classification assuming ground truth keypoints at test time. While Zhang *et al*. [31] reports an oracle accuracy of 82.0%, we compare with the highest we were able to achieve with our implementation: 81.5%. This is likely due to minor differences in network

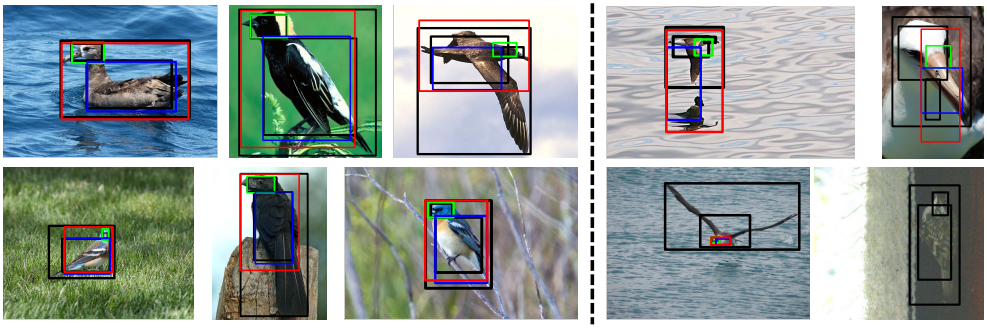
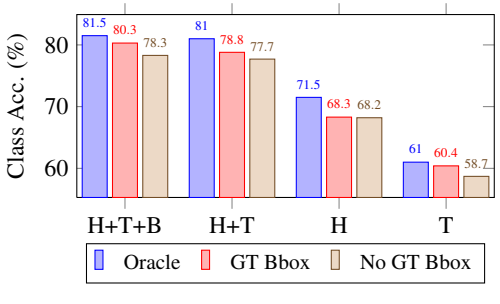


Figure 3: Examples of good (left) and failed (right) localization results: The ground truth boxes are in solid black. The head, torso, and whole body boxes are in green, blue and red respectively. The head is correctly localized in most of the above examples. In the top row middle example, even though the whole body box IOU is low, most of the missed area is actually background due to the bird extending its wings. In the bad examples, we show that we mostly fail in rare close-ups and when there are multiple instances.

training parameters. We also tried both fc6 and fc7 features and found that fc6 performed a little better. Although Zhang *et al*. [31] and Branson *et al*. [5] noted that their drops in accuracy from using ground truth parts to predicted parts were surprisingly small, our relative improvements suggest that it is still worthwhile to focus on better localization. Further, we perform at least as well as the contemporary Deep LAC model [21], likely due to our better localization of the more discriminative head regions.

In the left half of Fig. 4, we show how our accuracy is affected from the ground truth keypoint ideal case (Oracle) to the use of predicted keypoints (GT Bbox), and finally with the GT Bbox removed (No GT Bbox). Unsurprisingly, the better localization at test time allows for a significantly smaller drop as annotations are removed.

The same plot also shows an ablation test of individual parts. It appears that the bulk of our performance comes from discriminating localized bird heads. This is also supported by [5] which observed that of their learned poses, the one that corresponded to the head was the most discriminative. This suggests that most of our improvement over our base method of [31] comes from significantly improving our head part localization (shown in Table 3).



	Method	Acc.
Oracle	Oracle Parts + SVM	**81.5**
GT Bbox	DPD [30]	51.0
	Symbiotic [2]	59.4
	Alignment [15]	62.7
	DeCAF [10]	65.0
	POOF [2]	56.8
	Part-Based RCNN [31]	76.4
	Deep LAC [21]	80.3
	Ours (mult, medoid)	80.3
	Ours (mult, mdshft)	**80.3**
No GT Bbox	Pose Norm [5]	75.7
	Part-Based RCNN [31]	73.9
	Ours (mult, medoid)	78.2
	Ours (mult, mdshft)	**78.3**

Figure 4: On the left we show a comparison of classification accuracies obtained using combinations of parts localized under different conditions (H: Head, T: Torso, B: Whole Body). On the right, we compare our classification accuracy with other works.

Figure 5: Qualitative results for a subset of the keypoints. Predictions for most of the images cluster tightly. Therefore, simple prediction methods such as medoids work well. Medoid shift adds to the robustness, leading to further improvements (second last column). Primary failure mode is when visibility thresholding fails to rule out clusters of false positives (bottom right).

5 Conclusion

We presented a method for obtaining state-of-the-art keypoint predictions on the CUB dataset. We demonstrated that conditioning the predictions on multiple object proposals for sufficient image support can reliably improve localization predictions without using a cascade of coarse-to-fine networks. We tackle the problem of fixed-size inputs when using neural networks by sampling predictions from several boxes and determining the "peak" of the predictions with medoids. We then use our predictions to significantly improve state-of-the-art methods on the fine-grained classification task on the CUB dataset. In future work, we intend to apply this method to human datasets as well as to combine it with more sophisticated inference methods to deal with multiple instances.

6 Acknowledgements

This work is supported by NSF CAREER awards 1053768 and 1228082, NSF Award IIS-1029035, ONR MURI Award N00014-10-10934 and the Sloan Fellowship. In addition, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPUs used for this research.

References

[1] Peter N Belhumeur, David W Jacobs, David Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

[2] Thomas Berg and Peter N Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[4] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010.

[5] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014.

[6] Joao Carreira and Cristian Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3241–3248. IEEE, 2010.

[7] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[8] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3286–3293. IEEE, 2014.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[11] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):222–234, Feb 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.122.

[12] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *CVPR*. IEEE, 2014.

[13] Ryan Farrell, Om Oza, Ning Zhang, Vlad I Morariu, Trevor Darrell, and Larry S Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[14] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3294–3301. IEEE, 2013.

[15] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[18] Boris Iglewicz and David Caster Hoaglin. *How to detect and handle outliers*, volume 16. Asq Press, 1993.

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1666–1674, 2015.

[22] Jiongxin Liu and Peter N Belhumeur. Bird part localization using exemplar-based models with enforced pose and subcategory consistency. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2520–2527. IEEE, 2013.

[23] Jiongxin Liu, Yinxiao Li, and Peter N Belhumeur. Part-pair representation for part localization. In *Computer Vision–ECCV 2014*, pages 456–471. Springer, 2014.

[24] Yaser Ajmal Sheikh, Erum A Khan, and Takeo Kanade. Mode-seeking by medoidshifts. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[25] Kevin Shih, Ian Endres, and Derek Hoiem. Learning discriminative collections of part detectors for object recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (1):1–1.

[26] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3476–3483. IEEE, 2013.

[27] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653–1660. IEEE, 2014.

[28] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104 (2):154–171, 2013.

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

[30] Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013.

[31] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[32] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.