Global Descriptors

(Session 3)

Francisco Roman

January 28^{th}, 2019

Outline

1. Image Description
2. Histogram of Oriented Gradients (HOG)
3. GIST

Global Descriptors 1/2

- Describe the image as a whole.
- Measure statistics of the intensities.
- Including contours, shapes, textures, colors.

Global Descriptors 2/2

- Generalize the full image into a single description.
- Includes context (background and other objects).
- Is this right or wrong? (car or person detection, digit recognition)

Popular descriptors

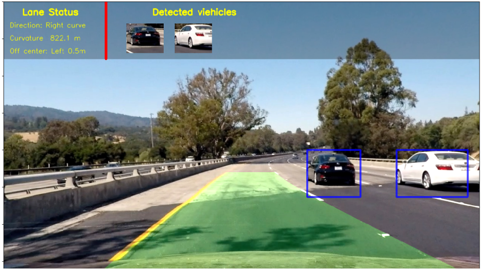
- Invariant Moments [Hu, Zernike]
  (Shapes, aka binary images).
- Histogram of Oriented Gradients (HOG) [Dalal and Triggs]
  (Human and car detection).
- GIST [Oliva and Torralba]
  (Scene recognition).

Outline

HOG

- (Dalal and Triggs, 2005).
  Navneet Dalal and Bill Triggs. "Histogram of Oriented Gradients for Human Detection". *CVPR*. 2005.
- Image descriptor for object detection.

- Detect pedestrians on static images.
- Also used for human and car detection on videos.



[Images by Satya Mallick, from Histogram of Oriented Gradients at Learn OpenCV].

Differences

Q: Remember the difference between recognition, classification, identification, localization, detection, segmentation?

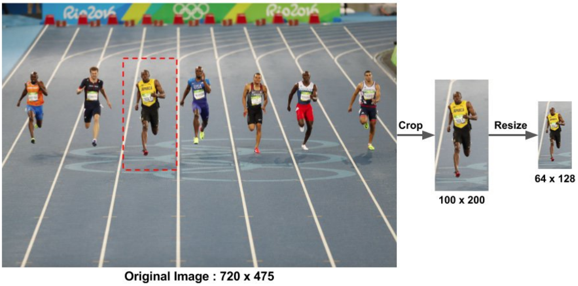
Differences

Q: Remember the difference between recognition, classification, identification, localization, detection, segmentation?

- Recognition: Broad ambiguous concept.
  Encompasses many specific tasks.
- Classification: Assign a label. Decide its class.
  Often the object spans most of the image.
- Identification: Finer (individual) twist of classification.
  i.e., who is this?
- Detection: Binary answer. Possible many objects in an image.
  i.e., is there a car?
- Localization: Find the coordinates $[x, y, w, h]$ of an object.
- Segmentation: Find all pixels representing an object.

HOG

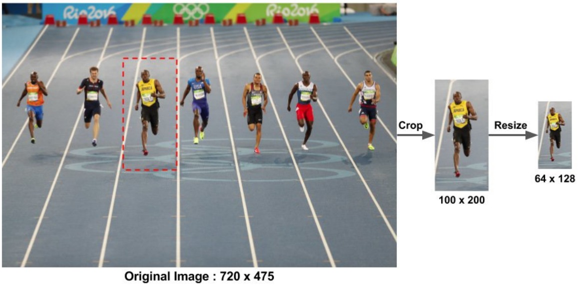
- Input (image section) 64×128 pixels.
  Sliding window and heat-map for larger images.



[Images by Satya Mallick, from HOG at learnopencv.org].

HOG

- Input (image section) 64×128 pixels.
  Sliding window and heat-map for larger images.

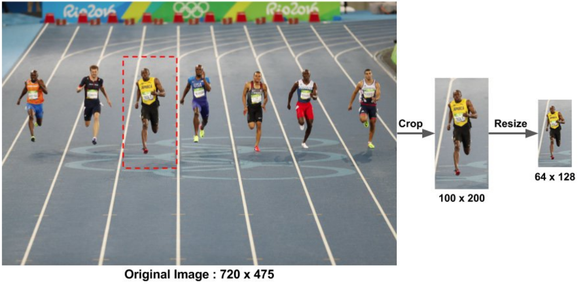


[Images by Satya Mallick, from HOG at learnopencv.org].

- Based on the counting of gradient orientation occurrences.
- Computed on a dense grid of uniformly spaced cells.

HOG

- Input (image section) 64×128 pixels.
  Sliding window and heat-map for larger images.



[Images by Satya Mallick, from HOG at learnopencv.org].

- Based on the counting of gradient orientation occurrences.
- Computed on a dense grid of uniformly spaced cells.

Q: What is the gradient orientation of a 2D1C signal (gray-scale image)?

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

What is m (what does it represent)?

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

What is m (what does it represent)?

$$\tan(\theta) = m$$

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

What is m (what does it represent)?

$$\tan(\theta) = m$$

So, how do we compute the orientation?

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

What is m (what does it represent)?

$$\tan(\theta) = m$$

So, how do we compute the orientation?

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

Gradient of 2D1C signal

Remember the formula?

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

What is m (what does it represent)?

$$\tan(\theta) = m$$

So, how do we compute the orientation?

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

The resulting orientation is bounded,

$$0^\circ \leq \theta \leq 180^\circ.$$

Intuition

- Characterize an image as a function of its gradient directions or edge orientations.

"Local object appearance and shape can be characterized by the distribution of local intensity changes or edge directions. Even without knowing the precise gradient magnitude or edge position".

– Dalal and Triggs.

Overview

Pipeline:

[Image by R. Cosman and T. Wang - The Cutting Edge of CV course at Stanford].

Gradient computation

Applying derivative kernels on horizontal and vertical directions.

$$k_{\delta_x} = [-1, 0, 1] \qquad \text{and} \qquad k_{\delta_y} = [-1, 0, 1]^T$$

- These vectors where the most promising kernels.
- Unsigned gradients, i.e., [0, 180] degrees.

Gradient computation

Applying derivative kernels on horizontal and vertical directions.

$$k_{\delta_x} = [-1, 0, 1] \qquad \text{and} \qquad k_{\delta_y} = [-1, 0, 1]^T$$

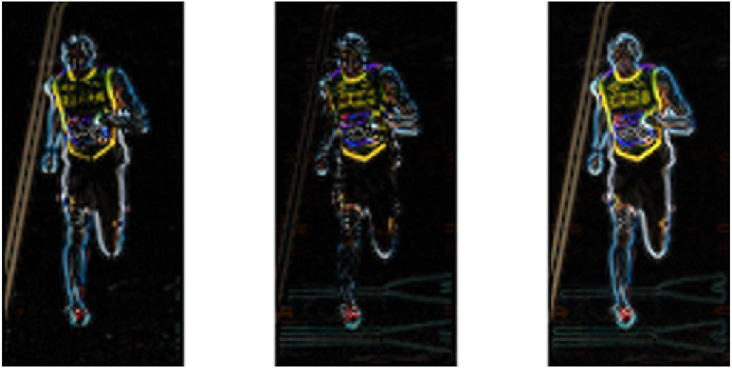
Input image | Horizontal gradient | Vertical gradient

- These vectors where the most promising kernels.
- Unsigned gradients, i.e., [0, 180] degrees.

Q: What about diagonal derivatives?

Gradients

Absolute value of x gradient, y gradient, and magnitude.



[Images by Satya Mallick, from HOG at learnopencv.org].

Q: How do we do for RGB images?

A: Proceed by channel. Characterize each pixel by the highest magnitude among the three channels and its corresponding angle.

Orientation binning

Quantification: creation of a frequency histogram of orientations.

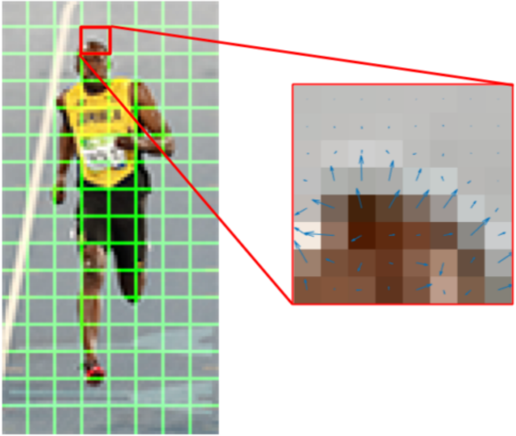
Cell: group of 8×8 pixels.

For each cell:

- Define the number of bins, i.e., 9.
- Count the number of pixel orientations within each bin.
- Use a weighted aggregation scheme (**magnitude**).
- Option: Split contribution between adjacent bins.

Orientation binning

Cells and gradients per pixel.



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

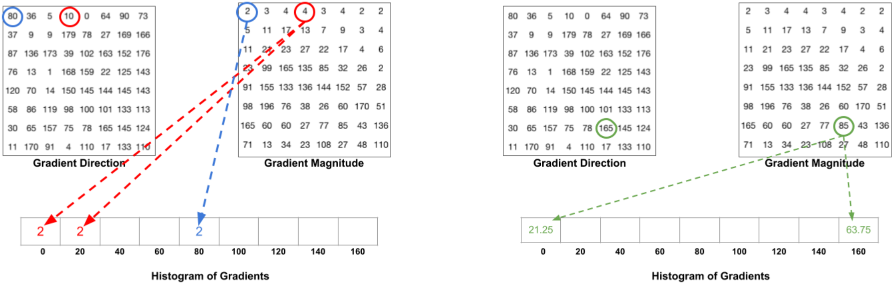
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

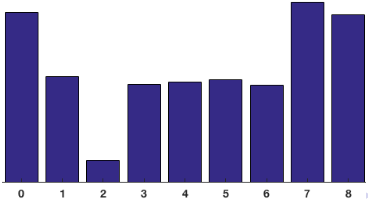
[Images by Satya Mallick, from HOG at learnopencv.org].

Orientation binning

Cell histogram construction:



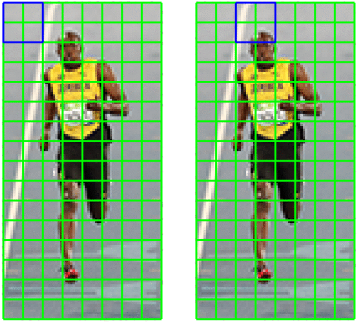
Histogram:



Block normalization

To add robustness against lighting changes.

- Define blocks of 16×16 pixels, i.e., 2×2 cells.
  This is a 36-D vector.
- Divide each element by the magnitude of the full vector.

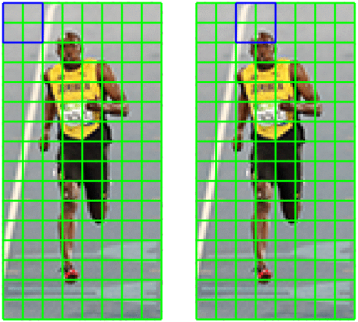


[Images by Satya Mallick, from HOG at learnopencv.org].

Block normalization

To add robustness against lighting changes.

- Define blocks of 16×16 pixels, i.e., 2×2 cells.
  This is a 36-D vector.
- Divide each element by the magnitude of the full vector.



[Images by Satya Mallick, from HOG at learnopencv.org].

Q: What is the result of summing up all 36 elements of the vector after normalization?

Final descriptor

Image representation, aka., HOG descriptor.

Concatenate vectors using an overlapping-block scheme:

- Slide the block definition by half its size, i.e., 8 pixels.
- At every step, concatenate the 36-D vector corresponding to the current block.
- Overlap $\implies$ repetition of information.

Final descriptor

Image representation, aka., HOG descriptor.

Concatenate vectors using an overlapping-block scheme:

- Slide the block definition by half its size, i.e., 8 pixels.
- At every step, concatenate the 36-D vector corresponding to the current block.
- Overlap $\implies$ repetition of information.

Q: What is the length of the final vector?

Final descriptor

Image representation, aka., HOG descriptor.

Concatenate vectors using an overlapping-block scheme:

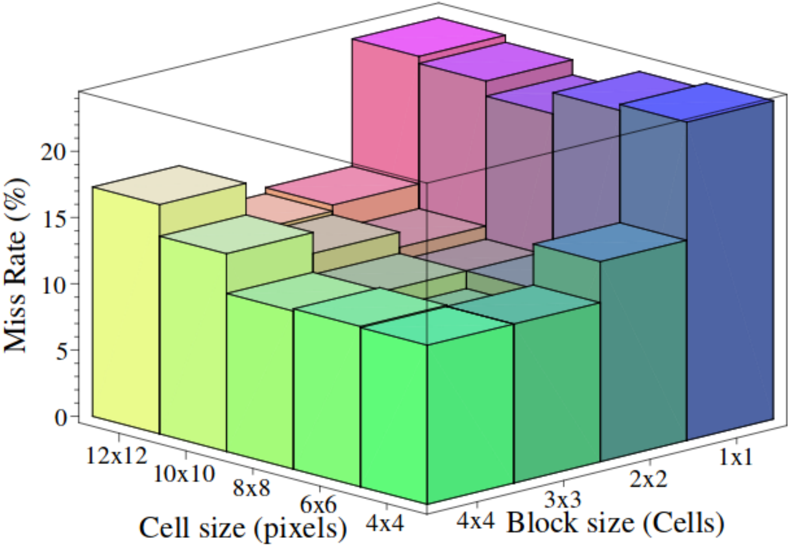
- Slide the block definition by half its size, i.e., 8 pixels.
- At every step, concatenate the 36-D vector corresponding to the current block.
- Overlap $\implies$ repetition of information.

Q: What is the length of the final vector?

A: Moving 8 pixels at a time, across 64×128 space, gives 7 horizontal steps and 15 vertical steps, thus 105 overlapping blocks. This results in a $105 \times 36 = 3780$-D vector.

Parameters

Best combination reported:



HOG in learnopencv.org

Look at:

https://www.learnopencv.com/histogram-of-oriented-gradients/

Outline

The GIST of an image

Spatial envelope

Aude Oliva and Antonio Torralba. "Modeling the shape of the scene: a holistic representation of the spatial envelope". *International Journal of Computer Vision*, 42(3):145–175, 2001.

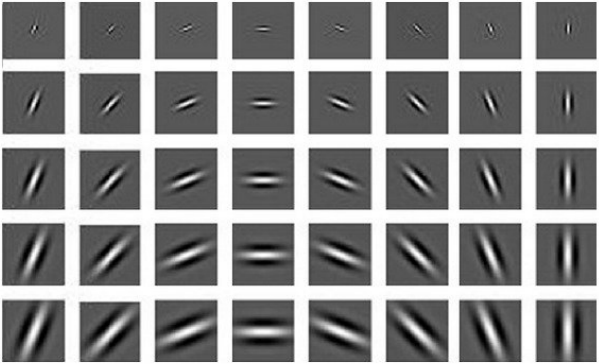
Applies a bank of filters and averages their local responses to produce an image signature (GIST).

- Global processing.
- Local (regional) description.
- Global representation.

Bank of filters

Gabor filters

- 2-D Gaussian kernel.
- Linear filter often used for texture analysis.
- Apply filters of different amplitude and orientation.
- Obtain different responses for different filters.



GIST descriptor

1. Convolve the image with 32 Gabor filters:
   4 scales, 8 orientations.
2. Produces 32 different feature maps (size?).
3. Divide each map into 4×4 regions.
4. Average the feature values within each region (output?).
5. Concatenate.

GIST descriptor

1. Convolve the image with 32 Gabor filters:
   4 scales, 8 orientations.
2. Produces 32 different feature maps (size?).
3. Divide each map into 4×4 regions.
4. Average the feature values within each region (output?).
5. Concatenate.

Q: What is the final length of the descriptor?

GIST descriptor

1. Convolve the image with 32 Gabor filters:
   4 scales, 8 orientations.
2. Produces 32 different feature maps (size?).
3. Divide each map into 4×4 regions.
4. Average the feature values within each region (output?).
5. Concatenate.

Q: What is the final length of the descriptor?
A: 16 averaged values per feature map, times 32 maps $\implies$ 512-D.

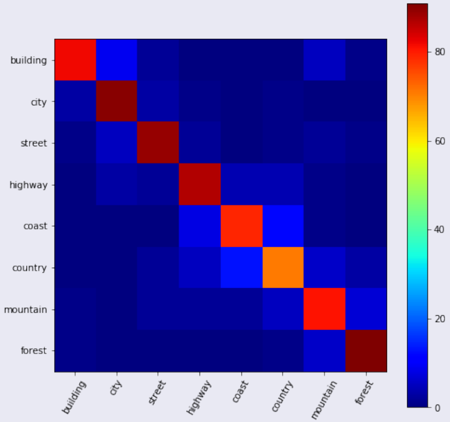
GIST application

Classification of outdoor scenes.
Dataset of 2600 images (256×256 pixels).
8 classes $\implies$ 325 images per class.

GIST application

Training an SVM classifier with 100 samples, and testing on the rest.

Confusion Matrix



Average along the diagonal: 83.7.