Improving the HoG descriptor

Carl Doersch
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh PA
cdoersch@cs.cmu.edu

Alexei Efros
Carnegie Mellon University
5000 Forbes Ave. Pittsburgh PA
efros@cs.cmu.edu

Abstract

The HoG descriptor has become one of the most popular low-level image representations in computer vision: even a small improvement in its ability to represent images would be useful. In this project, we explore several ways to enhance HoG at minimal performance cost.

One approach is to separate high-frequency gradients ('step edges'), which tend to represent edges, from low-frequency gradients ('diffuse gradients'), which tend to represent shading. We hypothesize that these two types of gradients should give different and complimentary information: the first indicates boundaries whereas the second gives clues about smooth 3-d shapes. As it is, however, HoG only represents the orientation of an edge, rather than its spatial extent, and so the distinction is lost. We propose several algorithms to separate these types of edges.

We also attempt to more strongly separate texture from contours. In the current implementation of HoG, an SVM cannot become sensitive a black object on a white background and also to a white object on a black background without also becoming somewhat sensitive to ordinary texture. We argue that a very simple modification to HoG can help with this problem.

1. Introduction

The HoG descriptor [2] has gained traction in the vision community, and particularly in object recognition, for several reasons. First, it is a vector-space model, where perceptual similarity is approximated by euclidan (or cosine) distance between two HoG vectors. This means that many off-the-shelf learning and database algorithms can work directly on HoG representations. Second, it appears to be a reasonably good model of perceptual similarity: it uses intensity gradients rather than intensity directly, which means that the responses of edges are localized; it is sensitive to local but not global contrast due to its normalization scheme; it can handle minor misalignment due to the bilinear interpolation between HoG cells; and many other reasons also apply. Third, it is very fast to compute: computing a HoG pyramid for a 500-by-500 image can take less than 2 seconds on a single core, and firing a sliding-window template at all positions and scales can happen equally fast via fast-fourier-transform convolution.

However, just because HoG is currently one of the best low-level descriptors for object recognition does not mean that it is as good as it could be. In this work, we propose that HoG can be improved by avoiding what we call *aliasing* in certain situations. By aliasing, we mean that two image patches which are perceptually very different can end up with very similar HoG descriptors. We focus on two cases of aliasing. The first is the case of diffuse gradients versus step edges. A diffuse gradient tends to indicate shading on some smooth surface: notably, animals under natural illumination tend to display very characteristic shading patterns on their bodies. A step edge is a place where one intensity or color abruptly gives way to another. These tend to occur wherever there is an object boundary. Intuitively these two patterns should be separated, since they tend to have different semantic meanings, but HoG is only sensitive to the direction of gradients. Within one cell, the histogram of a given orientation is a simple sum, which does not take into account the positions of the gradients relative to each other or the distribution of large versus small gradients in the given direction. To counter this type of aliasing, we propose a method which can separate the step edges from the diffuse gradients prior to histogramming, while not significantly changing the magnitudes of either type of gradient.

The second form of aliasing we address happens when the HoG descriptor is used in conjunction with a linear classifier. Say that we want to build a classifier that can detect both black dogs on a white background and white dogs on black backgrounds. Now consider a cell in our HoG vector which appears near the boundary of the dog: WLOG say it is near the rightmost edge of the dog. Thus, vertical edges in this region are highly informative. Which HoG dimensions should our linear classifier give a high weight to? It could give high weight to the dimension corresponding to black on the left, white on the right (one of the 'contrast-sensitive' dimensions of HoG); this would help it detect black dogs. To detect white dogs, though, it would also need to give high weight white on the left, black on the right. However, this means that it would also give high weight to a white background and a narrow vertical stripe of black, or to a texture with many vertical lines. It is the classic X-OR problem for linear classifiers. Note that we have the same problem with the 'contrast-insensitive' dimensions of the HoG descriptor as well: these dimensions respond equally to the boundary of the dog and to texture or the vertical stripe.

2. Methods

We solve the first case by separating the diffuse gradients from the step-edges (Algorithm 1). The standard HoG algorithm begins by computing a gradient image in each of 9 orientations. Then at each pixel, it finds the gradient image with the largest gradient magnitude at that pixel, and then add the gradient's magnitude to the histograms corresponding to the maximal orientation. Algorithm 1 operates on the gradient images, before the histogram step. We separate the diffuse gradients from the step edges separately in each gradient image. Given a gradient image I, we perform the following convex minimization:

$$\underset{\alpha_1,\alpha_2}{min}||I - \alpha_1 - \alpha_2 \otimes \mathcal{G}(\sigma)||_2 + \lambda_1||\alpha_1||_1 + \lambda_2||\alpha_2||_1 \tag{1}$$

Here, α_1 becomes sensitive to the step edges, and α_2 the diffuse gradients. $\mathcal{G}(\sigma)$ is a Gaussian filter of size σ, and $\otimes$ denotes convolution. $||\cdot||_2$ denotes the standard Frobenius norm, while $||\cdot||_1$ is the elementwise $L1$ norm, which induces sparsity. λ_1 and λ_2 are constants, where generally $\lambda_2 < \lambda_1$, since α_1 can explain any gradient whereas α_2 can only explain diffuse ones. Given the solution to this minimization, we combine the set of α_1's for every gradient orientation and compute a HoG descriptor as if the α_1's were the gradient representation of an image. We repeat the process for the α_2's, and concatenate the two HoG representations. Thus, the dimensionality for each patch doubles. For a single 500-by-500 gradient image, the minimization step tends to take about 5 seconds in Matlab, and must be repeated 9 times at every scale of the HoG pyramid. Thus, while it is considerably slower than computing the original HoG representation, it is not prohibitively slow for many applications. An example result is given in Figure 1. Naturally there is a large space of similar algorithms, for example using different norms, or using greedy methods such as the retinex [4] or matching pursuit [1] to separate the types of gradients, and each has advantages and disadvantages. Greedy algorithms tend to give similar results but are less stable.

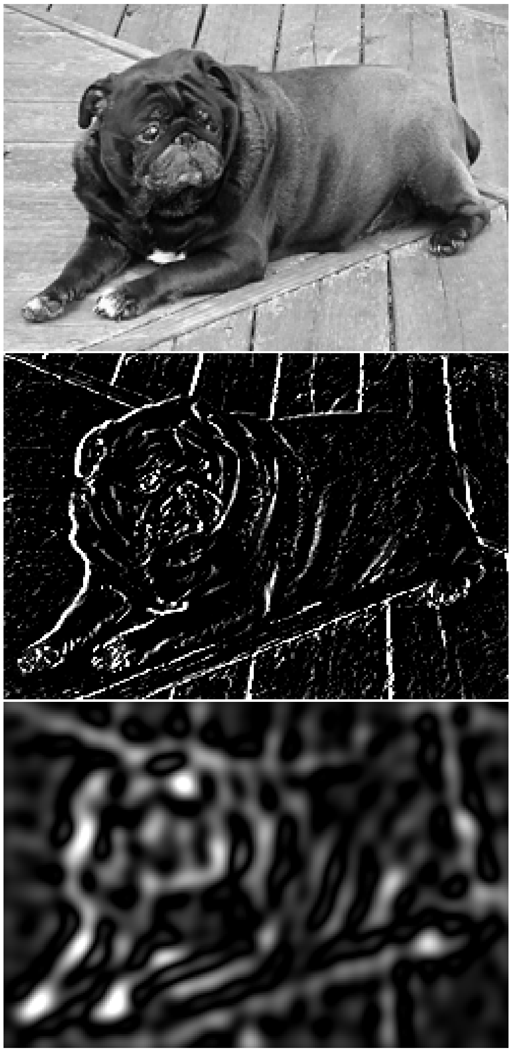

Figure 1. Example run of the first algorithm on a gradient image at one orientation. Top is the original image. From it, we extracted one gradient image (in this case, the gradient was from the top-left to bottom right, meaning edges running diagonally from the bottom left to top right are accentuated). The middle shows the extracted step edges, and the bottom shows the extracted diffuse gradients.

Our proposed solution to the second case of aliasing (Algorithm 2) is considerably simpler. We must only solve the X-OR problem. To do this, we add an extra dimension to the HoG descriptor of each cell at each orientation (thus, 9 extra dimensions per cell, approximately a 30% increase in dimensionality). The value for a given cell at a given orientation is the minimum of the two 'contrast-sensitive' dimensions at that orientation for that cell. Thus, a linear classifier that wishes to distinguish between object boundaries and stripes or texture may simply put a *negative* weight on this feature dimension.

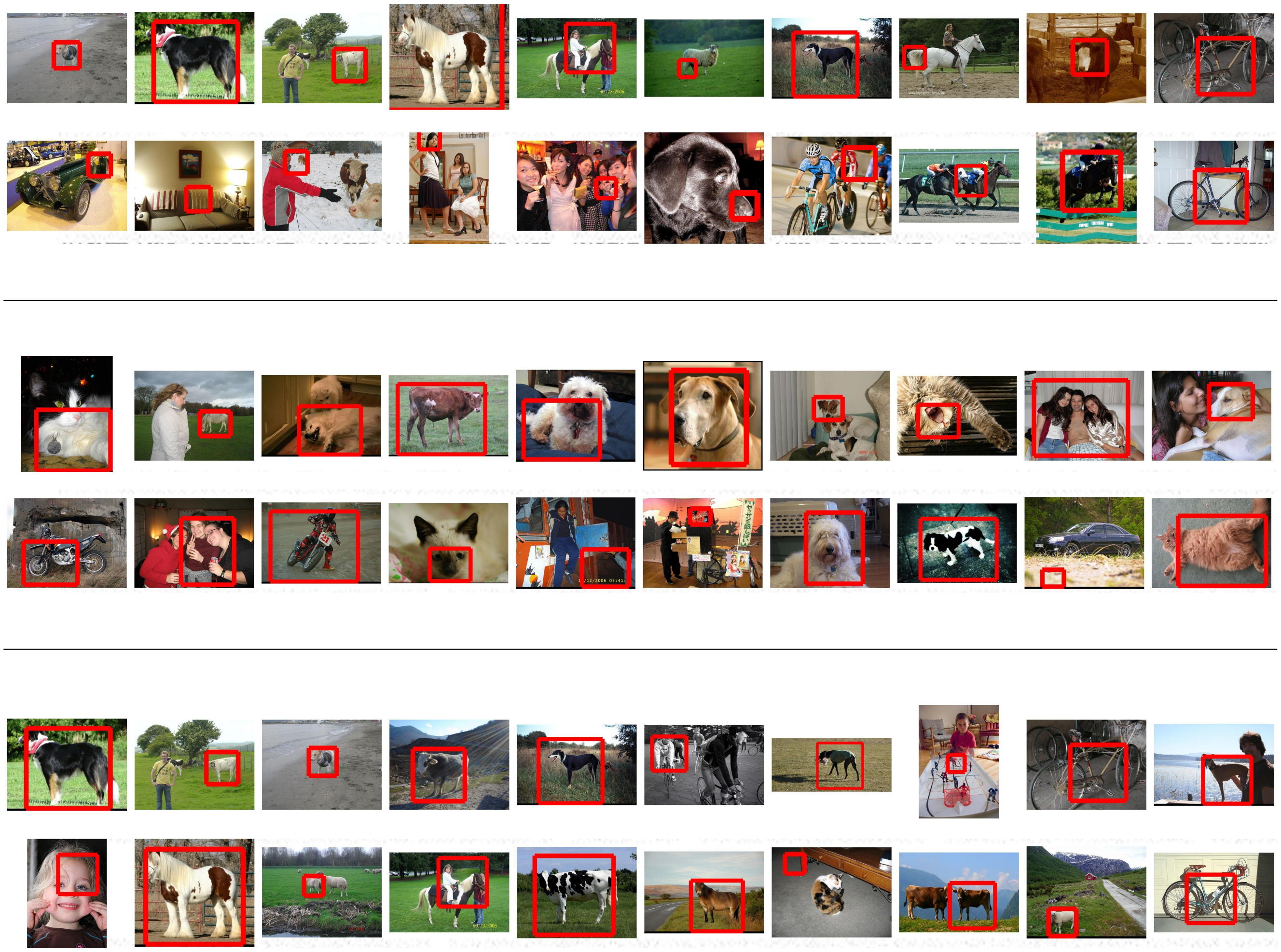

Figure 2. The top 2 rows show the top 20 detections for a detector using the baseline HoG implementation, ordered (left-to-right, first row before second) by the SVM score for the window. The third and fourth rows show the same for the descriptor computed with Algorithm 1, which attempted to separate diffuse gradients from step edges. The fifth and sixth show the results for the descriptor computed via Algorithm 2.

3. Results

The Algorithm 1 was unsuccessful. Figure 1 shows that the algorithm is somewhat successful in separating step edges from diffuse gradients, but the separation isn't nearly as clean as one would like. Unfortunately, the cleaner the separation becomes, the more sensitive it becomes to noise or blur.

To test both algorithms, we implemented a simple Dalal-Triggs-style detector [2] and trained it using the Pascal VOC 2007 [3] training set. We then fired these detectors on the Pascal test set, and showed the top detections (Figure 2). Qualitatively, the most Dog-like bounding boxes for Algorithm 1 are worse than the original. There are three likely reasons why this might have happened. First, the increased dimensionality may have led to overfitting. Second, the instability of the representation may have led to poor generalization. Third, the diffuse and step-edge components do not necessarily give different information: for example, most convex objects will have diffuse gradients that are nearly parallel to the step edges that indicate the object boundary. In these cases, separating the two types of gradients may be conterproductive.

Algorithm 2 was somewhat more successful: qualitatively, more of the top 20 detections seem to be for the bodies of animals, and fewer of the top detections seem to involve busy textured regions. This supports the idea that the algorithm is now able to differentiate edges from texture. However, more research is needed to verify that this is indeed useful.

4. Conclusions

We have proposed two methods for improving the HoG descriptor. This first attempted to separate diffuse gradients from step edges. While some success was attained in separating the two types of gradients, we have not yet found evidence that it is useful for object recognition. We theorize that a complete separation of the two types of gradients may be unhelpful, but modeling the relationships between the them may be useful.

We also proposed a method of separating contours from textures in certain cases. Overall this method appears promising, both based on results and on the simplicity of its implementation.

References

[1] F. Bergeaud and S. Mallat. Matching pursuit of images. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 53–56. IEEE, 1995.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, 2005.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[4] Z. Rahman, D. Jobson, and G. Woodell. Multi-scale retinex for color image enhancement. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 1003–1006. IEEE, 1996.