TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

CS231. Nhập môn Thị giác máy tính

Histogram of Oriented Gradient
HOG

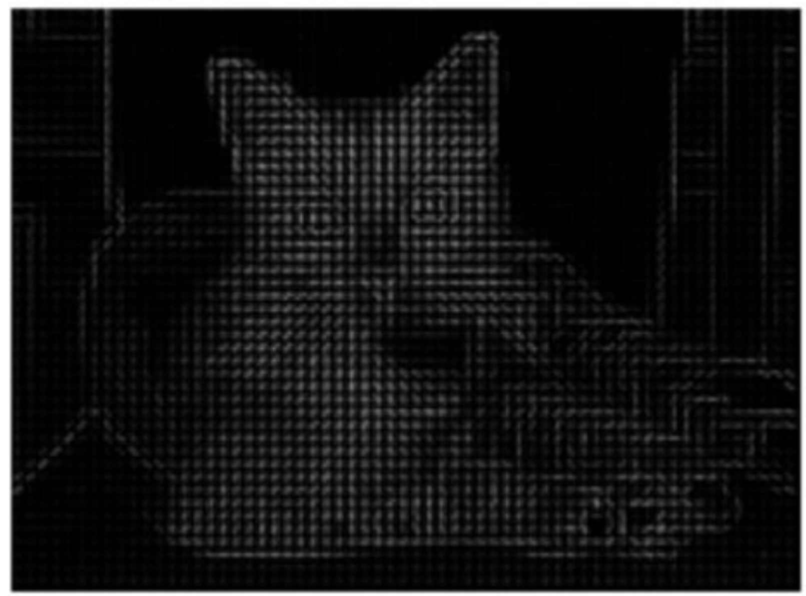
Mai Tiến Dũng

Introduction

- HOG (Histogram of Oriented Gradient) : là một phương pháp mô tả đặc trưng

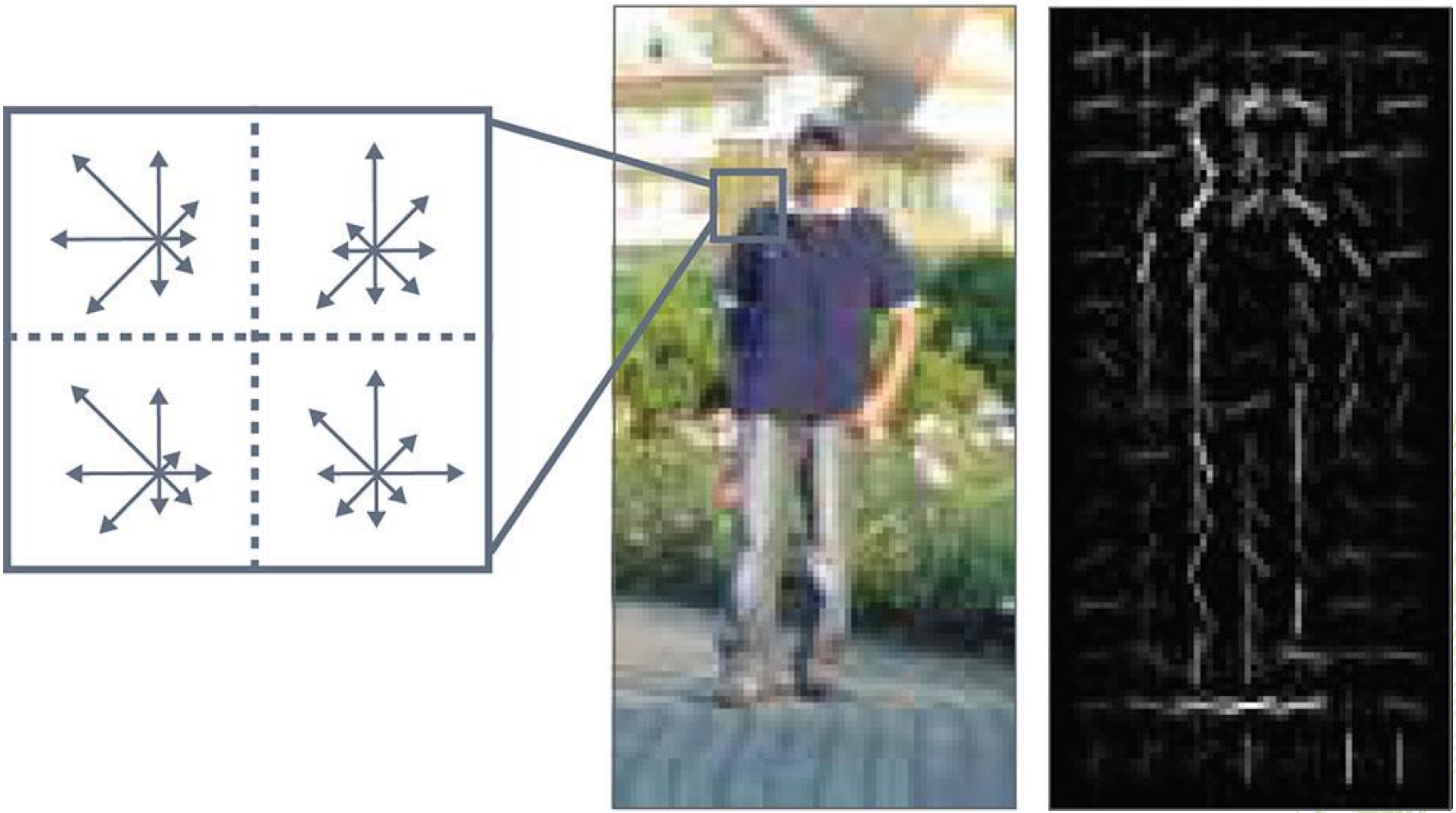
Input image

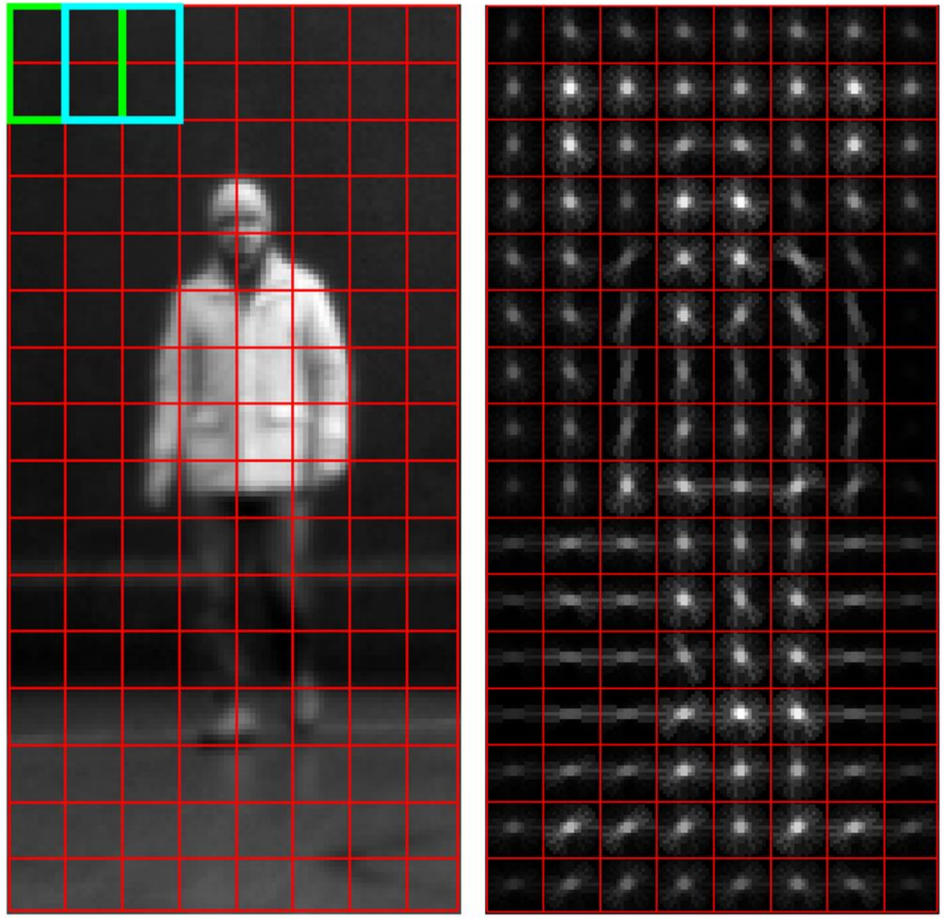
Histogram of Oriented Gradients



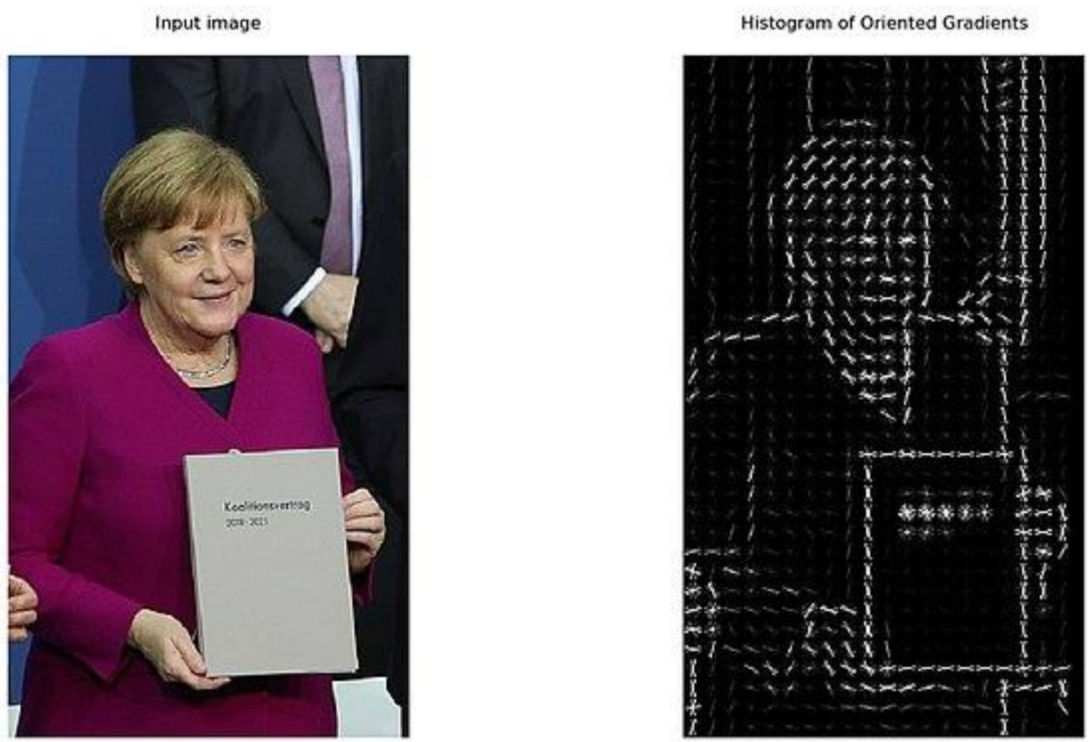
Nguyên lý

- Sử dụng gradient magnitude và gradient orientation để lưu thông tin ảnh.
- Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con(cells)
- Mỗi cell,tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell.
- Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu.



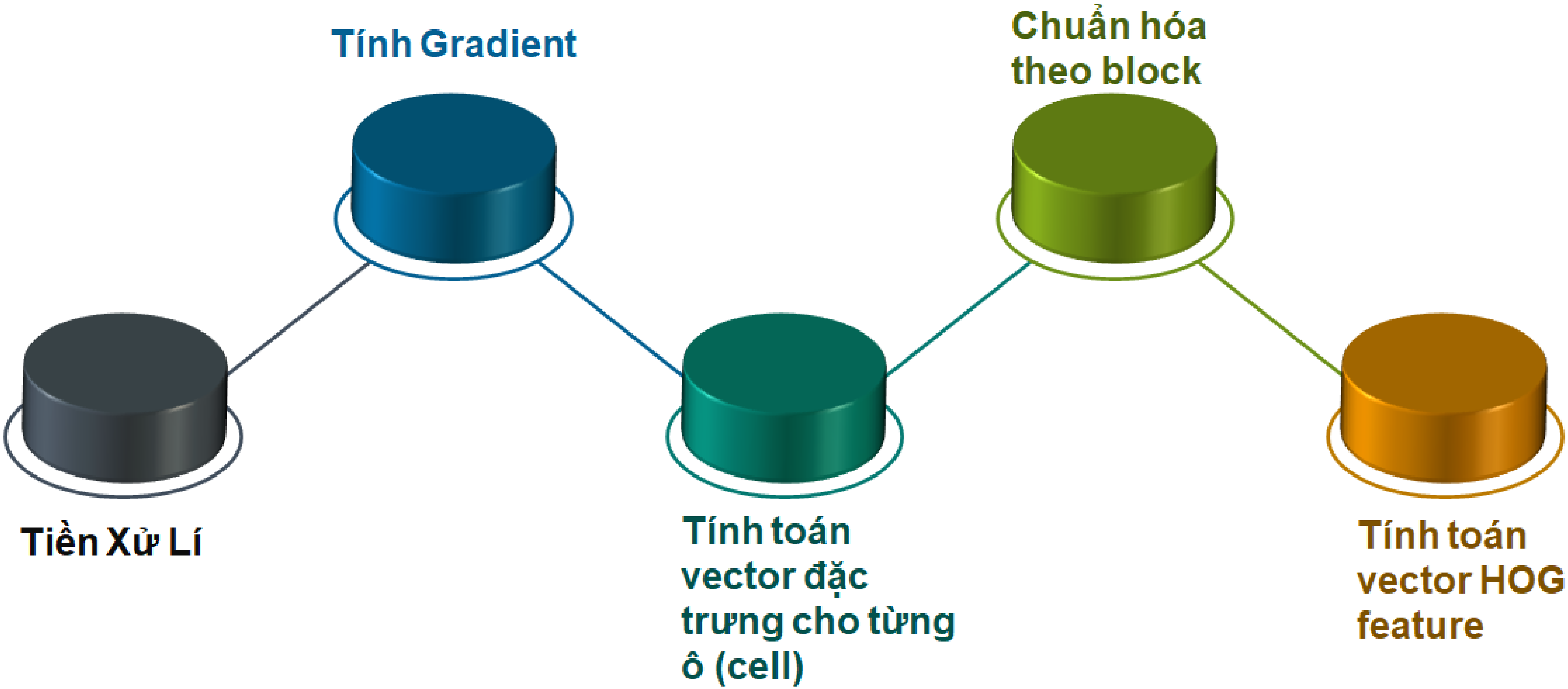


(a) (b)



https://de.wikipedia.org/wiki/Histogram_of_oriented_gradients

Các bước thực hiện

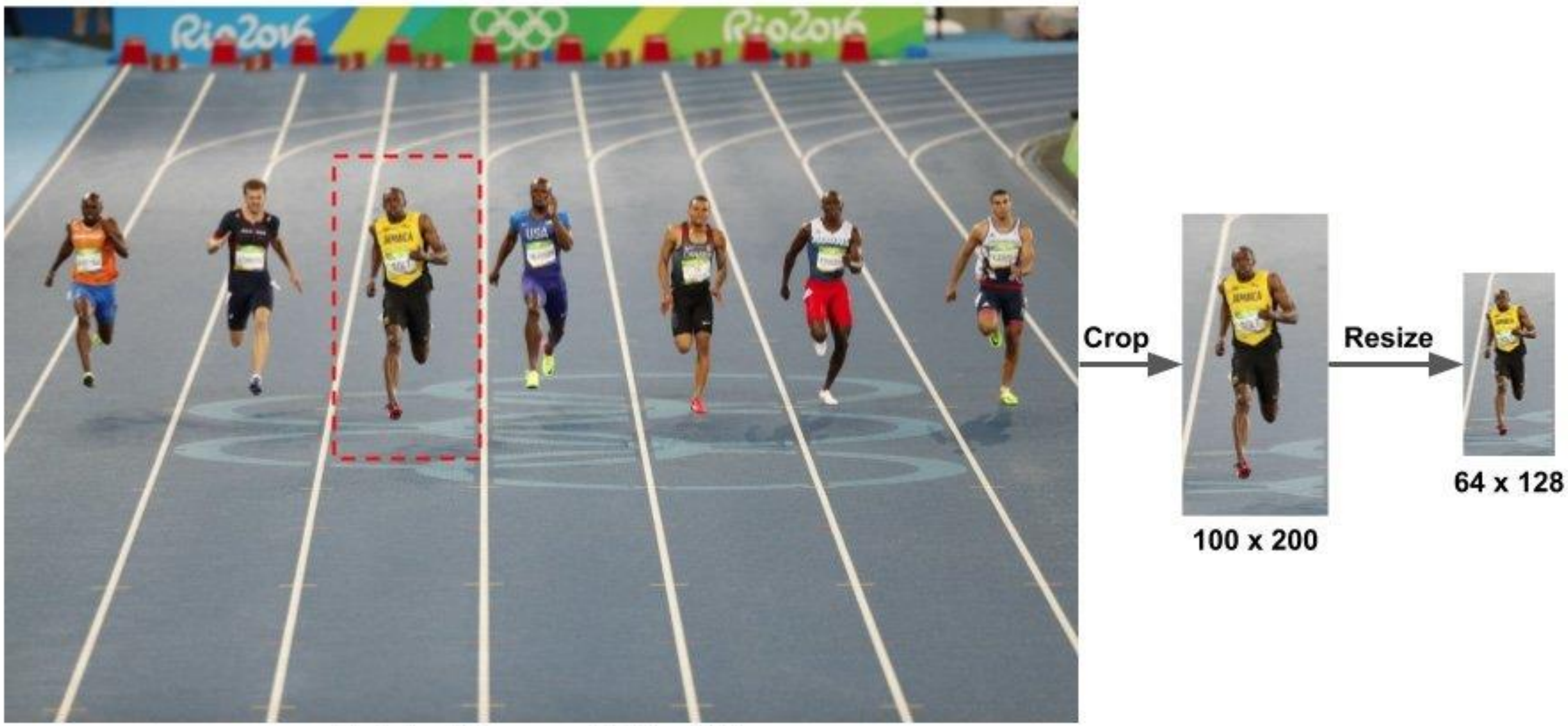


Step 1 : Preprocessing

- As mentioned earlier HOG feature descriptor used for pedestrian detection is calculated on a 64×128 patch of an image.
- The patches need to have an aspect ratio of 1:2. For example, they can be 100×200, 128×256, or 1000×2000 but not 101×205.

Step 1 : Preprocessing

- A large image of size 720×475 → selected a patch of size 100×200 → cropped out of an image and resized to 64×128



Original Image : 720 x 475

Step 2 : Calculate the Gradient Images

Đạo hàm theo chiều ngang: $G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{I}$

Đạo hàm theo chiều dọc: $G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \mathbf{I}$

Gradient Direction: $\theta = \mathrm{acrtan}(\frac{G_y}{G_x})$ → Là độ lớn góc giữa vector gradient x và y

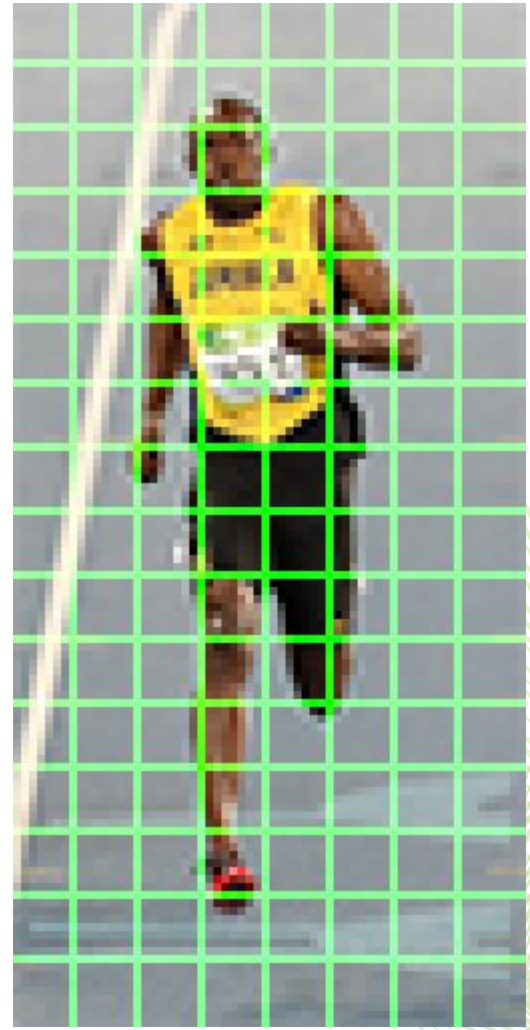
Gradient Magnitude: $G = \sqrt{G_x^2 + G_y^2}$ → Là chiều dài của vector Gradient theo phương x và y

Step 2 : Calculate the Gradient Images

Left : Absolute value of x-gradient. Center : Absolute value of y-gradient. Right : Magnitude of gradient.

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- In this step, the image is divided into 8×8 cells and a histogram of gradients is calculated for each 8×8 cells.
- An 8×8 image patch contains 8x8x3 = 192 pixel values. The gradient of this patch contains 2 values (magnitude and direction) per pixel which adds up to 8x8x2 = 128 numbers.
- These 128 numbers are represented using a 9-bin histogram which can be stored as an array of 9 numbers.



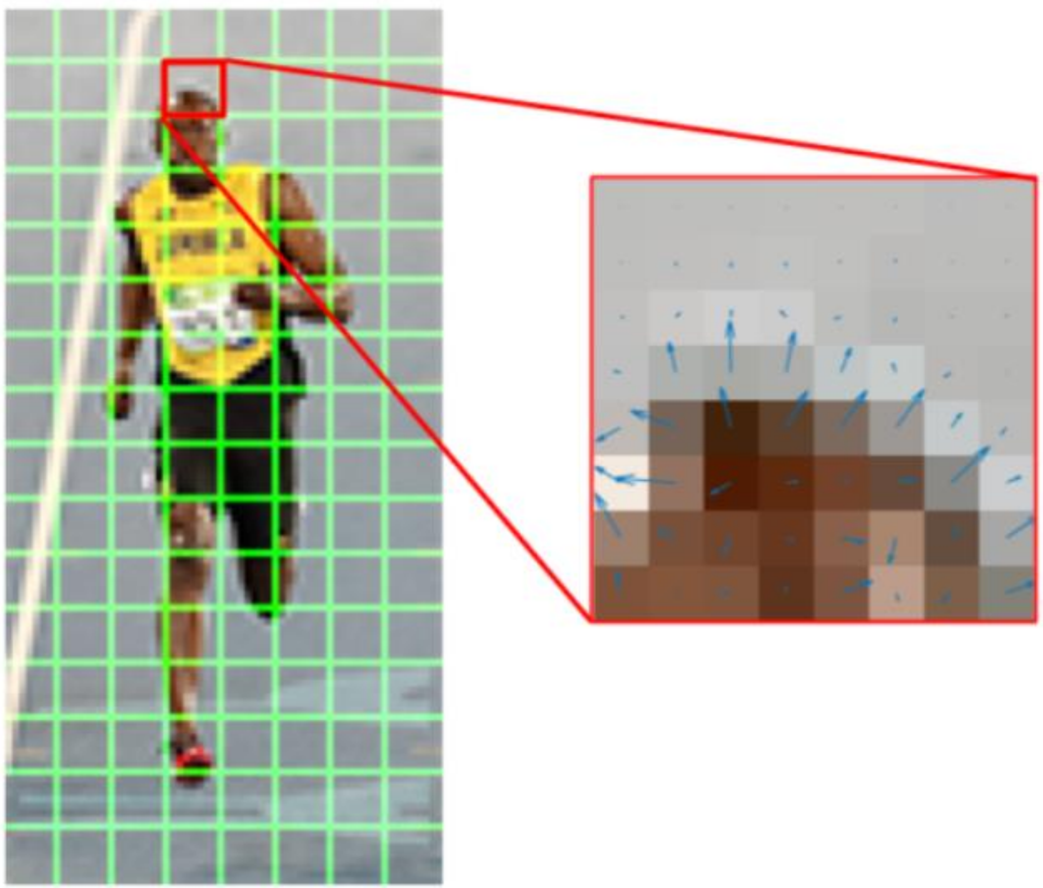
8×8 cells of HOG. Image is scaled by 4x for display.

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- Why we have divided the image into 8×8 cells
  - provides a compact representation
  - less sensitive to noise

- But why 8×8 patch ? Why not 32×32 ? It is a design choice informed by the scale of features we are looking for. HOG was used for pedestrian detection initially. 8×8 cells in a photo of a pedestrian scaled to 64×128 are big enough to capture interesting features (e.g. the face, the top of the head etc.).

Step 3 : Calculate Histogram of Gradients in 8×8 cells



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

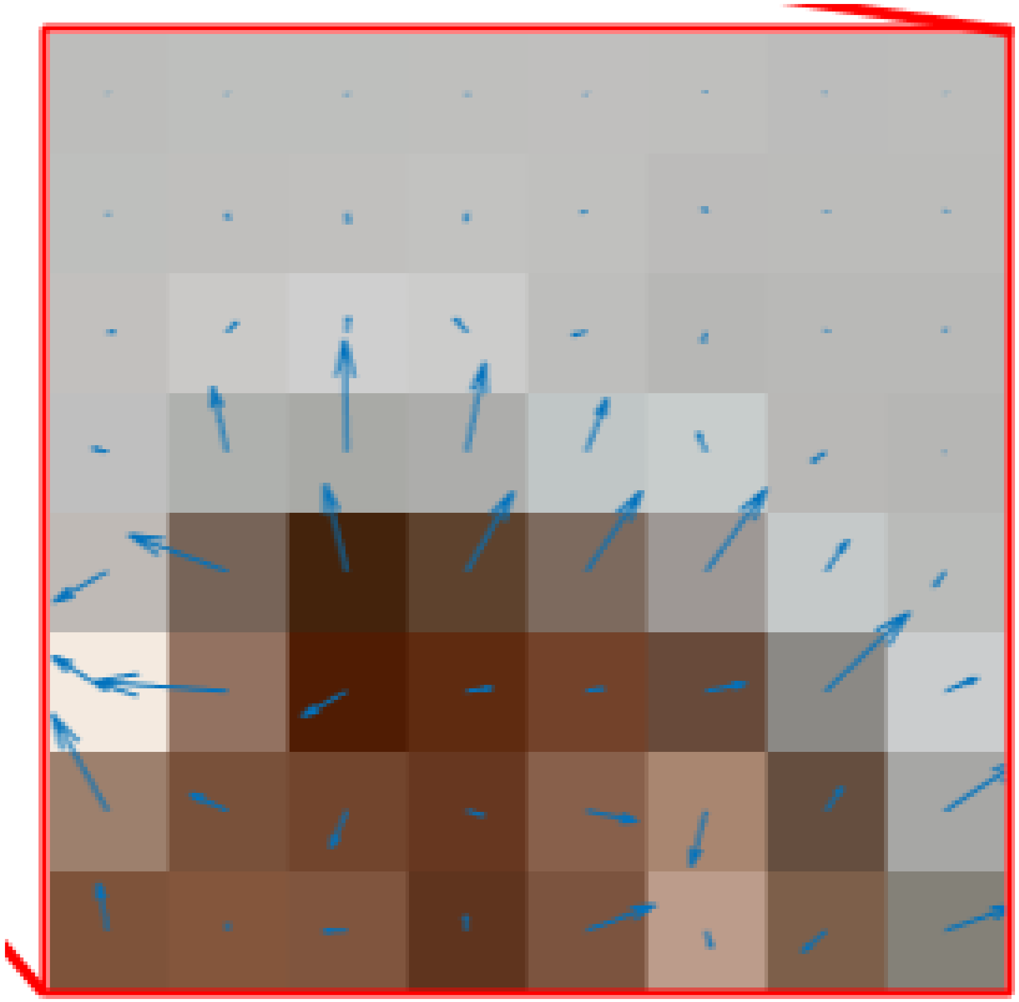
Gradient Direction

Center : The RGB patch and gradients represented using arrows. Right : The gradients in the same patch represented as numbers

Step 3 : Calculate Histogram of Gradients in 8×8 cells

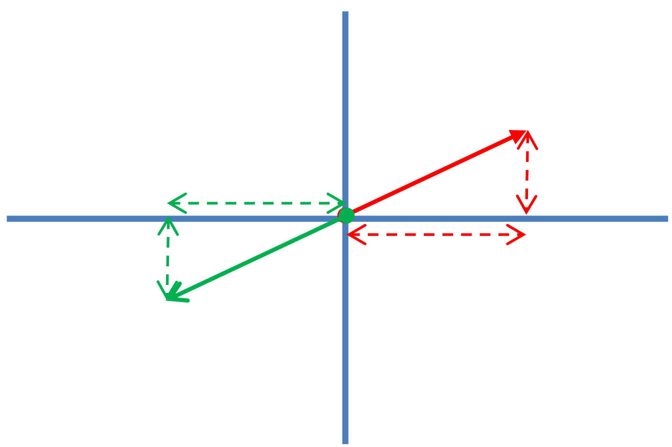
- Each patch: the arrow shows the direction of gradient and its length shows the magnitude.

- Notice how the direction of arrows points to the direction of change in intensity and the magnitude shows how big the difference is.



Step 3 : Calculate Histogram of Gradients in 8×8 cells

- the angles are between 0 and 180 degrees instead of 0 to 360 degrees. These are called **“unsigned” gradients** because a gradient and it’s negative are represented by the same numbers. In other words, a gradient arrow and the one 180 degrees opposite to it are considered the same.



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

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- But, why not use the 0 – 360 degrees ?

→ Empirically it has been shown that unsigned gradients work better than signed gradients for pedestrian detection. Some implementations of HOG will allow you to specify if you want to use signed gradients.

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

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- The histogram of gradients in these 8×8 cells is essentially a vector (or an array) of 9 bins (numbers) corresponding to angles 0, 20, 40, 60 … 160.
- A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude.

Step 3 : Calculate Histogram of Gradients in 8×8 cells

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

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

0	20	40	60	80	100	120	140	160
2	2			2				

Histogram of Gradients

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- Let's first focus on the pixel encircled in blue. It has an angle (direction) of 80 degrees and magnitude of 2. So it adds 2 to the 5th bin.

- The gradient at the pixel encircled using red has an angle of 10 degrees and magnitude of 4. Since 10 degrees is half way between 0 and 20, the vote by the pixel splits evenly into the two bins.

- If $x \in [x_0, x_1]$ then

$$x_{l-1} = \frac{(x_1 - x)}{x_1 - x_0} * y$$

$$x_l = \frac{(x - x_0)}{x_1 - x_0} * y$$

Step 3 : Calculate Histogram of Gradients in 8×8 cells

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

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

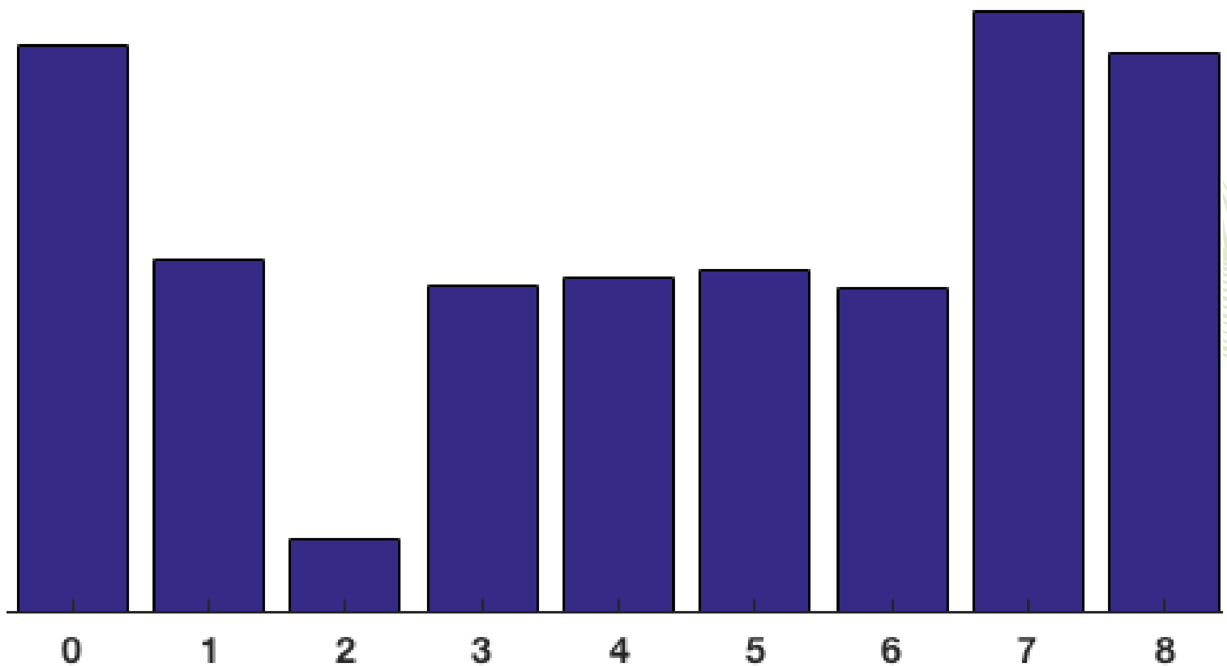
Gradient Magnitude

0	20	40	60	80	100	120	140	160
21.25								63.75

Histogram of Gradients

Step 3 : Calculate Histogram of Gradients in 8×8 cells

- The contributions of all the pixels in the 8×8 cells are added up to create the 9-bin histogram.



Step 4 : 16×16 Block Normalization

- Gradients of an image are sensitive to overall lighting.
  - If you make the image darker by dividing all pixel values by 2, the gradient magnitude will change by half, and therefore the histogram values will change by half.
- we want our descriptor to be independent of lighting variations. In other words, we would like to “normalize” the histogram so they are not affected by lighting variations.

Step 4 : 16×16 Block Normalization

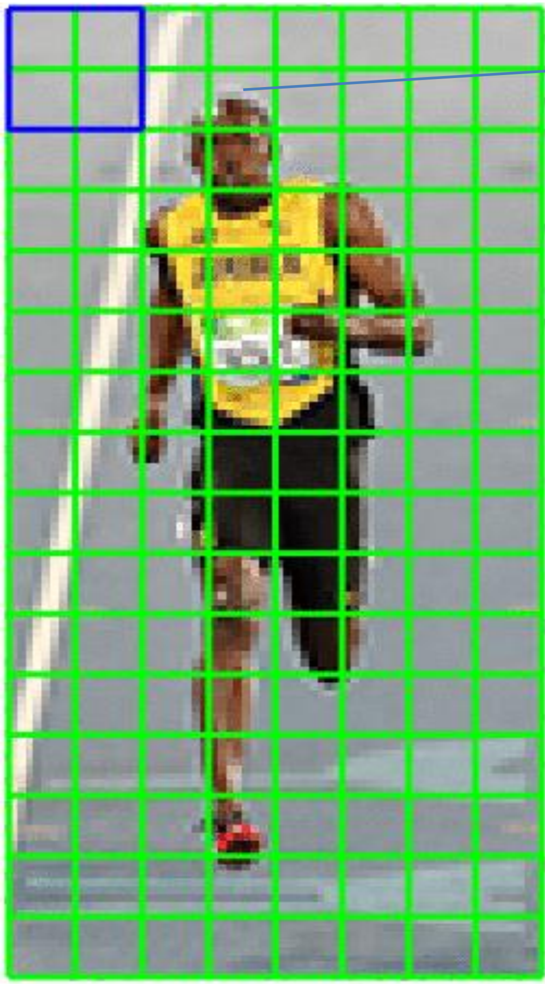
- Let's say we have an RGB color vector [128, 64, 32]. The length of this vector is $\sqrt{128^2 + 64^2 + 32^2} = 146.64$. This is also called **the L2 norm** of the vector. Dividing each element of this vector by 146.64 gives us a normalized vector [0.87, 0.43, 0.22].
- Now consider another vector in which the elements are twice the value of the first vector 2 x [128, 64, 32] = [256, 128, 64]. You can work it out yourself to see that normalizing [256, 128, 64] will result in [0.87, 0.43, 0.22], which is the same as the normalized version of the original RGB vector.

- You can see that normalizing a vector removes the scale.

Step 4 : 16×16 Block Normalization

- A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector and it can be normalized just the way a 3×1 vector is normalized.

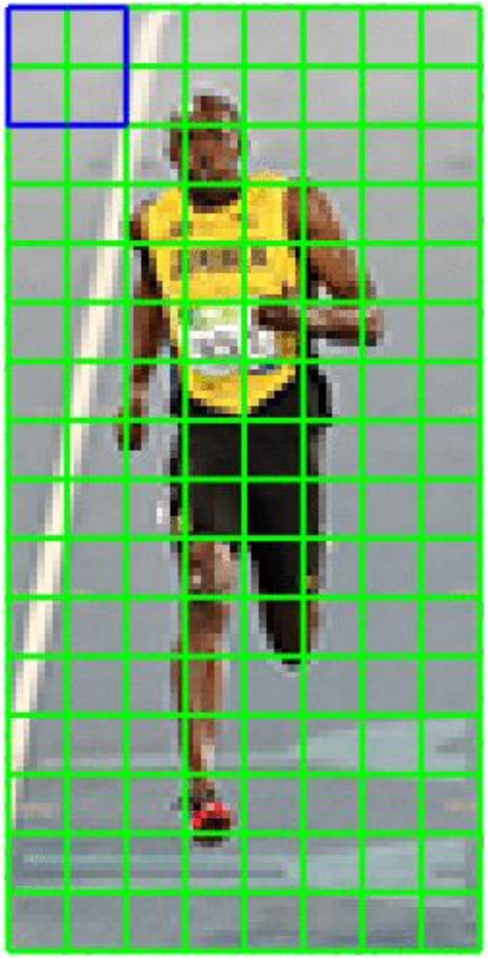
Step 4 : 16×16 Block Normalization



$$\text{normalize}(\mathbf{h}) = \frac{\mathbf{h}}{||\mathbf{h}||_2}$$

- The window is then moved by 8 pixels (see animation) and a normalized 36×1 vector is calculated over this window and the process is repeated.

Step 5 : Calculate the Histogram of Oriented Gradients feature vector



64x128

- To calculate the final feature vector for the entire image patch, the 36×1 vectors are concatenated into one giant vector. What is the size of this vector ? Let us calculate
  - How many positions of the 16×16 blocks do we have ? There are 7 horizontal and 15 vertical positions making a total of 7 x 15 = 105 positions.
  - Each 16×16 block is represented by a 36×1 vector. So when we concatenate them all into one gaint vector we obtain a 36×105 = **3780** dimensional vector.

Visualizing Histogram of Oriented Gradients

- See image on the side. You will notice that dominant direction of the histogram captures the shape of the person, especially around the torso and legs.

```
from skimage import feature

def compute_hog_train():
  train_features =[]
  for img in train_images:
    img =cv2.resize(img, (128, 256))
    (hog, hog_image) = feature.hog(img, orientations=9,
                    pixels_per_cell=(8, 8), cells_per_block=(2, 2),
                    block_norm='L2-Hys', visualize=True, transform_sqrt=True)


    train_features.append(hog)
  return train_features
```

Example

Input image

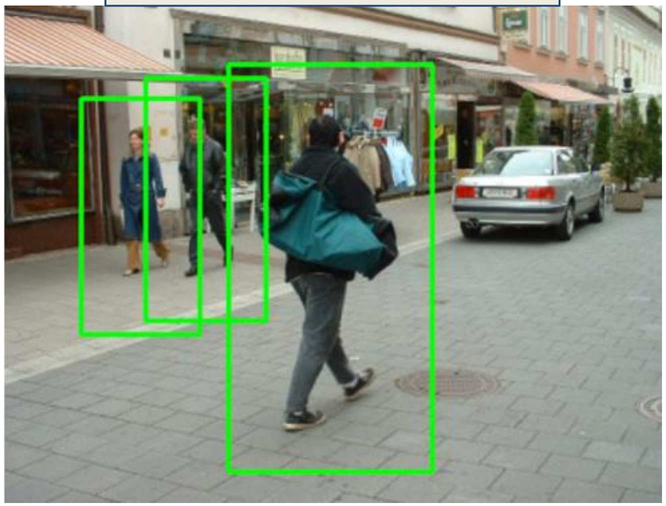
Histogram of Oriented Gradients

```
Kích thước ảnh gốc:  (657, 956, 3)
Kích thước bức ảnh crop theo winSize (pixel):  (952, 656)
Kích thước của 1 block (pixel):  (16, 16)
Kích thước của block stride (pixel):  (8, 8)
Kích thước lưới ô vuông (ô vuông):  (82, 119)
Kích thước hog feature :  (344088, 1)
```

```
[[0.04920654]
 [0.01065263]
 [0.03187659]
 ...
 [0.09042571]
 [0.1952392 ]
 [0.22843766]]
```

Ứng dụng

Human Detection

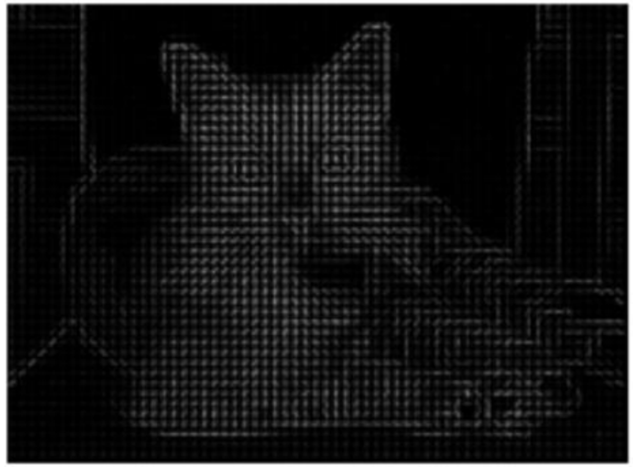


Face Detection

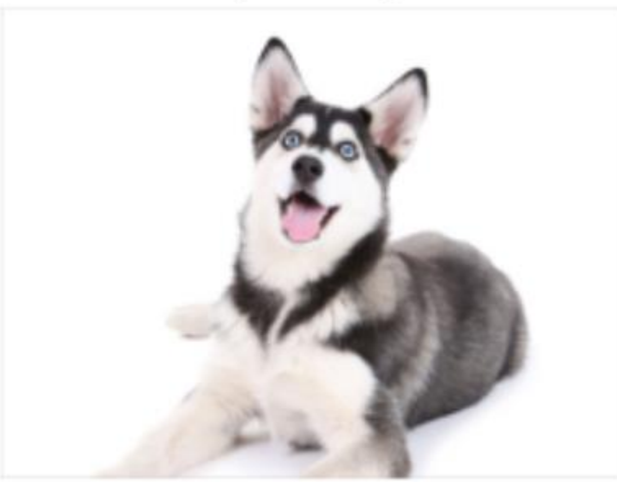
Classification

Input image

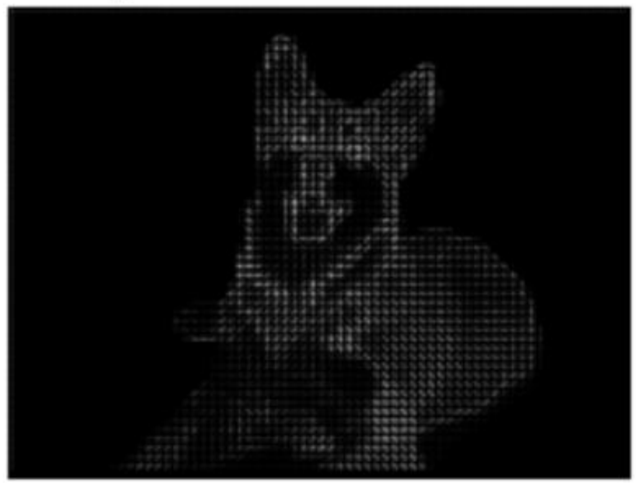
Histogram of Oriented Gradients



Input image



Histogram of Oriented Gradients

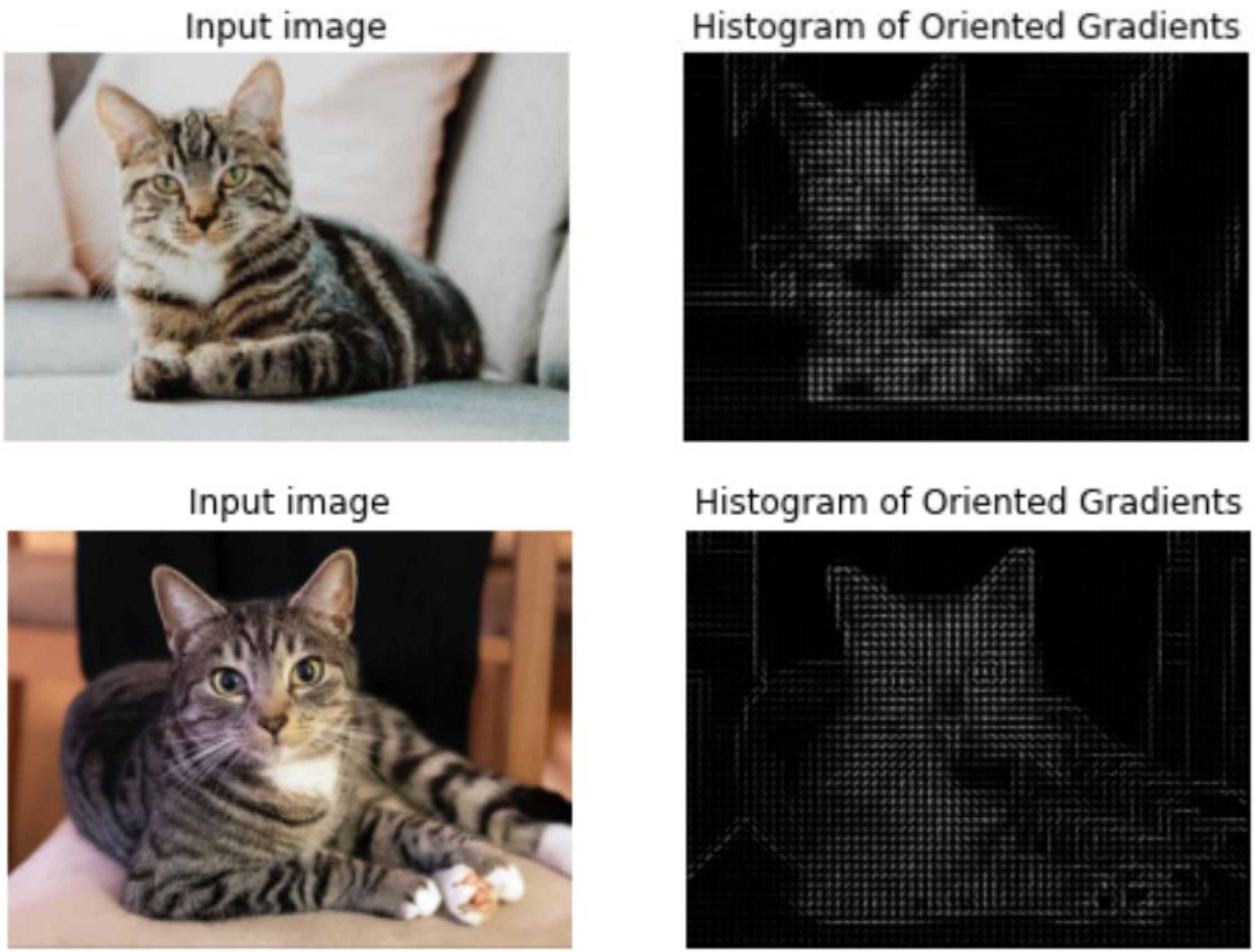


Hellinger distance:

0.679392633717499

Intersection:

0.329444963057603 6



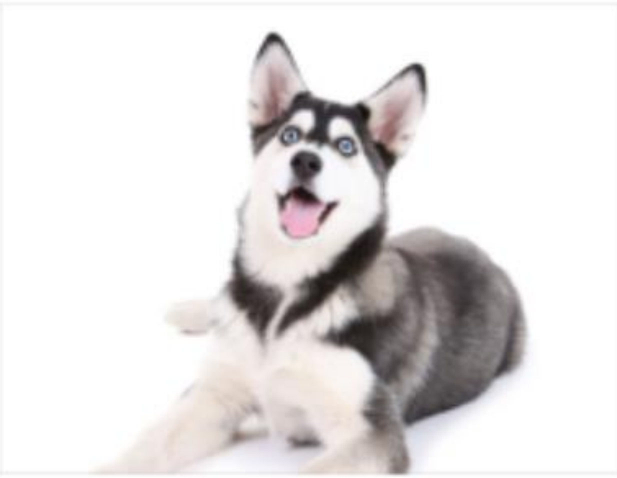
Hellinger distance:

0.4141894629252713

Intersection:

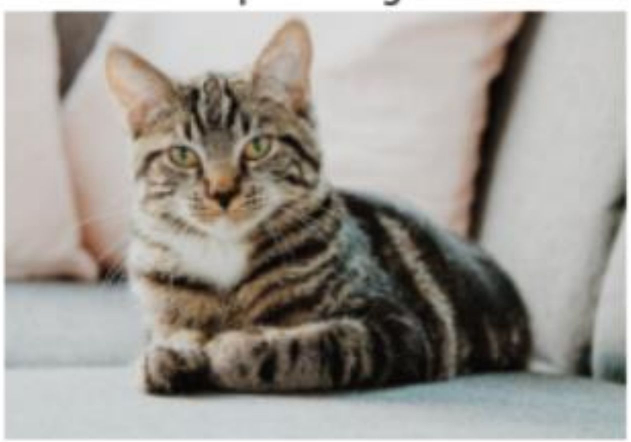
0.5983348276543636

Input image

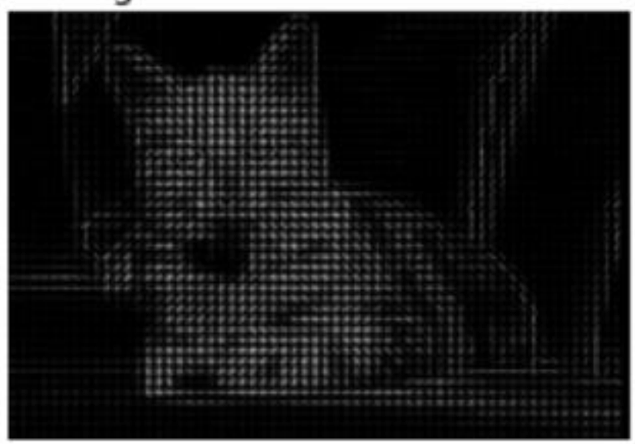


Histogram of Oriented Gradients

Input image



Histogram of Oriented Gradients



Hellinger distance:

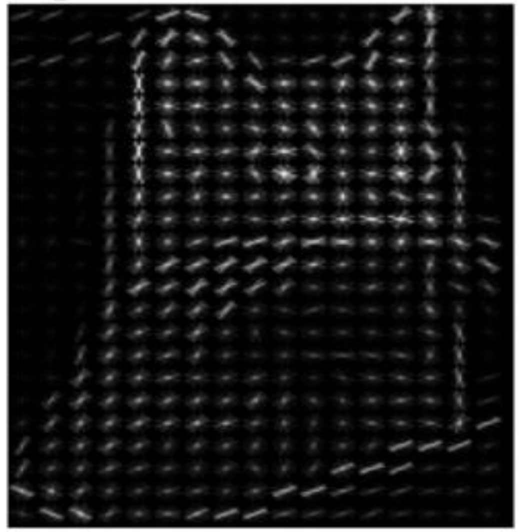
0.670362631797811

Intersection:

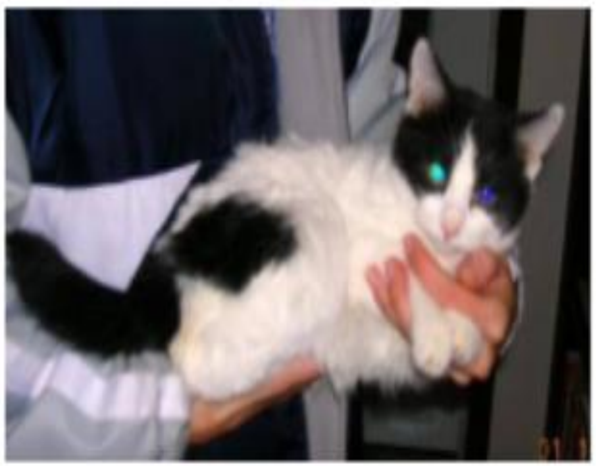
0.336479775409112

Input image

Histogram of Oriented Gradients



Input image



Histogram of Oriented Gradients

Hellinger distance:

0.5279926384038126

Intersection:

0.4713276896843683

Tài liệu tham khảo

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

Yêu cầu thực hành

- Tập dữ liệu Dog vs Cat:
  - Áp dụng Linear classification cho đặc trưng Histogram
  - Áp dụng KNN và Linear classification cho đặc trưng HOG