Examining Ambiguity in Human Pointing with Computer Vision

Arjun Mani
CS Dept, Princeton University
arjuns@princeton.edu

Abstract

Pointing is a fundamental communicative gesture in humans that can convey a great deal of complexity. It can involve both multi-level reasoning and ambiguity of reference. Computer vision systems today for tasks like semantic segmentation and object detection cannot extend to tasks that are inherently ambiguous and involve inter-annotator disagreement. Thus, pointing represents a new frontier in computer vision. We describe the object-part inference task, which entails identifying whether a point refers to a specific part of an object or the object as a whole. We use a joint supervision approach to train a deep learning model, and show the feasibility of this model for this task. In particular, it achieves 68.5% accuracy in distinguishing between ”whole object” and ”part” queries. We then examine the ambiguity of the task by first training the model with soft labels, which improves the accuracy on ambiguous points. We then train a model to explicitly predict points as ambiguous, which achieves reasonable results and for which we show the model uncertainty can be roughly tuned. Finally, we discuss future directions for ambiguity modeling, such as Bayesian inference.

1. Introduction

Pointing is an important and fundamental communicative gesture in humans. It is one of the first gestures that develops in infants and plays a central role in language acquisition [9]. This suggests that modeling pointing might be a promising approach to improve current computer vision systems, and to encourage potential AI-human collaboration in the visual domain.

The task of pointing represents a new frontier in computer vision. Deep learning systems have achieved great success in tasks like image classification, semantic segmentation, and object detection. These tasks, however, do not involve both multi-level reasoning and ambiguity. Pointing in the wild is a complex gesture that could convey a wide range and depth of meaning (e.g., look at what action that part of that object is doing). In addition, and our focus in this paper, pointing can be reasonably interpreted in different ways by humans themselves. Deep learning has so far succeeded at tasks that involve human agreement, but pointing by nature is ambiguous.

The operative question is what it would mean for a computer to succeed at the task of pointing. Given an image and a pointing gesture, the computer should be able to answer ”What’s that?” with the desired level of granularity. This task is difficult to model, but we can introduce an approximation of this task that retains the central features of this task.

1.1. The Object-Part Inference Task

The object-part inference task is the following: given an image I and a point (x, y) in the image, identify both the object that the point is on and whether if a human is pointing to that point, they are referring to a specific part of the object or the object as a whole. Figure 1 shows an example of two query points, one ”part” query and one ”whole object” query, as they will be referred to hereafter.

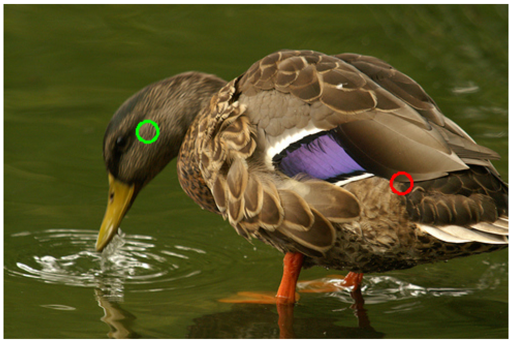


Figure 1: An example of a part query point (green) and a whole object query point (red).

This task is more complex than semantic segmentation, which involves identifying only the object is on, and better approximates the task of human pointing. When humans point to an object, they are often referring to a specific part: for example, pointing to the foot of a person likely refers to a specific part, while pointing to the torso might refer to the person as a whole. Crucially, humans can reasonably disagree on whether a point refers to a part or the whole object. Thus, the object-part inference task captures the complexity and the ambiguity of human pointing more effectively. Ir allows us to progress towards a more general understanding of pointing, while still being a feasible enough task to define and model practically.

1.2. Outline

In this paper, I present new results on the object-part inference task from my previous work [24]. I build deep learning models using a joint supervision framework to show the feasibility of the task on an expanded dataset. I then focus my paper on understanding and modeling the ambiguity in the task. I first transform the labels into soft probability distributions over the classes and evaluate this compared to a one-class approach. Then, I train models to specifically predict the existence of ambiguous points and present both qualitative and quantitative results. I end with a discussion of other possible approaches to modeling the ambiguity of the task.

2. Related Work

The task of pointing has not been studied in-depth. However, we review several research areas in image understanding as well as ambiguity.

2.1. Segmentation

Semantic segmentation is the task of annotating every pixel in the image with a label, rather than the image as a whole. The encoder-decoder architecture introduced by [23] has become the state-of-the-art deep learning model for semantic segmentation. This approach uses an image classification model like VGG-16 [30] to downsample the image followed by an upsampling. [23] used simple bilinear upsampling, while more complex approaches involve several upsampling steps [15, 3, 27].

Instance segmentation involves disambiguating different object classes as well as different instances of the same object class (e.g. multiple people in an image). The task is far less solved than semantic segmentation, but several approaches have been proposed [17, 25, 26]. The most notable is DeepMask [26], which learns semantic segmentation masks for image patches jointly with their likelihood to contain objects. Our model is based off the encoder-decoder architecture approach developed for semantic segmentation.

2.2. Saliency

Semantic segmentation generally does not involve inter-annotator disagreement, unlike the object-part inference task. Predicting image saliency, or which regions of the image draw more attention than others, is somewhat more subjective. Given an image, the goal is to output a pixel-wise saliency map indicating the saliency of each image region. Many methods use low-level features such as contrast and brightness to solve the task [7, 32]. However, deep learning has shown to be the state-of-the-art on this task. The general approach involves a CNN with convolutional layers at multiple scales and/or contexts [10, 21, 31]. It is possible that saliency is correlated with the object-part inference task in that part regions of an object may be more salient than whole object regions. However, my previous work [24] shows that this correlation is quite weak.

2.3. Ambiguity

A growing body of research has focused on modeling uncertainty in deep learning. This work is motivated by several real-world scenarios, such as medical diagnosis, where the uncertainty of a prediction is important to understand. An active area of research is Bayesian deep learning, which seeks a probability distribution over the set of weights in a model. The most popular method thus far is variational Bayesian inference, which in practice is equivalent to a dropout network with Monte-Carlo sampling [20, 13]. The model predicts an uncertainty value, which is learned in an unsupervised fashion from a modified loss function. This approach has shown to yield improved results on tasks like image classification and semantic segmentation [20]. Although we do not use this approach, we briefly discuss it in the conclusion.

Some alternative approaches to uncertainty have been proposed. These include predicting multiple hypotheses and learning probability distributions over the set of labels [28, 14]. In this work, we first evaluate the performance of soft labels for an ambiguous task. Second, we develop a model to explicitly predict ambiguous instances of the task, a novel contribution in relation to previous research.

2.4. Pointing

The task of human pointing has not been studied extensively in computer vision. There has been work in robotics on understanding and using pointing as a means of human-robot collaboration, but most focus on understanding the direction of pointing rather than intent [5, 2]. [19] creates a cost optimization model to make robot pointing more 'legible' to humans, but this model still focuses on understanding the object being pointed to, which is a superficial understanding of human pointing. Other research has focused on pointing as means of improving HCI interfaces [1, 16].

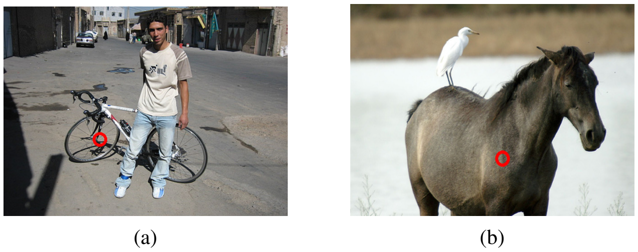
Points have been used as sparse supervisory signals in image classification and semantic segmentation [4, 22]. With time and budget constraints, labeling a few points per image has shown to be reasonably effective at semantic segmentation [4]. Hinthorn was the first to study pointing from a human perspective, and specifically the object-part inference task [18]. His contributions include a pilot dataset for the object-part inference task, extensive analysis of this dataset, and preliminary work on deep learning models. In this paper, I build off my previous work, which built deep learning models to demonstrate the feasibility of the task [24]. I train and evaluate these models on a significantly expanded dataset, and then focus on modeling the ambiguity in the task through the two different approaches described above.

3. Methods

3.1. Dataset and Tasks

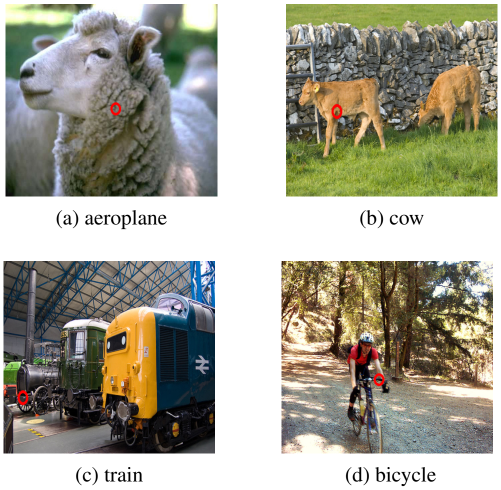
I use the Pascal Points Dataset [12], which is a dataset of human-labeled points based off the Pascal VOC dataset. Pascal VOC is a seminal dataset in computer vision that consists of 11,530 images and 20 object classes (bus, car, cat, etc.), and is popular for semantic segmentation [6]. One iteration of the dataset is Pascal Parts, which consists of pixel-wise annotations for each part of each object in the images.

The Pascal Points Dataset is built using Pascal Parts. It consists of $\sim$ 50000 points across over 8000 images, each point annotated with the corresponding object and whether it is referring to part or whole object. 15 of the 20 object classes are used. At least three human labelers annotate each point. Notably, approximately 30% of the points in the dataset are ambiguous, meaning that at least one annotator disagreed with the consensus. This indicates the inherent ambiguity of the task. Figure 2 shows some sample points in the dataset.


(a)

(b)

Figure 2: Examples of points in the dataset. (a) An unambiguous part point; (b) An unambiguous whole object point.


(a) aeroplane

(b) cow

(c) train

(d) bicycle

Figure 3: Examples of ambiguous points. The top two points are on ambiguous regions of the object. The bottom left is in the image background, and the bottom right is located on object boundaries.

3.1.1 Ambiguous Points

The high percentage of ambiguous points in the dataset underlines the inherent ambiguity of the object-part inference task. Figure 3 shows a selection of ambiguous points, which tend to fall into three general categories. The first is points on genuinely ambiguous regions of the object (e.g. neck), such as in the first image. The second is points that clearly refer to parts of objects that are far in the background of the image. Humans tend to often identify all points on these objects as 'whole object' points given the apparent distance of the object from the observer. The last category of ambiguous points are those where it is unclear which object the point is referring to, either due to occlusion or simply being located at the boundaries of objects. These points are very rare in the dataset.

Given this dataset, we can define the following two tasks:

1. Task 1: Given the ground truth object class, predict whether the point refers to a specific part or the whole object. This is a binary classification problem.

2. Task 2: Predict both the object class and the reference of the point. This is an N-way classification problem.

Since the focus of this paper is on ambiguity, we can also define a third task. Task 3 is the accuracy of the model on ambiguous points, i.e. points with inter-annotator disagreement. As we will see, the notion of this accuracy is defined differently for different models.

3.2. Model Design

We can frame object-part inference as an image-to-image problem for the purpose of designing models. For each image I in the dataset, there are k points in Pascal Points that are labeled. We can use a fully-convolutional network to output pixel-wise predictions for

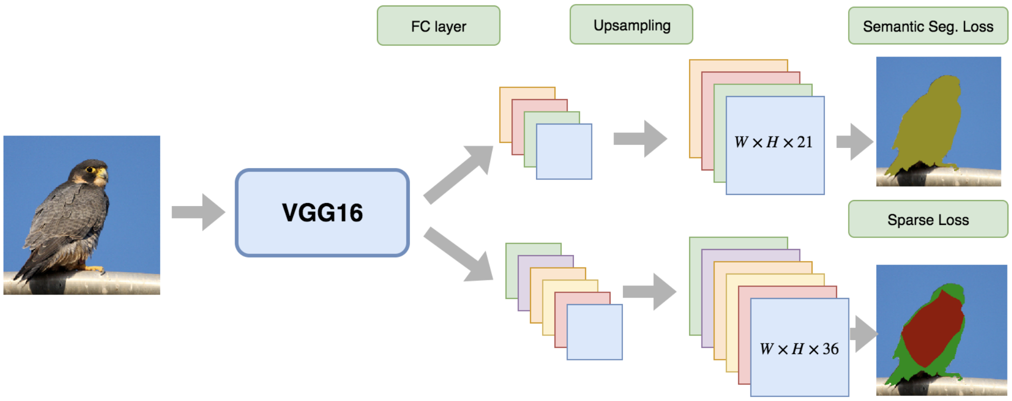


Figure 4: The model architecture. The top head computes the segmentation loss with respect to dense labels. The bottom head computes the object-part loss and is supervised on the points in Pascal Points. Both heads use cross-entropy loss. The dimension of the bottom head is 20 object classes + 15 part classes + background = 36.

Task 2, and then supervise over the k labeled pixels. Fully-convolutional networks are an extension of convolutional neural networks that involve downsampling via an image classifier followed by upsampling. We use the FCN-32s architecture developed by the original authors [23], consisting of VGG-16 followed by a bilinear upsampling. However, this approach yields low accuracy due to the sparsity of the labels. Our approach must take advantage of existing annotations to augment the ground truth provided to the model.

Thus, we use the joint supervision approach depicted in Figure 4. The model consists of two heads, which share weights until the final fully-connected layer. Each head then consists of a separate fully-connected layer and upsampling. For purposes of clarity, our model has 36 classes (20 object classes + 15 part classes + background). This is because only 15 objects are present in Pascal Points: for each of these 15, there is one class for ”whole object” and another for ”part”. The ”object-part head” of the model outputs a $W \times H \times 36$ prediction over all possible classes, which is then supervised over the labeled points in Pascal Points. The ”semantic head” of the model outputs a $W \times H \times 21$ prediction over the possible semantic segmentation classes, which is then supervised over the dense masks in the Pascal VOC dataset. Each head of the model is trained using cross-entropy loss. The final loss is the sum of the two individual losses, i.e. $L = L_{sparse} + L_{semantic}$.

The advantages of this approach are two-fold. First, it allows the model to approximately separate the task of disambiguating between objects and distinguishing between whole object and part, while still enabling each head of the model to learn from each other. Second and related, it enables efficient debugging of the model, since the issue can be isolated to a particular head of the model and by extension a particular sub-task. All the following models are built using this joint supervision approach.

3.3. Soft Labels

As mentioned, roughly 30% of the points in the dataset are ambiguous. In order to allow the model to deal better with ambiguity, we can use soft labels instead of one-hot encoding. A soft label is a probability distribution over the set of classes. Consider a toy problem with 2 classes and 5 human annotators per example. If four labeled an example 1 and one 0, then the one-hot label would be $[0\ 1]$ and the soft label would be $[0.2\ 0.8]$. Soft labels allows the model to better capture the uncertainty associated with training examples. We can train the model using soft labels for ambiguous points and keep the other aspects of training identical.

For a theoretical justification of why we can use cross-entropy, consider the same toy problem. The cross-entropy loss is $L = \sum_c y_c \log(p_c)$, where the sum is over the classes and p_c is the predicted probability of the c-th class. For the one-hot encoding, the loss would be $\log(p_1)$. However, for the soft label, the loss is $0.2 \log(p_0) + 0.8 \log(p_1)$. Taking the derivative and setting to 0, we have $\frac{0.2}{p_0} = \frac{0.8}{p_1}$, and thus the function is minimized at $p_0 = 0.2, p_1 = 0.8$. Thus, given soft labels, cross-entropy loss motivates the model to predict the correct probabilities of each class. We train a 36-way model using soft labels.

3.4. Predicting Ambiguous Points

Since ambiguity is a feature of the object-part inference task, we can train our model to explicitly predict ambiguous points. In this formulation, each point can be categorized into its object class and then as either whole object, part, or ambiguous. For the original 36-way model, this now becomes a 51-way model (+15 ambiguous classes). Since this is a large number of classes relative to the size of the dataset, we introduce the 9-way model.

The 20 object classes in Pascal VOC can be grouped into 4 ”super-classes”: person, animal, vehicle, and indoor. We can train the model to predict part/whole object for these 4 super-classes, for a total of 9 classes when including background. The advantage is that the model may potentially be able to perform object-part inference more successfully with a smaller number of classes, since object disambiguation is easier. This advantage is particular evident when we introduce an ambiguous category for each object class: the original model has 51 classes, while the super-class model has 13 classes. After training a 9-way model, we can then train a 13-way model to explicitly predict the existence of ambiguous points.

We can use a weighted cross-entropy loss to prioritize the model's performance on ambiguous points. The loss on the i-th example would read as $L_i = \alpha_i \sum_c y_c \log(p_c)$, where α_i is the weight of the i-th example in the loss. Intuitively, we penalize those examples more on which we prioritize correct classification. We experiment with weighting ambiguous points both higher and lower and measure how this both quantitatively and qualitatively affects performance.

3.5. Implementation Details

From my previous work [24], we significantly expand the dataset. The Pascal Points Dataset consists of 8,204 images. In the previous work, only these images were used with an 80/20 split for training and validation, resulting in 6,563 training images. However, the Berkeley version of the Pascal VOC dataset consists of 11,504 images with associated segmentation masks [12]. The semantic head of the joint supervision model can trained with all of these images, even those without annotations in Pascal Points. Thus, we incorporate all 11,504 images with a 90/10 training validation split, for a total of 10,353 images in the training set. This is more 1.5 times the size of the original training set.

The models were trained using stochastic gradient descent with momentum of 0.9 and a learning rate of either 10^{-3} or 10^{-4}. We experimented with adaptive learning rate methods like Adam but did not find a measurable increase in performance. The weights were initialized to the pretrained VGG-16 weights on the ImageNet classification task [29, 11]. The models were built in Keras, a high-level deep learning API based off of Tensorflow [8]. Custom losses and metrics were written to only supervise on labeled pixels for the object-part head of the model.

4. Results

4.1. Original Tasks

Since we use a new dataset, we first compare the results to my work in [24]. Table 1 shows the results. We can see that the 36-way model performs slightly better on both Task 1 and Task 2, but vice versa for the 9-way model. This result is not altogether surprising. Since the dataset was augmented with images that have semantic segmentation masks but no points for object-part inference, the semantic head of the model benefits most from the augmentation. In other words, the model improves at disambiguating between objects but not at distinguishing between whole object and part points. This is likely to benefit the 36-way model more, since it contains a greater number of object classes.

	36-way T1	36-way T2	9-way T1	9-way T2
Prev	67.53%	53.56%	69.93%	63.46%
Now	68.45%	57.36%	68.02%	59.96%

Table 1: Results of the 36-way and 9-way models on Task 1 and Task 2 using the expanded dataset compared to the old dataset.

Overall, this is a reasonable result and shows the feasibility of deep learning models on the object-part inference task. The accuracy is significant for Task 2, and the accuracy of 68.45% on Task 1, binary classification of whole object vs. part, demonstrates the ability of the model to generally distinguish between whole object and part points. Figure 5 shows qualitative results of the 36-way model on images in the validation set.

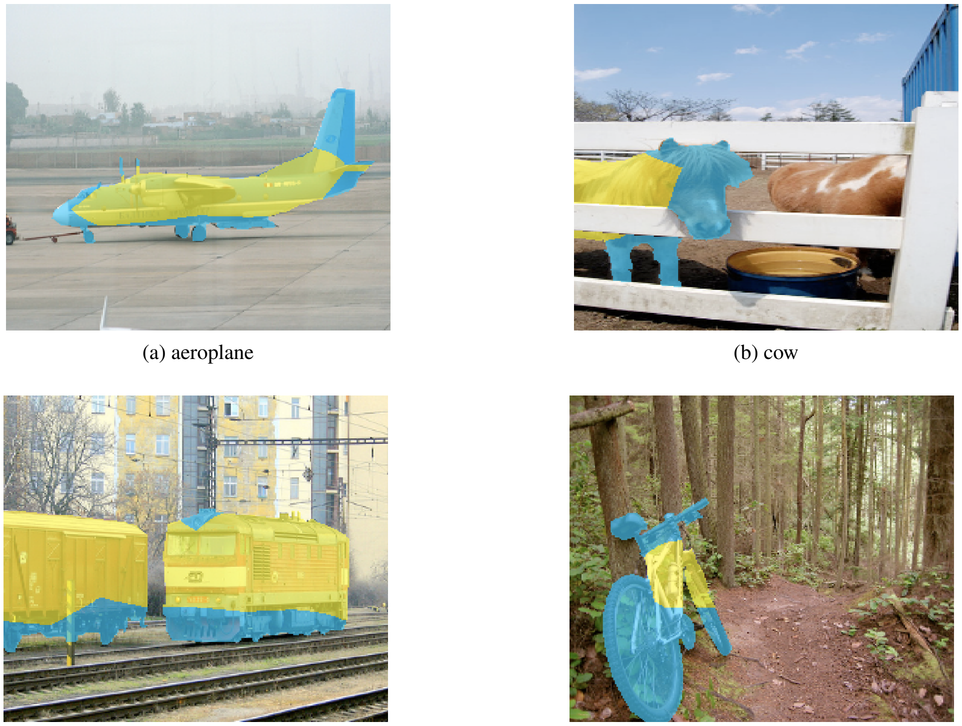
4.2. Soft Label Model

The next comparison point is the model with soft labels, which is intended to more accurately represent the ambiguity inherent in the task. This model performs comparably but slightly worse than the one-hot model, with an accuracy of 55.79% on Task 2. However, it performs particularly well on ambiguous points. The accuracy of the one-hot model on ambiguous points is 47.44%, while the accuracy of the soft label model is 53.24%. Table 2 summarizes these results.

	Task 2	Task 3
One-hot	57.36%	47.44%
Soft Label	55.79%	53.24%

Table 2: Comparing the one-hot and soft label methods on Task 2 and Task 3.

The results indicate that encoding labels for ambiguous points as probability distributions has little effect overall, but increases the model's performance on ambiguous



(a) aeroplane

(b) cow

(c) train

(d) bicycle

Figure 5: Qualitative results of the 36-way model on images across several classes. Blue indicates predicted part regions of the object and yellow predicted whole object regions of the object.

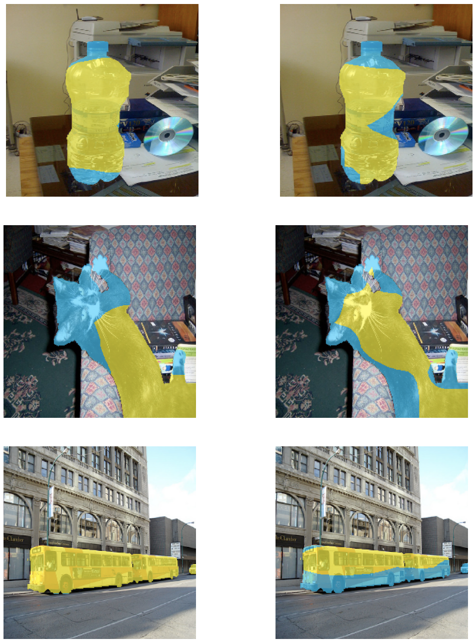
points. This is an interesting result, as it indicates that the soft label model is able to capture the uncertainty associated with ambiguous points and find the majority human consensus reasonably effectively. In particular, we draw from this that a soft label approach can be effective for instances of a task that are particularly ambiguous.

Figure 6 contrasts predictions of the one-hot model and the soft-label model on the same images. The soft label model predictions are generally more uncertain and arbitrary, as the first two images show. Although both models perform well on the two images, the soft label model is less smooth than the one-hot model and rather arbitrarily predicts some regions to be part. This is reasonable, since the soft label model is likely to output more uncertain predictions given the nature of the labels. The third image shows an example where the soft label model performs better than the one hot model. As discussed, objects are in the background generally cause ambiguity in the task, and so the ambiguity associated with the image might lead to the better performance of the soft label method.

4.3. Ambiguity Prediction

We also try to model ambiguity by adding a third category to the model, such that the model explicitly predicts ambiguous points. We use the 13-way ”super-class” model, which is the 9-way model with ambiguous points explicitly predicted. As might be expected, introducing ambiguity into the model decreases performance. On the 13-way classification task, the model achieves a reasonable accuracy of 50.83%. The accuracy on ambiguous points specifically is 49.54%, which interestingly is not significantly lower than the overall accuracy. Finally, the 3-way accuracy conditioned on the object class is 55.15%. This is significantly lower than the 2-way accuracy of the 9-way model, indicating that ambiguous points pose a particular challenge for the model. However, it is still far above the random expectation for a 3-way classification (33%), indicating that the model is still able to make some progress on the task. This is significant given the difficulty of classifying ambiguous points. Figure 7 shows the performance of the 9-way model and the 13-way model on two images. Both models appear to understand the object similarly, but the 13-way ”ambiguous model” characterizes the ambiguous regions reasonably well to provide a more nuanced understanding of object-part inference.

As discussed in Section 3.4, we can weight ambiguous points higher or lower when training the model, depending on our priority to classify them correctly. Since the accu-



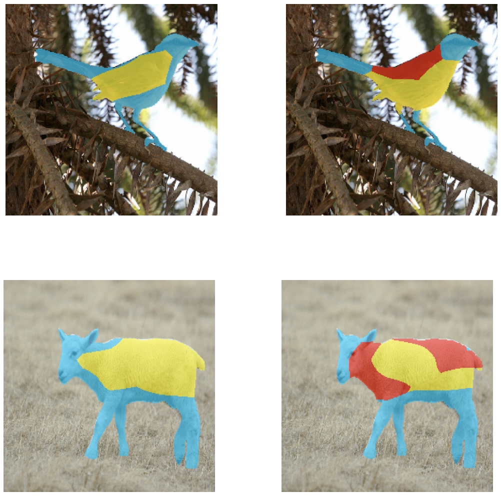
One-hot

Soft label

Figure 6: The left column shows object/part predictions of the one-hot model, and the right column object/part predictions of the soft label model. The soft label model predictions are generally more uncertain and arbitrary, as the first two images show. The third image is an example of where the soft label model performs better than the one-hot model.

racy on ambiguous points is comparable to the overall accuracy, we experiment with both approaches. First, we give all points with ambiguous classes a weight of 10 and other classes a weight of 1, which we call HI-AMB. We then give all points with ambiguous classes a weight of 0.1 and other classes a weight of 1, which we call LO-AMB.

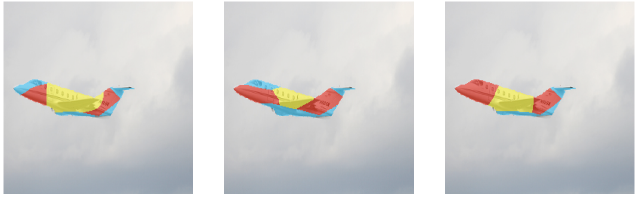
The 13-way accuracy is comparable for all three models. Surprisingly, however, HI-AMB does not yield a higher accuracy than LO-AMB on points in the validation set: 45.87% vs. 46.09%. A possible explanation for this is that the HI-AMB model is overfitting on the ambiguous points in the training set and not generalizing well to the ambiguous points in the validation set. Qualitative analysis, however, suggests that HI-AMB's predictions are more uncertain than LO-AMB. Figure 8 shows an example of the same image in increasing order of ambiguity weight. It and other images indicate that the model's uncertainty can be tuned, if not precisely.



9-way

13-way

Figure 7: Object/part predictions for the 9-way and 13-way model on two images. Red indicates regions of the object predicted to be ambiguous. The 13-way model appears to identify ambiguous regions reasonably effectively. Note that the 9-way model does not predict ambiguity.



(a) $w = 0.1$

(b) $w = 1$

(c) $w = 10$

Figure 8: An example of models with different ambiguous class weights evaluated on the same image. With increasing ambiguity weight appears increasing ambiguity.

5. Conclusion and Future Work

In this paper, we first describe the object-part inference task and motivate it by the possibility of a general understanding of human pointing. We then describe a joint supervision deep learning approach to the task that combines sparse and dense annotations. We show that this model performs reasonably well on an expanded dataset, indicating the feasibility of this task for computer vision systems to solve.

We focus the remainder of the paper on ambiguity. First, we show that using soft labels instead of one-hot encoding increases the model's accuracy on ambiguous points, while also generally increasing the uncertainty of the model. Second, we train a model to explicitly predict ambiguous points. We show that this model achieves significant accuracy on both an N-way and a 3-way classification task, and is able to characterize ambiguous regions of objects reasonably well. Finally, we over-weight and under-weight ambiguous points in the training set to demonstrate how we can tune the amount of model uncertainty.

The ability of the model to characterize ambiguous regions is a significant achievement. Deep learning systems thus far have relied on human labelers as objective ground truth. Given the ambiguity and complexity of human visual perception, this substantially limits the amount of tasks that computer vision systems can solve. By training models to predict where humans might disagree, we build a much more sophisticated understanding of human perception. Such understanding may also create opportunities for AI-human communication in the visual domain in real-world applications, such as disaster relief.

However, the accuracy for both the non-ambiguous models and ambiguous models can be improved significantly. In particular, if we treat the mode of human responses for each point as the correct label, then the "human accuracy" on this dataset is 84.6%, significantly higher than our non-ambiguous results. Humans can also do a very good job of identifying ambiguous regions of an object depending on size, point location, and location in image. This indicates that our work, while promising, has room for improvement.

Future work would first focus on expanding the size of the dataset, which is still relatively small (5-6 points per image on average). This would likely provide a greater boost in performance than training more sophisticated models. On the model-building side, we would continue to focus on models that characterize the ambiguity in the task. One such direction of future work would be Bayesian deep learning, which was addressed in related work. Training a deep learning model with variational inference would allow it to jointly learn the uncertainty of the task along with the task itself, potentially leading to a better understanding of ambiguity and higher classification accuracy. In the more distant future, we might look to increase the current object-part inference task by an order of complexity to better model human pointing, although the task here is complex enough and in many ways a new frontier in computer vision.

References

[1] A solution of computer vision based real-time hand pointing recognition. In *2008 27th Chinese Control Conference*, pages 384–388, July 2008.

[2] S. Abidi, M. Williams, and B. Johnston. Human pointing as a robot directive. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 67–68, March 2013.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Fei Fei Li. Whats the point: Semantic segmentation with point supervision. volume 9911, pages 549–565, 10 2016.

[5] Fei Chao, Zhengshuai Wang, Changjing Shang, Qinggang Meng, Min Jiang, Changle Zhou, and Qiang Shen. A developmental approach to robotic pointing via humanrobot interaction. *Information Sciences*, 283:288 – 303, 2014. New Trend of Computational Intelligence in Human-Robot Interaction.

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. *CoRR*, abs/1406.2031, 2014.

[7] M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):569–582, March 2015.

[8] François Chollet et al. Keras. `https://keras.io`, 2015.

[9] Cristina Colonnesi, Geert Jan J.M. Stams, Irene Koster, and Marc J. Noom. The relation between pointing and language development: A meta-analysis. *Developmental Review*, 30(4):352 – 366, 2010.

[10] Runmin Cong, Jianjun Lei, Huazhu Fu, Ming-Ming Cheng, Weisi Lin, and Qingming Huang. Review of visual saliency detection with comprehensive information. *CoRR*, abs/1803.03391, 2018.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[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. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[14] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *IEEE Transactions on Image Processing*, 26(6):2825–2838, 2017.

[15] Alberto Garcia-Garcia, Sergio Orts, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017.

[16] Y. Guan. Uncalibrated camera vision pointing recognition for hci. In *2010 13th IEEE International Conference on Computational Science and Engineering*, pages 204–207, Dec 2010.

[17] Bharath Hariharan, Pablo Arbelaez, Ross B. Girshick, and Jitendra Malik. Simultaneous detection and segmentation. *CoRR*, abs/1407.1808, 2014.

[18] Willian Hinthorn. Inferring intent from pointing with computer vision. 2018.

[19] Rachel M. Holladay, Anca D. Dragan, and Siddhartha S. Srinivasa. Legible robot pointing. *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 217–223, 2014.

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural In-*

formation Processing Systems 30, pages 5574–5584. Curran Associates, Inc., 2017.

[21] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. *CoRR*, abs/1603.01976, 2016.

[22] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, 2016.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[24] Arjun Mani. Understanding and modeling human pointing with computer vision. 2019.

[25] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *NIPS*, 2015.

[26] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollar. Learning to refine object segments. In *ECCV*, 2016.

[27] Aman Raj, Daniel Maturana, and Sebastian Scherer. Multi-scale convolutional architecture for semantic segmentation. Technical Report CMU-RI-TR-15-21, Carnegie Mellon University, Pittsburgh, PA, October 2015.

[28] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Nassir Navab, and Gregory D Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3591–3600, 2017.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[31] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, June 2015.

[32] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2814–2821, June 2014.