Determining Functional Attributes of Objects in Images

Christine Whalen
Brown University
christine_whalen@brown.edu

Abstract

I have used a data-driven approach to detect functional attributes of objects in images. Using ImageNet and Amazon Mechanical Turk, I have created a dataset of more than 7000 images with functional attribute labels. I have used these labels to train functional classifiers, and I compared the precision and recall of these classifiers to the precision and recall of the Amazon Mechanical Turk workers who provided the labels.

1. Introduction

When a person has a task to complete, missing the correct tool for the job is not an insurmountable barrier. If a person wants to carry water, any concave, impermeable object will do; if a person wants to sit down, a nearby tree stump will likely suffice. People are capable of improvisation: using objects in novel and unexpected ways. This means that objects can be used to complete many actions: some can be sat upon, some can carry water, and some can do both. These uses are functional attributes.

I attempted to capture the visual features of these functional attributes, using a data-driven approach. In order to determine the most common ways of interacting with objects, I looked at a list of the most common English verbs. From this set of common interactions, I created a set of 18 functional attributes, common to everyday objects. Functional attributes that I explore in this paper describe how a human can interact with an object or use the object to perform a task. From ImageNet, I collected URLs for all man-made objects. Using this data, I collected functional attribute labels using Amazon Mechanical Turk. Finally, I used these labels to train functional classifiers, and tested them on a held-out test set.

2. Related Work

Past work has explored both attribute detection in scenes [7] and also using attributes to classify objects [8], but neither has explored functional attributes of images.

2.1. Sun Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes

Patterson and Hays explored attribute-based representations for scenes [7]. Rather than classifying images as being depictions of kitchens or beaches, they chose to focus on scene attributes. Specifically, they explored five types of scene attributes: materials, surface properties, functions, spatial envelope attributes, and object presence.

They built the SUN Attribute Database, using images from the SUN categorical database and collecting labels using Amazon Mechanical Turk. They chose 102 scene attributes and collected labels on 14,000 images. While they crowdsourced their collection of attributes, I chose mine by hand. The authors used a combination of low-level features like gist and HOG to detect scene attributes.

Functional Features
Be turned on or off
Write or make a mark
Be heard
Be smelled
Be opened or closed
Be thrown
Hold liquid
Roll
Be sat upon
Float
Be broken
Be folded
Be climbed
Carry objects
Be eaten
Be worn
Protect
Harm

Table 1: This is the full list of functional attributes I considered. Example images of functional attribute occurrences in my dataset are in Figure 1.

Figure 1: These are example images for each of the functional features I am exploring. From left to right, they are "Be turned on or off", "Write or make a mark", "Be heard", "Be smelled", "Be opened or closed", "Be thrown", "Hold liquid", "Roll", "Be sat upon", "Float", "Be broken", "Be folded", "Be climbed", "Carry objects", "Be eaten", "Be worn", "Harm" and "Protect". Some of these images may have more than one valid functional feature.

This is a similar problem to the one I have tried to solve, though it is focused on scenes and scene attributes, rather than objects and object attributes. Both their approach and mine classify higher-level features of images without directly classifying the scene or object. My approach focuses specifically on functional features of objects, and rather than using low-level features for classification, I have used a convolutional neural net.

2.2. Recognition by Functional Parts

Rivlin, Dickinson, and Rosenfeld [8] worked on recovering shape and function from 2-D image data. They segmented images into sets of parts; possible functions were then assigned to groups of parts and to single parts. To do this, they used shape primitives, like sticks, strips, plates, and blobs, and enumerated the possible spatial relations which combine multiple shape primitives. Once they obtained a set of shape primitives, they were able to map functional primitives to shapes. For example, something which is intended to be picked up, like a handle, must be stick-like and small enough that a human hand can wrap around it.

This paper attempted to solve two problems: a bottom-up approach, where an unknown object was identified by its functional parts, and a top-down approach, where a specific object in an image was found by matching its functional parts to the expected attribute predictions.

Their bottom-up approach is similar to the problem I am attempting to solve: given an unknown object, what functions can it be used for? However, the authors attempt to determine the object category from the functions; while my classification scheme may also be used to identify object categories, the functional attributes are the goal. Their approach to solving the problem was also very different than my approach. They estimated specific relationships between object parts and used these relationships to predict functions, while my method abstracts away from these rules by collecting a large amount of data.

3. Data Collection

I collected functional labels for objects in images using ImageNet [5] and Amazon Mechanical Turk.

3.1. Data Acquisition

I determined the most common ways to interact with everyday objects from a list of the 500 most commonly used English verbs. I created a list of 18 interactions with or uses of an object to use as my functional features. For this list, I chose interactions which apply across object categories and which I believed may require visual features which distinguish them from other objects. Something that is "Able to hold water" may be a cup, a trash can lid, or a canoe, so this function applies across object categories. Something which is "Able to be turned on or off" will not be a natural object, may emit its own light (like a lamp or computer screen), and will likely be made out of plastic or metal; these features are visually distinctive.

In addition to features relating to how humans interact with the world, I was also interested in some features which relate to how humans perceive the world. When the typical person sees an object, he may be able to anticipate what other senses will be triggered by the same object. A study performed by Jadauji et al. [6] has shown that, in addition to there being a close connection between olfactory and visual responses, activation of the visual cortex improves performance on olfactory tasks. This is motivation for exploring the relationship between visual stimuli and other senses. To this end, I included "Able to be heard," "Able to be smelled," and "Able to be eaten" as functional attributes. I assumed all objects in my dataset are able to be seen and touched, as the objects appear in camera images, and "Able to be eaten" is a stand-in for the sense of taste.

I collected my dataset from ImageNet. I used the list of image urls from the Fall 2011 release of ImageNet[1], which contains approximately 14 million images. From the list of synset IDs, I chose the sensets for objects which were man-made. A synset in ImageNet is a collection of images for words occurring in the equivalent synset taken from WordNet. WordNet groups words into distinct concepts these sets of synonyms are called "synsets"[3]. WordNet also includes relationships between synsets; the relationship between a general synset, like the "man-made object" concept, and more specific synsets, like the "couch" set, allowed me to retrieve all images in ImageNet which were part of the general synset I selected. Because I was interested in how people can possibly interact with objects, I explored only objects which were labelled as man-made; my theory was that choosing objects which had been created by humans would result in objects which have a purpose, and therefore also have functional features.

3.2. Processing by Hand

For each synset that was a child of the manmade objects synset, I selected up to 5 images. Not all images could be downloaded from the ImageNet url provided, so any which did not load were skipped. This resulted in a set of approximately 30,000 images. I then cleaned the images by hand to remove any images where the focal point was unclear, functional features did not apply, the images were inappropriate for MTurk workers under the age of 18, or the image took too long to load from the external source. This process resulted in 11,749 images for which I collected labels.

(a) Functional attributes do not apply.

(b) Unclear focal point.

Figure 2: These images are examples of the kinds of images which were removed.

Please choose all actions which could be performed with this object. Select the actions which complete the sentence, "This object is able to _____." for any salient object in the image. People and animals are not objects.

☐ Be turned on or off	☐ Be smelled	☐ Hold liquid	☐ Float	☐ Be climbed	☐ Be worn
☐ Write or make a mark	☐ Be opened or closed	☐ Roll	☐ Be broken	☐ Carry objects	☐ Protect
☐ Be heard	☐ Be thrown	☐ Be sat upon	☐ Be folded	☐ Be eaten	☐ Harm

Figure 3: Example question from Qualtrics survey. A worker would be shown 20 random images, and for each image, the worker can select any combination of features.

4. Image Annotation

4.1. Pilot Tasks

I used Qualtrics and Amazon Mechanical Turk to collect functional labels for my images. I created a survey using Qualtrics, which displayed a single image and asked participants to make 18 binary choices for the image: whether or not an object in the image could be used for each of the functions I am exploring. The participants were given these instructions: "Please choose all actions which could be performed with this object. Select the actions which complete the sentence, 'This object is able to _____.' for any salient object in the image. People and animals are not objects." For each participant, I collected responses for 20 images randomly selected from my set of image URLs.

I tested using both binary choices and Likert scales for all features. Using a Likert scale allowed participants to make non-binary choices; rather than saying whether or not an object in an image could be used for a function, participants were able to describe how well the object could be used. I found this task to be significantly more challenging and time-consuming than the binary classification, so I decided to use classification instead. This decision was supported by a study at Brown in 2014, performed by Cavanagh et al. [4], which showed that decision conflict is a cost in selecting actions. As a result, answering Likert scale questions, which require more difficult decisions, could have become an aversive event causing an increase in the time and expense of paying Mechanical Turk workers.

4.2. Mechanical Turk Details

Using Amazon Mechanical Turk, I collected responses from 637 participants. Of these, 604 finished the task. Each participant gave functional labels for a random selection of 20 images, chosen from my set of 11,749 images.

Each participant was paid $0.02 per image labelled. After removing a few outliers who had required more than an hour, participants took an average of 8 minutes and 45 seconds to complete the task, so were paid an hourly wage of $2.84. Participants were allowed to take the survey more than once.

5. Dataset Statistics

5.1. Collecting Ground Truth from Humans

When I had collected responses from 637 participants, 7554 of my images had been seen and responded to by a person. I discarded data from humans who did not finish the task. In order to ensure that my ground truth responses were reasonable, I created a system which uses multiple participants' responses to create a correct answer. For each image, I determined the most common response for each feature, and I discarded any images where there were not at least two people who agreed. This became the ground truth label for the image. This resulted in 2896 images with labels, all of which had been independently verified by more than one person.

These ground truth labels allowed me to examine the types of mistakes humans make in labeling this sort of data. Not all disagreements are mistakes; some may occur because participants had a different understanding of what a functional feature means.

In Figure 4, below, there are some examples of images which at least three workers labeled as being able to hold liquid. There is quite a bit of variety in the object classes being labelled. Many of the objects which can hold liquid are mugs, but some are paint cans, thermoses, and artifacts.

Figure 4: These are all images of objects which can hold liquid.

Figure 5: This is an image that was labelled by four different Mechanical Turk workers. Three out of four of them agreed that it can be opened or closed, it can be climbed, and it can protect. Two of the workers believed it can be sat upon, and one worker believed that it can be turned on or off, be smelled, be broken, and harm.

6. Human Accuracy, Precision, and Recall

6.1. Human Errors

In general, my participants were more likely to miss labels than to give incorrect labels. Intuitively, this seems reasonable. Some of the features are difficult to determine, like "Be smelled," and some of them relate to small objects in images: Pens and pencils are often very small, and they are important to determine whether "Write or make a mark" is a valid classification. In addition, some humans may have many false negatives because they are not taking the time to look closely at images.

People were most likely to give false negatives for "able to be thrown," "able to be broken," and "able to protect." People were most likely to give false positives for "able to be turned on or off," "able to be opened or closed," "able to be thrown," and "able to be broken." "Able to be thrown" and "able to be broken" have both many false positives and false negatives, and were the two categories on which humans made the most mistakes. People made the fewest total mistakes on the feature, "able to be eaten."

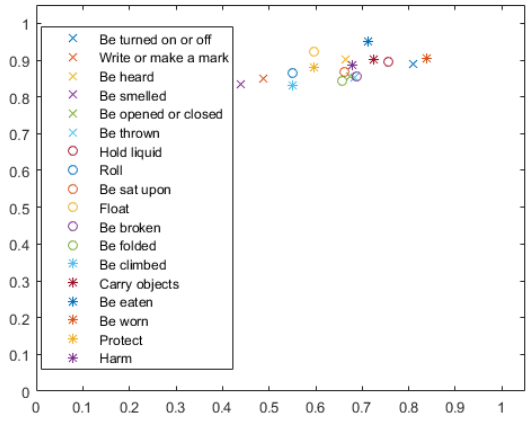


Figure 6: Mechanical Turk worker precision and recall for each functional feature. Some functional features had more consistent answers than others.

6.2. Recall and Precision

The errors made by humans are displayed in Table 2. ”Be worn” and ”Be turned on or off” have the highest recall, while ”Be eaten” has the highest precision.

7. Classification Method

7.1. CNN Features

I used the MatConvNet implementation of the neural net from Very Deep Convolutional Networks for Large-Scale Visual Recognition [10][9] to generate my features. I resized the images I was working from to 224x224x3 and removed the average image that the neural net was trained on. I then ran each image through the neural net, and extracted the outputs from the last five layers. Finally, I created an additional set of features, by combining the outputs of layers 36 and 38.

7.2. Linear SVM on Last Convolutional Layer

For each question and layer, I randomized the order of the image features and split them into training and test data. I used 80% of my data to train a linear SVM, and tested on the remaining 20% of my data. The results of the SVM classifier trained on each layer are shown in Figure 4, below. These are the precision and recall for a single set of features across all functional attributes. Additional images displaying the precision and recall of features from layers for specific attributes appear at the end of this paper.

While I was performing analysis on the classifier, I noticed some variance in the success of my models, and their performance in relation to models trained on features from earlier layers. I believe this variance was caused by the random splits of test and training data, resulting in different levels of success on the test set. To reduce the variance across runs, I performed ten different splits for each set of features, trained ten different SVMs, and calculated the precision and recall curve on the concatenated results for all SVMs on their respective test sets.

The combined feature of layers 36 and 38 performed better than either layer 38 or layer 36 on their own. Conceptually, layer 38 is the output of the softmax layer of the convolutional neural net, which takes in an image, and outputs the probabilities of the 1000 different object categories represented in the ImageNet Challenge, ILSVRC 2014 [2]. Layer 36 is the output of the final convolutional layer of the neural net, which is a 4096-element vector.

8. Classification Results

8.1. Recall, Precision, and Accuracy

The SVM classifier is good at the following features: ”Hold liquid”, ”Float”, ”Be eaten”, and ”Be worn”. It has greater than 50% precision and recall in all of these features.

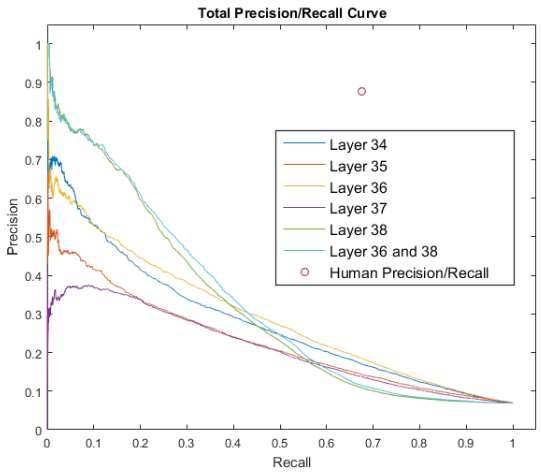


Figure 7: The result of multiple SVMs trained on features constructed from the output of different layers of the CNN. The legend shown applies to all precision and recall curves throughout this paper.

However, the classifier has much worse performance on the following features: ”Be smelled”, ”Roll”, ”Write or make a mark”, and ”Be climbed”.

In Figure 4, the precision and recall curves across all functional features have been plotted for each feature layer. Layer 38, the layer that performs object categorization, achieves high precision when recall is low. However, it performs less well than other layers when recall is greater than 0.4. Intuitively, this makes sense; consider the functional feature, ”be sat upon.” In general, most items that can be sat upon will be members of one of only a few categories, like chairs or couches. The classifier which uses object categories as features will learn a mapping between those categories and functions. However, it will be less likely to detect outliers for the function, like tables or beds, meaning that its performance with high recall will be worse than the performance of other layers.

8.2. Comparison to Human Precision and Recall

The categories that the SVM classifiers performed best at were in the top right section of the human workers' precision-recall plot, while the categories that the classifier performed worst at were in the bottom left section of the human precision-recall plot.

”Be worn” and ”Hold liquid”, which the SVM classifiers performed well on, are clearly functions that humans had high precision and recall while labeling. However, ”Be smelled”, ”Write or make a mark”, and ”Roll” have comparatively low precision and recall. This indicates that some categories may be harder to label than others. It is possible

Functional Features	Total Responses	Total Mistakes	Human Accuracy	# of Disputed Responses
Be turned on or off	6773	288	0.957478	971
Write or make a mark	7300	125	0.982877	444
Be heard	7002	248	0.964582	742
Be smelled	6983	215	0.969211	761
Be opened or closed	6519	386	0.940788	1225
Be thrown	6114	489	0.92002	1630
Hold liquid	7204	155	0.978484	540
Roll	6770	289	0.957312	974
Be sat upon	6794	354	0.947895	950
Float	7230	157	0.978285	514
Be broken	5857	550	0.906095	1887
Be folded	7106	226	0.968196	638
Be climbed	7079	206	0.9709	665
Carry objects	6728	325	0.95169	1016
Be eaten	7600	52	0.993158	144
Be worn	7276	170	0.976636	468
Protect	6541	384	0.941293	1203
Harm	7081	216	0.969496	663

Table 2: This table displays participant accuracy for each feature. The same number of possible responses were available for each function. The number of disputed responses is the number of responses where positive count equaled the negative count, so the image was thrown out of my dataset and was not considered in calculating ground truth.

that the instructions for labeling these functions were unclear, or that they are simply more difficult to see.

8.3. Results on the Output of Other Layers

Using the softmax layer as the set of features rather than the convolutional layer output made a vast improvement for "Be sat upon" and "Harm". In Figure 7, compare the green line to the yellow line. These may be features for which only a few types of objects can be used, or where there is little relationship between two images which share the same functional feature. For example, it is possible that only a few types of objects were labelled as being able to be sat upon. In contrast, it could be that guns and knives do not share very many visual characteristics, though they are both able to cause harm.

Some of the features that the category estimate performed poorly on were "Be eaten", "Be smelled", and "Hold liquid". The precision-recall curves for these layers are shown in Figure 6. The output of the final convolutional layer outperformed all other layers for both "Be eaten" and "Be smelled", while the combined feature of the final convolutional layer and the category estimation outperformed other layers for "Hold liquid".

"Hold liquid" is an attribute that relies on the shape and material of an object, more than on its object category. Some object categories, like cups and mugs, will almost always be able hold liquid, while others, like trash cans, may be able to hold liquid. A trash can made of wire will not be able to hold liquid, but a trashcan made of plastic likely can.

Some functions have a small set of typical object classes with which the functions are performed, like "Be sat upon" or "Harm". Other functions are more varied, and, in the case of "Be eaten" and "Be smelled", depend on the state of the objects in question: a warm apple pie straight out of the oven can be smelled, but a cold apple pie wrapped in tin foil cannot. Essentially, when information beyond object category is required, layers which encode more information are more useful than an object category estimate.

9. Discussion and Future Steps

For future work, there are a few modifications I would be interested in exploring: these include pre-training, using bounding boxes, and allowing more classifications per person.

9.1. Different Methods for Collecting Data

9.1.1 Pre-training

Not all Mechanical Turkers interpret instructions in the same ways. For many images, two or more different workers provided different labels for a given feature in an image. For the functional attribute "Be turned on or off", 417 of the 2718 images with two or more responses had an equal number of positive and negative votes. It is not clear why each disagreement occurred, but it is possible that many of these

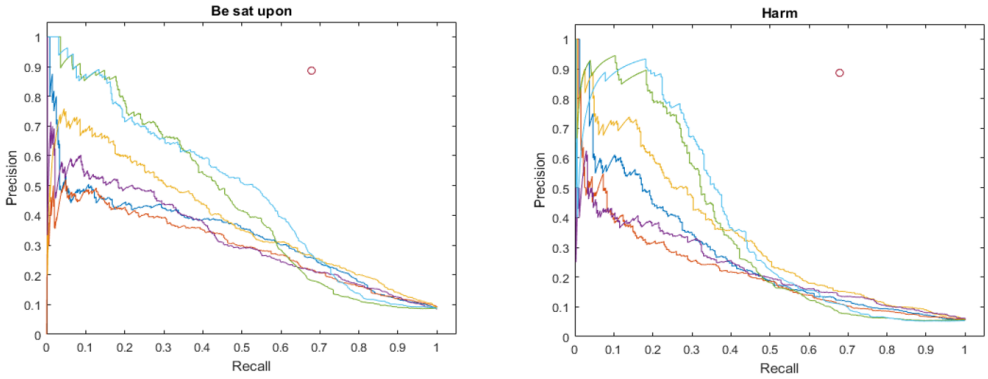


Figure 8: In this figure, compare the green line (the category estimate) to the yellow line (the output of the final convolutional layer).

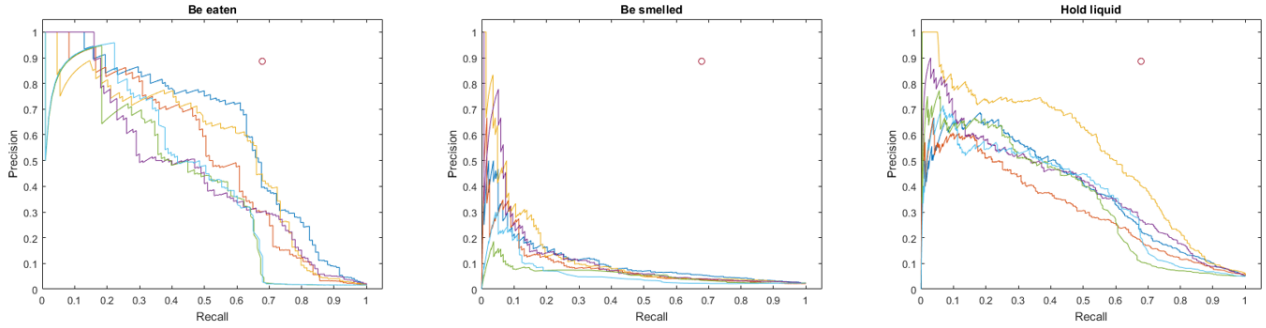


Figure 9: Layer 36 has good performance on both ”Be eaten” and ”Be smelled”, while the combined feature of Layer 36 and Layer 38 has good performance on ”Hold Liquid”. Both features outperform the category estimation layer for all three functions.

mistakes could have been avoided by using a pre-training task.

There are some attributes in particular for which the meaning could be unclear, or for which the boundary between binary classifications is blurry. For example, consider the attribute, ”Able to be thrown.” In general, a human conjuring images of objects which can be thrown comes up with small objects, which would typically be thrown. These objects, like balls or paper airplanes, are easily gripped and lifted in one hand. However, if one uses two hands, even an object like a desk chair could conceivably be lifted and thrown. In the case of a desk chair, whether or not it is classified as being able to be thrown depends on a few factors: is the worker considering using two hands? Does the worker have an accurate estimate of its weight? How strong is the worker? Not only do workers interpret instructions differently, they are also different people, with different physical capabilities.

In the case of ”Able to be heard” does the object in the image need to be in the process of making noise, or does it just need to be able to? For example, a trumpet on a desk would not be heard, but a trumpet held up to someone’s mouth would be; however, both trumpets are able to make noise.

The features with the highest rates of confusion among participants were, in order, ”Be broken”, ”Be thrown”, and ”Be opened or closed”. These three features were all features which my classifier performed poorly on. See Figure 10 for precision and recall curves for these three functions. The feature with the lowest rate of confusion was ”Be eaten”, and this was also a feature that my classifier performed well on.

Some sort of pre-training for each feature would likely result in more consistent responses. It would likely only require a few examples of potentially confusing images for each feature. In fact, I could use some of the images on which workers did not agree, which I ended up throwing out, like the ones in figures 11 and 12.

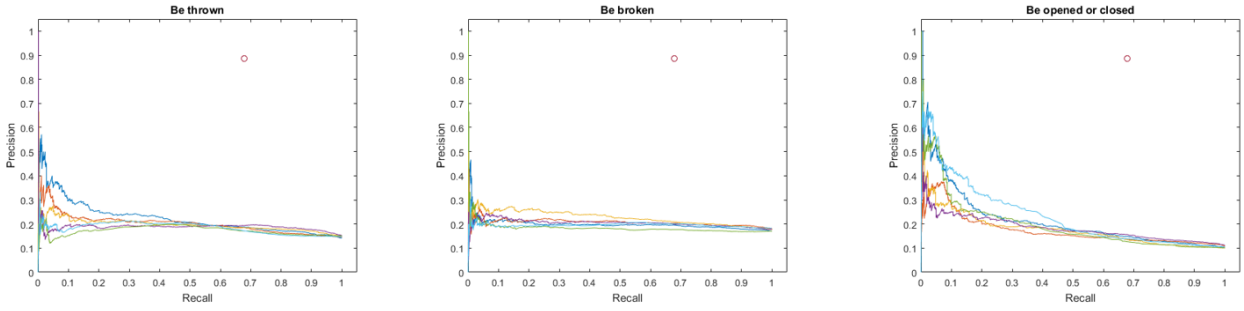


Figure 10: These three features were all confusing for humans and were also features which my classifier performed poorly on.

Figure 11: Three of the six people who saw this image believed that it could be thrown, and the other three believed that it could not be.

Figure 12: Two of the four people who saw this image believed that it could be heard. It is easy to see that the fan is not currently spinning, so in its current state, it is unlikely to be heard. However, one can easily imagine hearing it when it is spinning. The images in both Figure 10 and Figure 11 were thrown out before training due to this type of disagreement.

9.1.2 Bounding Boxes

All images associated with a given object category in ImageNet are guaranteed to have an instance of that object. However, not all instances of the object are the focus of the image, nor do they take up the majority of the pixels in the image, nor are they even guaranteed to be entirely contained within the image. There are typically also other objects in the image. This meant that it was not always clear what object or objects in the image should be labelled, and some small objects may have been lost or ignored.

Including bounding boxes around objects would likely help to mitigate this problem, by drawing the workers' attention to a specific object. This would remove the possibility that two workers who are looking at the same image may provide labels for two different objects within it, and would likely produce more consistent answers as a result. It would also result in labels for objects in images which are comprised of relatively few pixels: for example, a pen on top of a desk.

9.1.3 Allow More Classifications Per Person

The way that I performed data collection limited the number of labels that I collected per person. This meant that approximately 600 workers labelled 20 images each. This means that I had many different workers, all of whom may

Figure 13: Is this a picture of a canoe, a power tool in the background, the clamps holding the canoe together, or all of these things at once? One participant said that an object in this image could be turned on and that an object in this image could float. Both of these statements are true, but the object that is able to be turned on is not the clear focus.

have slightly different interpretations of the task. Allowing a single person to perform more classifications would improve the consistency of the labels.

9.2. Different Methods for Determining Ground Truth

The most significant increases in my classifier's accuracy, precision, and recall came from modifying my methods for determining the ground truth functional labels for images. There are two areas in which improvements could be made: the first is in adding verification measures when labels are collected, and the second is in removing labels from inconsistent workers.

In future iterations of data collection, it would be a good idea to include sentinel tasks: easy images which have predetermined ground truth labels, to test worker accuracy [7]. It could be that some workers are always inconsistent at labeling certain features, or it could be that some workers are missing possible positive labels, because they are moving too quickly. Of the 604 participants who completed the task, 68 took less than three minutes to complete the task. Completing the task in three minutes exactly requires that the participant view a new image every 9 seconds. Because there are 18 binary classifications to be made per image, a participant viewing an image every 9 seconds can only spend half a second on any single classification, and is much more likely to miss a functional feature than a participant who is moving through images more slowly. In fact, when I removed the labels from participants who finished the task in under three minutes, the total human precision and recall scores increased. Human recall improved by 4.3% from this change alone; this indicates that people who are completing the task quickly are likely to mislabel valid functions.

Minimum Time	Total Precision	Total Recall
0 minutes	85.96%	66.74%
3 minutes	87.59%	70.98%
5 minutes	89.33%	76.63%
7 minutes	92.84%	82.65%

Table 3: This table shows the total precision and recall values for humans across all functions. The humans who finished faster than the minimum time allowed were removed from consideration. Recall and precision both improve when fast workers are removed.

Alternatively, adding another layer of user input in which people check the accuracy of other's responses may also be helpful. In looking at worker responses, I observed a trend in my own reactions to the responses: it seems easier to confirm a positive classification than to invent one. In other words, when I saw a set of features for an image, if I saw a feature that had been labelled as positive that I would not have labelled as positive, it was relatively easy to understand why the classification had been made and consider it to be reasonable. This would have resulted in a different set of features than I would have supplied without another person's input. Because of this observation, it might be interesting to add one or more layers of worker verification in order to reduce the effect of differing opinions among workers, and to allow workers to converge to a single consensus. As a result, I think more positive classifications will be made, and more incorrect positive classifications will be caught.

While these measures are useful for future data, there may be more opportunities to remove problematic results even without sentinel tasks and verification steps: there may be people who are working significantly faster than others, or there may be people who disagree with other people often. My current voting strategy already removes labels when the majority of other workers who saw the image disagrees, but it acts on an image-by-image and feature-by-feature basis. It is possible that I have workers who are wrong on significantly more images than other workers, like my workers who completed the task too quickly. Including their results for any of their labels may be affecting the validity of my ground truth labels; it may be better to identify inaccurate workers and remove their labels entirely, as I have done with fast workers and those who did not complete the task.

Finally, there may be other voting methods which are reasonable. Initially, I had made the assumption that false negatives were going to be much more likely than false positives, based on the idea that it is easier to miss a positive feature than to mark an incorrect feature as positive. As a

result, I counted all positive votes as true, without considering the number of people who had supplied a negative vote. While this was less successful than my majority strategy, I think that this may be an interesting area to explore, given that there is a bias toward false negatives.

10. Conclusion

After collecting images and functional labels using ImageNet, Amazon Mechanical Turk and Qualtrics, I was able to train classifiers which detected these functional features in images. Some functions were more successful than others, both for my classifier and for my human workers. Layer 38, the feature that allowed for the best classification on most functional features, performed less well than layer 36 on the functions relating to human senses, like "Be eaten", "Be smelled", and "Be heard". However, the composite feature of 36 and 38 did very well across the board.

References

[1] Imagenet. Fall 2011 Release http://www.image-net.org/releases.

[2] Imagenet. Large Scale Visual Recognition Challenge 2014 http://image-net.org/challenges/LSVRC/2014/index.

[3] Wordnet. Princeton University http://wordnet.princeton.edu.

[4] J. Cavanagh, S. Masters, K. Bath, and M. Frank. Conflict acts as an implicit cost in reinforcement learning. *Nature Communications*, 5, 2014.

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Computer Vision and Pattern Recognition*, 2009.

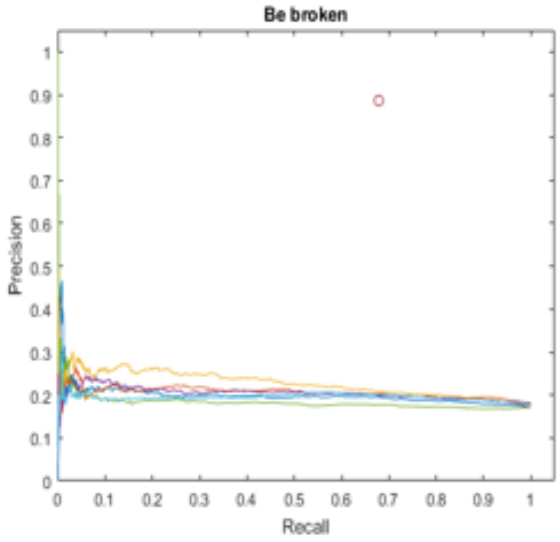
[6] J. B. Jadauji, J. Djordjevic, J. Lundstrom, and C. Pack. Modulation of olfactory perception by visual cortex stimulation. *Journal of Neuroscience*, 32(9):3095–3100, 2012.

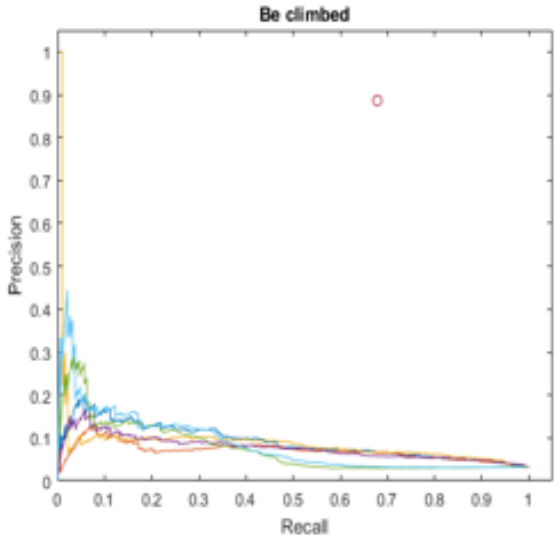
[7] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2012.

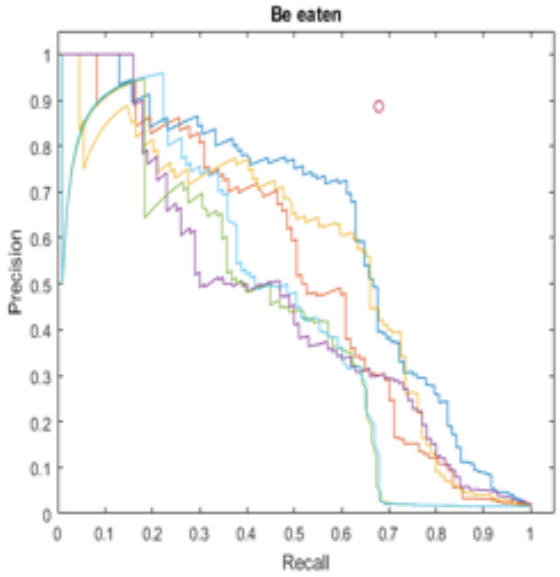
[8] Rivlin, Dickinson, and Rosenfeld. Recognition by functional parts [function-based object recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, 1994.

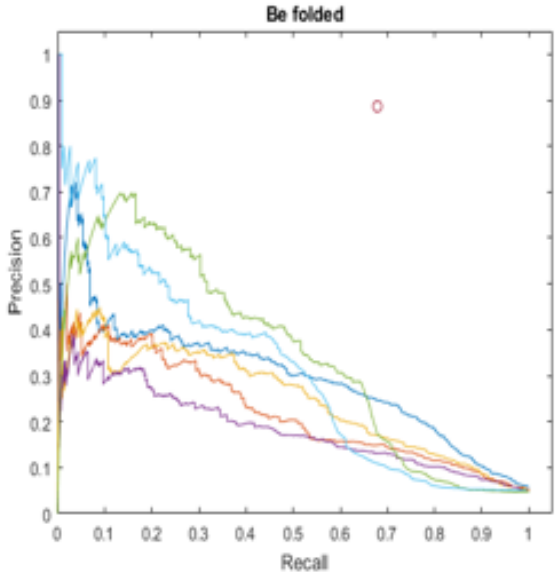
[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

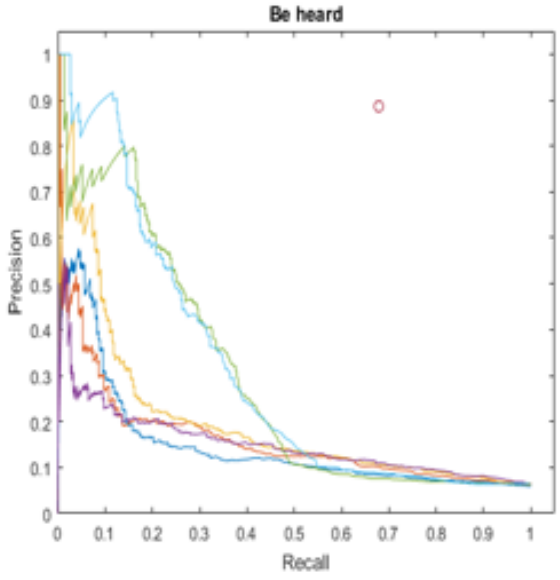
[10] A. Vedaldi. Matconvnet. Convolutional Neural Net in MATLAB http://www.vlfeat.org/matconvnet/#pretrained.

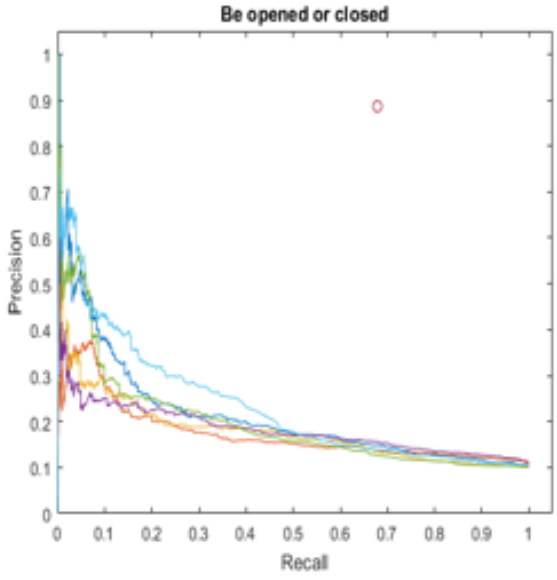

Be broken
Precision
Recall

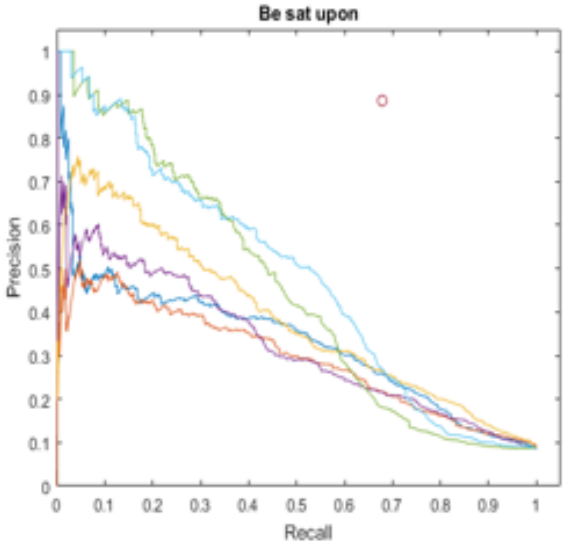

Be climbed
Precision
Recall

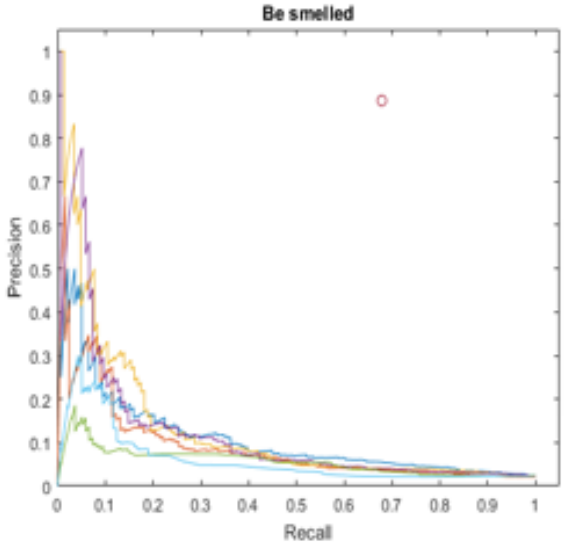

Be eaten
Precision
Recall

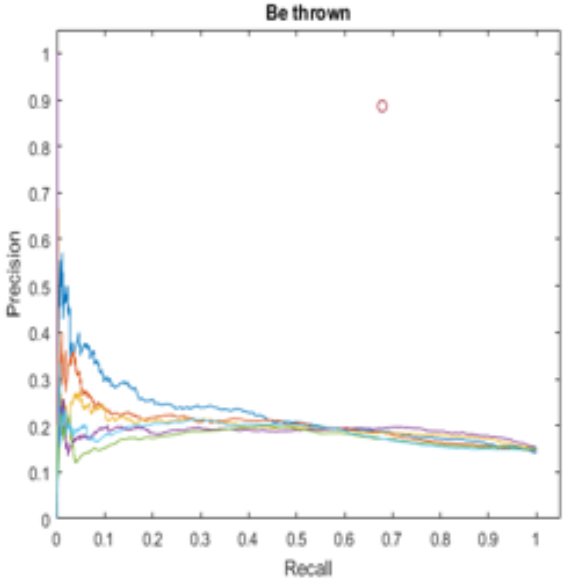

Be folded
Precision
Recall

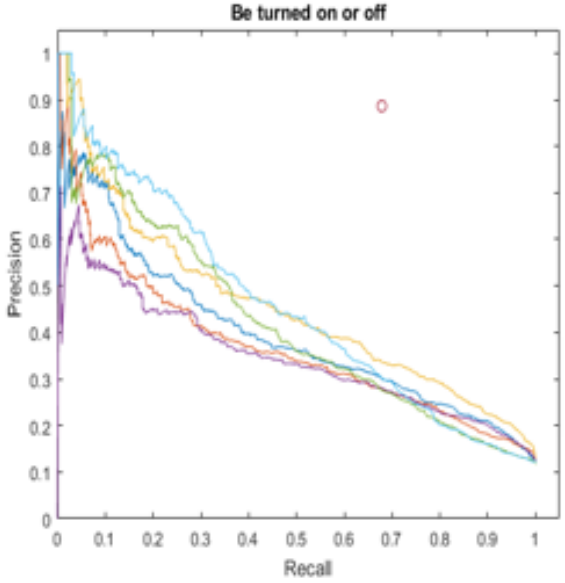

Be heard
Precision
Recall

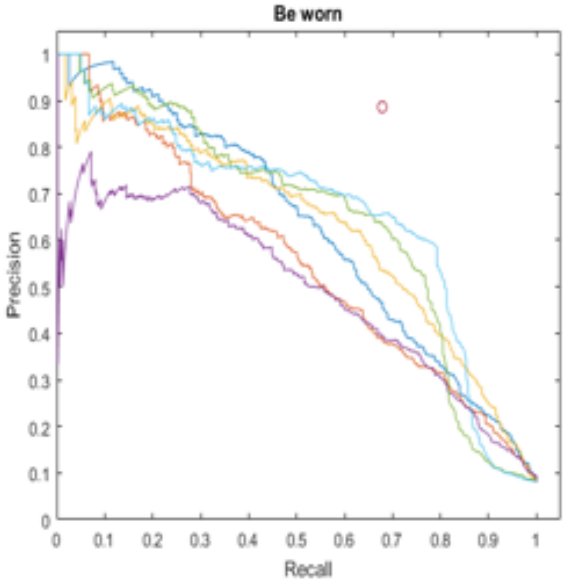

Be opened or closed
Precision
Recall

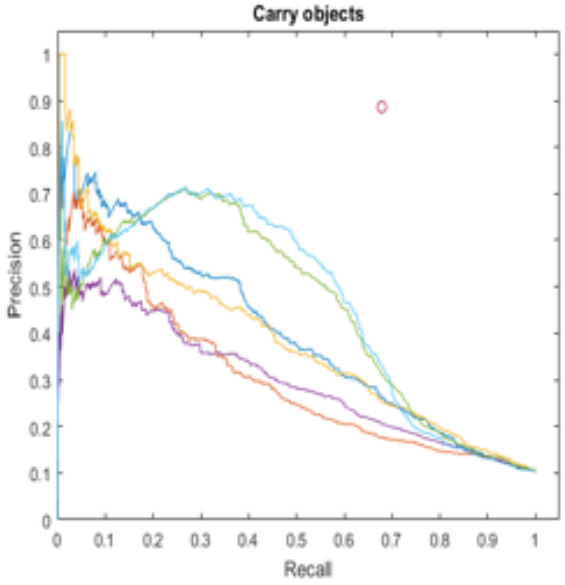

Be sat upon
Precision
Recall

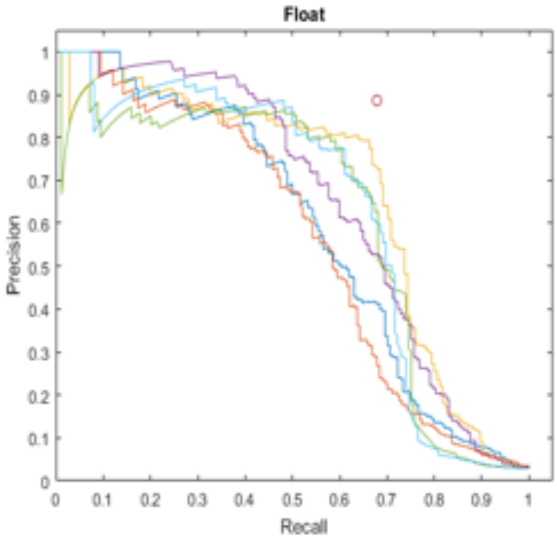

Be smelled
Precision
Recall

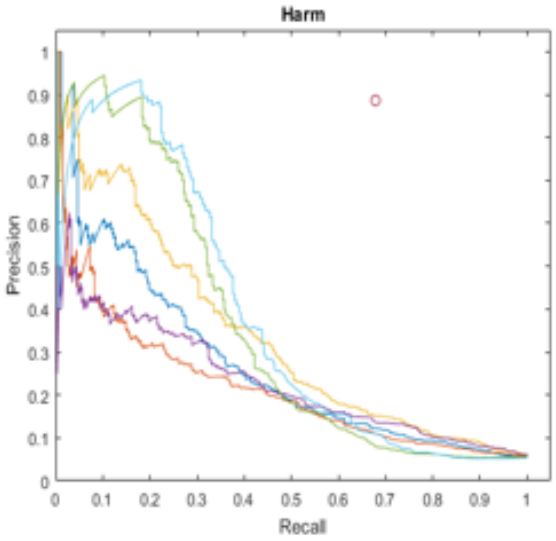

Be thrown
Precision
Recall

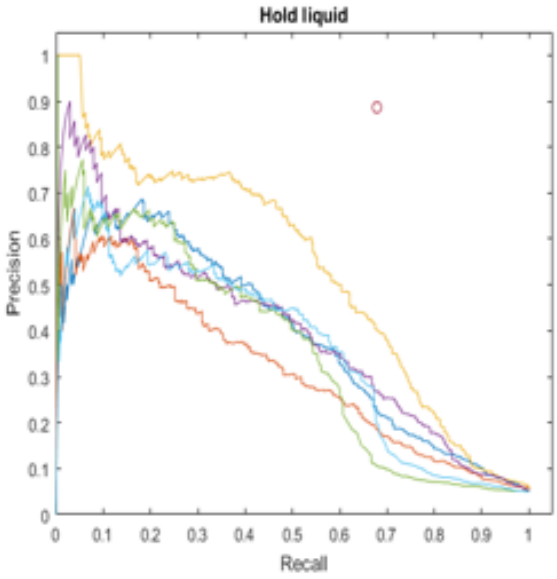

Be turned on or off
Precision
Recall

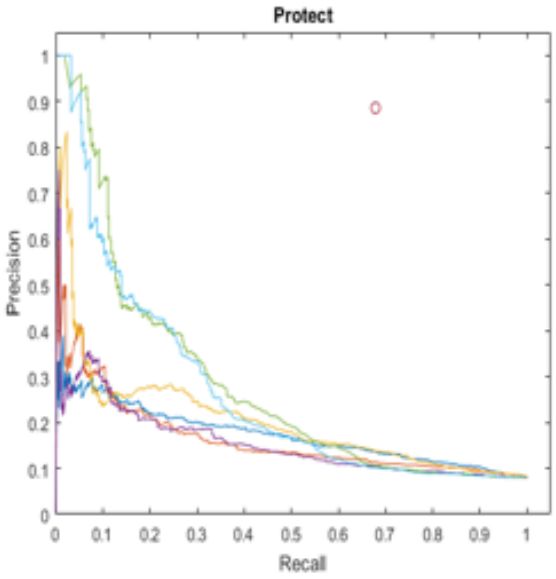

Be worn
Precision
Recall

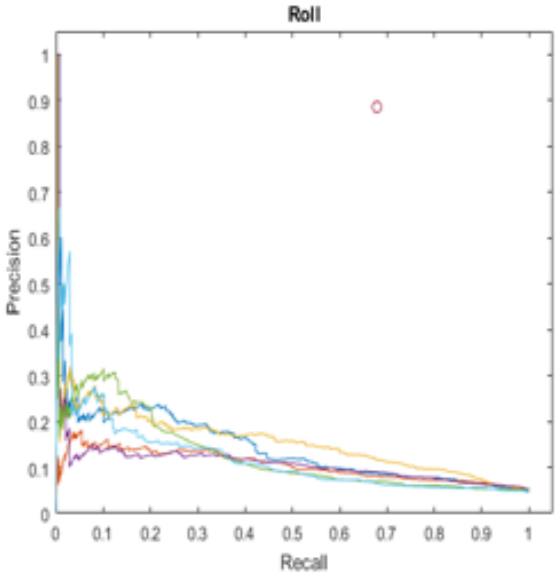

Carry objects
Precision
Recall

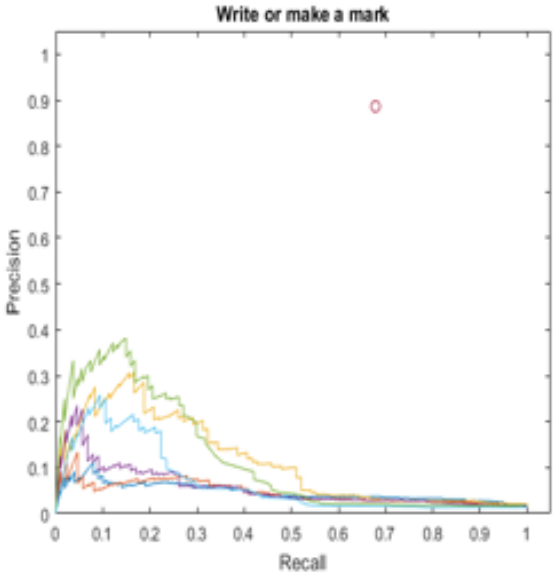

Float
Precision
Recall


Harm
Precision
Recall


Hold liquid
Precision
Recall


Protect
Precision
Recall


Roll
Precision
Recall


Write or make a mark
Precision
Recall