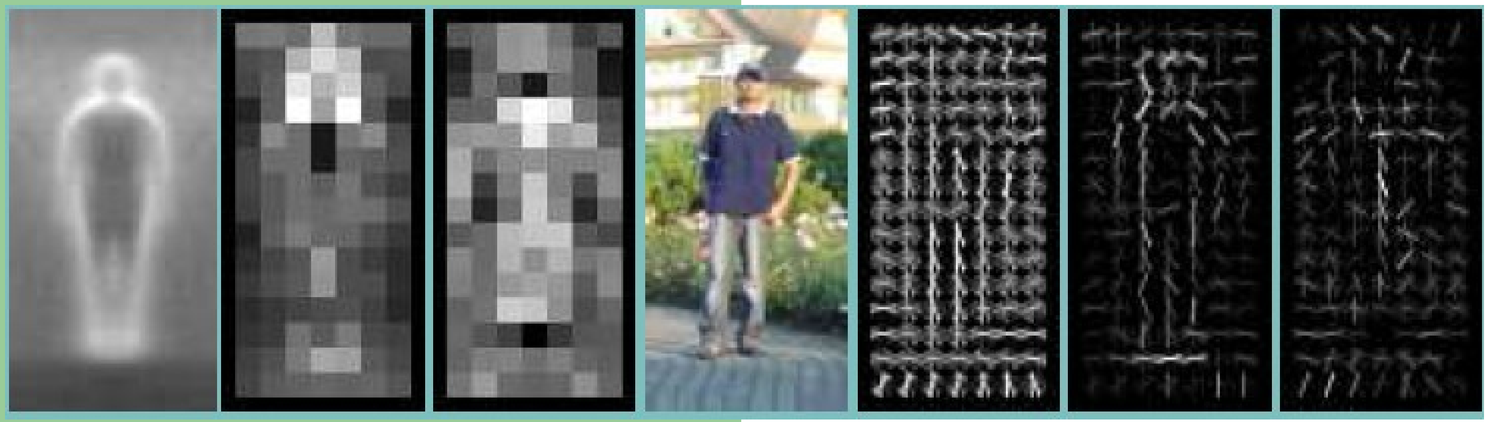
# Review and MATLAB Implementation of “Histogram of Oriented Gradients for Human Detection” by Dalal and Triggs



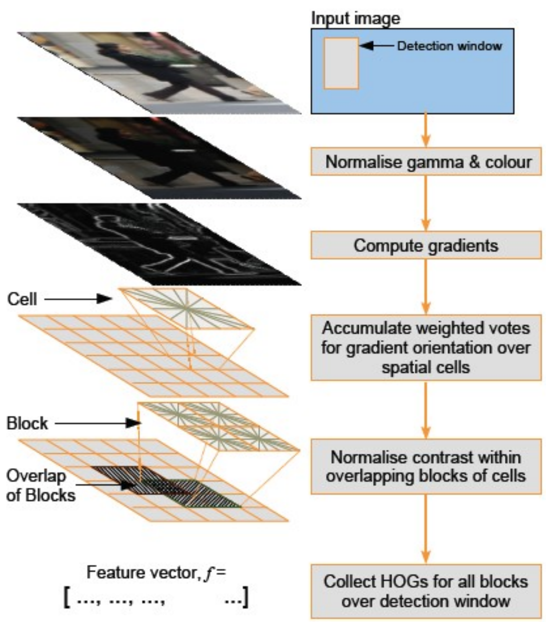
**Takeshi Tisdall-Yamada**

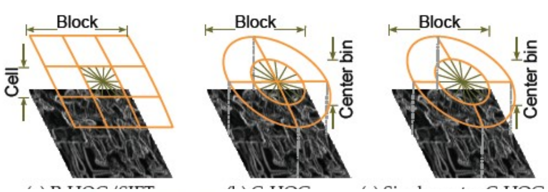
Advisors:
I.A. Kakadiaris, S.K. Shah, Z. Zeng
Computational Biomedicine Lab

# Introduction

- Histogram of Oriented Gradients (HOG): an image descriptor based on the image's gradient orientations
- HOG has been used by Dalal and Triggs for pedestrian identification
- Potential applications of pedestrian identification:
  -Pedestrian traffic monitoring, vehicle safety systems
  -Crowd control, pedestrian monitoring for security systems
- Algorithm has 2 main phases:
  -Descriptor formation
  -Training and classification

# Phase I : Descriptor Formation




- Normalize gamma and color
- 1D gradient filters compute orientation
- Form histograms of gradient orientations over spatial cells
- Group cells into overlapping blocks (R-HOG or C-HOG) and normalize
- Descriptor - HOGs of all blocks within the window

# Phase II : Training and Classification

- Dataset is split intro training and test sets
- Images containing pedestrians (positive) and images without pedestrians (negative)
- HOG descriptors formed
- Support Vector Machine (SVM) training
- Classification using the trained SVM

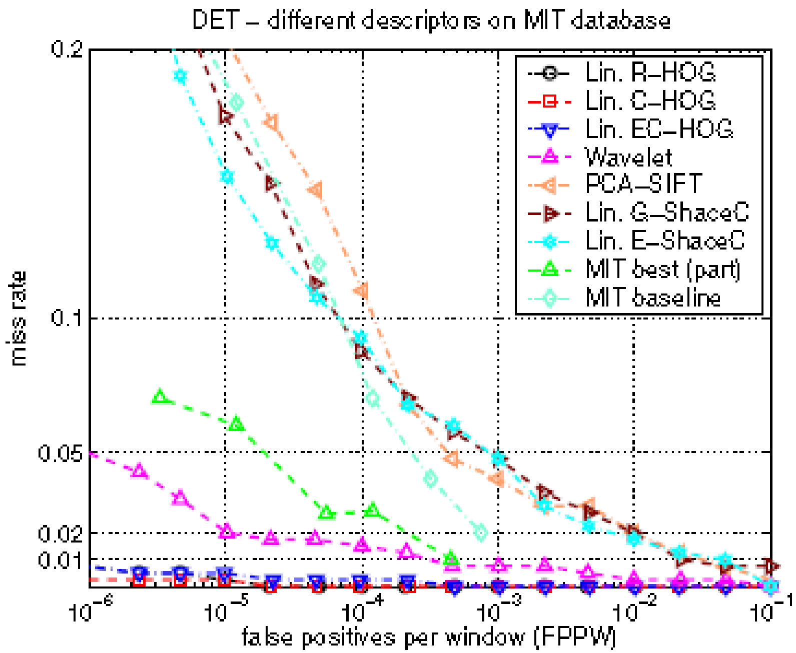
# Data Sets

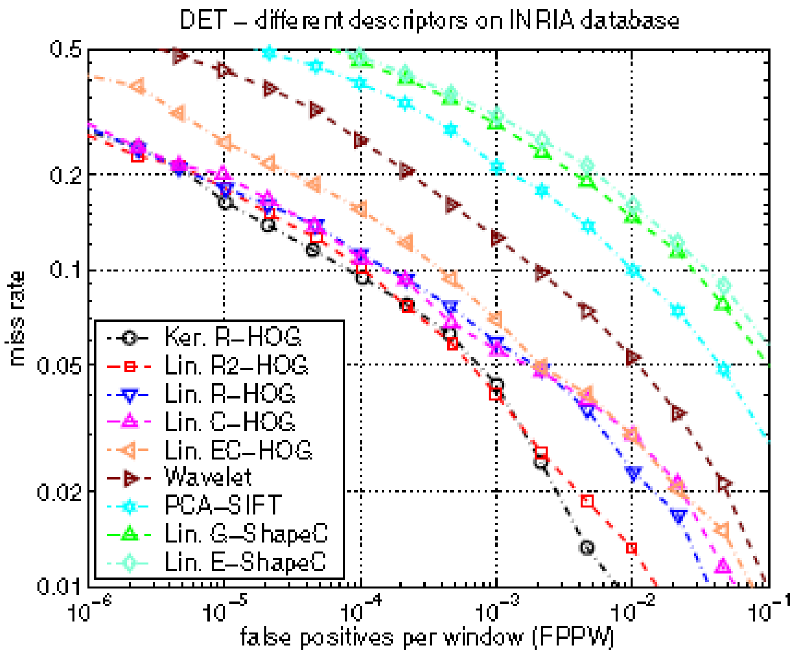
- MIT pedestrian database:
  - 509 training images, 200 test images
  - Urban setting, limited pose variation, viewed only directly from the front or back
- INRIA pedestrian database:
  - 1805 images
  - Wide variation in pedestrian pose, number of pedestrians, and setting

-5 sample images from the MIT database

-Sample images from INRIA database

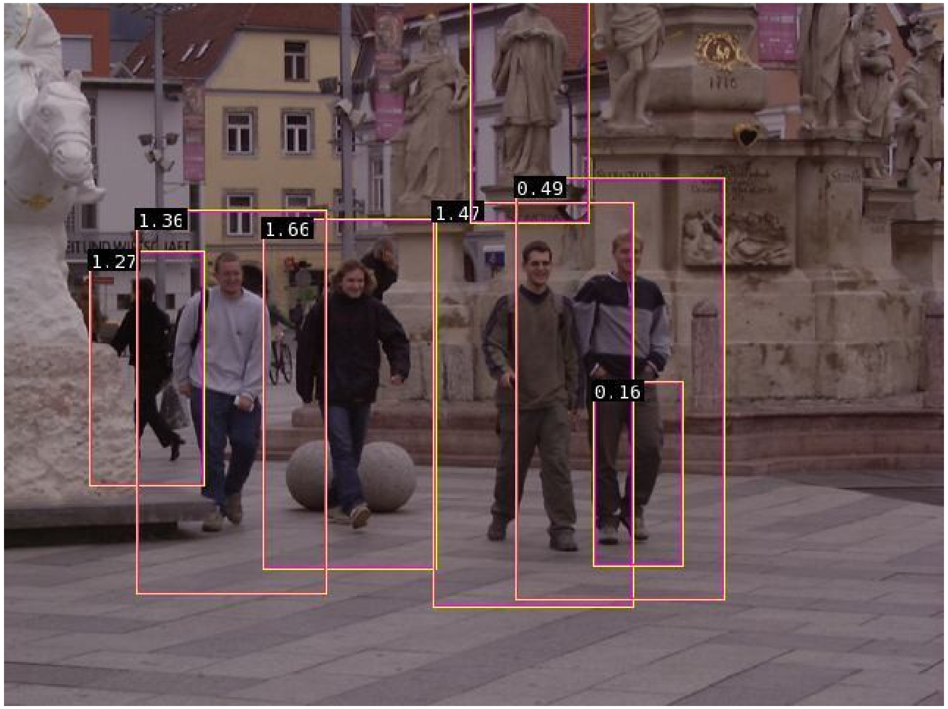
# Results




- Results using HOG on the MIT database (left) and INRIA database (right).
- HOG outperforms other descriptor techniques
- R-HOG block geometry is optimal

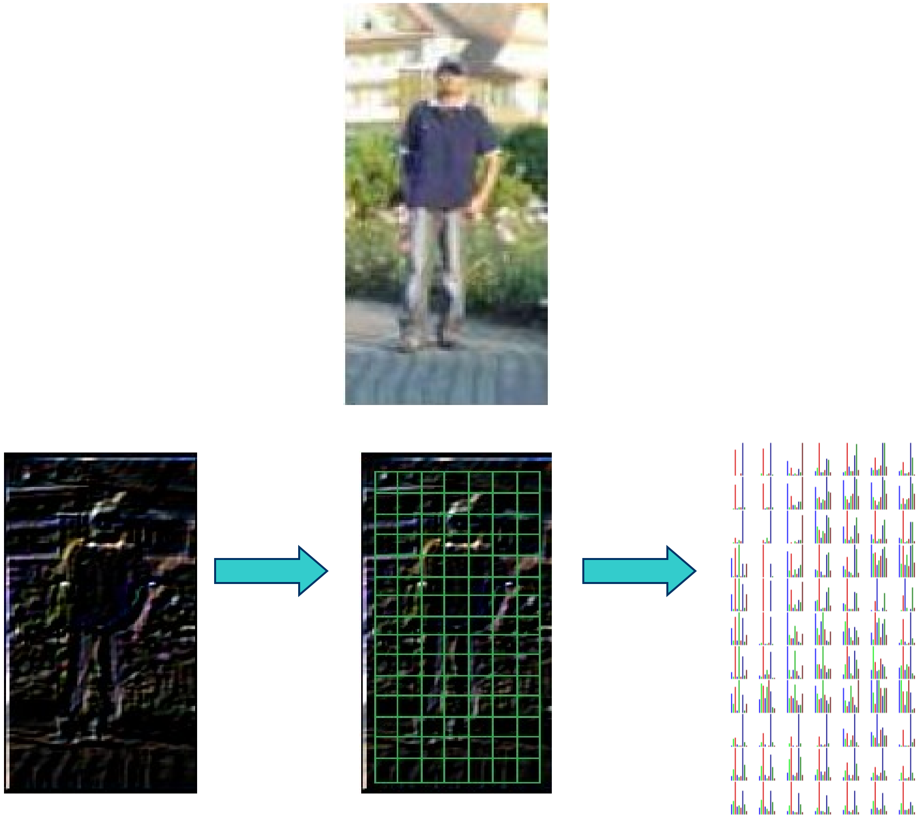
# Sample Positive Identification Result




- INRIA Object Localization Toolkit
- Image scanned at different scales using varying window size

# Implementation Details

- HOG descriptor is based on dominant edge orientations
- Edge detection applied
- Image divided into cells
- Histograms of edge orientations compiled



# Implementation Contd.

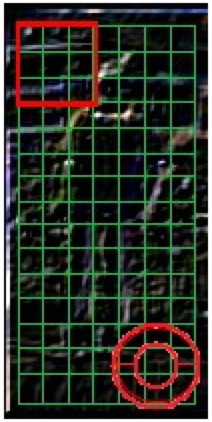
- Block normalization - corrects local contrast variation
- Histograms for the cells of each block are normalized:

L2-norm: $$f = \frac{v}{\sqrt{||v||_2^2 + e^2}}$$

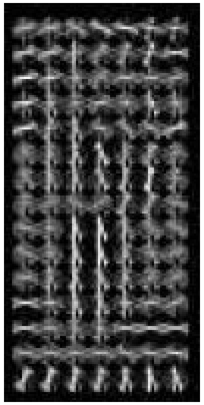
L1-norm: $$f = \frac{v}{(||v||_1 + e)}$$

L1-sqrt: $$f = \sqrt{\frac{v}{(||v||_1 + e)}}$$

- Block normalized histograms form the HOG descriptor



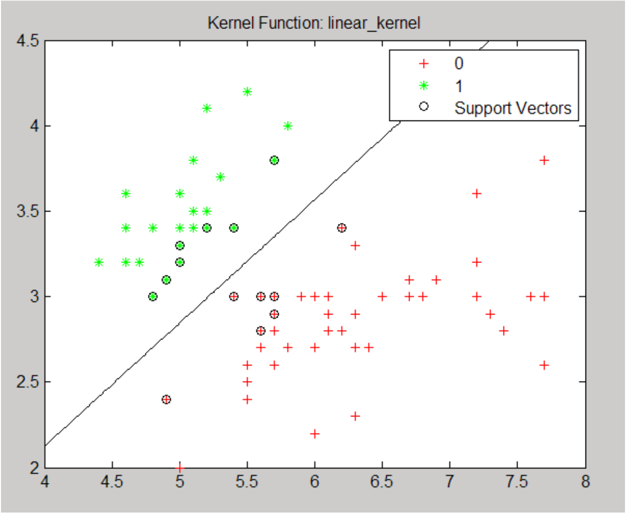
-Cells grouped into potential R-HOG or C-HOG blocks

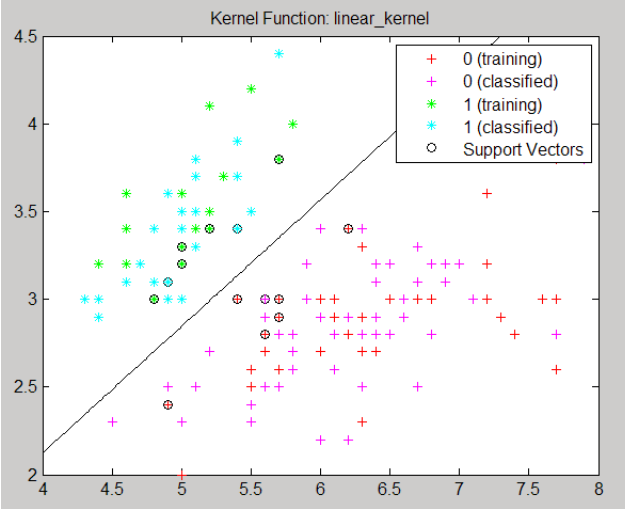


-Final image descriptor using R-HOG

# Support Vector Machines

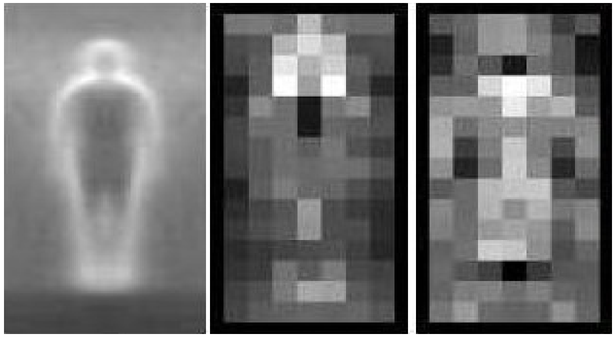
- Used extensively in machine learning, data-mining, classification and ranking
- Data clusters with other data of same class
- Optimal separating hyperplane between sets (or classes) of data
- Collection of data and calculation of the hyperplane = “learning” or “training” phase
- The “trained” SVM model accepts test data
- SVMs support multiple classes, N-dimensional data

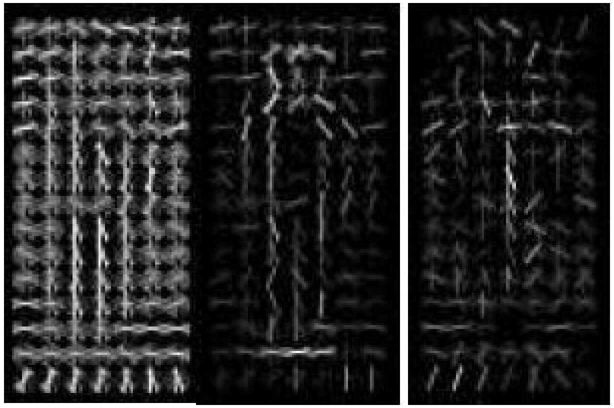


# Support Vector Machine Classification

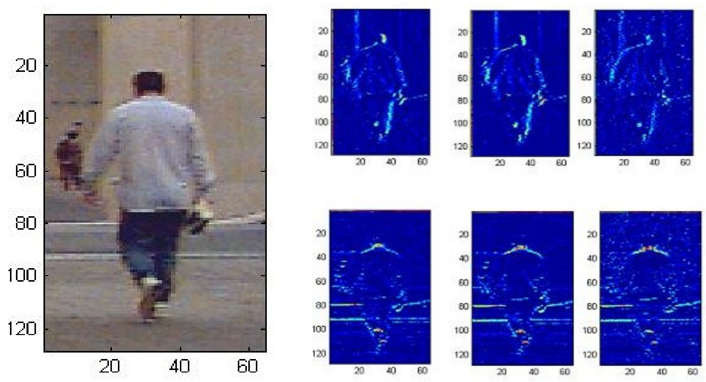
- The SVM is “trained”
- Negative training images scanned for false positives
- Descriptors for test images compiled
- Descriptor of unknown class is weighed against positive and negative SVM weights





# My MATLAB Implementation

- MATLAB – allows quick implementation, testing
- MATLAB - built in functions for image analysis, bioinformatics, statistics, etc.



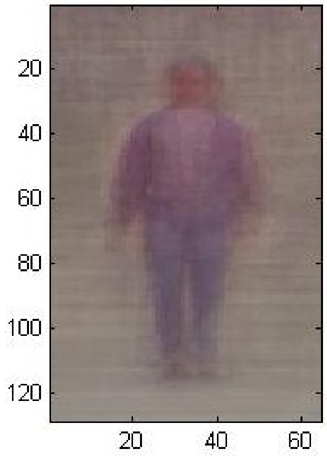
-Sample image from MIT pedestrian database (left)

-Image split into R, G, and B channels with horizontal gradient filter (top)

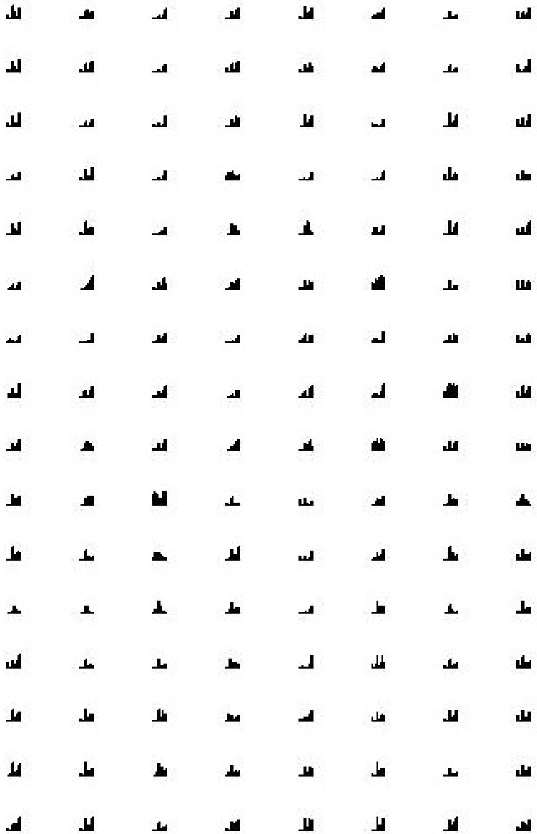
-R, G, B channels with vertical gradient filter applied (bottom)

# MATLAB implementation contd.

- Built in image/matrix manipulation, quick image processing.
- Images may be batch processed, combined, and displayed/output

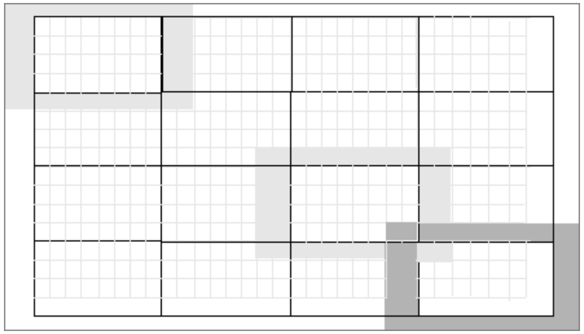



-"Average" image of first 75 images in the MIT pedestrian database



-Histograms of gradient orientations for each cell

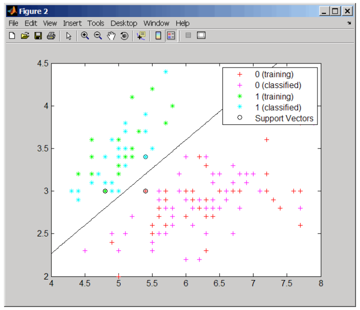
# Block Normalization and SVM



-Function for performing distinct block operations on overlapping blocks.

-blkproc(image, [m n],[mborder nborder],function)

```
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
svmStruct = svmtrain(data(train,:),groups(train));
classes = svmclassify(svmStruct,data(test,:),'showplot',true);
cp = classperf(groups);
classperf(cp,classes,test);
cp.CorrectRate
```

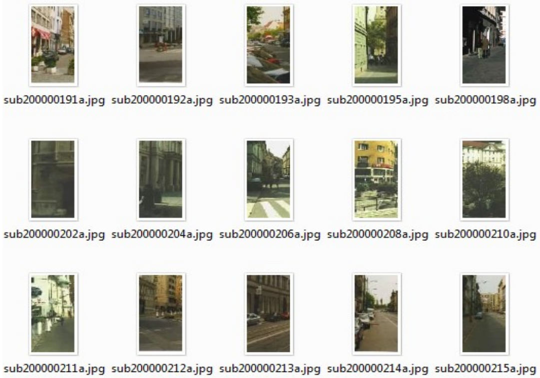



-Sample code shows MATLAB's built in SVM

-Plot of 2D SVM data

# SVM Data Sets

- MIT pedestrian database omits negative samples
- Used MATLAB to segment INRIA negative images into 64x128 samples
- Negative samples critical for robust SVM training




-Sample pedestrian-free image from the INRIA data set. Thumbnails of 64x128 subsections created with MATLAB.

# Next Steps

- Refine MATLAB implementation to more closely match the HOG paper
- Continue testing my datasets using the MATLAB generated descriptors
- Examine author's HOG C++ source code
- Test HOG descriptors on another database, investigating SVM classification accuracy for profile vs. face on images.

# Sources


- Dalal, N. Triggs, B. : *Histograms of Oriented Gradients for Human Detection,* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.
- Dalal, N. : *Finding People in Images and Videos*, PhD thesis 2006
- Wikipedia page on HOG: http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
- MATLAB help files
- Lab on HOG:

  http://users.utcluj.ro/~raluca/srf_2008/lab/04/lab_04e.pdf