Using humans to build mid-level features

Genevieve Patterson[1] Tsung-Yi Lin[2] James Hays[1]
Brown University[1] University of California, San Diego[2]

Abstract

Identifying the distinctive parts of an image is a challenging task for computer vision. In contrast to previous methods, we use human participants to discover mid-level discriminative features. Amazon Mechanical Turk (AMT) workers filter groups of patches to identify clusters that have strong visual and semantic similarity. We show that SVMs trained from human-defined discriminative patches outperform the patch classifiers discovered by Singh et al. and Doersch et al. [7, 2] when used as features for classification on the 15 scene dataset [6].

1. Introduction

Recently, several publications have demonstrated state of the art performance at scene classification using discriminative patches [7, 2, 4]. These papers introduce different methods for identifying salient visual elements in the form of mid-sized image patches, and training classifiers to detect the visual phenomena observed in training patches. Singh et al. and Juneja et al. each propose pipelines for training discriminative patch classifiers, but both employ a library of discovered patches to create "bags of parts" for scene classification. The methods shown in [7, 4] show state of the art performance compared to other single features on the MIT Scene 67 dataset. Doersch et al. shows impressive capability to discriminate between the architecture of different European cities.

The insight of these papers is that scene and object categories can be separated from each other by observing a small number of visual events that are highly discriminative. Unlike bag of words models, not all of the locations are equally discriminative – only a sparse set of locations in an image are useful for determining the category. Detectors trained on discriminative patches can build features that are more useful for scene classification than directly using low-level features. This would be similar to identifying only the most discriminatively powerful visual words, and only using those words in the bag of words codebook. Patches also have the advantage of being larger than the typical visual word, thus enabling them to capture a visual element that could have higher-level semantic significance.

While there are several proposed methods to discover discriminative patches, [7, 4], we examine an interesting alternative to the often time consuming methods of automatic discriminative patch discovery. We directly ask non-expert humans to select visually and semantically similar image patches. We train classifiers to recognize the human-identified visual elements.

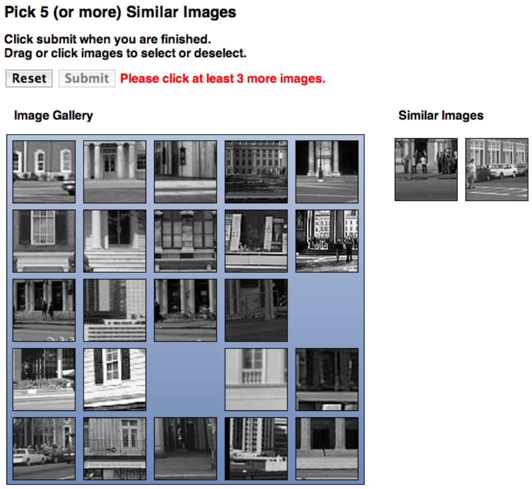
Other human-in-the-loop methods such as [3, 1, 5] have demonstrated success at visual classification tasks. Humans (often non-expert, crowdsourced humans) are commonly used in vision algorithms at two stages (a) annotation time, either exhaustively annotating a dataset or providing the most informative annotations in an "active learning" framework or (b) test time, coupled with a computational method to improve human accuracy and / or reduce human effort. In contrast, we put humans in the loop at neither annotation time nor test time, but rather at "representation discovery" time. The humans are directly telling the computer which visual elements should be discriminative. This has some similarity to part-based annotation of visual phenomena, except that our method does not require any explicit semantic meaning for the parts. In our experiments the humans never even see entire training images, only sets of candidate image patches. Putting humans "in the loop" at this stage in a recognition algorithm is to the best of our knowledge unexplored.

2. Building Human Patches

Our starting point for building human-in-the-loop patches is the automatic method presented in Singh et al. We first find candidate patches within the 15 scene dataset [6] using the code of [7, 2]. The algorithm selects 1200 random patches from each category and for each patch finds its 24 nearest neighbors. Each random patch and its nearest neighbors are a candidate discriminative patch model. Individual patches are 80x80 pixel windows represented as 2112 dimensional HoG features.

At this point the Singh et al. use an iterative cross-validation method to discover sets of 5 patches that when used to train a linear SVM result in discriminatively powerful classifiers. Instead of this computationaly expensive process, we present AMT workers with a page showing a group of 25 nearest neighbor patches. We create a patch selection task for 400 randomly selected patch groups from each category in the 15 scene dataset. Our user interface is shown in Fig. 1.

Using the interface in Fig. 1, we obtain 3 user responses for each nearest neighbor group. We manually examine the user responses to this task and only observe a few spurious responses out of thousands of HITs. This suggests that this HIT is not attractive to cheating Turkers. We discard


Pick 5 (or more) Similar Images

Click submit when you are finished.
Drag or click images to select or deselect.

Please click at least 3 more images.

Figure 1: *AMT patch cluster refinement interface.* Users are presented with a group of 25 nearest neighbor patches. These user-selected patches are used to train a model for a discriminative patch.

redundant responses where the selected patches are nearly the same and train discriminative patch models from the remaining responses. For each discriminative patch classifier, the user selected patches are the positive examples and a large set of randomly selected patches from other categories are the negative examples.

3. Results for Scene Classification

Figure 2 shows the performance of scene classification on the 15 scene dataset using automatically discovered and human-in-the-loop discriminative patch models. In the case of human-in-the-loop patches, we tried both linear and non-linear SVM patch classifiers. Each set of discriminative patch classifier generates a "bag of parts" histogram for every image which encodes how often a particular patch was found. A second SVM is trained to classify images into scene categories based on these "bags of parts". We set a detection threshold at a confidence of -1.0, as suggested in [7]. An equal number of patches are selected from each scene category. Fig. 2 shows that the patches discovered using human intervention perform better than the automatically discovered patches for scene classification on the 15 scene dataset.

Interestingly, our human patch discovery method is arguably cheaper to implement than the automatic method. In our experience, it took 300 CPU hours plus 1200 AMT Human Intelligence tasks to discover patches for one scene category using our method. It took us roughly 6000 CPU hours to automatically discover discriminative patches for one category. The human-generated patches method is approximately 20x faster than the iterative cross-validation method in [7]. Using the default cost of an Amazon AWS instance ($0.06 per instance per hour) and the cost of our AMT HITs ($0.04 per HIT), it would cost $66 to discover the human patches and $360 to discover the automatic patches for one of the 15 scene categories. In light of the efficiency and accuracy shown by human patch discovery, we believe that using humans to build mid-level patches deserves further inquiry.

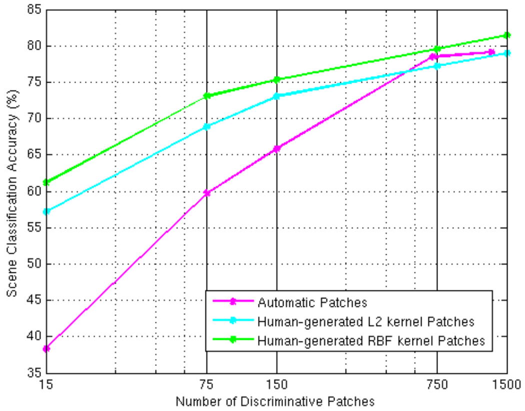


Figure 2: *Scene Classification Performance of Human and Automatic Patches on the 15 scene dataset.* The training set includes100 images from each of the 15 categories, and the test set contains 80-90 images from each category. In this plot each trend line shows the performance of a different kind of discriminative patch. The best performing scene classifier (either χ^2 or L1 kernel SVMs) are shown for each different kind of patch. The automatically generated patch models are linear SVMs (see [7]). The human generated patch models are trained using either linear or RBF kernel SVMs.

References

[1] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *Computer Vision–ECCV 2010*, pages 438–451. Springer, 2010. 1

[2] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes paris look like paris? *ACM Transactions on Graphics (TOG)*, 31(4):101, 2012. 1

[3] Y. Gingold, A. Shamir, and D. Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Transactions on Graphics (TOG)*, 31(5):119, 2012. 1

[4] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1

[5] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Image search with relative attribute feedback. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2973–2980. IEEE, 2012. 1

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006. 1

[7] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *Computer Vision–ECCV 2012*, pages 73–86. Springer, 2012. 1, 2