HOG: Histogram of Oriented Gradients

Olivia Wilson and Tom Ladyman

12^{th} February 2016

Introduction and Motivation

- Histogram of oriented gradients. Collect histograms of the gradient direction from areas of the image and use these histograms to classify and detect objects
- Key point detectors are inconsistent on humans even on consecutive frames [1]
- HOG provides a characteristic result for humans
- Can be optimised to run in parallel or fast HOG can be used for speed
- Output is easily classified with an SVM

[1] http://bengal.missouri.edu/ duanye/cs8690/lecture-notes/HoG.pdf

Method

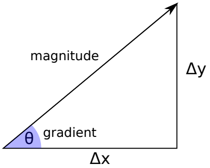
HOG is calculated through the following ordered steps:

1. Create gradient and magnitude images
2. Create overlapping blocks (size chosen by user) on the image
3. Split each block into cells (size chosen by user)
4. From each cell create orientation histogram
5. Normalise the histograms by block
6. Concatenate all the histograms together

The concatenated histograms can then be directly input into a classifier such as an SVM.

Method

Create gradient and magnitude images



The original image is convolved over using the 1-D point derivatives $[-1, 01]$ and $[-1, 01]^T$. This creates a first order differentiation images, gx (gradient of x) and gy (gradient of y) for the first and second kernels respectively. The following equations are then carried out element-wise on the arrays.

$$\text{gradient} = \arctan\left(\frac{gy}{gx}\right) \tag{1}$$

$$\text{magnitude} = \sqrt{gx^2 + gy^2} \tag{2}$$

Method

Sample blocks (size chosen by user) from image that overlap

- A block size is chosen by the user, a size of (16,16) is often used in the literature.
- R-HOG uses rectangular cells and C-HOG uses circular cells.

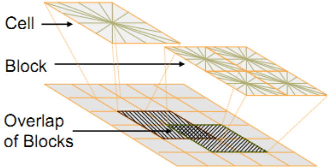


Figure 1: Demonstration of how blocks and cells are sampled and split.[2]

[2]Reproduced from
http://bengal.missouri.edu/ duanye/cs8690/lecture-notes/HoG.pdf

Method

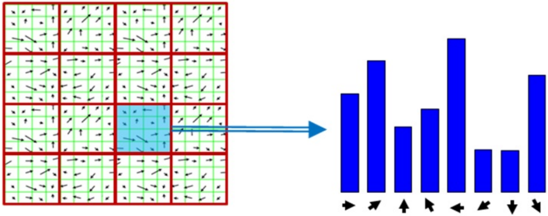


Figure 2: Demonstration of how a given cell is transformed into a histogram of orientations. [4]

[3] Reproduced from http://gilscvblog.com/2013/08/18/a-short-introduction-to-descriptors

[4] Reproduced from http://gilscvblog.com/2013/08/18/a-short-introduction-to-descriptors

Implementation and Choices

HOG was implemented from scratch using python and a number of python modules (skimage, numpy, scipy and PIL)

- First, the gradient and magnitude images are calculated before being split up to ensure less duplication of effort
- `skimage.util.view_as_windows` is used to create windows of data with a given overlap (avoids copying of data)
- Blocks, cells, histograms are split into their own objects and have only the methods pertaining to them, meaning this code could be used in its components as well. (i.e. the Cell code contains the cell array and an array of block positions compared to cell array)
- Using `%timeit` within `ipython` our version takes 813ms to create HOG compared to 75.3ms for the `skimage` version
- Hosted on github at https://github.com/tladyman/HOG

Demonstration

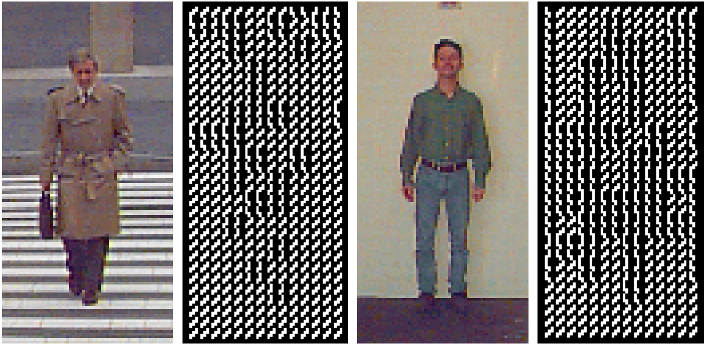


Figure 3: The largest angle for each cell plotted in place of the cell [5]

[5] Original images from MIT pedestrian database
http://cbcl.mit.edu/software-datasets/PedestrianData.html

Summary and Conclusions

- Image split into cells
- Cells are normalised by overlapping blocks
- Orientation histograms for each cell are produced
- Characteristic histogram pattern for upright humans
- Useful for other object detection
- Fairly fast and easily input to a classifier
- Used for pedestrian detection (autonomous cars) and object detection
- Not patented

Interesting Resources

- HOGgles: a system demonstrating how a computer sees an HOG image by reconstructing images from their HOG representations.

Figure 4: A HOG image, the reconstructed image and the original [6]

[6]Reproduced from http://web.mit.edu/vondrick/ihog/

HOGgles

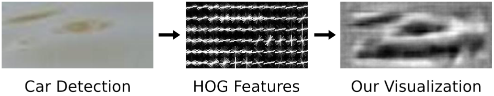


Figure 5: A misclassification and the reconstructed HOGgles image to show why [8]

[7] Reproduced from http://web.mit.edu/vondrick/ihog/
[8] Reproduced from http://web.mit.edu/vondrick/ihog/

References

- Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.
- Zhu, Qiang, et al. "Fast human detection using a cascade of histograms of oriented gradients." Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 2. IEEE, 2006.
- Vondrick, Khosla, et al. "HOGgles: Visualizing Object Detection Features" International Conference on Computer Vision (ICCV), Sydney, Australia, December 2013.